Acta Cryst. (1999). A**55**, 700–703

The efficient calculation of the normal matrix in least-squares refinement of macromolecular structures

Dale E. Tronrud

Howard Hughes Medical Institute and Institute of Molecular Biology, University of Oregon, Eugene, OR 97403, USA. E-mail: dale@uoxray.uoregon.edu

(*Received 31 July 1998; accepted 24 December 1998*)

Abstract

The optimization procedure with the greatest power of convergence is the full-matrix method. This method has not been utilized to a great extent in macromolecular refinement because of the great cost of both calculating and inverting the 'normal' matrix. This paper describes an algorithm that can calculate this matrix in a relatively short amount of computer time. The procedure requires two Fourier transforms, which can be performed with the fast-Fourier transformation (FFT) algorithm, as well as a large number of simple function products.

1. Introduction

Least-squares optimization is the procedure whereby values for the parameters of a model are determined by minimizing the residual function

$$f(\mathbf{p}) = \sum_i^{\rm asu} W(i)[Q_o(i) - Q_c(i,\mathbf{p})]^2. \qquad (1)$$

$Q_o(i)$ is the value for observation number i, $Q_c(i,\mathbf{p})$ is the model's prediction for observation i using the set of model parameters $\mathbf{p}$ and $W(i)$ is some weighting function.

The values of the parameters that produce the best fit to the data are found by expanding $f(\mathbf{p})$ into a Taylor series in the neighborhood of $\mathbf{p}_0$, calculating the first derivative with respect to each parameter and solving the resulting set of equations where these derivatives are equal to zero. Solving the resulting equation is the full-matrix method of minimization:

$$\mathbf{p} = \mathbf{p}_0 - |{\rm d}^2 f(\mathbf{p})/{\rm d}\mathbf{p}^2|_{\mathbf{p}_0}^{-1} |{\rm d}f(\mathbf{p})/{\rm d}\mathbf{p}|_{\mathbf{p}_0}. \qquad (2)$$

The matrix

$$|{\rm d}^2 f(\mathbf{p})/{\rm d}\mathbf{p}^2|_{\mathbf{p}_0} \qquad (3)$$

is usually called the 'normal' matrix. This matrix can be huge, having N^2 elements, where N is the number of parameters in the model. For a typical protein containing 2500 atoms each with four parameters, the normal matrix contains 10 000 × 10 000 or 100 000 000 elements. A protein such as betagalatosidase (Jacobson *et al.*, 1994), which contains 130 000 atoms in the asymmetric unit, with each atom having three parameters, would require a normal matrix with 152 100 000 000 elements. Obviously, the calculation of such a matrix is a sizable problem.

Because of the prohibitive time requirements, the refinement packages most commonly used for macromolecular refinement – *PROLSQ* (Hendrickson & Konnert, 1980), *TNT* (Tronrud *et al.*, 1987), *X-PLOR* (Brünger *et al.*, 1987) and *REFMAC* (Murshudov *et al.*, 1997) – each use methods of minimization that do not require the complete normal matrix. While the various methods implemented are computationally efficient and, for poor initial models, can be superior to the full-matrix method because of their larger radius of convergence, one would expect that the 'final' models produced by these programs could be improved by additional cycles of refinement using the full-matrix method.

While *SHELXL* (Sheldrick & Schneider, 1997) offers the full-matrix method, it has been applied only to smaller proteins because of the prohibitive amounts of computer time required by the calculations for large macromolecules.

This paper describes a means for calculating the normal matrix that is much faster than previous methods. The amount of time required is still significant and the amount of time required for the inversion (or approximation thereof) of this matrix is not reduced by the following method.

2. Derivation

The method used to calculate the first derivatives of the least-squares residual function in most refinement programs uses a short-cut which requires the calculation of the product of a particular function with an $F_o - F_c$ difference map (Agarwal, 1978; Agarwal *et al.*, 1981). Agarwal showed that, given the residual function

$$f = \sum_{\mathbf{h}}^{\rm asu} W(\mathbf{h})[|F_o(\mathbf{h})| - |F_c(\mathbf{h})|]^2, \qquad (4)$$

where

$$\mathbf{h} = hkl, \qquad (5)$$

the gradient function

$$\partial f/\partial \mathbf{p}_i = -2\sum_{\mathbf{h}}^{\text{asu}} W(\mathbf{h})[|F_o(\mathbf{h})| - |F_c(\mathbf{h})|]\partial|F_c(\mathbf{h})|/\partial \mathbf{p}_i \quad (6)$$

is equivalent to the expression

$$\partial f/\partial \mathbf{p}_i = \hat{T}^{-1}[W(\mathbf{h})(|F_o(\mathbf{h})| - |F_c(\mathbf{h})|)\exp i\varphi_c(\mathbf{h})] \underset{\mathbf{r}_i}{\otimes} \hat{T}^{-1}\left[-2V\begin{pmatrix}-2\pi ih\\ -2\pi ik\\ -2\pi il\\ -s^2/4\\ 1/O_i\end{pmatrix} g_i(\mathbf{h})\right]. \quad (7)$$

A number of terms must be defined. $\mathbf{p}_i$ is the vector containing the parameters describing atom i. $\hat{T}^{-1}$ is the inverse Fourier transform operator. $\underset{\mathbf{r}_i}{\otimes}$ denotes a convolution evaluated at the point $\mathbf{r}_i$, the position of the atom. V is the volume of the unit cell. $g_i(\mathbf{h}) = O_i f_i(\mathbf{h})\exp[-(B_i/4)s^2]$, where O_i is the atom occupancy, B_i is its B factor, $f_i(\mathbf{h})$ is the scattering factor for its atom type and s is the resolution of reflection $\mathbf{h}$ in Å^{-1}.

Whereas (6) requires a sum over all the reflections for each parameter, (7) is composed of two parts. The first of these is common to all parameters and the second can be calculated analytically. Because the calculation of the part of the equation that involves the crystallographic data is performed only once, the entire calculation is sped up considerably.

3. The normal matrix

The expression for the second derivative can be transformed in an equivalent fashion. The usual form for the second derivative (after linearization) is

$$\frac{\partial^2 f}{\partial \mathbf{p}_i \partial \mathbf{p}_j} = 2\sum_{\mathbf{h}}^{\text{asu}} W(\mathbf{h})\left(\frac{\partial|F_c(\mathbf{h})|}{\partial \mathbf{p}_i}\right)\left(\frac{\partial|F_c(\mathbf{h})|}{\partial \mathbf{p}_j}\right)^t. \quad (8)$$

The equivalent form is

$$\begin{aligned}\frac{\partial^2 f}{\partial \mathbf{p}_i \partial \mathbf{p}_j} &= V\hat{T}^{-1}[W(\mathbf{h})] \\ &\underset{(\mathbf{r}_i - \mathbf{r}_j)}{\otimes} \hat{T}^{-1}\left[g_i(\mathbf{h})g_j(\mathbf{h})\begin{pmatrix} 4\pi^2h^2 & 4\pi^2hk & 4\pi^2hl & \pi ihs^2/2 & -2\pi ih/O_j \\ 4\pi^2hk & 4\pi^2k^2 & 4\pi^2kl & \pi iks^2/2 & -2\pi ik/O_j \\ 4\pi^2hl & 4\pi^2kl & 4\pi^2l^2 & \pi ils^2/2 & -2\pi il/O_j \\ -\pi ihs^2/2 & -\pi iks^2/2 & -\pi ils^2/2 & s^4/16 & -s^2/(4O_j) \\ 2\pi ih/O_i & 2\pi ik/O_i & 2\pi ik/O_i & -s^2/(4O_i) & 1/(O_iO_j)\end{pmatrix}\right] \\ &- V\hat{T}^{-1}\big[W(\mathbf{h})\exp 2i\varphi_c(\mathbf{h})\big] \\ &\underset{(\mathbf{r}_i + \mathbf{r}_j)}{\otimes} \hat{T}^{-1}\left[g_i(\mathbf{h})g_j(\mathbf{h})\begin{pmatrix} 4\pi^2h^2 & 4\pi^2hk & 4\pi^2hl & -\pi ihs^2/2 & 2\pi ih/O_j \\ 4\pi^2hk & 4\pi^2k^2 & 4\pi^2kl & -\pi iks^2/2 & 2\pi ik/O_j \\ 4\pi^2hl & 4\pi^2kl & 4\pi^2l^2 & -\pi ils^2/2 & 2\pi il/O_j \\ -\pi ihs^2/2 & -\pi iks^2/2 & -\pi ils^2/2 & -s^4/16 & s^2/(4O_j) \\ 2\pi ih/O_i & 2\pi ik/O_i & 2\pi ik/O_i & s^2/(4O_i) & -1/(O_iO_j)\end{pmatrix}\right].\end{aligned} \quad (9)$$

While this expression appears very complicated, each component can be calculated quite efficiently. Each of the two terms begins with a Fourier transform which depends on data common to all the parameters. These two maps need only be calculated once requiring only two FFT's in the entire calculation. The second components in each term are very similar. Most are identical and those that differ differ only in sign. In addition, these Fourier transforms can be calculated analytically (see Appendix *A*).

The organization of the calculation is quite simple. First calculate the asymmetric unit of each of the two maps. The space group of the first map is the same as the Patterson function while the space group of the second map is the 'squared' space group.† Then, for each pair of atoms, i and j, one simply evaluates the 25 convolutions for each term. This results in one block of the normal matrix.

The blocks can be calculated in any order or not calculated at all. If the diagonal of the normal matrix is desired, only the blocks where $i = j$ need be calculated. A sparse matrix is as easy to calculate.

Since this method has the fixed overhead of the two FFTs, it will become less efficient when fewer blocks are needed. There is a break-even point below which it becomes more efficient to use other methods. The new method, however, is so efficient that the break-even point is smaller than the size of the diagonal of the matrix, which is the smallest portion of the matrix one would want to calculate.

A reviewer of this paper noted a similarity between the method described herein and that published by Murshudov *et al.* (1997). Equation (9) can be considered a specialization of their equations (46) and (47) for the least-squares residual.

† Take the Hermann–Mauguin symbol for the crystal's space group. Replace each subscript, i, with $2i$ (mod N) where N is the multiplicity of the screw axis. For example, $P2_1$ becomes $P2$ and $P3_2$ becomes $P3_1$.

4. Speed comparisons

The usual method for comparing the efficiency of two algorithms is to estimate the rate at which the size of the computation grows with an increase in the size of the problem. An algorithm for which the computation time grows more slowly is better. The rate of growth of the computation is expressed in 'big-oh' notation (Knuth, 1973, pp. 104–107). $O(\ldots)$ is usually read as 'on the order of ...'.

The traditional method for calculating the normal matrix requires a loop over all reflections for each pair of parameters in the model. In big-oh notation, the computation increases in size as $O(N^2n)$, where N is the number of parameters in the model and n is the number of reflections. In the proposed method, the calculation is segregated into two FFT's (the O for a three dimensional FFT is $n\ln n/3$) followed by a simple calculation for each pair of parameters (there are N^2 pairs). The speed of the new algorithm is, therefore, $O(N^2 + n\ln n/3)$. When $N^2 \gg n\ln n/3$, as is usually the case, the larger term dominates and the big-oh is approximated as $O(N^2)$. The size of the calculation for the traditional method will increase with the size of the problem faster than the proposed method by roughly a factor equal to the number of structure factors.

5. Limitations

The derivation of the optimized form of the normal-matrix equation involves several assumptions which limit its application. The first is the assumption that $W(\mathbf{h})$ obeys the space-group symmetry and does not depend on the parameter of the model ($\mathbf{p}$). The first part of this assumption will always be correct. The latter part will be violated by several weighting schemes including maximum likelihood (Bricogne, 1993). These types of target function can be accommodated by repeating the derivation of the optimized form and deriving different coefficients for the two maps.

This method calculates the normal matrix for the atoms as though the space group were $P1$. This assumption requires appropriate modification when calculating normal-matrix elements involving atoms occupying special positions in the true space group. If an atom overlaps a symmetry image of itself, one must use the chain rule to derive the corrected normal matrix element.

The derivation further assumes that Friedel's law holds. This method cannot be used when the diffraction data contain anomalous scattering. A partition of the data to $(F_+ + F_-)$ and $(F_+ - F_-)$ would allow the new method to be used for the averaged structure factors and most of the parameters, while the normal matrix for the fit to the Bijvoet differences could be calculated using the traditional method for the parameters of the anomalous scatterers alone.

APPENDIX *A*
Analytical Fourier transforms

The calculation of the normal matrix using (9) requires the evaluation of a number of analytical Fourier transforms. The results of these transforms are here listed.

In these equations, the scattering factors, $f(\mathbf{h})$, are modeled as a sum of Gaussians. This model results in the equation for $g_i(\mathbf{h})$ becoming

$$g_i(\mathbf{h}) = O_i \exp[-(B_i/4)s^2] \sum_{l=1}^{3} a_l \exp[-(b_l/4)s^2]. \quad (10)$$

The number of Gaussians summed in this equation is arbitrarily set to three, which is adequate for data to 1.5 Å resolution. It can be increased as required by the resolution of the diffraction data being fit.

The matrix $\mathbf{G}$ is the metric tensor for the coordinate system.

$$\begin{aligned}\hat{T}^{-1}&[4\pi^2\mathbf{hh}^t g_i(\mathbf{h})g_j(\mathbf{h})]\\ &= O_iO_j \sum_{l=1}^{3}\sum_{m=1}^{3} a_la_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times[4\pi^2/(B_i+B_j+b_l+b_m)]\\ &\quad\times[2\mathbf{G} - 4\pi^2/(B_i+B_j+b_l+b_m)4\mathbf{xGGx}^t]\\ &\quad\times\exp\{-[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\} \qquad (11)\end{aligned}$$

$$\begin{aligned}\hat{T}^{-1}&[s^4/16\, g_i(\mathbf{h})g_j(\mathbf{h})]\\ &= O_iO_j \sum_{l=1}^{3}\sum_{m=1}^{3} a_la_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times[1/(B_i+B_j+b_l+b_m)]^2\\ &\quad\times\big([4\pi^2/(B_i+B_j+b_l+b_m)r^2]^2\\ &\quad - 5[4\pi^2/(B_i+B_j+b_l+b_m)]r^2 + (15/4)\big)\\ &\quad\times\exp\{-[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\} \qquad (12)\end{aligned}$$

$$\begin{aligned}\hat{T}^{-1}&[g_i(\mathbf{h})g_j(\mathbf{h})/(O_iO_j)]\\ &= \sum_{l=1}^{3}\sum_{m=1}^{3} a_la_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times\exp\{-[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\} \qquad (13)\end{aligned}$$

$$\begin{aligned}\hat{T}^{-1}&[\pi i\mathbf{h}s^2/2\, g_i(\mathbf{h})g_j(\mathbf{h})]\\ &= O_iO_j \sum_{l=1}^{3}\sum_{m=1}^{3} a_la_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times[1/(B_i+B_j+b_l+b_m)]\\ &\quad\times[4\pi^2/(B_i+B_j+b_l+b_m)]\\ &\quad\times\{5/2 - [4\pi^2/(B_i+B_j+b_l+b_m)]r^2\}\\ &\quad\times 2\mathbf{Gx}\exp\{[-4\pi^2/(B_i+B_j+b_l+b_m)]r^2\} \qquad (14)\end{aligned}$$

$$\begin{aligned}\hat{T}^{-1}&[2\pi i\mathbf{h}/O_i\, g_i(\mathbf{h})g_j(\mathbf{h})]\\ &= -O_j\sum_{l=1}^{3}\sum_{m=1}^{3} a_l a_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times[4\pi^2/(B_i+B_j+b_l+b_m)]\\ &\quad\times 2\mathbf{Gx}\exp\{-[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\}\end{aligned} \qquad (15)$$

$$\begin{aligned}\hat{T}^{-1}&[-s^2/(4O_i)g_i(\mathbf{h})g_j(\mathbf{h})]\\ &= O_j\sum_{l=1}^{3}\sum_{m=1}^{3} a_l a_m[4\pi/(B_i+B_j+b_l+b_m)]^{3/2}\\ &\quad\times[1/(B_i+B_j+b_l+b_m)]\\ &\quad\times\{-3/2+[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\}\\ &\quad\times\exp\{-[4\pi^2/(B_i+B_j+b_l+b_m)]r^2\}.\end{aligned} \qquad (16)$$

This work was supported in part by NIH grant GM20066 to B. W. Matthews.

References

Agarwal, R. C. (1978). *Acta Cryst.* A**34**, 791–809.

Agarwal, R., Lifchitz, A. & Dodson, E. (1981). In *Refinement of Protein Structures*, edited by P. A. Machin, J. W. Campbell & M. Elder. SERC Daresbury Laboratory, Warrington, England.

Bricogne, G. (1993). *Acta Cryst.* D**49**, 37–60.

Brünger, A. T., Kuriyan, K. & Karplus, M. (1987). *Science*, **235**, 458–460.

Hendrickson, W. A. & Konnert, J. H. (1980). *Computing in Crystallography*, edited by R. Diamond, S. Ramaseshan & K. Venkatesan, ch. 13, pp. 13.01–13.26. Bangalore: Indian Academy of Sciences.

Jacobson, R. H., Zhang, X.-J., DuBose, R. F. & Matthews, B. W. (1994). *Nature (London)*, **369**, 761–766.

Knuth, D. E. (1973). *Fundamental Algorithms*, 2nd ed. *The Art of Computer Programming*, Vol. 1. Reading, MA: Addison Wesley.

Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**, 240–255.

Sheldrick, G. M. & Schneider, T. R. (1997). *Macromolecular Crystallography*, Part B, edited by R. M. Sweet & C. W. Carter Jr. *Methods in Enzymology*, Vol. 277. New York: Academic Press.

Tronrud, D. E., Ten Eyck, L. F. & Matthews, B. W. (1987). *Acta Cryst.* A**43**, 489–501.