

An Efficient Newton-like Method for Molecular Mechanics Energy Minimization of Large Molecules

Jay W. Ponder and Frederic M. Richards

Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut 06511

Received 16 September 1986; accepted 19 February 1987

Techniques from numerical analysis and crystallographic refinement have been combined to produce a variant of the Truncated Newton nonlinear optimization procedure. The new algorithm shows particular promise for potential energy minimization of large molecular systems. Usual implementations of Newton's method require storage space proportional to the number of atoms squared (i.e., $O(N^2)$) and computer time of $O(N^3)$. Our suggested implementation of the Truncated Newton technique requires storage of less than $O(N^{1.5})$ and CPU time of less than $O(N^2)$ for structures containing several hundred to a few thousand atoms. The algorithm exhibits quadratic convergence near the minimum and is also very tolerant of poor initial structures. A comparison with existing optimization procedures is detailed for cyclohexane, arachidonic acid, and the small protein crambin. In particular, a structure for crambin (662 atoms) has been refined to an RMS gradient of 3.6×10^{-6} kcal/mol/Å per atom on the MM2 potential energy surface. Several suggestions are made which may lead to further improvement of the new method.

INTRODUCTION

One of the major advantages of molecular mechanics compared to other computational techniques is the relative ease with which structures can be optimized via minimization of the corresponding potential energy functions.¹ A number of sophisticated methods are available which permit efficient minimization of potential energy for small molecular systems. Atom-by-atom block diagonal approximations of Newton's method are tolerant of poor initial geometries and quickly convergent, often to within 0.1 kcal/mol or less of a local energy minimum. Quadratic convergence to the exact minimum structure and energy, limited only by machine tolerance, can be achieved with full matrix Newton methods implemented via generalized matrix inversion² or imposition of Eckart constraints.³

With the tremendous advances in computer processing speed realized over the past decade, molecular mechanics computations on small to moderate sized proteins and comparably sized nucleic acids have become almost routine. Several program packages capable of performing such calculations have been reported.⁴⁻¹¹ All of these programs make

available one or more methods for energy minimization. Unlike the small molecule case, energy minimization of a large biopolymer can tax the resources of even the largest current computers. In particular, full matrix Newton methods are limited to systems of less than a few hundred atoms since standard implementations of these methods require storage proportional to the square of the number of atoms, $O(N^2)$, and CPU time of $O(N^3)$.

Most reported macromolecular energy minimizations make use of nonlinear conjugate gradient, variable metric (ECEPP¹¹ and Levitt's torsional angle optimization⁶) or limited memory Newton techniques (the ABNR method available in CHARMM⁵). These methods exhibit at best superlinear convergence (often only linear) and typically require many hundreds or thousands of iterations to reduce the RMS energy gradient per atom into the 0.1 to 0.01 kcal/mol/Å range.

In this article we present a variant of Newton's method adapted specifically for full Cartesian coordinate energy minimization of large molecules. The method is based on the Truncated Newton formalism elaborated recently by several numerical analysis groups. Alterations borrowed from the sparse matrix

techniques used in crystallographic refinement serve to greatly reduce total computational requirements.

ALGORITHM

The numerical analysis literature on Truncated Newton¹²⁻¹⁵ methods and particularly the preconditioned conjugate gradient¹⁶ (PCG) approach to linear equations is extensive. For convenience, we provide an algorithm for the basic method. Computational results and several possible extensions are discussed in subsequent sections. We assume access to a set of callable subroutines which, given the Cartesian coordinates x_i ($i = 1, 3N$), compute the potential energy (E), gradient vector (g , where $g_i = dE/dx_i$), and Hessian matrix (H , where $H_{ij} = d^2E/dx_i dx_j$). A Truncated Newton optimization using a sparse Hessian approximation and a preconditioned linear conjugate gradient method for the iterative solution of the resulting Newton equations consists of the following steps:

Truncated Newton Method

(1) Begin a Truncated Newton cycle by evaluating the current potential energy and gradient. If convergence criteria based upon these values have been met, then we are done. For small molecules ($N < 100$) we typically require the RMS gradient per atom to be 10^{-4} to 10^{-6} kcal/mol/Å or less at convergence. For larger structures, values of 10^{-3} to 10^{-4} are routinely obtained.

(2) Evaluate the Hessian matrix. For large systems where storage of the full Hessian is impossible, a sparse approximation (which we will still call H) is computed storing only elements on the diagonal or in the upper triangle of the full Hessian which have absolute value greater than some cutoff tolerance. Data structures needed for sparse matrix storage can be found in the literature.¹⁷

(3) Approximately solve the Newton equations: $Hp = -g$, where the vector p is a search direction that will (hopefully) lead to reduction of the current energy. These linear equations can be solved by a number of methods. The Truncated Newton idea is to use an iterative method to obtain a partial solution since exact solution is an $O(N^3)$ process and

quite unnecessary except in the immediate vicinity of the minimum. We currently use the preconditioned linear conjugate gradient method outlined below, although other iterative methods have been suggested.

(4) Use a line search technique to minimize the energy in the search direction P . Our current algorithm is a simple parabolic extrapolation/cubic interpolation procedure which uses both energy and gradient evaluations along the search direction. The search is terminated when the magnitude of the projection of the gradient on the search direction has been reduced by half from its value at the start of the current line search. Any number of alternative line search methods could be used since the overall optimization time is not very sensitive to the efficiency of this step.

(5) Increment the number of Truncated Newton cycles performed and set the current coordinates to be those located during the line search. Compute the RMS change in atom positions and any other quantities of interest. Return to step 1 to begin the next cycle.

As outlined above, the centerpiece of the Truncated Newton optimizer is its partial solution of the Newton equations. The algorithm is outlined below:

Preconditioned Linear Conjugate Gradient Method (PCG)

(1) Transform the Newton equations by scaling the gradient and Hessian with the exact Hessian diagonal. Create a diagonal scaling vector b such that $b_i = (\text{abs}(H_{ii}))^{0.5}$, then set $g_i = g_i/b_i$, and $H_{ij} = H_{ij}/(b_i b_j)$ for $i, j = 1$ to $3N$. Without this transformation, the Truncated Newton cycles very near the minimum were often observed to exhibit an oscillatory behavior.

(2) Initialize various vector and scalar variables prior to beginning conjugate gradient iterations. Compute the Euclidian lengths of the gradient vector, g -norm and the RMS gradient per atom, g -rms. Set the convergence tolerance epsilon to the minimum of g -rms and $1/\text{cycle}$, where cycle is the number of the current Truncated Newton cycle (see algorithm above). Set $p_i = 0$ and $r_i = g_i$ for $i = 1$ to $3N$.

(3) Solve the linear equations $Ms = r$ for the vector s , where M is some (simple) preconditioning matrix which approximates H . The

equations are solved directly via (incomplete) Cholesky factorization followed by substitution. One particularly simple choice is to set M to the identity matrix in which case we get $s = r$. Several other, much more efficient, choices for M are discussed later in this article.

(4) Set $d_i = s_i$ for $i = 1$ to $3N$ and equate rs -old with the dot product of the r and s vectors.

(5) Begin a conjugate gradient iteration by computing the vector q where $q = Hd$. If a sparse approximation to H is being used it is imperative to use that sparsity to compute q as efficiently as possible since for large molecules formation of this vector-matrix product is the rate limiting step in the PCG method.

(6) Set dq to the dot product of the d and q vectors. If dq is less than zero then negative curvature has been detected in the transformed Hessian matrix and we exit the PCG algorithm with the current unscaled search direction $p_i = p_i/b_i$ for $i = 1$ to $3N$. If dq is less than zero on the first conjugate gradient iteration then choose $p_i = d_i/b_i$ for $i = 1$ to $3N$.

(7) Set the scalar t equal to rs -old/ dq . Set $p_i = p_i + (t)d_i$ and $r_i = r_i - (t)q_i$ for $i = 1$ to $3N$.

(8) Compute r -norm, the Euclidian length of the r vector. If r -norm/ g -norm is less than epsilon, we exit the PCG algorithm with the current unscaled search direction $p_i = p_i/b_i$. This is the normal termination.

(9) Solve the new set of preconditioning equations $Ms = r$. Note that r has just been updated, but the matrix M need not change. Thus any factorization of M from step 3 can be reused here.

(10) Compute rs -new, the dot product of the current r and s vectors. Set $u = rs$ -new/ rs -old, then $d_i = s_i + (u)d_i$ for $i = 1$ to $3N$. Set rs -old = rs -new.

(11) If we have performed the maximum allowed number of conjugate gradient iterations, then exit the PCG algorithm with the current unscaled search direction $p_i = p_i/b_i$ for $i = 1$ to $3N$. Otherwise, increment the iteration counter and return to step 5 to begin the next iteration. Our current program allows a maximum of $10 \times (3N)^{0.5}$ PCG iterations per Truncated Newton cycle. In most cases far fewer than the allowed number of iterations are actually required to reach the desired convergence.

COMPUTATIONAL

In order to test the efficacy of the Truncated Newton method and compare it with other optimization methods, an explicit potential function and several test cases are required. We have developed a compact, modular set of potential energy routines, YALIE, which is easily interfaced to a number of different nonlinear optimizers. The present program implements an expanded and slightly modified version of the MM2 potential surface.¹⁸ Electrostatic interactions are modelled as a sum of dipole-dipole, charge-dipole and charge-charge terms. Various subroutines are available to evaluate the energy, gradient and Hessian in any combination.¹⁹ CPU times required for sample evaluations are shown in Table I. In general, gradient evaluations are slightly less than twice as costly as computation of the potential energy. Hessian evaluation requires CPU time equivalent to about 15 gradient calls for small structures decreasing to a factor of 10 or less as the number of atoms becomes large. Two comments are required with regard to the reported times. First, van der Waals and electrostatic interactions are computed from a pairwise search rather than neighbor list or grid partitioning techniques,²⁰ since we routinely use infinite cutoffs for such interactions. Secondly, the Hessian matrix is computed one atom at a time and only significant elements are retained for storage. This introduces an inefficiency of a factor of 2 to 4 into the Hessian CPU times compared to more direct methods, but permits Hessian computation on very large systems without the requirement for a large amount of computer memory. Table II indicates the number of Hessian elements stored as a function of various cutoff values. Since the largest Hessian elements for a typical molecule will have a magnitude of 1000 to 3000 kcal/mol Å², a Hessian cutoff of 0.01 would include elements five orders of magnitude smaller than the largest element.

Results from several different optimization techniques are reported in the next section. The standard Fletcher-Reeves nonlinear conjugate gradient method²¹ can be made self-restarting and less dependent on line search accuracy if the update formula is derived from a memoryless BFGS quasi-Newton update. Our scaled version of this method will be referred to as MQN.²²

Table I. CPU times required for potential energy functions^a.

Molecule	Atoms	Energy	Gradient	Hessian
Cyclohexane	18	0.09	0.18	3.00
Arachidonic Acid	56	0.69	1.26	14.62
Pentapeptide ^b	85	1.80	3.21	35.73
Gramacidin A	278	17.0	27.9	258.5
Crambin	662	94	164	1590
Lysozyme	2025	735	1369	13726
Crambin ^c	662	45	72	781
Lysozyme ^c	2025	308	436	4316

^aAll times are in CPU seconds on a VAX 11/750 with 3 Mbytes of memory and a floating point accelerator running under VMS Verison 4.2.

^bALA-PRO-TRP-MET-GLN in a turn conformation.

^cVan der Waals interactions were cutoff at 8 Å; dipole-dipole interactions were cutoff at 12 Å.

Table II. Hessian matrix storage^a.

Molecule	Value of Hessian Cutoff: ^b			
	0.0	0.01	0.1	1.0
	Number of Matrix Elements above Cutoff			
Cyclohexane	2916	2422	1984	1372
Arachidonic Acid	28224	7794	5932	4090
Pentapeptide	65025	20651	12299	7441
Gramacidin A	695556	90328	41670	23678
Crambin	3944196	392265	122328	57177
Lysozyme	36905625	1570559	403157	174549

^aRequirements for the full matrix; in practice only the diagonal and upper triangle elements need to be stored.

^bOnly those elements whose magnitudes are greater than the listed cutoffs are accounted; matrix elements have units of kcal/mol/Å².

Quasi-Newton (variable metric) methods are usually preferred over nonlinear conjugate gradients when space is available to store an approximation to the inverse Hessian. One particularly efficient version is the optimally conditioned method without line searches devised by Davidon.²³ The economy of this technique is due to the fact that most iterations require only one function/gradient evaluation. This method and the related VA09D routine from the NAG Library have already seen use in chemical applications. Our implementation will be referred to as DAVIDON.

The Truncated Newton method with linear conjugate gradient solution of the Newton equations (TNCG) was tested without scaling or preconditioning (TNCG-N), with only diagonal scaling (TNCG-D), with both diagonal scaling and 3-by-3 block diagonal preconditioning (TNCG-B), and with symmetric successive over-relaxation preconditioning²⁴ (TNCG-S).

In addition, a standard full matrix Newton method (NEWTON) was simulated by using the Truncated Newton formalism, but with an exact solution of the Newton equations at each cycle. A 3-by-3 block diagonal Newton method (BLOCK NEWTON) was achieved by passing a Hessian matrix containing only the block diagonal elements to the NEWTON method just described. Finally, the one atom at a time 3-by-3 block diagonal method used by the standard MM2 program was tested.

Three test problems were selected for investigation: (1) cyclohexane starting from a model with standard bond lengths and angles and +60/-60/+60 degree dihedral angles around the ring, (2) arachidonic acid starting from standard bond lengths and angles, +90 degree dihedral angles around all *sp*²-*sp*³ bonds and 180 degree dihedrals around all *sp*³-*sp*³ bonds, and (3) the 46-residue protein crambin starting from Protein Data Bank coordinates with hydrogen atoms added in idealized positions.^{25,26} All calculations involving

the potential functions and optimization methods were performed in double precision arithmetic on a VAX 11/750 computer.

RESULTS

The small cyclohexane test case is easily solved by all optimization methods. Statistical summaries of the minimizations are shown in Table III. It is clear even for this simple problem that block diagonal Newton methods are much closer to conjugate gradient (MQN) and quasi-Newton (DAVIDON) methods in terms of the total number of algorithm cycles required. In contrast, all variants of the TNCG procedure exhibit quadratic or near-quadratic convergence.

Despite its moderate size, 56 "atoms" including two MM2 lone pairs, the arachidonic acid problem proved to be quite difficult. The initial structure is an extended conformation and lies on an almost flat portion of the potential surface. Thus even the Newton-based methods tend to wander somewhat before locating a productive minimum. All of the optimizers do however eventually converge upon the same final structure. As shown in Table IV, the MQN conjugate gradient technique exceeds 7500 cycles and 13000 energy/

gradient evaluations before reaching the requested RMS gradient convergence of 10^{-4} kcal/mol/Å per atom. This is typical of the action of linearly convergent algorithms on difficult problems. The DAVIDON method exhibits superlinear convergence on the arachidonic acid test problem while needing only slightly more than one energy evaluation per iteration.

The 3-by-3 block diagonal methods perform poorly on the arachidonic acid test. The BLOCK NEWTON technique barely gets with 1 kcal/mol of the correct minimum energy in 100 cycles. The behavior of the original MM2 algorithm is only very slightly better; it reaches a potential energy of 13.2 kcal/mol after 100 iterations. In addition, convergence of the block diagonal methods is even slower than with MQN. At 1672 iterations the MM2 method results in an energy decrease of less than 10^{-5} kcal/mol over the previous 5 iterations. At this point, the energy was still almost 0.1 kcal/mol above the exact minimum and the RMS deviation in atom positions between the MM2 structure and the exact minimum was 0.29 Å.

Our full matrix Newton variant (NEWTON) converged to an RMS gradient under 10^{-4} kcal/mol/Å per atom in 22 cycles. Re-

Table III. Comparison of optimization methods for cyclohexane.

Method	Algorithm Iterations	Evaluations Ener/Grad	Hess	Hessian Cutoff	PCG Cycles	Final Energy	RMS Gradient
MQN	43	76	—	—	—	6.5510	$9.9 \cdot 10^{-5}$
DAVIDON	34	54/35	—	—	—	6.5510	$8.3 \cdot 10^{-5}$
Block Newton	40	63	(40)	—	—	6.5510	$1.3 \cdot 10^{-5}$
MM2 ^a	16	(16)	(16)	—	—	6.5510	$1.1 \cdot 10^{-3}$
Newton	3	7	3	0.0	136	6.5510	$5.0 \cdot 10^{-5}$
TNCG-N	6	11	6	0.0	30	6.5510	$3.3 \cdot 10^{-5}$
TNCG-D	5	10	5	0.0	72	6.5510	$3.7 \cdot 10^{-8}$
TNCG-B	5	10	5	0.0	37	6.5510	$1.4 \cdot 10^{-8}$
TNCG-S	5	10	5	0.0	40	6.5510	$7.5 \cdot 10^{-6}$
TNCG-N	6	11	6	0.01	30	6.5510	$2.9 \cdot 10^{-5}$
TNCG-D	5	10	5	0.01	76	6.5510	$3.3 \cdot 10^{-8}$
TNCG-B	5	10	5	0.01	44	6.5510	$7.1 \cdot 10^{-7}$
TNCG-S	5	10	5	0.01	42	6.5510	$9.2 \cdot 10^{-6}$
TNCG-N	6	11	6	0.1	33	6.5510	$2.8 \cdot 10^{-5}$
TNCG-D	5	10	5	0.1	93	6.5510	$9.8 \cdot 10^{-6}$
TNCG-B	5	10	5	0.1	57	6.5510	$9.3 \cdot 10^{-6}$
TNCG-S	5	13	5	0.1	46	6.5510	$8.3 \cdot 10^{-5}$
TNCG-N	9	21	9	1.0	182	6.5510	$6.1 \cdot 10^{-5}$
TNCG-D	6	12	6	1.0	163	6.5510	$4.9 \cdot 10^{-5}$
TNCG-B	6	12	6	1.0	119	6.5510	$5.1 \cdot 10^{-5}$
TNCG-S	9	16	9	1.0	141	6.5510	$5.3 \cdot 10^{-5}$

^aThe standard MM2 program was continued until a potential energy of 6.5510 kcal/mol was achieved, this criterion is less restrictive than the RMS gradient of 10^{-4} kcal/mol/Å per atom required of all other minimizations.

Table IV. Comparison of optimization methods for arachidonic acid.

Method	Algorithm Iterations	Evaluations		Hessian Cutoff	PCG Cycles	Final Energy	RMS Gradient
		Ener/Grad	Hess				
MQN	500	769	—	—	—	12.5384	9.4×10^{-2}
	1000	1520	—	—	—	12.4629	4.4×10^{-2}
	2500	3820	—	—	—	12.4495	3.7×10^{-3}
	7887	13754	—	—	—	12.4469	8.0×10^{-5}
DAVIDON	679	836/680	—	—	—	12.4469	6.6×10^{-5}
Block							
Newton	100	193	(100)	—	—	13.3705	1.2×10^{-1}
MM2 ^a	1672	(1672)	(1672)	—	—	12.5318	3.5×10^{-2}
Newton	22	33	22	0.0	3378	12.4469	5.9×10^{-5}
TNCG-N	35	51	35	0.0	3244	12.4469	6.8×10^{-5}
TNCG-D	29	41	29	0.0	3111	12.4469	7.9×10^{-6}
TNCG-B	17	25	17	0.0	1671	12.4469	5.6×10^{-5}
TNCG-S	16	26	16	0.0	1185	12.4469	2.9×10^{-5}
TNCG-N	33	58	33	0.01	2921	12.4469	9.8×10^{-5}
TNCG-D	28	35	28	0.01	3137	12.4469	5.9×10^{-5}
TNCG-B	20	34	20	0.01	2103	12.4469	7.4×10^{-5}
TNCG-S	29	54	29	0.01	2332	12.4469	8.8×10^{-5}
TNCG-N	43	69	43	0.1	4290	12.4469	8.4×10^{-5}
TNCG-D	32	47	32	0.1	3812	12.4469	9.5×10^{-5}
TNCG-B	38	76	38	0.1	3994	12.4469	3.3×10^{-5}
TNCG-S	42	76	42	0.1	3303	12.4469	9.9×10^{-5}
TNCG-N	100	168	100	1.0	6581	12.4472	1.1×10^{-2}
TNCG-D	100	178	100	1.0	5874	12.4472	1.8×10^{-2}
TNCG-B	100	182	100	1.0	4654	12.4473	9.1×10^{-3}
TNCG-S	100	171	100	1.0	2413	12.4471	4.8×10^{-3}

^aThe standard MM2 program was continued until the energy decrease over 5 iterations was less than 10^{-5} kcal/mol.

sults for the Truncated Newton algorithm show it to be very similar to the full matrix Newton in terms of cycles required and final gradient values. In fact, the best performance of any method is achieved by TNCG-S using the exact Hessian. That Truncated Newton, an approximation to full matrix Newton methods, should outperform the NEWTON routine is perhaps surprising. Similar behavior has been observed on other classes of numerical test problems.¹⁵ The initial steps of a Truncated Newton method interpolate between the exact Newton direction and the steepest descent direction for the transformed equations submitted to preconditioned linear conjugate gradient solution. In the case of relatively poor initial geometries, the truncated solution to the Newton equations will often result in a better search direction than the exact solution. We recall that a full Newton treatment will often diverge from an arbitrary structure while methods closer to steepest descent are much more tolerant of grossly incorrect initial geometries.

It is clear from inspection of Table IV that use of a Hessian cutoff degrades the resulting

TNCG method. However, it is important in the case of larger structures to choose the biggest possible cutoff that still affords the desired degree and rate of convergence. Increasing the size of the cutoff reduces the total amount of storage (Table II) and speeds the linear conjugate gradient iterations. The results for arachidonic acid (and many other test cases not reported here) indicate that a cutoff in the 0.1 to 0.01 kcal/mol/Å² range is often a good compromise. Use of a Hessian cutoff as large as 1.0 appears to severely damage performance of the algorithm. Finally, we note that different variants of TNCG seem to react differently to use of a non-zero cutoff. In particular, TNCG-B or TNCG-S is the preferred method when the full Hessian is retained. However, the performance of TNCG-D is degraded much more slowly than TNCG-B or TNCG-S by a cutoff in the range mentioned above. Thus, we currently use the TNCG-D method with a moderate Hessian cutoff for problems containing over a few hundred atoms.

The performance degradation observed with the use of large Hessian cutoffs can

be partially alleviated by replacing the vector-matrix product in step 5 of the PCG algorithm with a finite difference approximation.¹² Since this approximation requires a gradient evaluation at each PCG cycle, its higher time cost per Truncated Newton iteration will be justified only if an extremely precise answer is required (i.e., an RMS gradient per atom smaller than those reported in Tables III–V).

Results for the small protein crambin are provided in Table V. Since the original parameterization of the MM2 potential surface did not provide for amides, several small extensions have been made to enable peptide and protein computations. The TNCG-D method with a Hessian cutoff varied from 10^{-2} (first 39 cycles) to 10^{-4} (last 5 cycles) is able to reach an RMS gradient of 3.6×10^{-6} kcal/mol/Å per atom. This is the greatest reported degree of convergence for any Cartesian coordinate protein optimization. The energy at the minimum is determined to a precision of at least 0.0001 kcal/mol. The conjugate gradient MQN method converges within a few hundred iterations to a potential energy value 10 to 15 kcal above the local minimum. After 1200 cycles, the MQN energy is still nearly 2 kcal/mol above the TNCG-D minimum energy. Further minimization of the 1200 iteration MQN structure with the TNCG-D optimizer results in the same minimum as TNCG-D optimization of the initial x-ray coordinates. After 600 MQN iterations the RMS deviation in atom positions between the MQN structure and the exact minimum is 0.47 Å; at 1200 MQN cycles the deviation is reduced to 0.28 Å. Both the full matrix NEWTON and the quasi-Newton DAVIDON method are too expensive to apply

to the crambin problem. Block diagonal methods were not attempted in light of their poor performance on the arachidonic acid test case. In fact, the complete TNCG-D trial with a Hessian cutoff of 0.01 kcal/mol/Å required less total CPU time than the first 600 iterations of the MQN trial. An increase of the Hessian cutoff from 0.01 to 0.1 does degrade convergence to a small extent. However, use of the latter cutoff still results in convergence to within 0.1 kcal/mol of the exact minimum and yields a correspondingly small deviation in atom positions.

DISCUSSION

Use of the Truncated Newton formalism significantly reduces both the storage and computer time necessary for complete convergence. For proteins in the size range of crambin or lysozyme, use of a Hessian cutoff of 0.01 kcal/mol/Å² requires storage roughly proportional to $N^{1.5}$. A larger Hessian cutoff value of 0.1 further reduces the storage growth rate. Similar sparse matrix techniques coupled with linear conjugate gradient solution of the normal equations arising during least squares fitting have been used for crystallographic refinement.^{27,28} In this setting, approximately 1% of the corresponding matrix is retained to assure convergence for protein structures.

Overhead for the Truncated Newton procedure is dominated by the preconditioned linear conjugate gradient step. The PCG algorithm requires several scalar-vector products and vector dot products each involving $3N$ flops.²⁹ However, for large N the total PCG cost is dominated by the formation of the vector-matrix product in step 5. This product

Table V. Comparison of optimization methods for crambin.

Method	Algorithm Iterations	Evaluations Ener/Grad	Hess	Hessian Cutoff	PCG Cycles	Final Energy	RMS Gradient
MQN	300	467	—	—	—	-549.8590	9.9×10^{-1}
	600	993	—	—	—	-557.4671	9.2×10^{-2}
	900	1403	—	—	—	-561.6985	8.7×10^{-2}
	1200	1881	—	—	—	-562.4097	9.1×10^{-2}
TNCG-D	44	88	44	0.01 to 10^{-4}	7500	-564.2269	3.6×10^{-6}
TNCG-D	44	89	44	0.01	6402	-564.2264	6.5×10^{-3}
	58	119	58	0.01	11678	-564.2269	3.0×10^{-3}
TNCG-D	54	128	54	0.1	6993	-564.1116	1.3×10^{-1}
	67	152	67	0.1	10462	-564.1851	9.5×10^{-2}

needs a number of flops equal to the number of non-zero Hessian matrix elements. Since the number of elements is $O(N^{1.5})$ or less for large N and moderate Hessian cutoff, each PCG iteration should require $O(N^{1.5})$ flops. Since a properly preconditioned linear conjugate gradient method is reported to average $O(N^{0.5})$ iterations for convergence,¹⁶ the total overhead for the Truncated Newton cycles should be $O(N^2)$ as a worst case. In fact, many early cycles will require very few PCG iterations and significantly better the above analysis.

The ability of the Truncated Newton procedure to completely converge to a local minimum may be of assistance in comparing macromolecular potential energy functions. For example, the RMS deviation of the fully optimized crambin structure from the initial x-ray coordinates is 0.71 Å for all non-hydrogen atoms. After 600 MQN nonlinear conjugate gradient iterations, the corresponding deviation is only 0.47 Å while the RMS gradient is consistently below 0.1 kcal/mol/Å per atom and near 0.01 on some iterations. Thus, energy minimization results based upon convergence at the 0.1 kcal/mol/Å per atom level may significantly underestimate the deviation of the (exact) minimum from the starting structure. This bias is particularly severe for nonlinear conjugate gradient techniques since our experience indicates these methods follow "valley bottoms" much more closely than Newton-based methods and can return low RMS gradients while still far from a local minimum.

Further improvement in the Truncated Newton method will most easily be achieved by reducing the number of PCG iterations used when very close to the minimum. A preconditioning matrix M which is close to the full Hessian matrix H will reduce the number of PCG iterations. Note that use of $M = H$ would produce convergence in only one iteration at the expense of making solution of the preconditioning equations a $O(N^3)$ process. One attractive possibility involves use of an incomplete Cholesky factorization³⁰ of the sparse matrix H as the "exact factorization of the preconditioning matrix M (which need not be computed explicitly). This preconditioning greatly improves the PCG algorithm in application to certain classes of partial differential equations.³¹ With appropriate reor-

ganization of the steps in the algorithm outlined in the second section, the incomplete Cholesky factorization can be implemented with only one vector-matrix product per iteration.³²

Another potential improvement would involve setting the initial search direction to something other than the steepest descent direction upon entering the PCG algorithm. One option is to determine the initial direction by means of a nonlinear conjugate gradient or limited memory quasi-Newton update applied to the Truncated Newton cycles (i.e., outside the PCG iterations).¹⁵ An alternate method which also uses information gleaned from the previous Truncated Newton cycle has been suggested.³³ These ideas may be of great assistance very near to the minimum where the steepest descent direction is usually a poor approximation to the Newton direction.

CONCLUSION

The Truncated Newton method presented above appears to be competitive with the best optimization techniques reported for small molecules and superior to those commonly used for energy minimization of large structures. Shifting a large portion of the optimization procedure to solution of the Newton equations and away from repeated potential energy evaluations should encourage experimentation with computationally costly potential functions and ease the necessity to approximate currently used functional forms in the face of limited computer processing power.

Note Added in Proof: Since submission of this article, we have implemented a version of the incomplete Cholesky factorization mentioned above [see T. A. Manteuffel, *Math. Comp.*, **34** 473 (1980) for details]. This preconditioning method can greatly reduce the number of PCG iterations at the expense of additional preconditioning work. A current version of the VAX Fortran source code, including the incomplete Cholesky option, is available from the authors.

References

1. U. Burkert and N. L. Allinger, *Molecular Mechanics*, American Chemical Society, Washington, D.C., 1982, Ch. 3.
2. D. N. J. White, *Acta Cryst.*, **A33**, 1010 (1977).
3. B. van de Graaf and J. M. A. Baas, *J. Comput. Chem.*, **5**, 314 (1984).
4. S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, *J. Comput. Chem.*, **7**, 230 (1986).
5. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Comput. Chem.*, **4**, 187 (1983).
6. M. Levitt, *J. Mol. Biol.*, **170**, 723 (1983).
7. D. H. J. Mackay, P. H. Berens, K. R. Wilson, and A. T. Hagler, *Biophys. J.*, **46**, 229 (1984).
8. S. J. Wodak, P. Alard, P. Delhaise, and C. Renneboog-Squilbin, *J. Mol. Biol.*, **181**, 317 (1984).
9. A. Vedani and J. D. Dunitz, *J. Am. Chem. Soc.*, **107**, 7653 (1985).
10. T. Oie, G. M. Maggiora, R. E. Christoffersen, and D. J. Duchamp, *Int. J. Quant. Chem., Quant. Biol. Symp.*, **8**, 1 (1981).
11. G. Nemethy, M. S. Pottle, and H. A. Scheraga, *J. Phys. Chem.*, **87**, 1883 (1983).
12. R. S. Dembo and T. Steihaug, *Mathematical Programming*, **26**, 190 (1983).
13. R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *SIAM J. Numer. Anal.*, **19**, 400 (1982).
14. S. G. Nash, *SIAM J. Numer. Anal.*, **21**, 770 (1984).
15. S. G. Nash, *SIAM J. Sci. Stat. Comput.*, **6**, 599 (1985).
16. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, 1983, Ch. 10.
17. S. C. Eisenstat, M. H. Schultz, and A. H. Sherman, *SIAM J. Sci. Stat. Comput.*, **2**, 225 (1977).
18. N. L. Allinger, *J. Am. Chem. Soc.*, **99**, 8127 (1977).
19. Computer codes implementing both the Truncated Newton method and the MM2 potential surface in a set of callable energy, gradient and Hessian subroutines will be deposited with the Quantum Chemistry Program Exchange.
20. W. F. van Gunsteren, H. J. C. Berendsen, F. Colonna, D. Perahia, J. P. Hollenberg, and D. Lellouch, *J. Comput. Chem.*, **5**, 272 (1984).
21. R. Fletcher and C. M. Reeves, *Computer J.*, **6**, 163 (1964).
22. D. G. Luenberger, *Linear and Nonlinear Programming, 2nd Ed.*, Addison-Wesley, Reading, MA, 1984, section 9-7.
23. W. C. Davidon, *Mathematical Programming*, **9**, 1 (1975).
24. T. F. Chan and K. R. Jackson, *SIAM J. Sci. Stat. Comput.*, **5**, 533 (1984).
25. W. A. Hendrickson and M. M. Teeter, *Nature (London)*, **290**, 107 (1981).
26. F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, *J. Mol. Biol.*, **112**, 535 (1977).
27. J. H. Konnert, *Acta Cryst.*, **A32**, 614 (1976).
28. W. A. Hendrickson and J. H. Konnert, in *Biomolecular Structure, Function and Evolution*, R. Srinivasan, Ed., Pergamon, Oxford, 1981, p. 43.
29. Following the concept of C. B. Moler, we define a "flop" as roughly the amount of work associated with the statement: $x = x + a_{ik}b_{kj}$; a floating point add, a floating point multiply and associated subscripting.
30. J. A. Meijerink and H. A. Van der Vorst, *Math. Comp.*, **31**, 148 (1977).
31. D. S. Kershaw, *J. Comput. Phys.*, **26**, 43 (1978).
32. S. C. Eisenstat, *SIAM J. Sci. Stat. Comput.*, **2**, 1 (1981).
33. D. P. O'Leary, *Mathematical Programming*, **23**, 20 (1982).