SANDIA REPORT

SAND98-8251

Unlimited Release

Printed October 1998

Fast Energy Minimization of Large Polymers Using Constrained Optimization

Todd D. Plantenga

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

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.

Sandia National Laboratories

RECEIVED

APR 29 1999

OSTI

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

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

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

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

Prices available from (615) 576-8401, FTS 626-8401

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

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND98-8251
Unlimited Release
Printed October 1998

Fast Energy Minimization of Large Polymers Using Constrained Optimization

Todd Plantenga
Scientific Computing Department
Sandia National Laboratories
Livermore, CA

September 17, 1998

ABSTRACT

A new computational technique is described that uses distance constraints to calculate empirical potential energy minima of partially rigid molecules. A constrained minimization algorithm that works entirely in Cartesian coordinates is used. The algorithm does not obey the constraints until convergence, a feature that reduces ill-conditioning and allows constrained local minima to be computed more quickly than unconstrained minima. Computational speedup exceeds the 3-fold factor commonly obtained in constrained molecular dynamics simulations, where the constraints must be strictly obeyed at all times.

Acknowledgment

The author acknowledges support by the Department of Energy through Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Fast Energy Minimization of Large Polymers Using Constrained Optimization

1 Introduction

The minimization of empirical potential energy models (e.g., CHARMM22 [1, 2]) is an important and computationally challenging problem. Large molecules have an enormous number of local energy minima [3], each corresponding to a particular three-dimensional structure. The ability to rapidly calculate many local minima allows researchers to investigate molecular structures near equilibrium and to generate suitable starting configurations for molecular dynamics simulations.

It is difficult to compute a local minimum for large covalently connected molecules because of **ill-conditioning** in the potential energy model (by this we mean the model has widely disparate eigenvalues in its Hessian matrix of second derivatives). Most of the ill-conditioning can be eliminated by fixing the distances between selected pairs of atoms. This paper describes a new computational method for finding local energy minima of constrained molecules. The new method is a superior option for certain molecular structures.

The idea of fixing bond lengths and valence angles to simplify calculations was proposed by Scheraga et al. [4]. Fixing the covalent structure eliminates the ill-conditioned harmonic potential energy terms, and restricts the domain to a more meaningfully differentiable set of molecular structures [3, 5, 6]. This approach has been used both for computing energy minima and for simulating molecular dynamics. Computational techniques generally fall into two categories: methods that transform the Cartesian energy model to a new unconstrained set of variables [5, 6, 7, 8, 9, 10, 11], and methods such as SHAKE [12] that retain the Cartesian coordinate energy model and treat constraints as additional equations to be satisfied [13, 14, 15, 16, 17]. This paper proposes an algorithm in the latter category, unique in that the constraints are enforced only as minimization converges.

Constraints are added to eliminate ill-conditioning associated with terms in the potential energy function. However, a long chain polymer with partially fixed covalent structure exhibits a new form of ill-conditioning when backbone dihedrals can rotate. Backbone rotations are leveraged by the rigid constraints, so that a slight rotation causes large movements in atoms located farther down the chain. This ill-conditioning becomes worse as the polymer chain lengthens.

We can avoid the new form of ill-conditioning by “loosening” the constraints. Our algorithm applies Newton’s method to the nonlinear distance constraints, but alternates the Newton iterations with minimization steps. As the algorithm converges to a structure of minimum energy, it simultaneously converges the structure to obey the constraint equations. Intermediate structures (possibly even the starting structure) are “less rigid”, and calculations are well-conditioned.

It is commonly reported that fixing bond lengths and bond angles leads to only a 3-fold speedup in molecular dynamics simulations [18, 19]. We obtain a better speedup factor for the task of energy minimization, in part because the constraints are temporarily relaxed. This paper focuses on computing structures of minimum energy, and does not address the difficulties of relaxing constraints to make molecular dynamics simulations run faster.

The chief problem with unconstrained minimization is discussed in §2 and the constrained minimization fix is presented in §3. The new algorithm is tested in §4 and §5. Details of the minimization method are specified in §6.

2 Unconstrained Minimization and Ill-Conditioning

The molecular structure corresponding to a local energy minimum can be found by applying a gradient descent method to the potential energy function. A properly constructed method will converge from any initial configuration of atoms; however, convergence is often slow for large covalently connected molecules. An example of slow convergence is shown in Figure 2.1.

Figure 2.1: Effects of ill-conditioning on unconstrained minimization

These graphs show the performance of a Newton-like algorithm as it minimizes the energy of the 455-atom protein 1ROP (see Appendix A). The upper graph plots potential energy vs. execution time; the lower plots the magnitude of the largest residual force component on any atom using a logarithmic scale. Note that atoms experience forces of more than 1.0 kcal/mol-Å for the first 2900 seconds of minimization, even though the energy barely decreases after the first 800 seconds. This region of slow progress is a sign of ill-conditioning, common for large covalently bonded molecules.

The minimization algorithm used in Figure 2.1 is a Hessian-free truncated Newton method [20, 21], an excellent choice for molecular potentials [22, 23, 24, 25]. This class of algorithm avoids computing the large Hessian matrix of second partial derivatives (see Table A.2), but still possesses the fast convergence rate of Newton methods. Even so, the algorithm struggles to reduce the energy after the first 800 seconds of execution. The next 3400 seconds produce a reduction of only 12.7 kcal/mol. It is important to realize that stopping minimization after 800 seconds does not produce a local minimum; forces are not relaxed, and the molecule undergoes significant structural changes at the 2500 second mark (see the QuickTime movie available at <http://midway.ca.sandia.gov/~tdplant/compchem/main.html>). A major goal of this paper is to completely relax the inter-atomic forces, but avoid spending computational effort on “plateau” regions where the energy hardly changes.

The undesirable “plateau” behavior can be traced to certain terms in the potential energy function. Given the Cartesian coordinates of atoms, one can compute the Euclidean distance

$r_{ij} = \|\vec{r}_i - \vec{r}_j\|_2$ between atoms i and j , the angle θ_{ijk} between three atoms, and various angles defined by arrangements of four atoms. Standard potential energy models [1, 2] take the form

$$E = \sum_{bonds} k_b(r_{ij} - r_{ij}^0)^2 + \sum_{angles} k_a(\theta_{ijk} - \theta_{ijk}^0)^2 + \sum_{dihedrals} E_{dih}(\phi_{ijkl}) \\ + \sum_{impropers} E_{imp}(\xi_{ijkl}) + \sum_{pairs\ i,j} \sigma_{cut}(r_{ij}) \left[\frac{Q_i Q_j}{4\pi\epsilon r_{ij}} + \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) \right]. \quad (2.1)$$

The terms in (2.1) model, respectively, covalent bond lengths, valence angles, dihedral rotation barriers between four atoms, improper angles (out-of-plane motions) between four atoms, and non-bonded coulombic and van der Waals potentials (multiplied by a smooth cut-off function σ_{cut}). Only r_{ij} , θ_{ijk} , ϕ_{ijkl} , and ξ_{ijkl} depend on atom positions; all other quantities in (2.1) are constants. E_{dih} and E_{imp} are not specified in this paper because they are not a significant source of computational difficulty.

Minimization converges slowly because of the covalent bond and angle terms in (2.1). Modeled as harmonic potentials, the terms possess spring “stiffness” constants k_b and k_a that generate Hessian eigenvalues many orders of magnitude larger than other eigenvalues. The extreme stiffness makes it difficult to reduce potential energy even though dihedral rotation barriers are small. This phenomenon can be quantified by computing eigenvalues of actual Hessian matrices, or by comparing second derivatives of each term in (2.1) with typical CHARMM values substituted for the constants. The eigenvalue gap is most pronounced for a relaxed molecule close to an energy minimum (unrelaxed structures may have very large van der Waals forces). In molecular dynamics this phenomenon is described as a gap between high-frequency and low-frequency components of motion, and it limits the size of integration time steps [26, §3.4].

An obvious strategy for improving minimization performance is to construct a preconditioner that accounts for ill-conditioning. This is certainly a viable alternative [24, 25, 27], and may sometimes be preferred to the constrained minimization approach of this paper (see §7).

3 Constrained Minimization with SQP

A more direct strategy is to eliminate ill-conditioning by fixing the bond lengths and angles of the molecule during minimization. The spring potential terms in (2.1) no longer change and can be dropped from the energy calculation. This technique requires a careful implementation for efficient minimization of large proteins and polymers.

To improve algorithm efficiency, we allow only distance constraints between pairs of atoms. Atoms i and j are fixed a distance R_{ij} apart by the holonomic equality constraint

$$r_{ij}^2 = \|\vec{r}_i - \vec{r}_j\|_2^2 = R_{ij}^2, \quad (3.1)$$

which is a quadratic function of the atoms’ Cartesian coordinates.

Distance constraints are sufficient for freezing the covalent structure of a molecule, leaving selected dihedrals free to rotate [28, 29]. Figure 3.1 provides a small example. These constraints can fix the structures of arbitrary branched polymers and molecules containing closed loops of atoms. Procedures for choosing a suitable set of distance constraints exist [28]. We have developed a slightly modified procedure in this paper that enhances minimization performance (see §6).

Figure 3.1: Distance constraints allow only dihedral angle rotations

Glycine (in zwitterion form $H_3N^+CH_2COO^-$) is shown schematically with covalent bonds as solid lines. Artificial distance constraints are dashed lines. The rigid constraints define a structure of three solid polyhedra free to rotate about the two axes that link them together. The structure has 30 unknowns and 22 constraints, leaving 8 degrees of freedom: 2 internal (rotatable dihedrals shown by arrows) and 6 external (translation and rotation with respect to Cartesian coordinates).

The minimization problem with distance constraints is expressed mathematically as

$$\min_x E^{con}(x), \quad x \in \mathbb{R}^{3n} \quad (3.2)$$

$$\text{subject to } c(x) = 0, \quad c \in \mathbb{R}^m. \quad (3.3)$$

In this notation the n atom coordinates are concatenated in a vector $x = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n)$, and m distance constraints are stacked in the vector c . The energy potential is now

$$E^{con} = \sum_{\text{dihedrals}} E_{dih}(\phi_{ijkl}) + \sum_{\text{pairs } i,j} \sigma_{cut}(r_{ij}) \left[\frac{Q_i Q_j}{4\pi\epsilon r_{ij}} + \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) \right] \quad (3.4)$$

and each distance constraint is of the form (3.1).

This paper advocates the use of a sequential quadratic programming (SQP) algorithm for solving (3.2)-(3.3). SQP methods are well established in the field of optimization [30], although codes for large scale problems are relatively recent [31, 32, 33, 34]. SQP algorithms use Lagrange multipliers to couple (3.2) and (3.3) into one system of nonlinear equations, then solve that system by a Newton-like method. SQP possesses a characteristic feature of Newton methods: fast convergence to a local minimum when second derivatives are used.

Each iteration of SQP solves a quadratic subproblem

$$\min_d E_k^{con} + d^T (\nabla E_k^{con}) + \frac{1}{2} d^T (\nabla^2 \mathcal{L}_k) d, \quad d \in \mathbb{R}^{3n} \quad (3.5)$$

$$\text{subject to } A_k^T d + c_k = 0, \quad (3.6)$$

where $A(x)$ is the $3n \times m$ constraint Jacobian matrix of system (3.3) (each column is the gradient of a constraint equation), and $\nabla^2 \mathcal{L}(x)$ is the $3n \times 3n$ Hessian matrix of the Lagrangian of

problem (3.2)-(3.3). All quantities in (3.5)-(3.6) are constant except the vector d ; the subscript k means that E^{con} , $\mathcal{L}$, A , and c are computed for a specific conformation x_k . In general, the solution d_k to the k th quadratic subproblem determines the next set of atom coordinates from the relation $x_{k+1} = x_k + d_k$.

The quadratic subproblem is a local approximation to (3.2)-(3.3) derived from a Taylor series expansion about the current iterate x_k . Note that the constraint equations (3.3) are treated as linear approximations in (3.6) (the second derivatives of (3.3) are present in $\nabla^2 \mathcal{L}_k$). The set of linearized constraints (3.6) define a linear manifold of dimension $n - m$. The solution d_k resolves into two components with respect to this manifold: the **horizontal** (tangent) component, found by projecting d_k onto the manifold, and the **vertical** (perpendicular) component. The vertical component makes x_{k+1} satisfy the linear constraints. The horizontal component minimizes (3.5) in the manifold; i.e., it makes the part of the force vector that is not constrained equal to zero. SQP eventually generates an iterate that satisfies the nonlinear constraints (3.3) and reduces the tangent force vector to zero.

A prominent feature of the SQP method is that constraints are not satisfied during intermediate iterations. This is because each subproblem provides only a linear approximation to the constraints. Figure 3.2 shows that the largest constraint violation is commensurate with the residual tangent force magnitude at each iteration. Both quantities converge at a quadratic rate in our implementation of SQP [34, 35]. Other constrained minimization methods choose iterates that stay closer to or always satisfy the constraints, but a serious disadvantage results from this strategy, as discussed in §5.

Figure 3.2: Convergence behavior of constrained minimization

These graphs show the performance of the SQP constrained minimization algorithm on the 455-atom protein from Figure 2.1. The upper graph plots potential energy vs. execution time (note the much shorter time scale compared to Figure 2.1). The lower logarithmic graph shows two quantities that converge to zero: the largest tangent component of force (in kcal/mol-Å), and the largest constraint violation (in Å). The two quantities are converged simultaneously by SQP.

4 Tests Between Constrained and Unconstrained Methods

The SQP algorithm performs well. It was first compared with the truncated Newton unconstrained minimization method described in §2 on a number of test proteins. For each molecule a set of 20 slightly different starting conformations were minimized, usually leading to 20 different local minima. Appendix A describes the test molecules in detail. Distance constraints were chosen to freeze all bond lengths, valence angles, and improper angles, leaving backbone and side chain dihedrals free to rotate. The full procedure for choosing constraints is described in §6.

Average algorithm performances are reported in Table 4.1. To facilitate the comparison, both algorithms used the same stop criteria and both approximated the Hessian matrix by an implicit conjugate gradient subiteration that was not preconditioned. The Hessian approximation determines the asymptotic rate of convergence to a local minimum, in this case a quadratic rate. Table 4.1 shows that constrained minimization was substantially faster for all molecules. Figure 4.1 indicates that the time savings resulted from cutting out plateau regions; thus, as expected, ill-conditioning has been eliminated by freezing covalent structure.

Table 4.1: Unconstrained and Constrained Algorithm Performance

<i>Molecule</i>	<i>Atoms</i>	<i>Free dihedrals</i>	<i>Unconstrained Results</i>		<i>Constrained Results</i>	
			<i>CPU time</i>	<i>E decrease</i>	<i>CPU time</i>	<i>E decrease</i>
P21	172	78	771 sec	245.5	169 sec	182.7
1PGB	436	207	2834 sec	852.7	401 sec	666.3
1ROP	455	231	4980 sec	690.7	646 sec	578.2
3IL8	559	276	4970 sec	860.1	903 sec	668.6
3CHY	979	484	12144 sec	1916.1	2047 sec	1601.3
153L	1432	701	33580 sec	2374.1	3962 sec	1855.0
1TIM	3742	1806	* 193100 sec	* 6987.7	53990 sec	4163.3

Molecules were minimized from identical starting conformations by an unconstrained truncated Newton method and the new constrained SQP algorithm. Both algorithms minimized until $\|force\| < 0.0001$ kcal/mol-Å; in addition, the SQP method required all constrained distances be within 10^{-7} Å of nominal. For the constrained test the ϕ and ψ protein backbone dihedrals and all significant side chain dihedrals were free to rotate. Each test molecule was minimized from 20 different starting conformations (see Appendix A). Average execution time and the average decrease in energy are reported. The constrained energy decreases less because bond lengths and angles do not relax.

* Unconstrained minimization data for 1TIM were averaged over 5 conformations instead of 20 because convergence was so time consuming.

The energy decreases in Table 4.1 suggest how much excess potential energy remains in the covalent bonds and angles of a constrained molecule. In some applications the extra bond energy locked up by distance constraints may be relatively unimportant; for example, when minimization is used to prepare a molecule for dynamic simulation. In this case constrained minimization will save significant processing time. For other applications the unrelaxed bond energies may be unacceptably large and unconstrained minimization is the only option.

Figure 4.1: Constraints eliminate ill-conditioning

Potential energy reduction is plotted as a function of execution time for the unconstrained (top) and constrained (bottom) algorithms. Each graph superimposes results from all 20 starting conformations of the test molecule (again 1ROP was used). The end of each algorithm run is shown by a circle. Unconstrained minimization exhibits long irregular “plateau” regions where little progress is made, which indicates ill-conditioning. Constraints eliminate the plateaus, but final energies are not as low.

5 Ill-conditioning Due to Constant Enforcement of Constraints

The SQP minimization algorithm treats distance constraint equations as explicit side conditions. A reasonable alternative uses the constraints to eliminate dependent variables, yielding an energy potential expressed in terms of **internal coordinates** instead of Cartesian atom positions. Freezing the covalent structure leaves free dihedral angles that define a “torsion space” of internal coordinates. In this space the energy minima of a partially rigid molecule can be computed using standard unconstrained minimization algorithms.

Internal coordinate methods have been applied successfully to molecular dynamics [9, 11, 36] and energy minimization [5, 6]. Like explicit distance constraints, they freeze bond lengths and angles to eliminate ill-conditioning of the energy potential. However, a new form of ill-conditioning can be introduced because the distance constraints are rigidly enforced throughout minimization. Consider a long chain polymer with frozen structure except for a free dihedral in the backbone. A slight rotation of the dihedral moves neighboring atoms only a slight distance, but causes atoms farther down the chain to shift much larger distances; thus, the potential energy can be highly sensitive to small changes in dihedral variables. This new ill-conditioning is caused by strict adherence to the constraints. (Rigid constraint enforcement is also known to unrealistically restrict molecular motions [19], and the torsion space potential is often modified to compensate for this distortion [37].) The SQP constrained minimization method allows constraints to relax until final convergence (see Figure 3.2) and therefore does not introduce new ill-conditioning.

To clearly demonstrate the phenomenon, a 455-atom protein was constrained to leave different sets of dihedrals free. Two cases were considered. In the first, the free dihedrals were evenly spaced over the protein backbone. In the second, the backbone was frozen and only side chain dihedrals were allowed to rotate. Each case was minimized with the SQP algorithm

and an internal coordinates method. The truncated Newton unconstrained technique of §2 was applied to internal coordinates, with torsion space forces calculated analytically using the chain rule (i.e., $\partial E^{con}/\partial\phi = \sum(\partial E^{con}/\partial r_i)/(\partial r_i/\partial\phi)$, the sum taken over all atoms shifted by rotation of the angle ϕ). Figure 5.1 shows that the internal coordinates method had increasing difficulty converging to an energy minimum as more backbone dihedrals were made free, while the SQP method degraded only mildly. Visual examination of internal coordinates results revealed the same plateau behavior seen in §2. Figure 5.2 shows the results when side chain dihedrals were left free. In this case both algorithms performed well, showing no signs of ill-conditioning. Comparing the two figures, we see that the SQP algorithm was insensitive to dihedrals being located in the backbone or side chain; it did not become ill-conditioned.

The test molecules considered in this paper were all proteins free to rotate about every ϕ and ψ backbone dihedral. Thus, the new form of ill-conditioning was present whenever internal coordinates were used. Table 5.1 shows how seriously it degraded performance.

Table 5.1: Constrained SQP and Internal Coordinates Performance

<i>Molecule</i>	<i>Atoms</i>	<i>Free dihedrals</i>	<i>SQP Results</i>		<i>Internal Coords Results</i>	
			<i>prob size</i>	<i>CPU time</i>	<i>prob size</i>	<i>CPU time</i>
P21	172	78	1002	169 sec	78	170 sec
1PGB	436	207	2523	401 sec	207	4313 sec
1ROP	455	231	2643	646 sec	231	3426 sec
3CHY	979	484	5636	2047 sec	484	49775 sec

Molecules were minimized from identical starting conformations by the new constrained SQP algorithm and an unconstrained truncated Newton method applied to internal coordinates. Both algorithms minimized until $\|force\| < 0.0001$ kcal/mol-Å; in addition, the SQP method required all constrained distances be within 10^{-7} Å of nominal. The ϕ and ψ protein backbone dihedrals and all significant side chain dihedrals were free to rotate. Each test molecule was minimized from 20 different starting conformations (see Appendix A). The *prob size* column contains the number of unknowns in each algorithm. For SQP this equals $3n + m$, where m is the number of distance constraints; for internal coordinates it is the number of free dihedrals. Molecules 3IL8 and 153L were not tested because they contain closed loops that cannot be handled by our internal coordinate transformation.

Internal coordinate minimization has the advantage of dealing with only the free dihedrals as unknowns. The SQP method creates a Lagrange multiplier variable for each constraint, and therefore manipulates a larger set of unknowns (see Table 5.1). Internal coordinates should be used when there are few free dihedrals and the polymer backbone is kept rigid.

6 Making SQP Efficient

Sequential quadratic programming was briefly described in §3. This section covers SQP in greater detail and explains how to tailor the method to exploit the special structure of distance constrained molecular minimization problems. The key is to efficiently solve linear systems derived from the constraint equations. These systems have a particular sparse matrix structure whose condition number depends on how the distance constraints are chosen. The last part of this section will discuss selection of the constraint set.

Let us return to the mathematical form (3.2)-(3.3) of the constrained minimization problem. In SQP each constraint equation is assigned a Lagrange multiplier. Collecting these multipliers

Figure 5.1: Constraints vs. internal coordinates with free dihedrals in protein backbone

Execution time of the two algorithms increased at different rates as more dihedrals were made free. The test molecule 1ROP was made rigid except for evenly spaced dihedrals in the protein backbone. Data reports the average time to converge to a local minimum for a set of 20 different starting conformations. Both algorithms stopped when the largest component of the unconstrained force was less than 0.0001 kcal/mol-Å. Minimization based on internal coordinates slowed far more as the number of free dihedrals was increased.

Figure 5.2: Constraints vs. internal coordinates with free dihedrals in side chains

Execution time of the two algorithms increased at similar rates as more dihedrals were made free. The test molecule 1ROP was made rigid except for evenly spaced dihedrals in side chain groups. Data reports the average time to converge to a local minimum for a set of 20 different starting conformations. Both algorithms stopped when the largest component of the unconstrained force was less than 0.0001 kcal/mol-Å.

into the vector $\lambda \in \mathbf{R}^m$, the Lagrangian of (3.2)-(3.3) is defined as $\mathcal{L} = E^{con}(x) + \lambda^T c(x)$. The fundamental quadratic subproblem (3.5)-(3.6) contains the gradient of the energy, ∇E^{con} , the gradient of each constraint, collected in the $3n \times m$ array A , and the full set of second partial derivatives of $\mathcal{L}$ with respect to x . As explained in §3, the solution d_k of (3.2)-(3.3) consists of a vertical and horizontal component.

Algorithm 6.1 describes the SQP implementation used in this paper. Each iteration of the main loop solves a quadratic subproblem by computing a vertical and horizontal step. The subproblem solution d_k is evaluated in lines 10 and 11, and either accepted or rejected in line 12. Rejections go back to line 7 where (3.5)-(3.6) is solved again with a tightened **trust region** constraint. This constraint is the inequality $\|d_k\|_2 \leq \Delta_k$, where Δ_k is a scalar parameter. The inequality restricts d_k to a hypersphere surrounding x_k in which the quadratic model (3.5)-(3.6) is presumed to be a valid approximation to (3.2)-(3.3). The trust radius is adjusted heuristically in line 12. The trust region inequality affects both the vertical and horizontal step lengths [38, 39] and is described more completely in [34].

Algorithm 6.1 *Outline of the SQP algorithm*

```

1  Choose an initial molecular conformation described by  $x_0$ , set  $\Delta_0 = 1$ 
2  Evaluate energy and forces to get  $E_0^{con}$  and  $\nabla E_0^{con}$ 
3  Compute constraint violations  $c_0$  and constraint gradients  $A_0$ 
4  begin loop with  $k = 0$ 
5      Compute Lagrange multiplier estimates  $\lambda_k$  from  $A_k$  and  $\nabla E_k^{con}$ 
6      if  $\|\nabla E_k^{con} - A_k \lambda_k\|_\infty < 10^{-4}$  kcal/mol-Å and  $\|c_k\|_\infty < 10^{-7}$  Å
          then exit
7      Compute the vertical step  $v_k$  from  $A_k$  and  $c_k$ 
8      Compute the horizontal step  $h_k$  from  $\nabla E_k^{con}$ ,  $A_k$ , and additional forces
          (see Algorithm 6.2 for details)
9      Set  $d_k = v_k + h_k$ 
10     Evaluate energy and forces to get  $E^{con}(x_k + d_k)$  and  $\nabla E^{con}(x_k + d_k)$ 
11     Compute  $c(x_k + d_k)$  and  $A(x_k + d_k)$ 
12     if the trial iterate is not a sufficiently good improvement over  $x_k$ 
        then decrease the trust region radius so  $\Delta_k < \|d_k\|_2$  and goto 7
        else set  $x_{k+1} = x_k + d_k$ ,  $\Delta_{k+1} \geq \Delta_k$ , and increment  $k$ 
13 end loop

```

The most difficult part of the SQP iteration is computation of the horizontal step. It involves the second derivatives $\nabla^2 \mathcal{L}$, and must comply with the linear constraint equations (3.6) already solved by the vertical step. The latter condition means that $A_k^T h_k = 0$. Substituting $d = v_k + h$ into (3.5)-(3.6), we obtain the subproblem solved by the horizontal step:

$$\min_h \quad h^T [\nabla E_k^{con} + (\nabla^2 \mathcal{L}_k) v_k] + \frac{1}{2} h^T (\nabla^2 \mathcal{L}_k) h, \quad h \in \mathbf{R}^{3n} \quad (6.1)$$

$$\text{subject to} \quad A_k^T h = 0 \quad (6.2)$$

$$\|h\|_2 \leq \sqrt{\Delta_k^2 - \|v_k\|_2^2}. \quad (6.3)$$

The constraint (6.2) states that h is the projection of some vector onto the null space of the matrix A_k^T . This suggests that h can be calculated by first finding the vector that

minimizes (6.1), then projecting that vector to satisfy (6.2). The orthogonal projection is accomplished by solving a linear least squares problem. If $r \in \mathbb{R}^{3n}$ is a vector, then its null space projection onto A^T is

$$\mathcal{P}_{null}\{A^T, r\} = r - Ay^* \quad \text{where } y^* \in \mathbb{R}^m \text{ solves } \min_y \|r - Ay\|_2.$$

We will consider the computation of $\mathcal{P}_{null}$ later in this section.

The challenge in (6.1) is to find the minimizing vector without calculating second derivatives. Explicit computation of the Hessian is usually avoided because of its size (see Table A.2 in Appendix A). We use the same Hessian-free truncated Newton technique that was applied to unconstrained minimization in §2. Outlined in Algorithm 6.2, it features a conjugate gradient subiteration with one extra test in line 8 to enforce the trust region inequality (6.3). Notice that $\nabla^2 \mathcal{L}_k$ appears only in lines 5, 6, and 9, each time multiplied by the search direction s_i . This Hessian-vector product is the curvature of $\mathcal{L}_k$ in the direction s_i . It is approximated by a finite difference [40]:

$$[\nabla^2 \mathcal{L}(x)]s \approx \frac{\nabla \mathcal{L}(x + \epsilon s) - \nabla \mathcal{L}(x)}{\epsilon}, \quad (6.4)$$

where $2 \max\{1, \|x\|\} \sqrt{\text{machine precision}}/\|s\|$ is a typical value for ϵ . Thus, the horizontal step computation makes one force calculation for each pass through the loop in Algorithm 6.2.

Algorithm 6.2 *Procedure for computing the horizontal step*

```

1   $x_k, \nabla E_k^{con}, A_k, v_k$ , and  $\Delta_k$  are constants supplied from Algorithm 6.1
2  Initialize  $h_0 = 0$  and  $r_0 = -\nabla E_k^{con}$ 
3  Compute  $t_0 = \mathcal{P}_{null}\{A_k^T, r_0\}$  and start with  $s_0 = t_0$ 
4  begin loop with  $i = 0$ 
5      Approximate  $[(\nabla^2 \mathcal{L}_k)s_i]$  by finite differences between gradients (see (6.4))
6       $\alpha = r_i^T t_i / s_i^T [(\nabla^2 \mathcal{L}_k)s_i]$ 
7       $h_{i+1} = h_i + \alpha s_i$ 
8      if  $\|h_{i+1}\|_2^2 > \Delta_k^2 - \|v_k\|_2^2$ 
9          then find  $\tau \geq 0$  so that  $\|h_i + \tau s_i\|_2 = \Delta_k$  and return  $h \leftarrow h_i + \tau s_i$ 
10      $r_{i+1} = r_i - \alpha [(\nabla^2 \mathcal{L}_k)s_i]$ 
11      $t_{i+1} = \mathcal{P}_{null}\{A_k^T, r_{i+1}\}$ 
12     if  $\sqrt{r_{i+1}^T t_{i+1}} < 0.01 \sqrt{r_0^T t_0}$ 
13         then return  $h \leftarrow h_{i+1}$ 
14      $s_{i+1} = t_{i+1} + (r_{i+1}^T t_{i+1} / r_i^T t_i) s_i$ 
15     Increment  $i$ 
16 end loop
```

Algorithm 6.2 requires projection operations in lines 3 and 10. When A is full rank, the solution to the orthogonal projection subproblem can be written as $y^* = (A^T A)^{-1} A^T r$, and hence

$$\mathcal{P}_{null}\{A^T, r\} = [I - A(A^T A)^{-1} A^T]r. \quad (6.5)$$

This result comes from a normal equations formulation and may be computationally inaccurate if A is not well-conditioned [41, §2.2]; however, corrective refinements can be applied [42]. The advantage of (6.5) is that the Cholesky factor of $A_k^T A_k$ can be computed once per iteration

of Algorithm 6.1, then reused to quickly calculate the projections in Algorithm 6.2. The Cholesky factor is also used to compute the vertical step in line 7 of Algorithm 6.1 and the Lagrange multiplier estimates in line 5. Table 6.1 testifies to the efficiency of the Cholesky factorization strategy: constraint Jacobian linear algebra costs account for only 5% of the total minimization process. The normal equations with corrective refinement also provide enough accuracy to satisfy the tight constraint tolerance required in line 6 of Algorithm 6.1.

Table 6.1: Division of Work in Algorithm 6.1

<i>Molecule</i>	<i>Atoms</i>	<i>Free dihedrals</i>	<i>CPU Time</i>			<i>Force Evaluations</i>	
			<i>force evals</i>	<i>A_k subprobs</i>	<i>other</i>	<i>in Alg 6.2</i>	<i>elsewhere</i>
P21	172	78	86 %	5 %	9 %	90 %	10 %
1PGB	436	207	89 %	4 %	7 %	90 %	10 %
1ROP	455	231	88 %	5 %	7 %	89 %	11 %
3IL8	559	276	88 %	5 %	7 %	91 %	9 %
3CHY	979	484	89 %	4 %	7 %	90 %	10 %
153L	1432	701	90 %	4 %	6 %	91 %	9 %
1TIM	3742	1806	87 %	5 %	8 %	91 %	9 %

This table identifies the most costly parts of the SQP constrained minimization algorithm. The middle columns indicate that energy/force evaluations dominate. The A_k subprobs column breaks out the percentage of CPU time spent on linear algebra subproblems involving the constraint Jacobian matrix (orthogonal projections in Algorithm 6.2, computation of the vertical step and Lagrange multipliers in Algorithm 6.1, and computation of the Cholesky factor of $A_k^T A_k$). The last two columns reveal that most force evaluations occur in line 5 of Algorithm 6.2. Other force calculations are made in lines 2 and 10 of Algorithm 6.1. Note that all percentages hold fairly constant over the range of molecule sizes.

The normal equations Cholesky factorization technique works well because A is sparse and well-conditioned. Each distance constraint equation is a function of two atoms' coordinates; hence, each column of A has just six nonzero elements. Table 6.2 shows that $A^T A$ is also sparse (it is formed explicitly) and that Cholesky factorization does not cause significant fill-in (the factors were computed using the code of Ng and Peyton from Oak Ridge National Laboratory [43]).

The condition number of A is influenced greatly by the choice of distance constraints. In Figure 3.1 we saw that distance constraints create rigid polyhedra linked by rotatable dihedral angles. A polyhedron with p atoms can have $(p^2 - p)/2$ possible distance constraints, but as a rigid body it has only $3p - 6$ degrees of freedom; thus, when $p \geq 5$ the experimenter must choose a subset of the possible constraints. The chosen set should maximize linear independence to keep the condition number of A small. Such a subset can be determined by performing a rank-revealing QR factorization [44, §5.4.1]. Rank-revealing factorization is a Gram-Schmidt orthogonalization procedure that chooses the next column to be eliminated by examining the size of all remaining pivot elements. It returns an optimal ordering of constraints with their pivot sizes. For each rigid polyhedron in a molecule, the full set of $(p^2 - p)/2$ constraint gradients is assembled in a matrix, arranged in any order, then factored with the LAPACK subroutine dgeqpf [45]. The first $3p - 6$ constraints are chosen. Note that this is a preprocessing step, performed once before invoking the minimization algorithm.

Table 6.2: Constraint Jacobian Matrix Sparsity

Molecule	Size of A		A		$A^T A$		Factor of $A^T A$	
	rows	cols	nnz	fill-in	nnz	fill-in	nnz	fill-in
P21	543	459	2,754	1.1 %	2,829	1.3 %	4,127	2.0 %
1PGB	1368	1155	6,930	0.4 %	7,009	0.5 %	10,075	0.8 %
1ROP	1440	1203	7,218	0.4 %	7,134	0.5 %	10,130	0.7 %
3IL8	1755	1485	8,910	0.3 %	9,135	0.4 %	15,335	0.7 %
3CHY	3063	2573	15,438	0.2 %	15,530	0.2 %	22,936	0.4 %
153L	4491	3796	22,776	0.1 %	23,160	0.2 %	36,729	0.3 %
1TIM	11694	9876	59,256	0.1 %	60,239	0.1 %	97,432	0.1 %

The Jacobian matrix A has $3n$ rows and m columns corresponding to a molecule with n atoms and m distance constraints (n does not agree with Table 4.1 because constrained problems have extra “fictitious” atoms to improve conditioning; see Table 6.3). There are six nonzero elements in each column of A . Linear subproblems based on normal equations require forming and factoring the $m \times m$ matrix $A^T A$. The columns marked *nnz* contain the number of nonzero elements in the matrix, and *fill-in* expresses the sparsity as the percentage of matrix elements which are nonzero. Note that the Cholesky factor of $A^T A$ (obtained using [43]) remains sparse, apparently scaling as $O(m)$.

A simple example demonstrates the impact made by QR factorization on the condition number of the constraint Jacobian matrix. Nine atoms form a rigid three-dimensional helix in the two figures at left. Twenty-one constraints are needed to make the helix rigid. The upper figure shows a set of distance constraints chosen to connect nearest neighbors; in particular, it includes all covalent bonds (solid lines). The bottom figure shows the optimal set of constraints chosen by rank-revealing QR factorization; only one covalent bond was selected. The constraint Jacobian of the upper set has a condition number of 1790, compared with 30.1 for the optimal set (condition number is computed from singular values of the rectangular Jacobian matrix). As atoms are added to the make the helix longer, the condition number disparity becomes more extreme; for example, a helix of 18 atoms has condition numbers of 9070 and 60.5.

QR factorization determines the optimal set of constraints for each rigid polyhedron in isolation. The combined set from all polyhedra is not necessarily optimal over the full set of m constraints, but it is much easier to make many small rank-revealing factorizations. We must also be sure to control the rotational degree of freedom between connecting rigid polyhedra. Each polyhedron must constrain the length of the bond that forms the axis of a connecting rotatable dihedral; otherwise, the bond length could shift even though polyhedra remain rigid. Therefore, dihedral axis bonds are included in the constraint set even if the condition number increases.

Sometimes the experimenter wants to make a nearly flat segment of molecule into a rigid body. In this case distance constraints are nearly dependent and even the optimal set has a large condition number. The QR factorization indicates near dependency when the magnitudes of the last pivots remain similar [44, §5.5.7]. If a polyhedron is nearly planar, then a fictitious atom is positioned 1 Å out of the plane and included in the polyhedron. A new well-conditioned set of constraints is then computed. The fictitious atom is assigned null force field parameters

(zero charge, etc.); it affects only the constraints. For the proteins tested in this paper the QR factorization indicated a need for fictitious atoms in side chains of most Arg, Asn, Asp, Gln, Glu, His, Phe, Trp, and Tyr residues. Table 6.3 shows the total number of fictitious atoms used.

Table 6.3: Fictitious Atoms in Test Molecules

	<u>True Atoms</u>	<u>Fictitious Atoms</u>
P21	172	9
1PGB	436	20
1ROP	455	25
3IL8	559	26
3CHY	979	42
153L	1432	65
1TIM	3742	156

The *true* number of atoms matches Tables 4.1 and 6.1. It should be increased by *fictitious* atoms to match the number of rows in Table 6.2. Constrained minimization includes fictitious atoms, but force field calculations involve only the true atoms.

This completes the lengthy description of how distance constraints are chosen and used. Since Table 6.1 shows that A_k subproblems consume only 5% of total CPU time, it might seem that the attention given to this topic has been excessive. Our original implementation [46] used no QR factorization or fictitious atoms, and there the CPU time fraction exceeded 80%. Constrained minimization was not even as fast as the unconstrained minimization algorithm of §2 because the condition number of A was so large. Thus, the attention is warranted. The procedure described here can generate distance constraints for molecules with arbitrary branching, closed loops (both 3IL8 and 153L have disulfide bridges), and any number of free dihedrals (including zero for a completely rigid structure). Multiple interacting molecules can also be minimized (1TIM is a dimer).

7 Conclusions

This paper has shown that a specialized sequential quadratic programming algorithm can efficiently compute local energy minima of partially rigid molecules. Fixing covalent structure with distance constraints eliminates ill-conditioning caused by the harmonic bond length and valence angle potentials. SQP, unlike other algorithms, relaxes the constraints until convergence and thus prevents new ill-conditioning that may appear in long chain polymers with rotatable backbone dihedrals. Experiments show that the SQP algorithm performs well on proteins with up to 3700 atoms. The SQP algorithm is particularly efficient because it uses a sparse Cholesky factorization of the constraint system Jacobian matrix. The Jacobian is well-conditioned if constraints are chosen from a QR factorization that analyzes each rigid molecular substructure.

At the end of §2 it was suggested that applying a preconditioner to the basic unconstrained minimization algorithm might serve to reduce ill-conditioning. Schlick et al. [24] constructed a preconditioner from the analytic Hessian of the bonding energy terms. They obtained a speedup factor of 5-9 for molecules with up to 66 atoms. This author is examining a quasi-

Newton preconditioner [27, 47] which has so far produced a 5-fold speedup for test molecules 1ROP and 3CHY. Table 4.1 indicates that distance constraints using SQP speeds up minimization by a factor of 4-8 for molecules with up to 3700 atoms. Of course, distance constraints do not relax fixed covalent structure, so a direct comparison is not meaningful. In any event, constrained minimization is important for restricting the search space of local minima [5, 6], keeping bond lengths within the domain of a harmonic potential model [6], and preparing equilibrium structures for constrained dynamics simulations.

An intriguing possibility is to apply an appropriate preconditioner to the CG subiteration of Algorithm 6.2. Table 6.1 shows that 90% of all force evaluations occur in Algorithm 6.2, accounting for roughly 80% of total CPU time. Any reduction in the number of subiterations will proportionally reduce the number of force calculations. A preconditioner for Algorithm 6.2 will have to combine with the projection operation in line 10 to honor the linearized constraint equations.

Many thanks go to Richard Judson and Mike Colvin for sharing their expertise. More information related to this work is at <http://midway.ca.sandia.gov/~tdplant/compchem/main.html>.

References

- [1] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminatha, and M. Karplus. CHARMm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.
- [2] Lennart Nilsson and Martin Karplus. Empirical energy functions for energy minimization and dynamics of nucleic acids. *J. Comp. Chem.*, 7:591–616, 1986.
- [3] M. Saunders, K. N. Houk, Y.-D. Wu, W. C. Still, M. Lipton, G. Chang, and W. C. Guida. Conformations of cycloheptadecane. A comparison of methods for conformational searching. *J. Am. Chem. Soc.*, 112:1419–1427, 1990.
- [4] F. A. Momany, R. F. McGuire, A. W. Burgess, and H. A. Scheraga. Geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *J. Phys. Chem.*, 79:2361–2381, 1975.
- [5] K. A. Palmer and H. A. Scheraga. Standard-geometry chains fitted to X-ray derived structures: Validation of the rigid-geometry approximation. *J. Comp. Chem.*, 12:505–526, 1991.
- [6] R. Abagyan, M. Totrov, and D. Kuznetsov. ICM – a new method for protein modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comp. Chem.*, 15:488–506, 1994.
- [7] H. Abe, W. Braun, T. Noguti, and N. Gō. Rapid calculation of first and second derivatives of conformational energy with respect to dihedral angles for proteins: General recurrent equations. *Comp. and Chem.*, 8:239–247, 1984.
- [8] A. K. Mazur, V. E. Dorofeev, and R. A. Abagyan. Derivation and testing of explicit equations of motion for polymers described by internal coordinates. *J. Comput. Phys.*, 92:261–272, 1991.
- [9] S. Sunada and N. Gō. Small-amplitude protein conformational dynamics: Second-order analytic relation between Cartesian coordinates and dihedral angles. *J. Comp. Chem.*, 16:328–336, 1995.
- [10] J. Eisenfeld, S. Vajda, I. Sugar, and C. DeLisi. Constrained optimization and protein structure determination. *Am. J. Physiology*, 261:C376–C386, 1991.
- [11] K. D. Gibson and H. A. Scheraga. Variable step molecular dynamics: An exploratory technique for peptides with fixed geometry. *J. Comp. Chem.*, 11:468–486, 1990.
- [12] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of *n*-alkanes. *J. Comput. Phys.*, 23:327–341, 1977.
- [13] W. F. van Gunsteren and M. Karplus. A method for constrained energy minimization of macromolecules. *J. Comp. Chem.*, 1:266–274, 1980.
- [14] A. Jain, N. Vaidehi, and G. Rodriguez. A fast recursive algorithm for molecular dynamics simulation. *J. Comput. Phys.*, 106:258–268, 1993.
- [15] L. M. Rice and A. T. Brunger. Torsion angle dynamics: Reduced variable conformational sampling enhances crystallographic structure refinement. *PROTEINS: Str. Func. Gen.*, 19:277–290, 1994.
- [16] Y. Duan, S. Kumar, J. M. Rosenberg, and P. A. Kollman. Gradient SHAKE: An improved method for constrained energy minimization in macromolecular simulations. *J. Comp. Chem.*, 16:1351–1356, 1995.
- [17] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije. LINCS: A linear constraint solver for molecular simulations. *J. Comp. Chem.*, 18:1463–1472, 1997.
- [18] W. F. van Gunsteren. Constrained dynamics of flexible molecules. *Molec. Phys.*, 40:1015–1019, 1980.
- [19] Wilfred F. van Gunsteren and Martin Karplus. Effect of constraints on the dynamics of macromolecules. *Macromolecules*, 15:1528–1544, 1982.
- [20] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.
- [21] D. P. O’Leary. A discrete Newton algorithm for minimizing a function of many variables. *Math. Prog.*, 23:20–33, 1982.
- [22] T. Schlick and M. Overton. A powerful truncated Newton method for potential energy minimization. *J. Comp. Chem.*, 8:1025–1039, 1987.

- [23] T. D. Plantenga. Parallel unconstrained minimization of potential energy in LAMMPS. Technical Report SAND98-8201, Sandia National Laboratories, Livermore, CA, 1997.
- [24] P. Derreumaux, G. Zhang, and T. Schlick. A truncated Newton minimizer adapted for CHARMM and biomolecular applications. *J. Comp. Chem.*, 15:532-552, 1994.
- [25] Tamar Schlick. Optimization methods in computational chemistry. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 3, pages 1-72. VCH Publishers, New York, 1992.
- [26] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford UP, Oxford, U.K., 1987.
- [27] G. Gray, T. Plantenga, and M. Viswanathan. Effectiveness of Hessian-free unconstrained algorithms for minimization of molecular potential energy. In progress, 1998.
- [28] G. Ciccotti. Molecular dynamics of rigid systems in Cartesian coordinates: A general formulation. *Molec. Phys.*, 47:1253-1264, 1982.
- [29] G. Ciccotti and J.-P. Ryckaert. Molecular dynamics simulation of rigid molecules. *Comp. Phys. Rep.*, 4:345-392, 1986.
- [30] R. Fletcher. *Practical Methods of Optimization*. Wiley & Sons, Chichester, UK, second edition, 1990.
- [31] J. T. Betts and P. D. Frank. A sparse nonlinear optimization algorithm. *J. Opt. Th. and Appl.*, 82:519-541, 1994.
- [32] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Numerical Analysis Report 96-2, Dept of Mathematics, University of California at San Diego, La Jolla, CA, 1996.
- [33] R. Franke and E. Arnold. On the integration of a large-scale nonlinear optimization tool with open modeling and simulation environments for dynamic systems. In *10th European Simulation Multiconference*, Budapest, Hungary, June 1996.
- [34] M. Lalee, J. Nocedal, and T. D. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. To appear in *SIAM J. Optimization*, 1993.
- [35] Todd D. Plantenga. *Large-Scale Nonlinear Constrained Optimization Using Trust Regions*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Northwestern University, 1994.
- [36] N. Gö and H. A. Scheraga. Ring closure and local conformational deformations of chain molecules. *Macromolecules*, 3:178-187, 1970.
- [37] E. Helfand. Flexible vs. rigid constraints in statistical mechanics. *J. Chem. Phys.*, 7:5000-5007, 1979.
- [38] R. H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, Houston, TX, May 1987.
- [39] Emmanuel O. Omojokun. *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*. PhD thesis, Dept. of Computer Science, University of Colorado, 1989.
- [40] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Computing forward-difference intervals for numerical optimization. *SIAM J. Sci. Stat. Comp.*, 4:310-321, 1983.
- [41] Ake Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, Philadelphia, 1996.
- [42] Mary E. Hribar. *Large-Scale Constrained Optimization*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Northwestern University, 1996.
- [43] E. G. Ng and B. W. Peyton. Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM J. Sci. Stat. Comp.*, 14:1034-1056, 1993.
- [44] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins UP, Baltimore, second edition, 1991.
- [45] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostronchov, and D. Sorensen. *LAPACK User's Guide*. SIAM, Philadelphia, 1992.
- [46] T. D. Plantenga and R. S. Judson. Energy minimization along dihedrals in Cartesian coordinates using constrained optimization. Technical Report SAND95-8724, Sandia National Laboratories, Livermore, CA, 1995.
- [47] J. L. Morales and J. Nocedal. Automatic preconditioning by limited memory quasi-Newton updating. Technical Report OTC 97/08, Optimization Technology Center, Argonne National Laboratory and Northwestern University, 1997.

- [48] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, 112:535-542, 1977.
- [49] T. Gallagher, P. Alexander, P. Bryan, and G. L. Gilliland. Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR. *Biochemistry*, 33:4721-4729, 1994.
- [50] D. W. Banner, M. Kokkinidis, and D. Tsernoglou. Structure of the ColE1 rop protein at 1.7 Angstroms resolution. *J. Mol. Biol.*, 196:657-675, 1987.
- [51] E. T. Baldwin, I. T. Weber, R. St. Charles, J.-C. Xuan, E. Appella, M. Yamada, K. Matsushima, B. F. P. Edwards, G. M. Clore, A. M. Gronenborn, and A. Wlodawer. Crystal structure of interleukin 8: Symbiosis of NMR and crystallography. *Proc. Natl. Acad. Sci. USA*, 88:502-506, 1988.
- [52] K. Volz and P. Matsumura. Crystal structure of Escherichia coli cheY refined at 1.7 Angstroms resolution. *J. Biol. Chem.*, 266:15511-15519, 1991.
- [53] L. H. Weaver, M. G. Gruetter, and B.W. Matthews. The refined structures of goose lysozyme and its complex with a bound trisaccharide show that the 'goose-type' lysozymes lack a catalytic aspartate residue. *J. Mol. Biol.*, 245:54-68, 1995.
- [54] D. W. Banner, A. C. Bloomer, G. A. Petsko, D. C. Phillips, and I. A. Wilson. Atomic coordinates for triose phosphate isomerase from chicken muscle. *Biochem. Biophys. Res. Commun.*, 72:146-155, 1976.
- [55] QUANTA/CHARMM. Waltham MA, 1993. The results published were generated in part using the program QUANTA. This program was developed by Molecular Simulations, Inc.
- [56] R. Judson, D. Barsky, T. Faulkner, D. McGarrah, C. Melius, J. Meza, E. Mori, T. Plantenga, and A. Windemuth. CCEMD - Center for Computational Engineering Molecular Dynamics: Theory and user's guide, version 2.2. Technical Report SAND95-8258, Sandia National Laboratories, Livermore, CA, 1995.

Appendix A Experimental Procedures

The molecules tested in this paper are listed in Table A.1. All except P21 were proteins taken from the Brookhaven database [48] using the code shown in the first column. P21 was an artificial protein with amino acid sequence Gly-Ala-Val-Leu-Ile-Met-Asp-Glu-Asn-Gln-Cys-Ser-Thr-His-Phe-Tyr-Trp-Lys-Arg-Pro-Gly, arranged in the all α -helical conformation.

Table A.1: Test Molecules

<i>ID Code</i>	<i>Name</i>	<i>num residues</i>	<i>num atoms</i>
P21	special test case (see text)	21	172
1PGB	Immunoglobulin Binding Protein [49]	56	436
1ROP	Transcription Regulation Protein [50]	56	455
3IL8	Interleukin 8 [51]	72	559
3CHY	Signal Transduction Protein [52]	129	979
153L	Hydrolase [53]	185	1432
1TIM	Isomerase [54]	496	3742

Each molecule was read into QUANTA [55] and assigned charges and atom types. No hydrogens were used. Each molecule initially underwent about ten steps of unconstrained gradient minimization to partially relax any severely stretched bond lengths or angles. In all experiments the long range forces were truncated at 10Å with a smooth differentiable cutoff function. Finite cutoff distances are commonly used to reduce the computation costs of the energy and forces. The choice of 10Å is partially justified by the data in Table A.2.

Table A.2: Number of Atom-Atom Interactions

<i>Molecule</i>	<i>Atoms</i>	<i>max possible</i>	<i>20Å cutoff</i>	<i>10Å cutoff</i>	<i>5Å cutoff</i>
P21	172	14,706	12,000	6,000	1,000
1PGB	436	94,830	80,000	25,000	4,000
1ROP	455	103,285	68,000	23,000	4,000
3IL8	559	155,961	112,000	30,000	5,000
3CHY	979	478,731	296,000	63,000	9,000
153L	1432	1,024,596	509,000	103,000	14,000
1TIM	3742	6,999,411	1,592,000	295,000	39,000

This table lists the number of long range interactions between pairs of atoms for different cutoff distances. The cutoff truncates coulomb and van der Waals forces to limit the number of interactions that must be calculated. The *max possible* column contains the number of interactions if no cutoff were used; it equals $(n^2 - n)/2$. Other columns contain typical numbers of interactions for the given cutoff distance. Sparsity of the potential energy Hessian matrix is determined by the cutoff distance. Each atom contributes three nonzero Hessian elements on the main diagonal (one for each Cartesian coordinate), and each atom-atom interaction adds six unique off-diagonal elements. For example, the Hessian matrix for 1ROP contains roughly $3 \times 455 + 6 \times 23,000 \approx 139,000$ unique nonzero elements.

Most experiments in this paper report an averaged value resulting from 20 separate minimizations that started from slightly different conformations. The 20 starting structures were generated by randomly rotating amino acid side chains. Rotations were about the dihedral angle connecting the α and β carbon atoms in the side chain (Gly, Ala, Pro, and Cys residues were

not rotated). The procedure created 20 conformations with identical backbone placements, covalent bond lengths, and valence angles. Initial energies differed as shown in Table A.3.

Table A.3: Initial Energies for each Set of 20 Conformations (in kcal/mol)

	<u>Lowest Energy</u>	<u>Highest Energy</u>
P21	-166.0	-149.4
1PGB	-445.6	-306.8
1ROP	-864.1	-295.5
3IL8	-678.7	-621.4
3CHY	-1131.8	-499.0
153L	-1839.4	-1090.1
1TIM	-1168.9	470.9

All numerical experiments were performed using CCEMD [56], a general purpose molecular mechanics program developed at Sandia National Laboratories. Among other features, CCEMD implements the CHARMM22 [1, 2] force field model and accepts PDB input files (XPLOR or CHARMM formats). CCEMD is written in C, and was executed on a Silicon Graphics workstation (180 MHz MIPS R5000 processor, 128 Mbytes RAM).

DISTRIBUTION:

1	MS 9402	T. O. Hunter	8000
		Attn: J. B. Wright	2200
		J. F. Ney (A)	5200
		M. E. John	8100
		L. A. West	8200
		W. J. McLean	8300
		R. C. Wayne	8400
		P. N. Smith	8500
		P. E. Brewer	8600
		T. M. Dyer	8700
		L. A. Hiles	8800
1	MS 9003	D. R. Henson	8909
1	MS 9011	J. C. Meza	8950
1	MS 9214	L. M. Napolitano	8130
40	MS 9201	T. D. Plantenga	8950
1	MS 1111	S. J. Plimpton	9221
1	MS 1111	J-L. M. Faulon	9225
1	MS 1110	W. E. Hart	9222
1	MS 9214	C. F. Melius	8130
1	MS 0899	Technical Library, 4916	
3	MS 9018	Central Technical Files, 8940-2	
1	MS 9021	Technical Communications Department, 8815/Technical Library, MS 0899, 4916	
2	MS 9021	Technical Communications Department, 8815, for DOE/OSTI	

This page intentionally left blank