

- BODE, W. & SCHWAGER, P. (1975). *J. Mol. Biol.* **98**, 693–717.
- CHAMBERS, J. L. & STROUD, R. M. (1977). *Acta Cryst.* **B33**, 1824–1837.
- COOLEY, J. W. & TUKEY, J. W. (1965). *Math. Comput.* **19**, 297–301.
- CRUICKSHANK, D. W. J. (1949). *Acta Cryst.* **2**, 65–82.
- CUTFIELD, J. F., DODSON, E. J., DODSON, G. G., HODGKIN, D. C., ISAACS, N. W., SAKABE, K. & SAKABE, N. (1975). *Acta Cryst.* **A31**, S21.
- DEISENHOFER, J. & STEIGEMANN, W. (1975). *Acta Cryst.* **B31**, 238–250.
- DIAMOND, R. (1971). *Acta Cryst.* **A27**, 436–452.
- DIAMOND, R. (1974). *J. Mol. Biol.* **82**, 371–391.
- DODSON, E. J., ISAACS, N. W. & ROLLETT, J. S. (1976). *Acta Cryst.* **A32**, 311–315.
- DODSON, G. G., DODSON, E. J., HODGKIN, D. C. & VIJAYAN, M. (1978). To be published.
- FREER, S. T., ALDEN, R. A., CARTER, C. W. & KRAUT, J. (1975). *J. Biol. Chem.* **250**, 46–54.
- HODGKIN, D. C. (1974). *Proc. R. Soc. London Ser. A*, **338**, 251–275.
- HUBER, R., KUKLA, D., BODE, W., SCHWAGER, P., BARTELS, K., DEISENHOFER, J. & STEIGEMANN, W. (1974). *J. Mol. Biol.* **89**, 73–101.
- LUZZATI, V. (1952). *Acta Cryst.* **5**, 802–810.
- MOEWS, P. C. & KRETSINGER, R. H. (1975). *J. Mol. Biol.* **91**, 201–228.
- SAYRE, D. (1972). *Acta Cryst.* **A28**, 210–212.
- SAYRE, D. (1974). *Acta Cryst.* **A30**, 180–184.
- TAKANO, T. (1977a). *J. Mol. Biol.* **110**, 537–568.
- TAKANO, T. (1977b). *J. Mol. Biol.* **110**, 569–584.
- TEN EYCK, L. F. (1973). *Acta Cryst.* **A29**, 183–191.
- WATENPAUGH, K. D., SIEKER, L. C., HERRIOT, J. R. & JENSEN, L. H. (1973). *Acta Cryst.* **B29**, 943–956.

*Acta Cryst.* (1978). **A34**, 791–809

## A New Least-Squares Refinement Technique Based on the Fast Fourier Transform Algorithm

BY RAMESH C. AGARWAL\*

*IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA*

(Received 25 October 1977; accepted 2 May 1978)

A new atomic-parameters least-squares refinement method is presented which makes use of the fast Fourier transform algorithm at all stages of the computation. For large structures, the amount of computation is almost proportional to the size of the structure making it very attractive for large biological structures such as proteins. In addition the method has a radius of convergence of approximately 0.75 Å making it applicable at a very early stage of the structure-determination process. The method has been tested on hypothetical as well as real structures. The method has been used to refine the structure of insulin at 1.5 Å resolution, barium beauvericin complex at 1.2 Å resolution, and myoglobin at 2 Å resolution. Details of the method and brief summaries of its applications are presented in the paper.

### Introduction

The conventional least-squares refinement method is prohibitively expensive for large structures such as proteins. In this paper, we present a new method which is inexpensive even for a very large structure.

### The least-squares minimization

The process of least-squares minimization is discussed in *International Tables for X-ray Crystallography* (1959), Rollett (1965) and Lipson & Cochran (1966). It is being restated here to provide a uniform notation

for the paper. In least-squares minimization, one minimizes the sum of squares of  $N$  error functions (differences between the observed and the calculated values)  $E(s)$  with respect to (w.r.t.)  $R$  variables  $u_r$  ( $N > R$ ). The function to be minimized is called  $P$  and is given by

$$P = \frac{1}{2} \sum_s E^2(s). \quad (1)$$

In (1), we have introduced the factor of  $\frac{1}{2}$  to simplify the expressions for the derivatives. In normal least-squares minimization, the error function is approximated as a linear function of the variables as given by:

$$\Delta E(s) = \sum_r \frac{\partial E(s)}{\partial u_r} \Delta u_r \quad (2)$$

\* Present address: Centre for Applied Research in Electronics, Indian Institute of Technology, New Delhi 110029, India.

where  $\Delta E(\mathbf{s})$  is the change produced in  $E(\mathbf{s})$  due to small changes  $\Delta u_r$  in  $u_r$ . Defining

$$J(\mathbf{s}, u_r) \equiv \frac{\partial E(\mathbf{s})}{\partial u_r}, \quad (3)$$

(2) becomes

$$\Delta E(\mathbf{s}) = \sum_r J(\mathbf{s}, u_r) \Delta u_r \quad (4)$$

We would like a solution of (4) such that  $E(\mathbf{s}) + \Delta E(\mathbf{s}) = 0$ , or in other words we would like to solve the following set of linear equations for  $\Delta u_r$ :

$$-E(\mathbf{s}) = \sum_r J(\mathbf{s}, u_r) \Delta u_r; \quad (5)$$

or, in matrix notation

$$-E = \mathbf{J} \Delta \mathbf{u} \quad (6)$$

where  $E$  is the length- $N$  error vector,  $\Delta \mathbf{u}$  is the length- $R$  vector of desired displacement in variables, and  $\mathbf{J}$  is the  $N \times R$  Jacobian matrix. Equation (6) is a set of  $N$  over-determined equations in  $R$  unknowns. According to linear algebra, the least-squares solution of (6) which minimizes  $P$  is obtained by premultiplying (6) by the transpose of the Jacobian matrix and solving the resulting normal equations:

$$-\mathbf{J}^T E = \mathbf{J}^T \mathbf{J} \Delta \mathbf{u}. \quad (7)$$

Let us define

$$\mathbf{G} \equiv \mathbf{J}^T E \quad (8)$$

and,

$$\mathbf{H} \equiv \mathbf{J}^T \mathbf{J}, \quad (9)$$

where  $\mathbf{G}$  is a length- $R$  gradient (derivatives) vector and  $\mathbf{H}$  is an  $R \times R$  normal matrix. Then (7) becomes

$$-\mathbf{G} = \mathbf{H} \Delta \mathbf{u}. \quad (10)$$

These are  $R$  normal equations which can be solved by matrix inversion.

$$\Delta \mathbf{u} = -\mathbf{H}^{-1} \mathbf{G}. \quad (11)$$

This solution of  $\Delta \mathbf{u}$  will minimize  $P$  if the linearity assumption of (2) is valid, *i.e.* if the required displacement  $\Delta \mathbf{u}$  is small. The following are explicit expressions for  $\mathbf{G}$  and  $\mathbf{H}$ :

$$G(u_r) = \sum_{\mathbf{s}} J(\mathbf{s}, u_r) E(\mathbf{s}) \quad (12)$$

$$H(u_q, u_r) = \sum_{\mathbf{s}} J(\mathbf{s}, u_q) J(\mathbf{s}, u_r). \quad (13)$$

If one wishes to minimize a weighted least-squares function as given by

$$P = \frac{1}{2} \sum_{\mathbf{s}} W(\mathbf{s}) E^2(\mathbf{s}), \quad (14)$$

then (12) and (13) become

$$G(u_r) = \sum_{\mathbf{s}} J(\mathbf{s}, u_r) E(\mathbf{s}) W(\mathbf{s}) \quad (15)$$

$$H(u_q, u_r) = \sum_{\mathbf{s}} J(\mathbf{s}, u_q) J(\mathbf{s}, u_r) W(\mathbf{s}). \quad (16)$$

Now, we apply this to the problem of least-squares refinement of the atomic parameters.

### Least-squares refinement of the atomic parameters

In least-squares refinement of the atomic parameters, we minimize the following function,

$$P = \frac{1}{2} \sum_{hkl} W(hkl) [ |F_c(hkl)| - |F_o(hkl)| ]^2 \quad (17)$$

where  $W(hkl)$  is any desired weighting function. Strictly speaking, the summation in (17) should be carried out over only the unique reflections but, in order to derive simple closed-form expressions for the gradient and normal matrix terms, we consider the summation over all the reflections. This gives mathematically identical results. Also for brevity, most of the time we will denote  $(hkl)$  by  $(\mathbf{s})$ , where  $\mathbf{s}$  represents a point in the reciprocal lattice and  $F_c(\mathbf{s})$  by simply  $F(\mathbf{s})$ . In this notation, (17) becomes:

$$P = \frac{1}{2} \sum_{\mathbf{s}} W(\mathbf{s}) [ |F(\mathbf{s})| - |F_o(\mathbf{s})| ]^2. \quad (18)$$

This equation is similar to (14) with

$$E(\mathbf{s}) = |F(\mathbf{s})| - |F_o(\mathbf{s})|. \quad (19)$$

Let us assume there are  $M$  atoms in the asymmetric unit of the structure. Their thermal motion is assumed to be isotropic. In this case, there are  $3M$  positional and  $M$  thermal parameters ( $B$ 's). Equation (18) can be minimized w.r.t. these  $4M$  variables. It is assumed that the number of unique reflections  $N$  is much larger than  $4M$  variables. For the time being, we assume that all atomic sites have full occupancies, although partial occupancies can be easily taken into consideration. Also, at present we assume that the structure is in space group  $P1$ , although similar expressions can be derived for other space groups also. We introduce the following notations:

$\mathbf{r}_m \equiv (x_m, y_m, z_m)$  – fractional cell coordinates of the  $m$ th atom,

$B_m$  – isotropic thermal parameter of the  $m$ th atom,

$\mathbf{s} \equiv hkl$  – a diffraction point in reciprocal space,

$f_m(\mathbf{s}) \equiv f_m(hkl)$  – scattering factor of the  $m$ th atom at reciprocal distance  $\mathbf{s}$ ,

$\mathbf{s} \cdot \mathbf{r}_m \equiv hx_m + ky_m + lz_m$ ,

$s^2 \equiv |\mathbf{s}|^2 = (4 \sin^2 \theta) / \lambda^2$ .

With this notation the expression for calculated structure factors is

$$F(\mathbf{s}) = \sum_m f_m(\mathbf{s}) \exp(-B_m s^2/4) \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m). \quad (20)$$

To simplify the equations further, we introduce the notation

$$g_m(\mathbf{s}) \equiv f_m(\mathbf{s}) \exp(-B_m s^2/4). \quad (21)$$

Here  $g_m(\mathbf{s})$  represents the contribution of the  $m$ th atom to structure factors after taking into account its thermal motion. This notation greatly simplifies the expressions for gradient and normal-matrix terms. With this notation (20) becomes

$$F(\mathbf{s}) = \sum_m g_m(\mathbf{s}) \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m). \quad (22)$$

To evaluate the gradient vector and the normal matrix, we need expressions for the Jacobian as given by (3). First, consider the Jacobian w.r.t. positional coordinates:

$$J(\mathbf{s}, x_m) = \frac{\partial E(\mathbf{s})}{\partial x_m}.$$

Because  $F_0(\mathbf{s})$  is independent of the atomic variables,

$$\begin{aligned} J(\mathbf{s}, x_m) &= \frac{\partial |F(\mathbf{s})|}{\partial x_m} \\ &= \frac{1}{2|F(\mathbf{s})|} \frac{\partial |F(\mathbf{s})|^2}{\partial x_m}. \end{aligned}$$

In our derivations, we neglect the effect of anomalous dispersion. The atoms having anomalous dispersion can be treated separately. Neglecting anomalous dispersion, because of the Hermitian symmetry  $F(-\mathbf{s}) = F^*(\mathbf{s})$ , where \* represents the complex conjugate, giving

$$\begin{aligned} J(\mathbf{s}, x_m) &= \frac{1}{2|F(\mathbf{s})|} \frac{\partial [F(\mathbf{s})F(-\mathbf{s})]}{\partial x_m} \\ &= \frac{1}{2|F(\mathbf{s})|} \left[ F(\mathbf{s}) \frac{\partial F(-\mathbf{s})}{\partial x_m} + F(-\mathbf{s}) \frac{\partial F(\mathbf{s})}{\partial x_m} \right]. \end{aligned}$$

Since,  $F(-\mathbf{s}) = F(\mathbf{s})^*$ ,  $J(\mathbf{s}, x_m)$  can be rewritten as

$$J(\mathbf{s}, x_m) = \frac{1}{2|F(\mathbf{s})|} \left\{ F(\mathbf{s}) \left[ \frac{\partial F(\mathbf{s})}{\partial x_m} \right]^* + F^*(\mathbf{s}) \frac{\partial F(\mathbf{s})}{\partial x_m} \right\}. \quad (23)$$

Taking the partial derivative of (22) w.r.t.  $x_m$ , we obtain

$$\frac{\partial F(\mathbf{s})}{\partial x_m} = g_m(\mathbf{s})(i2\pi h) \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m). \quad (24)$$

Let  $\varphi(\mathbf{s})$  be the phase associated with  $F(\mathbf{s})$ , then

$$\begin{aligned} F(\mathbf{s}) &= |F(\mathbf{s})| \exp[i\varphi(\mathbf{s})] \\ F(-\mathbf{s}) &= |F(\mathbf{s})| \exp[-i\varphi(\mathbf{s})]. \end{aligned}$$

Substituting these in (23) and after simplification, we obtain

$$\begin{aligned} J(\mathbf{s}, x_m) &= \frac{1}{2} g_m(\mathbf{s})(i2\pi h) \{ \exp[-i\varphi(\mathbf{s})] \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m) \\ &\quad - \exp[i\varphi(\mathbf{s})] \exp(-i2\pi\mathbf{s} \cdot \mathbf{r}_m) \}. \end{aligned} \quad (25)$$

Similar expressions can be obtained for  $J(\mathbf{s}, y_m)$  and  $J(\mathbf{s}, z_m)$ . Also similarly we derive the expression for  $J(\mathbf{s}, B_m)$ :

$$\begin{aligned} J(\mathbf{s}, B_m) &= \frac{1}{2} g_m(\mathbf{s})(-s^2/4) \{ \exp[i\varphi(\mathbf{s})] \exp(-i2\pi\mathbf{s} \cdot \mathbf{r}_m) \\ &\quad + \exp[-i\varphi(\mathbf{s})] \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m) \}. \end{aligned} \quad (26)$$

Substituting (25) in (15), we obtain the expression for gradient w.r.t.  $x_m$ :

$$\begin{aligned} G(x_m) &= \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s})(i2\pi h) W(\mathbf{s}) E(\mathbf{s}) \{ \exp[-i\varphi(\mathbf{s})] \\ &\quad \times \exp(i2\pi\mathbf{s} \cdot \mathbf{r}_m) - \exp[i\varphi(\mathbf{s})] \\ &\quad \times \exp(-i2\pi\mathbf{s} \cdot \mathbf{r}_m) \}. \end{aligned}$$

The above equation can be simplified to

$$\begin{aligned} G(x_m) &= \sum_{\mathbf{s}} g_m(\mathbf{s})(-i2\pi h) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})] \\ &\quad \times \exp(-i2\pi\mathbf{s} \cdot \mathbf{r}_m). \end{aligned} \quad (27)$$

In the above two equations if the summation terms corresponding to  $\mathbf{s}$  and  $-\mathbf{s}$  are grouped together, it can be easily proved that they are identical. Similar expressions can be derived for  $G(y_m)$  and  $G(z_m)$ . The expression for  $G(B_m)$  is as follows.

$$\begin{aligned} G(B_m) &= \sum_{\mathbf{s}} g_m(\mathbf{s})(-s^2/4) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})] \\ &\quad \times \exp(-i2\pi\mathbf{s} \cdot \mathbf{r}_m). \end{aligned} \quad (28)$$

By substituting (25) in (16), we can derive expressions for the normal-matrix terms corresponding to interaction between  $x_m$  and  $x_n$ :

$$\begin{aligned} H(x_m, x_n) &= \sum_{\mathbf{s}} \frac{1}{4} g_m(\mathbf{s}) g_n(\mathbf{s})(-4\pi^2 h^2) W(\mathbf{s}) \\ &\quad \times \{ \exp[i2\varphi(\mathbf{s})] \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)] \\ &\quad + \exp[-i2\varphi(\mathbf{s})] \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)] \\ &\quad - \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \\ &\quad - \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \}. \end{aligned}$$

Again, as above, by grouping together the summation terms corresponding to  $\mathbf{s}$  and  $-\mathbf{s}$ , this simplifies to

$$\begin{aligned} H(x_m, x_n) &= \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s})(4\pi^2 h^2) W(\mathbf{s}) \\ &\quad \times \{ \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] - \exp[i2\varphi(\mathbf{s})] \\ &\quad \times \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)] \}. \end{aligned} \quad (29)$$

We introduce another notation. Normal-matrix terms are written as the sum of two terms  $H_1$ , corresponding to  $\mathbf{r}_m - \mathbf{r}_n$  and  $H_2$  corresponding to  $\mathbf{r}_m + \mathbf{r}_n$ .

$$H(u_r, u_q) = H_1(u_r, u_q) + H_2(u_r, u_q). \quad (30)$$

With this notation, expressions for  $H_1(x_m, x_n)$  and  $H_2(x_m, x_n)$  are as follows:

$$H_1(x_m, x_n) = \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (4\pi^2 h^2) W(\mathbf{s}) \times \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \quad (31)$$

$$H_2(x_m, x_n) = \sum_{\mathbf{s}} -\frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (4\pi^2 h^2) W(\mathbf{s}) \times \exp[i2\varphi(\mathbf{s})] \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)]. \quad (32)$$

Expressions for  $H(y_m, y_n)$  and  $H(z_m, z_n)$  are similar. Similarly, we derive an expression for  $H(B_m, B_n)$ . Below are expressions for the two parts of  $H(B_m, B_n)$ :

$$H_1(B_m, B_n) = \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (s^4/16) W(\mathbf{s}) \times \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \quad (33)$$

$$H_2(B_m, B_n) = \sum_{\mathbf{s}} -\frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (s^4/16) W(\mathbf{s}) \times \exp[i2\varphi(\mathbf{s})] \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)]. \quad (34)$$

Expressions for  $H(x_m, y_n)$  are identical to those of  $H(x_m, x_n)$  with  $h^2$  replaced by  $hk$ :

$$H_1(x_m, y_n) = \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (4\pi^2 hk) W(\mathbf{s}) \times \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \quad (35)$$

$$H_2(x_m, y_n) = \sum_{\mathbf{s}} -\frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (4\pi^2 hk) W(\mathbf{s}) \times \exp[i2\varphi(\mathbf{s})] \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)]. \quad (36)$$

Expressions for  $H(x_m, B_n)$  are given below:

$$H_1(x_m, B_n) = \sum_{\mathbf{s}} \frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (i\pi h s^2) W(\mathbf{s}) \times \exp[i2\pi\mathbf{s} \cdot (\mathbf{r}_m - \mathbf{r}_n)] \quad (37)$$

$$H_2(x_m, B_n) = \sum_{\mathbf{s}} -\frac{1}{2} g_m(\mathbf{s}) g_n(\mathbf{s}) (i\pi h s^2) W(\mathbf{s}) \times \exp[i2\varphi(\mathbf{s})] \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_m + \mathbf{r}_n)]. \quad (38)$$

This completes closed-form expressions for all entries in the normal matrix.

### Fast computation of structure factors

Structure factors are three-dimensional Fourier transforms of the real-space electron density. Therefore,

structure factors can be calculated by Fourier inversion of a model electron density map. With the advent of the fast Fourier transform (FFT) algorithm (Cooley & Tukey, 1965), the Fourier inversion of the model electron density can be carried out very efficiently. To make use of the FFT method, electron density must be sampled on a uniform grid parallel to the cell axes. In order to reduce the amount of computation, the number of grid points should also be reduced. This introduces the sampling problem caused by overlapping of the periodic Fourier spectrum. Sayre (1951) first discussed these problems. Recently, Ten Eyck (1977) discussed the FFT computation of structure factors in great detail, including discussion of the sampling problem. He also suggests a method of reducing the number of grid points by artificially increasing the  $B$  values of all the atoms by a fixed value. The reader is urged to read this paper. In the FFT computation of structure factors, a major computational step is setting up the model electron density map. This aspect of the structure factor computation has not received enough attention. In this section, we will discuss it in detail.

### Modeling of the atomic electron density

As shown by (21),  $g_m(\mathbf{s})$  represents the contribution of the  $m$ th atom to structure factors. This is a product of two functions, the atomic form factor  $f_m(\mathbf{s})$  and the Gaussian function  $\exp(-B_m s^2/4)$  representing isotropic thermal motion of the atoms. Gaussian functions have two important properties which make them very useful in modeling atomic electron density. First, the product of two Gaussian functions is another Gaussian function. Thus, if  $f(\mathbf{s})$  is modeled as a sum of two Gaussian functions,  $g(\mathbf{s})$  is also a sum of Gaussian functions. Secondly, the Fourier transform of a Gaussian function is also a Gaussian function. Thus, the atomic electron density  $\rho(\mathbf{r})$ , which is the Fourier transform of  $g(\mathbf{s})$ , is also modeled as a sum of Gaussian functions. The amount of computation involved in setting up the electron density array is proportional to the number of different Gaussian functions used in approximating  $g(\mathbf{s})$ .

Vand, Eiland & Pepinsky (1957) approximated  $f(\mathbf{s})$  as a sum of a constant and two Gaussian functions. Forsyth & Wells (1959) further improved on their approximation. For this approximation, after accounting for the thermal motion of the atom,  $g(\mathbf{s})$  is represented as a sum of three Gaussian functions. Thus, for the purpose of computation, this approximation is equivalent to a three-Gaussian approximation. In deriving optimal parameters for the approximation, Forsyth & Wells (1959) used a weighting function  $w(\mathbf{s})$  of the form  $\exp[-(s-1)^2]$ . For this approximation, they calculated the weighted and normalized (divided

by the number of electrons) root-mean-square (r.m.s.) error, defined as

$$\sigma = \left\{ \left[ \sum_{\mathbf{s}} w(\mathbf{s}) \Delta f^2(\mathbf{s}) \right] / \left[ \sum_{\mathbf{s}} w(\mathbf{s}) \right] \right\}^{1/2} / Z, \quad (39)$$

where  $\Delta f(\mathbf{s})$  is the difference between tabulated and approximated values of the atomic form factors, and  $Z$  is the number of electrons in the atom. For atoms of biological significance (C, N, O and S)  $\sigma$  ranges from 0.0039 to 0.0052.

*International Tables for X-ray Crystallography* (1974) gives another approximation to atomic form factors as a sum of a constant and four Gaussian functions. For this approximation,  $\sigma$  values are much smaller, but at the cost of increased computation.

For most biological structures, observed X-ray intensities do not extend beyond 1.5 Å resolution. It is possible to get a better approximation to atomic form factors in this limited region of reciprocal space. For biological structures, we propose a two-Gaussian approximation to the form factors as given by (40)

$$f(\mathbf{s}) \simeq C^1 \exp(-B^1 s^2/4) + C^2 \exp(-B^2 s^2/4) \quad (40)$$

where  $C^1 + C^2$  is not necessarily equal to  $Z$ . The four parameters ( $C^1, C^2, B^1, B^2$ ) associated with this approximation are chosen so as to minimize the weighted sum of squared differences  $\sum_{\mathbf{s}} w(\mathbf{s}) \Delta f^2(\mathbf{s})$ . Since the number of reflections in each shell is proportional to  $s^2$ , the weighting function  $w(\mathbf{s})$  is taken as  $s^2$ . This weighting function gives very low weights to low-resolution form factors; to avoid this, we use a fixed value of 0.04 for  $w(\mathbf{s})$ , for  $s$  values up to 0.2. We carried out the minimization to obtain the parameters for some common atoms and biologically significant ions. The values of  $f(\mathbf{s})$  used were taken from Table 2.2A of *International Tables for X-ray Crystallography* (1974). Since we were primarily concerned with protein crystallog-

raphy, we used the scattering data only up to  $s = 0.68$ , corresponding to 1.47 Å resolution. Table 1 summarizes the results of this approximation. Column 7 gives the values of  $\sigma$  as defined by (39), with  $w(\mathbf{s})$  as explained above. The normalized average error in the approximation is given in column 8. In column 9, we give the normalized maximum error in the approximation. Since low-resolution form factors are given very low weights, the maximum error in the approximation tends to occur in that region. For protein crystallography, because of the disordered solvent structure, observed data up to about 10 Å resolution ( $s < 0.1$ ) are highly erroneous and are of little interest. Therefore, we have taken  $|\Delta f|_{\max}$  as the maximum error in the range of  $2 \sin \theta/\lambda = 0.10$  to  $0.68 \text{ \AA}^{-1}$  (corresponding to 10 to 1.47 Å resolution). The corresponding value of  $2 \sin \theta/\lambda$  is given in column 10. For all atoms in the table except  $C_{\text{val}}$  the  $\sigma$  values in this approximation are less than 0.001. This is a much better approximation than that of Vand *et al.* (1957) and Forsyth & Wells (1959) and it requires about one third less computation.

We have also developed a single-Gaussian approximation for the atomic form factors. Most of the atoms in a protein molecule are C, N, O and S. The atomic form factors for these atoms are similar and to a first degree they can be approximated as the product of the number of electrons and a normalized form-factor curve. To further improve this approximation, we multiply this normalized form-factor curve by a Gaussian function which depends on the atom type. Thus we propose the following approximation

$$f(\mathbf{s}) \simeq f_{\text{norm}}(\mathbf{s}) C^3 \exp(-B^3 s^2/4), \quad (41)$$

where  $f_{\text{norm}}(\mathbf{s})$  is common to all the atoms and  $C^3$  and  $B^3$  depend on the atom type. In this approximation only  $C^3 \exp(-B^3 s^2/4)$  is accounted for in the electron density computation. After the FFT computation of the

Table 1. Two-Gaussian approximation of atomic form factors for  $0.0 < (2 \sin \theta)/\lambda < 0.68 \text{ \AA}^{-1}$

1	2	3	4	5	6	7	8	9	10
Atom type	Number of electrons $Z$	$C^1$	$B^1$	$C^2$	$B^2$	$\sigma$ $\times 10^{-4}$	$\frac{ \Delta f _{\text{ave}}}{Z}$ $\times 10^{-4}$	$\frac{ \Delta f _{\max}^*}{Z}$ $\times 10^{-4}$	$(2 \sin \theta)/\lambda$ at $ \Delta f _{\max} (\text{\AA}^{-1})$
H	1	0.4866	34.1284	0.5098	8.8996	9.78	11.50	21.0	0.24
C	6	3.0102	29.9132	2.9705	2.8724	8.34	10.36	21.7	0.22
$C_{\text{val}}$	6	3.1055	30.1306	2.8623	2.4793	13.54	16.83	33.8	0.24
N	7	3.0492	25.0383	3.9432	3.4059	3.75	4.10	8.8	0.22
O	8	3.2942	20.0401	4.6968	3.1184	3.16	3.78	6.9	0.26
S	16	5.6604	33.0400	10.3140	1.8160	5.36	6.07	12.5	0.22
$\text{Fe}^{3+}$	23	10.3568	8.1324	12.6329	0.8137	1.81	2.01	2.9	0.32
$\text{Fe}^{2+}$	24	11.6635	9.0361	12.3057	0.5749	5.13	5.57	7.4	0.34
$\text{Zn}^{2+}$	28	5.7826	11.7082	22.2163	1.8234	0.32	0.27	0.8	0.42
$\text{Ba}^{2+}$	54	12.1432	21.7090	41.8442	1.4090	2.68	2.69	3.2	0.48

\* The maximum is for  $0.10 < (2 \sin \theta)/\lambda < 0.68 \text{ \AA}^{-1}$ .

structure factors, all structure factors are multiplied by  $f_{\text{norm}}(\mathbf{s})$ . For C, N, O and S we obtained  $f_{\text{norm}}(\mathbf{s})$ , given in Table 2, and the parameters ( $C^3$  and  $B^3$ ) of the single-Gaussian approximation which minimize the least-squares error in the approximation. For this approximation  $w(\mathbf{s})$  was unity. Table 3 summarizes the results of the approximation. The errors in the approximation are about 1%. This may be acceptable in the early stages of the refinement when the errors in the structure factors are very high. This approximation requires only about half the computation required for the two-Gaussian approximation. In this approximation, the FFT computes  $F_c(\mathbf{s})/f_{\text{norm}}(\mathbf{s})$ . As a function of  $\mathbf{s}$ , these do not drop as fast as  $F_c(\mathbf{s})$ ; therefore this approximation has somewhat more aliasing error. This can be compensated for either by reducing the grid spacing or alternatively by artificially increasing the  $B$  values of all the atoms (Ten Eyck, 1977). Also, this approximation will be used only in the early phase of the refinement when somewhat more error in high-order  $F_c$  would be permissible. For some of the heavier atoms the single-Gaussian approximation is very poor. Since a protein molecule has only a few heavy atoms, their contribution to the structure factors can be computed using the direct method and then added to the contribution (computed by the FFT method) from

the rest of the atoms. Alternatively, they can be approximated by  $f_{\text{norm}}(\mathbf{s})$  times a sum of two Gaussian functions.

### Radius of the model atoms

Deciding the radius of the model atoms is the next important step in setting up the electron-density array. The radius of the  $m$ th atom,  $\text{rad}_m$ , is defined as the maximum distance from its center beyond which its contribution to the electron density is neglected. The computation time in setting up the array is proportional to the number of grid points encompassed by a sphere of radius  $\text{rad}_m$ . Therefore, to minimize the computation, we should choose the minimum possible radius without leaving out too much electron density. If we decide the fraction of the number of electrons which may be left out, we can analytically compute the required radius of the atom. To compute this, we need the expression for the model electron density.

The Fourier relations for a Gaussian function are as follows:

real space		reciprocal space
electron density		atomic scattering factors
$\rho(r)$	$\leftarrow \text{FT} \rightarrow$	$f(\mathbf{s})$
$(4\pi/B)^{3/2} \exp(-4\pi^2 r^2/B)$	$\leftarrow \text{FT} \rightarrow$	$\exp(-B s^2/4)$

where  $r$  represents the distance from the center of the atom. Using the two-Gaussian approximation, the contribution of the  $m$ th atom to structure factors is obtained by multiplying (40) by the isotropic thermal motion term  $\exp(-B_m s^2/4)$ :

$$g_m(\mathbf{s}) \simeq C_m^1 \exp[-(B_m^1 + B_m)s^2/4] + C_m^2 \exp[-(B_m^2 + B_m)s^2/4]. \quad (42)$$

By taking the Fourier transform of  $g_m(\mathbf{s})$ , using the above Fourier relations, we obtain the expression for the model electron density of the  $m$ th atom.

$$\rho_m(r) = C_m^1 [4\pi/(B_m^1 + B_m)]^{3/2} \exp[-4\pi^2 r^2/(B_m^1 + B_m)] + C_m^2 [4\pi/(B_m^2 + B_m)]^{3/2} \exp[-4\pi^2 r^2/(B_m^2 + B_m)]. \quad (43)$$

Table 2. Normalized atomic form factors for the single-Gaussian approximation

$(2 \sin \theta)/\lambda$ ( $\text{\AA}^{-1}$ )	$f_{\text{norm}}$	$(2 \sin \theta)/\lambda$ ( $\text{\AA}^{-1}$ )	$f_{\text{norm}}$	$(2 \sin \theta)/\lambda$ ( $\text{\AA}^{-1}$ )	$f_{\text{norm}}$
0.02	1.0000	0.26	0.8114	0.50	0.5487
0.04	0.9934	0.28	0.7881	0.52	0.5308
0.06	0.9866	0.30	0.7645	0.54	0.5141
0.08	0.9773	0.32	0.7410	0.56	0.4975
0.10	0.9657	0.34	0.7176	0.58	0.4824
0.12	0.9518	0.36	0.6945	0.60	0.4674
0.14	0.9357	0.38	0.6718	0.62	0.4539
0.16	0.9181	0.40	0.6496	0.64	0.4404
0.18	0.8988	0.42	0.6283	0.66	0.4283
0.20	0.8784	0.44	0.6071	0.68	0.4164
0.22	0.8567	0.46	0.5873		
0.24	0.8344	0.48	0.5674		

Table 3. Single-Gaussian approximation of atomic form factors for  $0.0 < (2 \sin \theta)/\lambda < 0.68 \text{\AA}^{-1}$

1	2	3	4	5	6	7	8
Atom type	Number of electrons $Z$	$C^3$	$B^3$	$\sigma$ $\times 10^{-3}$	$  \Delta f  _{\text{ave}}$ $Z$ $\times 10^{-3}$	$  \Delta f  _{\text{max}}^*$ $Z$ $\times 10^{-3}$	$(2 \sin \theta)/\lambda$ at $  \Delta f  _{\text{max}}$ ( $\text{\AA}^{-1}$ )
C	6	5.9074	1.2913	10.70	9.55	18.00	0.68
N	7	7.0411	0.2065	3.11	2.72	4.43	0.68
O	8	8.1561	-0.8941	12.40	11.06	20.87	0.68
S	16	15.8448	-2.1392	5.55	4.85	8.25	0.32

\* The maximum is for  $0.10 < (2 \sin \theta)/\lambda < 0.68 \text{\AA}^{-1}$ .

If we omit the electron density beyond  $r = \text{rad}_m$ , the number of electrons left out (due to the truncation of the atom) is obtained by integrating (43) from  $r = \text{rad}_m$  to  $\infty$ :

$$\Delta Z_m = \int_{\text{rad}_m}^{\infty} \rho_m(r) 4\pi r^2 dr. \quad (44)$$

For reasonable values of  $\text{rad}_m$ ,  $\Delta Z_m$  can be approximated by the following expression

$$\Delta Z_m \simeq 4(\pi)^{1/2} \text{rad}_m \left[ (B_m^1 + B_m)^{-1/2} C_m^1 \exp\left(\frac{-4\pi^2 \text{rad}_m^2}{B_m^1 + B_m}\right) + (B_m^2 + B_m)^{-1/2} C_m^2 \exp\left(\frac{-4\pi^2 \text{rad}_m^2}{B_m^2 + B_m}\right) \right]. \quad (45)$$

Since  $B_m^1 \gg B_m^2$  (see Table 1), the second term in this approximation is much smaller compared with the first term. Thus, it can be further approximated to

$$\Delta Z_m \simeq 4(\pi)^{1/2} \text{rad}_m (B_m^1 + B_m)^{-1/2} C_m^1 \exp\left(\frac{-4\pi^2 \text{rad}_m^2}{B_m^1 + B_m}\right). \quad (46)$$

Let

$$d = \frac{4\pi^2 \text{rad}_m^2}{B_m^1 + B_m} \quad (47)$$

then

$$\Delta Z_m \simeq 2C_m^1 (d/\pi)^{1/2} \exp(-d). \quad (48)$$

The fraction of the number of electrons left out is given by

$$\Delta Z_m/Z_m \simeq 2(d/\pi)^{1/2} \exp(-d) C_m^1/Z_m, \quad (49)$$

where  $Z_m$  is the number of electrons in  $m$ th atom. This equation can be solved for the required radius  $\text{rad}_m$  for a desired value of  $\Delta Z_m/Z_m$  and other parameters of the atom. For some of the atoms, Table 4 gives the fraction of electrons left out for some values of  $d$ . Table 5 gives the required radius  $\text{rad}_m$  for carbon atoms for some values of  $B_m$  and  $\Delta Z_m/Z_m$ . For other atom types (N, O, S, etc.) the required radius will be even smaller because for these atoms both  $B_m^1$  and  $C_m^1/Z_m$  are smaller.

A similar analysis can be carried out for the single-Gaussian approximation. For this approximation, Table 6 gives the required radius  $\text{rad}_m$  for carbon atoms

for some values of  $B_m$  and  $\Delta Z_m/Z_m$ . For other atom types also the required radius would be about the same. Comparing Tables 5 and 6 we note that the single-Gaussian approximation requires a much smaller radius compared with the two-Gaussian approximation. This results in further savings in computation time for setting up the electron-density array. The reason for this is that  $B^3$  is much smaller than  $B^1$ .

To summarize, the fast computation of structure factors consists of two steps. (1) Building up of the electron-density array by adding up the contribution from each atom. (2) Three-dimensional FFT of the electron-density array to give  $F_c$ . The amount of computation required in the first step is clearly proportional to the number of atoms and the amount of computation required in the second step is proportional to  $N_s \log N_s$  where  $N_s$  is the number of grid points. For most of our refinement work the computation time required in both the steps was about the same. For most space groups (perhaps for all space groups) in step 1, only the atoms lying in the asymmetric unit of the unit cell need be considered. [For example in  $R_3$  the asymmetric unit chosen was  $(0 - \frac{1}{3}, 0 - \frac{1}{3}, 0 - 1)$ .] This also minimizes the storage requirement to that for the asymmetric-unit grid points only. Also, in most cases symmetries in the space group can be very well utilized in the FFT computation. See the excellent paper by Ten Eyck (1973). The computation of electron density using (43) requires many exponential computations. To reduce computation, exponentials can be precomputed in the form of a table of  $\exp(-a)$  for  $a = 0$  to 7.0, at an interval of 0.002 or more depending on the desired accuracy. For purposes of density computation

Table 5. *The required radius  $\text{rad}_m$  (Å) for carbon atoms for some values of  $\Delta Z_m/Z_m$  and  $B_m$  for the two-Gaussian approximation*

$\Delta Z_m/Z_m$	0.02	0.01	0.005	0.002
$B_m$				
0	1.74	1.90	2.06	2.24
5	1.89	2.06	2.22	2.42
10	2.02	2.21	2.38	2.58
20	2.26	2.47	2.68	2.89
50	2.84	3.12	3.36	3.65

Table 4. *The fraction of the electrons left out ( $\Delta Z_m/Z_m$ ) ( $\times 10^3$ ) for some values of  $d$  for the two-Gaussian approximation*

$d$	5.0	5.5	6.0	6.5	7.0
Atom type					
C	8.5	5.4	3.4	2.2	1.4
N	7.4	4.7	3.0	1.9	1.2
O	7.0	4.4	2.8	1.8	1.1
S	6.0	3.8	2.4	1.5	1.0

Table 6. *The required radius  $\text{rad}_m$  (Å) for carbon atoms for some values of  $\Delta Z_m/Z_m$  and  $B_m$  for the single-Gaussian approximation*

$\Delta Z_m/Z_m$	0.02	0.01	0.005
$B_m$			
10	1.17	1.26	1.37
20	1.61	1.74	1.89
50	2.50	2.69	2.93

$\exp(-a)$  can be considered 0 for  $a > 7.0$ . Also, since each atom has hundreds of sampling points, the constants of (43) which are independent of  $r$  should be computed only once for each atom. Also some savings can be achieved during the three-dimensional FFT computation. Since the sampling interval is approximately one third of the resolution of the data, after FFT computation along one dimension superfluous data in reciprocal space are obtained along that index, which need not be carried over for FFT computation for the next dimension. For example, if the first FFT is computed along  $x$ , after the FFT computation the  $h$  index will run up to approximately  $1.5 h_{\max}$ , where  $h_{\max}$  is the maximum  $h$  index in the  $F_o$ . Therefore, for the next transform computation, data beyond  $h_{\max}$  can be discarded, resulting in a computational saving. This concludes the discussion on fast computation of structure factors.

### Fast computation of gradient

Fast computation of gradient is similar to fast computation of structure factors, but, in this case the order of computation is reversed. First, a three-dimensional FFT is computed for each set of parameters (*i.e.* all  $x$  parameters) followed by individual computation for each atom in real space. Equation (27) is a closed-form expression for gradient w.r.t.  $x$  parameters. This equation can be rewritten as

$$G(x_m) = \sum_{\mathbf{s}} D_x(\mathbf{s}) g_m(\mathbf{s}) \exp(-i2\pi \mathbf{s} \cdot \mathbf{r}_m) \quad (50)$$

where  $D_x(\mathbf{s})$  is a function common to all the atoms and is defined by:

$$D_x(\mathbf{s}) \equiv (-i2\pi h) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})]. \quad (51)$$

The subscript  $x$  denotes gradient w.r.t.  $x$  parameters. Equation (50) represents the Fourier transform of the product of two functions  $D_x(\mathbf{s})$  and  $g_m(\mathbf{s})$ , evaluated at  $\mathbf{r}_m$  (position of  $m$ th atom). According to the convolution theorem, multiplication in reciprocal space is equivalent to convolution in real space. Let  $d_x(r)$  be the Fourier transform of  $D_x(\mathbf{s})$ . Since  $D_x(\mathbf{s})$  contains  $E(\mathbf{s}) \exp[i\varphi(\mathbf{s})]$ ,  $d_x(r)$  can be thought of as a modified difference density function. As before  $\rho_m(r)$  given by (43) is the Fourier transform of  $g_m(\mathbf{s})$ . Then by the convolution theorem  $G(x_m)$  is the convolution of  $d_x(r)$  and  $\rho_m(r)$ , evaluated at  $r = r_m$ :

$$G(x_m) = \int d_x(r) \rho_m(r - r_m) dr.$$

For computation purposes, the integration can be replaced by a summation:

$$G(x_m) = \sum_r d_x(r) \rho_m(r - r_m). \quad (52)$$

Here summation is to be carried out over all grid points in real space and  $r - r_m$  is the distance of the grid point from the center of the  $m$ th atom. Since atomic electron density  $\rho_m(r - r_m)$  is assumed to be non-zero only for grid points within a distance  $\text{rad}_m$  from  $r_m$ , the summation of (52) needs to be carried out only within the radius of the atom.

The computation of all  $x$  derivatives can be summarized as follows. (1) Compute a three-dimensional FFT of  $D_x(\mathbf{s})$  to obtain a modified difference density function  $d_x(r)$ . (2) For each atom, integrate  $d_x(r)$  with atomic electron density function as given by (52). Note that these steps are similar to those required for fast computation of structure factors, but in reverse order. The computation required in these steps is also identical. Computation in the first step is proportional to  $N_s \log N_s$  and in the second step is proportional to the number of atoms. Steps 1 and 2 have to be repeated for each set of parameters with different  $D(\mathbf{s})$  functions which are given below:

$$D_y(\mathbf{s}) = (-i2\pi k) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})] \quad (53)$$

$$D_z(\mathbf{s}) = (-i2\pi l) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})] \quad (54)$$

$$D_B(\mathbf{s}) = (-s^2/4) W(\mathbf{s}) E(\mathbf{s}) \exp[i\varphi(\mathbf{s})]. \quad (55)$$

In computing the FFT of  $D(\mathbf{s})$  functions their space group symmetries should be properly accounted for.  $D_B(\mathbf{s})$  has space group symmetry identical to that of  $F(\mathbf{s})$ , but for many space groups, symmetries in  $D_x(\mathbf{s})$ ,  $D_y(\mathbf{s})$ , or  $D_z(\mathbf{s})$  are different. Therefore, FFT computation for these may have to be modified somewhat. For both space groups ( $P2_1$  and  $R_3$ ) for which we have programmed, the FFT computation is about the same as that for structure factor computation. Also, since the accuracy requirement in the gradient computation is not as great as in the structure factor computation, for this purpose, the single-Gaussian approximation (41) can be used and the grid size and radii of the atoms can be somewhat reduced, all resulting in reduced computation. For the single-Gaussian approximation,  $f_{\text{norm}}(\mathbf{s})$ , which is a part of  $g_m(\mathbf{s})$ , has to be lumped with  $D_x(\mathbf{s})$  so that  $\rho_m(r)$  continues to remain a Gaussian function.

### Approximation and fast computation of the normal matrix

In conventional least-squares refinement, the calculation of the normal-matrix terms accounts for most of the computation time. Therefore, it is very important to understand the nature of the normal-matrix terms and devise fast algorithms to calculate them. As shown in a previous section [equations (29)–(38)], a normal matrix can be represented as a sum of two terms; the  $H_1$  term which is independent of the phases and is a function of  $(\mathbf{r}_n - \mathbf{r}_m)$ , the vector distance between two atoms, and

the  $H_2$  term which depends on the phases and is a function of  $(\mathbf{r}_n + \mathbf{r}_m)$ . To illustrate the nature of these terms, we will consider  $H(x_m, x_n)$  the normal-matrix term corresponding to the interaction between the  $x$  parameters of the  $m$ th and  $n$ th atoms. This is written as the sum of  $H_1(x_m, x_n)$  and  $H_2(x_m, x_n)$  which are given by (31) and (32) respectively. These can be rewritten as

$$H_1(x_m, x_n) = \sum_{\mathbf{s}} A_{xx}(\mathbf{s}) g_m(\mathbf{s}) g_n(\mathbf{s}) \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_n - \mathbf{r}_m)] \quad (56)$$

$$H_2(x_m, x_n) = \sum_{\mathbf{s}} -A_{xx}(\mathbf{s}) g_m(\mathbf{s}) g_n(\mathbf{s}) \exp[i2\varphi(\mathbf{s})] \times \exp[-i2\pi\mathbf{s} \cdot (\mathbf{r}_n + \mathbf{r}_m)], \quad (57)$$

where

$$A_{xx}(\mathbf{s}) \equiv 2\pi^2 h^2 W(\mathbf{s}). \quad (58)$$

The subscripts  $xx$  denote interactions between  $x$  parameters.

$H_1(x_m, x_n)$  represents the Fourier transform of  $A_{xx}(\mathbf{s})g_m(\mathbf{s})g_n(\mathbf{s})$  evaluated at  $(\mathbf{r}_n - \mathbf{r}_m)$ , the vector distance between two atoms. Since  $A_{xx}(\mathbf{s})g_m(\mathbf{s})g_n(\mathbf{s})$  is always real and positive, its Fourier transform has a very large positive peak at the origin (corresponding to diagonal terms or  $\mathbf{r}_n - \mathbf{r}_m = 0$ ) and then drops rapidly and alternates in sign, as the distance between the atoms increases. Our test results show that even for the nearest-neighbor atoms  $H_1(x_m, x_n)$  terms are very small in magnitude compared with the diagonal terms [ $H_1(x_m, x_m)$  and  $H_1(x_n, x_n)$ ]. The  $H_1$  terms corresponding to interactions between atoms other than nearest neighbors are much smaller and can be neglected. The magnitude distribution of  $H_1$  terms is heavily concentrated near the diagonal of the matrix and drops sharply as we go away from the diagonal. This is characteristic of all  $H_1$  terms [(31), (33), (35), and (37)].

$H_2(x_m, x_n)$  represents the Fourier transform of  $-A_{xx}(\mathbf{s})g_m(\mathbf{s})g_n(\mathbf{s})\exp[i2\varphi(\mathbf{s})]$  evaluated at  $(\mathbf{r}_m + \mathbf{r}_n)$ . Since this Fourier transform involves the phase term also, unlike  $H_1(x_m, x_n)$ ,  $H_2(x_m, x_n)$  terms have no significant peaks. The magnitude distribution of  $H_2$  terms is likely to be the same in all parts of the normal matrix. Diagonal or near-diagonal  $H_2$  terms are not likely to be any bigger than off-diagonal terms. This is particularly true for large structures. These influences have been confirmed by test calculations. Therefore, unless one is calculating the entire normal matrix, it is not worthwhile to calculate any  $H_2$  terms. This argument applies to all  $H_2$  terms [(32), (34), (36), and (38)]. Therefore, in subsequent discussion, we will entirely neglect  $H_2$  terms and approximate the normal matrix by only  $H_1$  terms. The resulting inaccuracy in the normal matrix affects only the rate of convergence, not the final result.

Next we discuss fast algorithms to calculate  $H_1$  terms. For this two possibilities exist. Either we

calculate only the diagonal terms (interactions between parameters of the same atom) or we compute off-diagonal terms (interactions between parameters of two different atoms) also. In all our refinement work we used only the diagonal terms because they require much less computation, and in addition the inversion of the normal matrix becomes trivial. We will discuss both these alternatives.

### Fast computation of the diagonal terms

First, we discuss the computation of the diagonal terms only. For this case  $m = n$  and (56) becomes

$$H_1(x_m, x_m) = \sum_{\mathbf{s}} A_{xx}(\mathbf{s}) g_m^2(\mathbf{s}). \quad (59)$$

Since  $g_m(\mathbf{s})$  is modeled as a sum of two Gaussian terms (42),  $g_m^2(\mathbf{s})$  becomes a sum of three Gaussian terms as shown below:

$$g_m^2(\mathbf{s}) = \sum_{i=1}^3 c_i \exp(-b_i s^2/4), \quad (60)$$

where  $c_i$ 's and  $b_i$ 's depend on the atomic parameters and are given below:

$$\begin{aligned} c_1 &= C_m^{1^2} \\ b_1 &= 2B_m^1 + 2B_m \\ c_2 &= C_m^{2^2} \\ b_2 &= 2B_m^2 + 2B_m \\ c_3 &= C_m^1 C_m^2 \\ b_3 &= B_m^1 + B_m^2 + 2B_m \end{aligned} \quad (61)$$

Let us define

$$A'_{xx}(b_i) \equiv \sum_{\mathbf{s}} A_{xx}(\mathbf{s}) \exp(-b_i s^2/4). \quad (62)$$

Then  $H_1(x_m, x_m)$  becomes the sum of three terms

$$H_1(x_m, x_m) = \sum_{i=1}^3 c_i A'_{xx}(b_i). \quad (63)$$

We calculate and tabulate  $A'_{xx}(b_i)$  for several dummy values of  $b_i$  in the expected range of  $b_i$ 's, which can be obtained from (61). Then  $H_1(x_m, x_m)$  can be calculated using (63). For this purpose the values of  $A'_{xx}(b_i)$ , for actual values of  $b_i$ , have to be obtained by linear interpolation between two nearest tabulated dummy values.

Most of the computation in deriving  $H_1(x_m, x_m)$  terms is in computing  $A'_{xx}(b_i)$  for dummy values of  $b_i$ . This computation is proportional to the number of dummy  $b_i$  values selected. Therefore, to minimize the computation in this phase, the fewest possible dummy  $b_i$  values should be selected.  $A'_{xx}(b_i)$  is a monotonically decreasing function of  $b_i$ . For large values of  $b_i$  (say greater than 100.0)  $A'_{xx}(b_i)$  becomes very small and can be neglected. For small values of  $b_i$   $A'_{xx}(b_i)$  changes faster than for large values of  $b_i$ . Therefore, to reduce error in the linear interpolation and at the same time to

reduce the computation, we suggest that  $A'_{xx}(b_i)$  be tabulated at a small interval for small values of  $b_i$  (say an interval of 1.0 up to  $b_i = 20.0$ ) and a larger interval for large values of  $b_i$  (say an interval of 5.0 up to  $b_i = 100.0$ ).

There are some relations between  $H_1$  terms corresponding to interactions between coordinates of the same atoms:

$$\begin{aligned} \frac{H_1(x_m, x_m)}{a^2} &= \frac{H_1(y_m, y_m)}{b^2} = \frac{H_1(z_m, z_m)}{c^2} = \frac{H_1(x_m, y_m)}{abc \cos \gamma} \\ &= \frac{H_1(x_m, z_m)}{acc \cos \beta} = \frac{H_1(y_m, z_m)}{bcc \cos \alpha} \end{aligned} \quad (64)$$

where  $a$ ,  $b$ ,  $c$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  are the conventional cell parameters. These relations can be derived by assuming that the number of reciprocal-lattice points in reciprocal space are infinitely dense, thus, converting a summation in reciprocal space to an integration. For most practical situations, the above relations hold to a fairly good degree of accuracy. These relations can be used to compute other diagonal  $H_1$  terms from only one set of  $H_1$  terms [say  $H_1(x_m, x_m)$ ]. This further reduces the computation in this step. Another implication of (64) is that for an orthogonal set of coordinates, there is no interaction between different coordinates of the same atom. This is an expected result.

The expressions for the  $H_1$  terms involving the isotropic thermal parameters are similar with  $A_{xx}(s)$  replaced by

$$A_{B_B}(s) = (s^4/32)W(s) \quad (65)$$

$$A_{B_x}(s) = (1/2)(i\pi h s^2)W(s), \quad (66)$$

where the subscripts are self explanatory. The expressions for  $A_{B_y}(s)$  and  $A_{B_z}(s)$  are similar to (66) with  $h$  replaced by  $k$  and  $l$  respectively. Since  $A_{B_x}(s)$  is an odd (antisymmetric) function, it can be proved that  $H_1(B_m, x_m) = 0$ . Therefore, we only need to calculate  $H_1(B_m, B_m)$ , which can be calculated similarly to  $H_1(x_m, x_m)$  as outlined above.

The computation in calculating the diagonal terms is proportional to the number of unique reflections. Our results indicate that this computation is a very small fraction of the total least-squares computation. If one wants to reduce further this computation, the single-Gaussian approximation (41) can be used.

As pointed out before, an important advantage of the diagonal approximation is that the computation of the inverse of the normal matrix becomes trivial. In this approximation, the only terms which are not self interactions are  $H_1(x_m, y_m)$ ,  $H_1(x_m, z_m)$ ,  $H_1(y_m, z_m)$ . These are non-zero only if the unit-cell axes are non-orthogonal. Even in that case, the normal matrix consists of  $3 \times 3$  diagonal blocks which are easy to invert. If the weighting function  $W(s)$  and the thermal parameters do not change for the next refinement cycle, the diagonal

$H_1$  terms remain the same and a precomputed inverse of the normal matrix can be used.

### Fast computation of off-diagonal terms

Fast computation of off-diagonal terms is similar to the fast computation of the gradient. Equation (56) is very similar to (50) with  $D_x(s)$  replaced by  $A_{xx}(s)$ ,  $g_m(s)$  replaced by  $g_m(s)g_n(s)$ , and  $\mathbf{r}_m$  replaced by  $\mathbf{r}_n - \mathbf{r}_m$ . If the two-Gaussian approximation (42) is used for the form factors,  $g_m(s)g_n(s)$  is represented as a sum of four Gaussian functions. On the other hand if the single-Gaussian approximation (41) is used,  $g_m(s)g_n(s)$  is represented as  $f_{\text{norm}}^2(s)$  times a single Gaussian function. Computation of off-diagonal terms takes much more time than the computation of gradient or the diagonal terms, but we do not need very good accuracy in their computation. Because of these two reasons, to reduce significantly the computation time, we recommend using the single-Gaussian approximation. Our further discussion will be based on this approximation, in which,  $g_m(s)g_n(s)$  is represented by

$$g_m(s)g_n(s) = f_{\text{norm}}^2 C_m^3 C_n^3 \times \exp[-(B_m^3 + B_n^3 + B_m + B_n)s^2/4]. \quad (67)$$

Since  $f_{\text{norm}}^2$  is independent of the atoms, similar to the gradient computation, we will lump it together with  $A_{xx}(s)$ . For the purpose of this section, let us redefine  $A_{xx}(s)$  as

$$A_{xx}(s) = 2\pi^2 h^2 W(s) f_{\text{norm}}^2(s). \quad (68)$$

With this notation (56) becomes

$$\begin{aligned} H_1(x_m, x_n) &= \sum_{\mathbf{s}} A_{xx}(\mathbf{s}) C_m^3 C_n^3 \\ &\times \exp[-(B_m^3 + B_n^3 + B_m + B_n)s^2/4] \\ &\times \exp[-i2\pi \mathbf{s} \cdot (\mathbf{r}_n - \mathbf{r}_m)]; \end{aligned} \quad (69)$$

$H_1(x_m, x_n)$  is the Fourier transform of  $A_{xx}(s)$  times  $C_m^3 C_n^3 \exp[-(B_m^3 + B_n^3 + B_m + B_n)s^2/4]$  evaluated at  $(\mathbf{r}_n - \mathbf{r}_m)$ . Let  $a_{xx}(\mathbf{r})$  be the Fourier transform of  $A_{xx}(s)$ , and  $\rho_{mn}(\mathbf{r})$ , the 'joint' Gaussian electron density function of  $m$ th and  $n$ th atoms, be the Fourier transform of  $C_m^3 C_n^3 \exp[-(B_m^3 + B_n^3 + B_m + B_n)s^2/4]$ . Then by the convolution theorem,  $H_1(x_m, x_n)$  is the convolution of  $a_{xx}(\mathbf{r})$  and  $\rho_{mn}(\mathbf{r})$  evaluated at  $\mathbf{r} = \mathbf{r}_n - \mathbf{r}_m$ . Similar to (52), this can be expressed as a summation,

$$H_1(x_m, x_n) = \sum_{\mathbf{r}} a_{xx}(\mathbf{r}) \rho_{mn}(\mathbf{r} - \mathbf{r}_n + \mathbf{r}_m). \quad (70)$$

Here the summation is to be carried out over all the grid points in real space and  $(\mathbf{r} - \mathbf{r}_n + \mathbf{r}_m)$  is the distance of the grid point from the point  $(\mathbf{r}_n - \mathbf{r}_m)$ . Again as before, the joint electron density  $\rho_{mn}(\mathbf{r} - \mathbf{r}_n + \mathbf{r}_m)$  is assumed to be non-zero only for the grid points

within some radius from the point  $(\mathbf{r}_n - \mathbf{r}_m)$ . The summation of (70) needs to be carried out only within this radius.

The computation of all  $H_1(x_m, x_n)$  terms can be summarized as follows. (1) Compute a three-dimensional FFT of  $A_{xx}(\mathbf{s})$  to obtain  $a_{xx}(r)$ . (2) For each pair of atoms for which  $H_1(x_m, x_n)$  is desired, integrate  $a_{xx}(r)$  with the joint electron density function  $\rho_{mn}(r)$  as given by (70). Note that these steps are similar to those required for fast computation of gradient. The computation required in these steps is also similar. Computation in the first step is proportional to  $N_s \log N_s$  and in the second step is proportional to the number of atom pairs for which  $H_1(x_m, x_n)$  values are desired. Thus,  $r_n - r_m$  is only of the order of interatomic distances. Therefore, for the purpose of calculating the summation of (70), in real space  $a_{xx}(r)$  is needed only for a small fraction of the unit cell around the origin. This will further reduce the FFT computation of  $a_{xx}(r)$ . If  $W(\mathbf{s})$  does not change then the  $a_{xx}(r)$  function also does not change and can be used for the next cycle, but off-diagonal terms have to be recalculated if  $(\mathbf{r}_n - \mathbf{r}_m)$ ,  $B_m$ , or  $B_n$  change.

For non-diagonal terms, there is no relation similar to (64) and  $H_1(B_m, x_n)$  terms are in general non-zero. Therefore  $H_1$  terms corresponding to interactions between other parameters have to be computed similar to  $H_1(x_m, x_n)$  with  $A_{xx}(\mathbf{s})$  replaced by an appropriate function. As in the case of the gradient computation, the space group symmetry of  $A_{BB}(\mathbf{s})$  is the same as that for the space group. But for other functions [for example,  $A_{xx}(\mathbf{s})$ ,  $A_{xy}(\mathbf{s})$ ,  $A_{Bx}(\mathbf{s})$  etc.] the space group symmetries may be different and the FFT computation for these may have to be modified.

Much more computation is required for non-diagonal  $H_1$  terms than for the rest of the computation in least-squares refinement. Also for this case, the normal matrix cannot be decomposed in diagonal blocks, thus, requiring inversion by an iterative process. For these reasons, we opted for the diagonal approximation for which the convergence is somewhat slower, but perhaps the overall computation time is less.

The  $H_2$  terms can also be computed by the FFT method in a manner similar to that used for the  $H_1$  terms.

### Weighting function and the limits of the data to be used

For a successful least-squares refinement, starting from large errors in atomic coordinates, it is necessary that the data used for the refinement be properly restricted and an appropriate weighting function be used.\* This also helps in increasing the radius of convergence of the

refinement technique, making it applicable at an early stage of the structure determination process.

The least-squares refinement is based on the assumption that errors in the structure factors are linearly related to errors in the parameters of the refinement. If the linearity assumption breaks down then the least-squares refinement would not give correct results. The structure factors are related to atomic positions by a sinusoidal function  $\exp(i2\pi\mathbf{s}\cdot\mathbf{r})$  which is linearized around the position of the atom. If  $\Delta\mathbf{r}$  is the error in the position then the linearity assumption approximates  $\exp(i2\pi\mathbf{s}\cdot\Delta\mathbf{r}) = \cos(2\pi\mathbf{s}\cdot\Delta\mathbf{r}) + i\sin(2\pi\mathbf{s}\cdot\Delta\mathbf{r})$  by  $1 + i2\pi\mathbf{s}\cdot\Delta\mathbf{r}$ . The validity of this assumption depends on how close  $2\pi\mathbf{s}\cdot\Delta\mathbf{r}$  is to zero. If  $2\pi\mathbf{s}\cdot\Delta\mathbf{r}$  is greater than  $\pi/2$  then clearly the reflection is meaningless for refinement of that atom. Since we do not know the magnitude and direction of  $\Delta\mathbf{r}$  for each atom we cannot decide if that reflection should be used for refining a particular atom. But, if we know the r.m.s. (root-mean-square) error in coordinates ( $\sigma_r$ ), we can calculate the r.m.s. value of  $2\pi\mathbf{s}\cdot\Delta\mathbf{r}$  and then decide the maximum  $s$  value of the data to be used for the refinement. Assuming that  $\Delta\mathbf{r}$  is a vector with random orientation, for a three-dimensional structure it can be proved that the r.m.s. value of  $2\pi\mathbf{s}\cdot\Delta\mathbf{r}$  is  $2\pi s\sigma_r/\sqrt{3}$ . This increases linearly with  $s = (2 \sin \theta/\lambda)$ . Therefore, the linearity assumption is very good for small values of  $s$  but breaks down for high-angle terms if  $\sigma_r$  is large. For a meaningful refinement (number of unique reflections  $\gg$  number of parameters) we need at least 2 Å data ( $s_{\max} = 0.5$ ). If  $\sigma_r = 0.75$  Å and  $s_{\max} = 0.5$ , the maximum value of  $2\pi s\sigma_r/\sqrt{3}$  becomes 78°. This is about the maximum value of  $2\pi s\sigma_r/\sqrt{3}$  which can be tolerated. Thus with 2 Å data, one can refine structures with a r.m.s. positional error of 0.75 Å. In our test refinements we have achieved this radius of convergence. As the refinement progresses the value of  $\sigma_r$  will decrease and we can gradually increase the limit of data used ( $s_{\max}$ ) and towards the end of the refinement we can use all the available data. Similarly, for the refinement of isotropic thermal parameters, the linearity assumption holds better for small values of  $s$  and for small errors in the parameters.

In conventional least-squares refinement, the '2 $\sigma$ ' criterion ( $\sigma$  values of the observed data) is used to discard reflections. For protein crystallography work this means discarding a large fraction of the reflections. For most proteins, the ratio of unique reflections measured to the number of parameters is not very large and if a large number of reflections are discarded based on the 2 $\sigma$  criterion, this ratio becomes even smaller. Therefore, we suggest an alternative criterion to discard a reflection, which can be used by itself or along with the 2 $\sigma$  criterion. If the ratio of  $F_o$  to  $F_c$  (or  $F_c$  to  $F_o$ ) is very high then it indicates one of two possibilities: the value of  $F_o$  is incorrect, or because of errors in parameters  $F_c$  is highly erroneous. In any case, either  $\Delta F$  or  $\phi_c$  would

\* This was suggested by Dr R. L. Garwin.

be highly erroneous, making the reflection unsuitable for the purpose of calculating the gradient. At the beginning of the refinement the limiting ratio should be set at a high value (3 to 4) and then as the refinement progresses it should be gradually reduced to say about 2.

As mentioned above, for large errors in parameters the linearity assumption holds better for low-angle reflections. This suggests that at the beginning of the refinement the weighting scheme should give higher weights to low-angle terms and lower weights to high-angle terms. Towards the end of the refinement when the errors in the parameters are small the linearity assumption holds for all reflections and therefore they should all be unit weighted. The weighting scheme which seems to work best is of the form  $W(s) = |s|^{-n}$ . If the initial errors in the parameters are very large then an  $n$  value of 1 to 1.5 should be used. As the refinement progresses this should be gradually reduced, making it zero (corresponding to unit weights) towards the end of the refinement. This weighting scheme can be used in conjunction with weights based on  $\sigma$  values.

#### Other steps in the least-squares refinement

For the diagonal approximation of the normal matrix there is no interaction between positional and thermal parameters. Therefore we can refine coordinates ( $xyz$ ) and thermal parameters ( $B$ ) separately. There are several reasons for doing this. (1) Because of the non-linearity in high-angle terms, it is not advisable to refine  $B$  until  $\sigma_r$  has reduced appreciably. (2) Usually many fewer cycles of  $B$  refinement are required compared with  $xyz$  refinement. Therefore, on the whole this may reduce the total computation. (3) Since  $xyz$  and  $B$  are different kinds of parameters the optimum step sizes (to be discussed below) for them are different. Therefore it is difficult to determine the optimum step size while refining all parameters at the same time. For these reasons we decided to refine  $xyz$  and  $B$  in separate cycles of the refinement.

There are some other steps in the least-squares refinement and these are discussed below.

#### Scale factor calculation

To match better the relative magnitudes of the observed and calculated data, the optimum scale factor  $k$  should be determined so as to minimize  $P = \sum_s [k|F_c(s)| - |F_o(s)|]^2$ :

$$k = 1 - \frac{\sum_s |F_c(s)| [ |F_c(s)| - |F_o(s)| ]}{\sum_s |F_c(s)|^2} \quad (71)$$

Throughout a refinement cycle  $F_c$  should be multiplied by this scale factor, which should be calculated at the beginning of the cycle. Another method to calculate  $k$  is to make  $\sum_s k |F_c(s)| = \sum_s |F_o(s)|$ .

#### Modifications of the displacement vector

In conventional least-squares refinement the new parameter vector  $\mathbf{u}'$  is given by  $\mathbf{u} + \Delta\mathbf{u}$ , where  $\Delta\mathbf{u}$  is the displacement vector given by (11) and  $\mathbf{u}$  is either the coordinates vector or the thermal parameter vector. Because of the nonlinearity and the approximation of the normal matrix, this  $\Delta\mathbf{u}$  may not be the optimum displacement. If errors in the parameters are large, this may even lead to instability. For fast convergence of the refinement and to increase its radius of convergence, we suggest the following modifications to the displacement vector.

#### Truncation of the displacement vector

If an atom has a large  $B$  value, its diagonal normal-matrix terms are small and for a diagonal approximation of the normal matrix, the shifts in its parameters may be large. These large shifts may lead to instability or oscillations. The least-squares procedure is not valid for large displacements and they should be avoided. If an atom needs a large shift, it is safer to achieve this by small shifts in many cycles. For each cycle of refinement, we decide that the shift for any atom will not be more than a 'ratio' times the r.m.s. shift for all the atoms. If the calculated shift for any atom is more than this maximum, its magnitude is truncated to the maximum allowed. In the initial stages of the refinement, when errors in the parameters are large, shifts also tend to be large. Since we do not want any atom to move 'too far' from its position, we set this ratio 'tight' (1.5 to 2.0). In the final stages of the refinement most of the atoms have refined fairly well and do not move much, making the r.m.s. shifts small. At this stage we make the ratio loose (3 to 4) so that poorly refined atoms or atoms with reassigned positions can move towards their 'correct' positions.

For  $xyz$  refinement we consider the magnitude of the displacement vector  $|\Delta\mathbf{r}|$  for each atom and set  $|\Delta\mathbf{r}|_{\max} < \text{ratio} \times |\Delta\mathbf{r}|_{\text{r.m.s.}}$ . For  $B$  refinement, atoms with large  $B$ 's tend to have large  $\Delta B$ 's and those with small  $B$ 's tend to have small  $\Delta B$ 's. Therefore a meaningful criterion is to consider  $|\Delta B/B|$  and set  $|\Delta B/B|_{\max} < \text{ratio} \times |\Delta B/B|_{\text{r.m.s.}}$ .

#### Optimum step size calculation

Let  $\Delta\mathbf{u}'$  be the truncated version of  $\Delta\mathbf{u}$ , after truncating large shifts. Even this may not be the optimum

displacement. A displacement of the form  $\alpha \Delta u'$  will be optimum, where  $\alpha$  is the optimum step size (a scalar and not to be confused with the cell parameter  $\alpha$ ) which minimizes  $P(\mathbf{u} + \alpha \Delta u')$ , the minimization function. The optimum step size has to be obtained by a linear search (Luenberger, 1973) along the search direction. At this point we introduce another notation which will be helpful for the next subsection. We denote the search vector by  $\mathbf{S}$ . For the time being  $\mathbf{S}$  is taken as  $\Delta u'$ . Our displacement will be along the search direction  $\mathbf{S}$  with a step size  $\alpha$ , to be determined by a linear search so as to minimize  $P$ . All this can be expressed as follows

$$\mathbf{S} = \Delta u' \tag{72}$$

$$\mathbf{u}' = \mathbf{u} + \alpha \mathbf{S} \tag{73}$$

where  $\mathbf{u}'$  is the new parameter vector and  $\alpha$  is chosen to minimize  $P(\mathbf{u}')$ .

The minimization function  $P$  plotted as a function of  $\alpha$  typically looks like Fig. 1. For small values of  $\alpha$  the function can be assumed to be quadratic. A quadratic function can be completely specified by three parameters. One of the parameters is the original value of the function  $P_0$  (corresponding to  $\alpha = 0$ ), the second parameter is chosen as the slope of the function at  $\alpha = 0$  which is the slope of the line  $P_0A$  and is given by the inner product of the gradient vector and the search direction vector:

$$\text{SLOPE}_0 = \mathbf{G} \cdot \mathbf{S}. \tag{74}$$

For a non-orthogonal coordinate system, care should be taken to include cross-terms also in computing (74). The third parameter is obtained by calculating the function  $P_1$  corresponding to a step size  $\alpha_1$  (point  $B$  on the curve). This involves calculating the structure factors with parameters  $\mathbf{u}' = \mathbf{u} + \alpha_1 \mathbf{S}$ . With these three parameters we can reconstruct the parabolic function and obtain its minimum  $P_{\text{opt}}$  corresponding to the

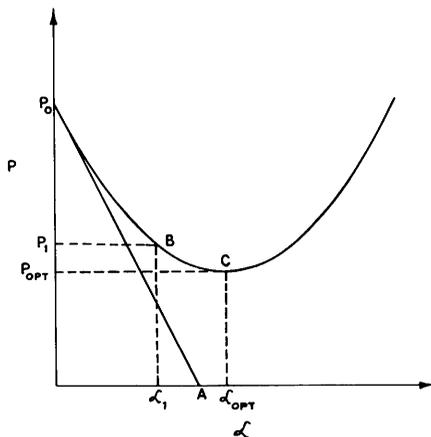


Fig. 1. Plot of the minimization function  $P$  vs the step size  $\alpha$ .  $P_0A$  is the tangent to the function at  $\alpha = 0$ .

optimum step size  $\alpha_{\text{opt}}$  (point  $C$  on the curve). Let  $\text{SLOPE}_1$  be the slope of the line  $P_0B$ .

$$\text{SLOPE}_1 = (P_0 - P_1)/\alpha_1. \tag{75}$$

Then we define a relative slope by

$$\text{RSLOPE}_1 = \text{SLOPE}_1/\text{SLOPE}_0, \tag{76}$$

and it can be proved that the optimum step size is given by

$$\alpha_{\text{opt}} = \frac{\alpha_1}{2(1 - \text{RSLOPE}_1)}. \tag{77}$$

For a very small step size the relative slope is close to 1.0 and for the optimum step size it is 0.5. The initial step size  $\alpha_1$  should be chosen close to the optimum step size, which is around 1.0. If the initial step size is too far from the optimum step size, we recommend that another trial step size, close to the optimum step size, be chosen and the above calculation be repeated. As shown in Fig. 1, in the neighborhood of the optimum step size the function is very shallow. Therefore, we recommend taking the actual step size  $\alpha_{\text{act}}$  (for the purpose of calculating the parameters for the next refinement cycle) as about two thirds of  $\alpha_{\text{opt}}$ . With this step size we avoid large shifts and at the same time achieve about 91% of the maximum possible drop in the minimization function. Corresponding to this step size, the relative slope is about 0.67.

### Conjugate directions

For the diagonal approximation of the normal matrix, this refinement method behaves somewhat like the steepest-descent method (Luenberger, 1973). Therefore, in the first few cycles, this method gives a very good drop in the minimization function, but after that the convergence becomes slow. Also, the optimum step size deviates widely from cycle to cycle. If many cycles of the refinement are to be carried out without idealization of the structure or manual adjustment in the parameters, it is advisable to use some kind of conjugate-directions method (Luenberger, 1973), to increase the rate of convergence. In the conjugate-directions method the search direction vector  $\mathbf{S}$  is modified to include part of the search direction vector from the previous refinement cycle:

$$\mathbf{S} = \Delta u' + \beta \mathbf{S}', \tag{78}$$

where  $\mathbf{S}'$  is the search direction vector from the previous cycle and  $\beta$  (not to be confused with the cell parameter  $\beta$ ) is a scalar which can be calculated by many different methods. The definition of  $\beta$  which seems to work in our refinement is

$$\beta = \frac{\mathbf{S}' \cdot \mathbf{S}'}{\Delta u' \cdot \Delta u'}. \tag{79}$$

This is the square of the ratio of the norm of the old search direction vector to the norm of the present  $\Delta u'$ . In addition, we put a constraint that  $\beta$  be no greater than 0.4. This ensures that the present search direction vector is not greatly influenced by previous search direction vectors. The optimum step size is calculated using the modified search direction vector of (78). The conjugate-directions method should not be used ( $S'$  assumed to be zero) in the following situations: if the parameters are changed manually or by some other program or if the previous refinement cycle was on parameters of different type (for example,  $xyz$  refinement followed by a  $B$  refinement).

### Results on hypothetical test structures in space group $P1$

Before using the method on real data, we gained experience of the method by using it on hypothetical test structures in space group  $P1$  with orthogonal axes. The test structures were generated by a program of Sayre (1972). They obeyed simple requirements on bond lengths and distances between non-bonded atoms. All atoms had Gaussian electron density with atomic form factors (including thermal parameters) of the form  $\exp(-Bs^2/4)$  with  $B$  having a uniform distribution in the range of 6.0 to 18.0 with an average value of 12.0. In all the test refinements all structure factors up to a given  $s_{\max}$  value were used, the scale factor was not varied and a value of unity (the correct value) was used. A weighting function of the form  $W(s) = |s|^{-n}$  was used, with the initial value of  $n$  being 1.5 which was gradually reduced to zero, the diagonal approximation of the normal matrix was used, and as discussed before the maximum displacement of an atom was restricted

to twice the r.m.s. displacement. To test the method, at the beginning of the refinement the starting coordinates were those of the test structure plus random positional errors for all atoms. For all the cases, the initial r.m.s. error  $|\Delta r|_{\text{r.m.s.}}$  was about 0.7 Å while the initial peak error  $|\Delta r|_{\text{peak}}$  varied from 1.0 Å to 1.24 Å. Throughout the refinement, none of the positions were changed manually or by any other program. In all cases, this method moved all the atoms close to their correct positions, indicating a radius of convergence of at least 0.7 Å for the method. This is significantly higher than for a conventional method.

The results on some of the test cases are summarized in Table 7. Most of the entries in the table are self-explanatory. For the purpose of this table the  $R$  factor (the agreement index or residual) is defined as

$$R = \frac{\sum_s | |F_o(s)| - |F_c(s)| |}{\sum_s |F_o(s)|} \quad (80)$$

The CPU time per cycle quoted is on an IBM 370/168 with virtual memory. In cases 5 through 8, the refinement was still progressing when it was terminated because it had served its purpose. Next, we discuss this test experience in detail.

First, we discuss cases 1 through 4. For all these cases, correct  $F_o$  values (calculated from the correct positions of the atoms) and correct  $B$  values were used for the refinement, only atomic positions were refined, and conjugate directions were not used. Case 1 is for a 40-atom structure with 1 Å data and is typical of a small structure. For this case also the method worked successfully, starting with very large initial errors, showing its effectiveness on a small structure. Case 3 is

Table 7. Summary of some results on hypothetical test structures in space group  $P1$

See the text for a detailed discussion of these cases.

Case no.	1	2	3	4	Initial values			8	9	Refined values <sup>a</sup>		
					Number of atoms	Resolution of data $1/s_{\max}$ (Å)	Number of unique reflections			$ \Delta r _{\text{r.m.s.}}$ (Å)	$ \Delta r _{\text{peak}}$ (Å)	$R$ factor
1	40	1.0	2147	0.721	1.009	0.546	15	23	0.025	0.093	0.018	
2	100	1.5	1523	0.719	1.170	0.500	20	22	0.019	0.087	0.008	
3	400	1.5	6156	0.702	1.160	0.541	21	67	0.020	0.125	0.009	
4	400	2.0	2610	0.702	1.160	0.450	25	49	0.087	0.312	0.018	
5	100	1.5	1523	0.712	1.240	0.502	13	22	0.068	0.289	0.038	
6 <sup>b</sup>	100	1.5	1523	0.712	1.240	0.502	13	22	0.038	0.210	0.019	
7 <sup>c</sup>	100	1.5	1523	0.712	1.240	0.526	10	22	0.171	0.746	0.091	
8 <sup>d</sup>	100	1.5	1523	0.712	1.240	0.484	12	22	0.074	0.366	0.042	

Notes: (a) In many cases the refinement was still progressing when it was terminated. (b) This case is same as case 5 but with the use of conjugate directions, also used for all the cases following it. (c) In this case initial  $F_o$  values were incorrect. After six cycles of the refinement correct  $F_o$  values were used. (d) In this case initially all atoms were assigned the average  $B$  value. After nine cycles of the refinement correct  $B$  values were used.

typical of a small protein. Case 4 is of particular interest for proteins with limited data. In this case, with only 2 Å data, the method was able to refine all atomic positions within tolerable errors in the coordinates. For this case although the final  $R$  factor was less than 2%, the positional errors were much higher than for cases 1 through 3. This shows the problem of limited data in protein crystallography. In such situations, since the ratio of the number of unique reflections to the number of parameters is low, even though the  $R$  factor can be reduced to a fairly low value, the corresponding errors in the parameters tend to be relatively high.

Cases 5 and 6 show the effect of the conjugate-directions method. These cases are similar to cases 1 through 4 in all respects except that conjugate directions were used for case 6. Starting with identical initial errors, after 13 cycles of the refinement, case 6 achieved lower errors and a lower  $R$  factor compared with case 5. Also for cases 1 through 5, where the conjugate-directions method was not used, the optimum step size fluctuated widely, while for case 6 it was fairly close to unity. Because of these reasons, in all our further work we used the conjugate-directions method wherever applicable.

Case 7 shows the effect of errors in observed diffraction data. For this case, we used incorrect data,  $F'_o$ , which were obtained from the correct data,  $F_o$ :

$$F'_o(\mathbf{s}) = F_o(\mathbf{s}) \times (1 + |\mathbf{s}| \times \text{ERROR}), \quad (81)$$

where ERROR is a uniformly distributed random variable with zero mean and a  $\sigma$  value of 0.4. The above relation introduces more error in high-angle reflections compared with low-angle reflections. The  $R$  factor of  $F'_o$  [obtained by replacing  $F_c$  by  $F'_o$  in (80)] was 0.147. After six cycles of the refinement with the incorrect data,  $|\Delta r|_{\text{r.m.s.}}$ ,  $|\Delta r|_{\text{peak}}$ , and the  $R$  factor decreased to 0.362 Å, 0.852 Å, and 0.229 respectively. At that point all these parameters were consistently decreasing and they would have decreased much further. This shows that a meaningful refinement can be carried out with somewhat incorrect data. After cycle 6, correct  $F_o$  were used. The refined values indicated in Table 7 are after four more cycles of the coordinate refinement. The use of correct  $F_o$  increased the pace of the refinement. At this stage also the refinement was progressing smoothly.

Case 8 shows the effect of incorrect  $B$  values on the refinement. For this case, initially all atoms were assigned the average  $B$  value of 12.0 and only the positional refinement was carried out using correct  $F_o$  values. The  $R$  factor due to incorrect  $B$  values was 0.127 ( $F_c$  calculated with correct positions and incorrect  $B$  values). After nine cycles of the refinement,  $|\Delta r|_{\text{r.m.s.}}$ ,  $|\Delta r|_{\text{peak}}$ , and the  $R$  factor had decreased to 0.149 Å, 0.555 Å, and 0.135 respectively, and were decreasing consistently. This shows that a meaningful

positional refinement can be carried out with only average  $B$  values and in the initial phase of the refinement the  $B$  refinement is not necessary. As discussed before, the  $B$  refinement with large positional errors could actually lead to instability. After cycle 9, correct  $B$  values were used which immediately reduced the  $R$  factor to 0.094. The refined values indicated in Table 7 are after three more cycles of the coordinate refinement. Again the correct  $B$  values increased the pace of the refinement.

In case 9 (not shown in the table),  $B$  values were also refined in addition to the coordinates. For this case the initial parameters were identical to those for case 8. But, in this case, later in the refinement, instead of using the correct  $B$  values we refined  $B$  values also. The progress of this refinement is shown in Fig. 2. Cycles 7, 11, 12, 16, 17 and 20 were  $B$  refinement and the remaining 15 cycles were the coordinate refinement. The initial values of  $|\Delta B|_{\text{r.m.s.}}$  and  $|\Delta B|_{\text{peak}}$  were 3.55 and 6.0 respectively. After cycle 21, the values of  $|\Delta r|_{\text{r.m.s.}}$ ,  $|\Delta r|_{\text{peak}}$ ,  $|\Delta B|_{\text{r.m.s.}}$ ,  $|\Delta B|_{\text{peak}}$  and the  $R$  factor were 0.04 Å, 0.20 Å, 0.29, 1.39 and 0.017 respectively. In this case, the conjugate-directions method was not used for cycles 7, 8, 11, 13, 16, 18, 20 and 21, because the refinement cycles previous to these were on parameters of different type.

In case 10, we refined both coordinates and  $B$  values using incorrect data. The starting atomic parameters for this case were the same as in case 9. But, for the purpose of refinement,  $F'_o$  values (81), with the  $\sigma$  value

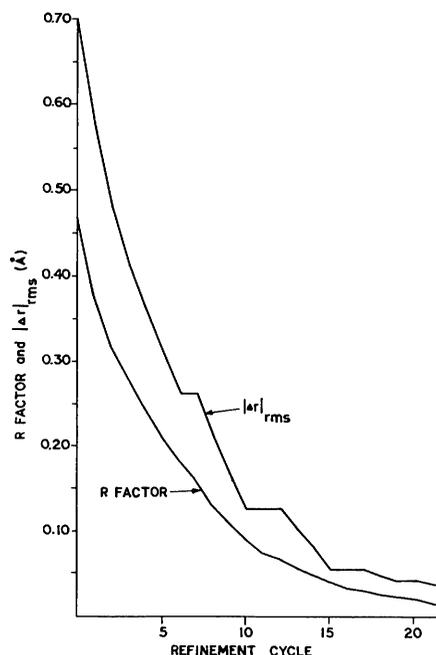


Fig. 2. Progress of the test refinement in case 9. Horizontal segments of the  $|\Delta r|_{\text{r.m.s.}}$  curve indicate  $B$  refinement.

of ERROR being 0.3, were used. The  $R$  factor of  $F'_o$  was 0.11. The progress of the refinement is shown in Fig. 3. Cycles 6, 10, 14, 19, 20 and 24 were  $B$  refinement and the remaining 18 cycles were coordinate refinement. At the end of cycle 16,  $|\Delta r|_{r.m.s.}$ ,  $|\Delta r|_{peak}$  and the  $R$  factor had reduced to 0.152 Å, 0.586 Å and 0.122 respectively. This value of the  $R$  factor is very close to the  $R$  factor of the incorrect data. This indicates that the refinement can be carried out to the limits of the accuracy of the data. This is a very encouraging result. After cycle 16, we used correct  $F'_o$  values. After eight more cycles of the refinement the values of  $|\Delta r|_{r.m.s.}$ ,  $|\Delta r|_{peak}$ ,  $|\Delta B|_{r.m.s.}$ ,  $|\Delta B|_{peak}$  and the  $R$  factor were 0.062 Å, 0.367 Å, 0.82, 4.08 and 0.033 respectively. If we had carried out the refinement further, all these parameters would have significantly decreased. In this case the conjugate-directions method was not used for cycles 6, 7, 10, 11, 14, 15, 17, 19, 21 and 24.

### Programming in space group $R3$

We programmed the new algorithm in space group  $R3$ . These programs were used for the refinement of insulin (Isaacs & Agarwal, 1978) and for a test refinement of a small structure. Here we mention some of the main features of the  $R3$  programs.

For these programs, we used the hexagonal set of coordinates ( $a = b$ ,  $\alpha = \beta = 90^\circ$ ,  $\gamma = 120^\circ$ ) and selected  $(0 - \frac{1}{3}, 0 - \frac{1}{3}, 0 - 1)$  as the asymmetric unit. For the

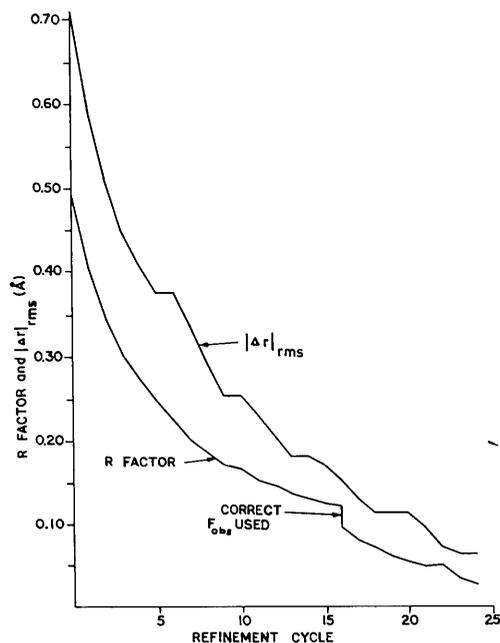


Fig. 3. Progress of the test refinement in case 10. Horizontal segments of the  $|\Delta r|_{r.m.s.}$  curve indicate  $B$  refinement. Incorrect  $F'_o$  were used up to cycle 16.

purpose of electron density computation, the grid points were selected parallel to the axes and included the  $xy$  edges of the asymmetric unit ( $x = 0$ ,  $y = 0$ ,  $x = \frac{1}{3}$ ,  $y = \frac{1}{3}$ ). Part of the electron density contribution of the atoms lying close to the  $xy$  edges may fall outside the asymmetric unit. By applying proper symmetry operations, this contribution is added to equivalent grid points within the asymmetric unit. Since the atoms on the  $z$  axis have only three symmetry positions in the unit cell, compared with nine for other atoms, they were assigned occupancies of  $3/9$  to account for this. The two-Gaussian approximation of the atomic form factors was used for the purpose of structure factor computation. In the FFT computation of electron density, the first transform was computed along the  $z$  axis for all grid points in the asymmetric unit. After this step, the transform points beyond  $l_{max}$  were discarded. Next, for each  $l$  index, transforms in the  $xy$  plane were computed, making full use of the symmetries. This approach minimizes the storage and computation requirement. The data expansion, beyond the asymmetric unit, is only for a plane at a time.

For the gradient computation, the single-Gaussian approximation of the atomic form factors was used. In  $R3$ ,  $D_z(s)$  and  $D_B(s)$  [(54) and (55)] have symmetries of  $R3$  and their FFT computation is identical to the electron density map calculation in  $R3$ . For this purpose the steps outlined above can be repeated in the reverse order. But,  $D_x(s)$  and  $D_y(s)$  do not have the symmetries of  $R3$ . Therefore the FFT routines for these have to be modified somewhat. In addition, this creates a problem for atoms lying close to the  $xy$  edges of the asymmetric unit. In this situation the grid points outside the asymmetric unit cannot be brought inside by applying symmetry operations. Therefore  $d_x(r)$  and  $d_y(r)$ , the Fourier transforms of  $D_x(s)$  and  $D_y(s)$ , have to be calculated for  $-\delta \leq x \leq \frac{1}{3} + \delta$ ,  $-\delta \leq y \leq \frac{1}{3} + \delta$ ,  $0 \leq z < 1$ , where  $\delta$  is the width of the rim around the asymmetric unit and is set approximately equal to the maximum diameter of the atoms. This is shown graphically in Fig. 4. Therefore, for  $D_x(s)$  and  $D_y(s)$ , after computing the transform in the  $xy$  plane, the 'electron density' was retained for the larger area

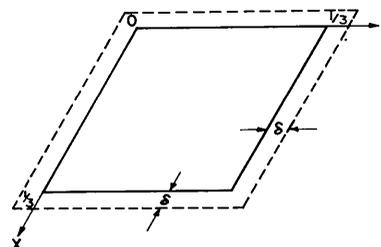


Fig. 4.  $xy$  plane of the asymmetric unit in  $R3$ . Solid heavy lines indicate the boundaries of the asymmetric unit and dashed lines indicate the boundaries of the rim used for the purpose of gradient computation;  $\delta$  is the width of the rim.

(shown by dashed lines in Fig. 4) and the transforms along the  $z$  axis were computed for all grid points within this area. The additional computation because of this is very small.

The single-Gaussian approximation was also used for computing the diagonal normal-matrix terms. The interaction between the  $x$  and  $y$  terms of the same atom was accounted for. Their fast computation was carried out according to (62) and (63). Since the single-Gaussian approximation was used, the summation in (63) consisted of only one term. The multiplicities of the symmetries were accounted for by multiplying (62) by an appropriate constant.

#### Test refinement of a small structure

We used the *R3* programs to test the radius of convergence of the method on a small structure, 6-acetyldolatriol,  $C_{22}H_{34}O_4$ . The hexagonal cell parameters for this structure are  $a = 24.124$  and  $c = 9.552$  Å. This structure has been solved by Von Dreele (1977) using anisotropic least-squares refinement. He kindly supplied us with the X-ray diffraction data and the anisotropically refined atomic parameters. Although the diffraction data extended up to  $s_{\max} = 1.69$  (0.77 Å resolution data), the  $R$  factor *vs*  $s$  curve for the anisotropically refined structure appeared most consistent only up to  $s = 1.0$ . Therefore, for our refinement we used only 1 Å data ( $s_{\max} = 1.0$ ). The anisotropically refined structure had an  $R$  factor of 0.07 for all 1054 reflections up to  $s_{\max} = 1.0$ . For the test refinement, we introduced random errors ( $|\Delta r|_{r.m.s.} = 0.75$  Å,  $|\Delta r|_{\text{peak}} = 1.27$  Å) in the positions of all non-hydrogen atoms of the structure, assigned the average  $B$  values to these atoms, and removed all hydrogen atoms from the structure. Because of the large errors, the starting model bore no resemblance to the actual structure and did not even look like a model of an atomic structure. In this structure, some of the bonds were as short as 0.64 Å, some atoms had as many as five atomic contacts of less than 2 Å, three of those with non-bonded atoms, many of the bonded atoms were more than 2 Å apart. We refined this structure initially using  $s_{\max} = 0.55$  (1.82 Å data) and as the refinement progressed this was gradually increased to 1.0 (1.0 Å data). The initial  $R$  factor was 0.475 for the 1.82 Å data and 0.5 for the 1 Å data. After ten cycles of the  $xyz$  refinement and one of  $B$ , the  $R$  factor for 1041 reflections ( $s_{\max} = 1.0$ , limiting value of  $|F_o|/|F_c| = 4.0$ ) dropped to 0.138. At this stage, we included 34 hydrogen atoms at appropriate positions, which reduced the  $R$  factor to 0.129. Three more cycles of the  $xyz$  refinement and one of  $B$  reduced the  $R$  factor to 0.096, to be compared with the value of 0.07 for the anisotropically refined structure. The refinement converged to essentially the same positions. This test shows that

the structures with very large initial errors could be refined without any human intervention as long as all non-hydrogen atoms are included in the model. The average CPU time per cycle for the refinement was 20 s on an IBM 370/168.

#### Use of the method to obtain a higher-resolution protein map

The large radius of convergence of the method can be used to obtain a high-resolution protein map starting from a low-resolution map. The method consists in refining by this least-squares method the positions and thermal parameters of a set of dummy atoms placed in the initial low-resolution electron density map. Phases calculated from these refined atomic positions are used to extend the resolution and improve the quality of the electron density map. The large radius of convergence, together with the severe restrictions placed on the initial positions of the dummy atoms by the requirement that they lie within limited regions of the isomorphous electron density map account for the success of the method. The method has been successfully used to phase the structure factors of 2-zinc insulin at a resolution of 2 Å and 1.5 Å, starting from a set of isomorphous phases at 3 Å resolution. The details of the method are covered in the paper by Agarwal & Isaacs (1977).

#### Refinement of the insulin structure at 1.5 Å resolution

The *R3* programs were also used to refine the crystal structure of 2-zinc insulin at 1.5 Å resolution. The details of the refinement are contained in the paper by Isaacs & Agarwal (1978). Rhombohedral 2-Zn insulin crystallizes in the space group *R3*, with hexagonal cell parameters  $a = 82.5$  and  $c = 34.0$  Å. There are two zinc atoms and two insulin molecules each of MW 5780 daltons in the asymmetric unit. The stoichiometric solvent content of the asymmetric unit is equivalent to 280 water molecules. Data to a resolution of 1.5 Å were available. For the refinement, the initial model consisted of 853 non-hydrogen atoms including 74 solvent atoms. The refinement gave consistent convergence from an initial  $R$  factor of 0.282 for 6572 reflections at 1.83 Å resolution, to a final  $R$  factor of 0.113 for 11890 terms (limiting ratio of  $|F_o|/|F_c| = 1.8$ ) to 1.5 Å resolution (0.148 for all 13424 terms). The final refined model consisted of all 813 non-hydrogen protein atoms (including 2 zinc atoms and 10 atoms assigned half occupancy), 264 solvent atoms (of which 82 were assigned half occupancy), and 749 hydrogen atoms. The average CPU time per cycle was 3 min on an IBM 370/168.

### Programming in space group $P2_1$

We programmed the method in space group  $P2_1$  also. These programs were used for the refinement of beauricin barium complex and myoglobin. Here we mention some of the main features of the  $P2_1$  programs. We used the two-Gaussian approximation for all calculations and selected  $(1, \frac{1}{2}, 1)$  as the asymmetric unit. For atoms close to the  $y = \frac{1}{2}$  plane, the part of the electron density contribution lying outside the asymmetric unit was added to equivalent grid points within the asymmetric unit, by applying the symmetry operation  $(\bar{x}, \frac{1}{2} + y, \bar{z})$ . In  $P2_1$ ,  $D_y(s)$  and  $D_B(s)$  have symmetries of  $P2_1$  and their Fourier transforms are computed in a normal manner. But  $D_x(s)$  and  $D_z(s)$  do not have the symmetries of  $P2_1$ ; therefore the FFT computation for these had to be modified somewhat. Both  $D_x(s)$  and  $D_z(s)$  have twofold symmetries, but of different types. Their Fourier transforms have the symmetry relation,  $d_x(x, y, z) = -d_x(\bar{x}, \frac{1}{2} + y, \bar{z})$ , (similar expression for  $d_z$ ). Therefore, in the computation of  $G(x_m)$  and  $G(z_m)$  (52), for points outside the asymmetric unit,  $d_x(r)$  and  $d_z(r)$  were obtained by the above relation. The normal-matrix terms corresponding to the interaction between the  $x$  and  $z$  coordinates of the same atom were also calculated.

### Refinement of the barium beauricin complex at 1.2 Å resolution

The barium complex of the antibiotic beauricin (space group  $P2_1$ , data supplied by Dr Geddes, Leeds University) has been refined by this procedure and in an independent study by Dr Geddes using conventional methods. The manuscript comparing results is in preparation. The asymmetric unit contains 370 non-hydrogen atoms (628 total) and there are 7352 unique reflections in the 1.2 Å data set used. The starting model for this refinement had an  $R$  factor of 0.206 (7178 reflections, limiting ratio of  $F_o/F_c = 4.0$ ) and the conventional individual-atom least-squares refinement had failed. This refinement procedure was used on this model in a straightforward fashion, six cycles of  $xyz$  and seven of  $B$  refinement brought the  $R$  factor down to 0.13 (6677 reflections, limiting ratio of  $F_o/F_c = 4.0$ ,  $2\sigma$  criterion used). Adding two solvent molecules (benzene), obvious from the  $F_o - F_c$  map, further refinement with eight  $xyz$  and three  $B$  cycles (with regularization before the last two cycles) produced an  $R$  factor of 0.121. Randomizing these coordinates with the introduction of a r.m.s. error of 0.15 Å (peak error of 0.25 Å) five times, followed by three  $xyz$  cycles each, brought the atoms to the original positions with an average (over all atoms) r.m.s. (over five cycles) difference of 0.029 Å. The refinement times for each cycle were 59 s for  $xyz$  and 45 s for  $B$  on an IBM 370/168. After

increasing the sampling interval, by using the method of Ten Eyck (1977),  $xyz$  refinement time was 38 s per cycle.

### Preliminary refinement of myoglobin at 2 Å resolution

Preliminary results with sperm whale myoglobin (~1400 atoms, ~9000 reflections to 2 Å resolution, data supplied by Dr T. Takano, MRC, Cambridge) show that the method is very effective with more limited data also. Starting with the coordinates refined by Takano (1977) using the Diamond method (1966, 1971, 1974), four  $xyz$  and three  $B$  refinement cycles reduced the  $R$  factor from 0.265 to 0.165, with very little distortion. The r.m.s. shift for all the atoms including solvent was about 0.26 Å. The refinement time was about 80 s per  $xyz$  cycle on an IBM 370/168. Use of the Ten Eyck (1977) method to increase the sampling interval could reduce this time to 55 s per cycle and extrapolations indicate that times for 1.5 Å data (~19 000 reflections) and 1.2 Å data (~37 000 reflections) would be about 2 and 4.2 min respectively.

### Summary and conclusions

In this paper a new least-squares atomic-parameter refinement technique has been presented for which the computation requirement is  $K_1 N \log N + K_2 M$ , where  $K_1$  and  $K_2$  are constants, and  $N$  and  $M$  are numbers of reflections and atoms respectively. The technique uses the fast Fourier transform algorithm (FFT) at all stages of the computation,  $K_1 N \log N$  represents the FFT computation and  $K_2 M$  represents the computation required in setting up the electron density array *etc.* The tests of the method indicate a radius of convergence of approximately 0.75 Å, which is considerably greater than that for the conventional least-squares refinement method. The method has been successfully applied for the refinement of insulin at 1.5 Å resolution (Isaacs & Agarwal, 1978) and barium beauricin complex at 1.2 Å resolution (manuscript in preparation), and has also been used for preliminary refinement of myoglobin at 2 Å resolution.

Our experience with the method indicates that it is not very effective in refining atoms with high  $B$  values. The reason for this is that the scattering contribution of high- $B$  atoms drops very sharply with the  $s$  value. The scattering intensity [ $g^2(s)$ ] of a carbon atom with a  $B$  value of 30 drops to 0.56% (at 2 Å resolution) and 0.02% (at 1.5 Å resolution) of its value at  $s = 0$ . The corresponding numbers for an atom with a  $B$  value of 5 are 13% (at 2 Å resolution) and 5% (at 1.5 Å resolution). Thus, in effect high- $B$  atoms are refined using only low-resolution data. This makes a very strong case for collecting low-temperature X-ray diffraction data. This has two advantages. First, at low temperatures the

$B$  values are lower, making it easier to refine a structure. Second, at low temperatures the data extends to a higher resolution thus increasing the ratio of the number of observations to parameters, leading to a higher accuracy in refined parameters. The tests of the method also indicate that the accuracy of the refinement is largely limited by the accuracy of the data available. Therefore, we also recommend that the data be collected on the fewest possible number of crystals and over the shortest possible time, thus leading to a more consistent data set.

I am very grateful to Drs Richard Garwin and David Sayre for introducing me to the wonderful world of structure determination using X-ray crystallography. I also thank them for continued guidance, encouragement, and many useful suggestions. Thanks are also due to Drs Karl Hardman and Neil Isaacs for many fruitful discussions and suggestions. Professor Dorothy Hodgkin of Oxford University and Dr Guy Dodson of York University provided the moral support and the data for insulin refinement. Dr. A. J. Geddes of the University of Leeds provided the data on barium beauricin complex, Dr T. Takano of Cambridge University provided the data on myoglobin, and Dr R. B. Von Dreele of the University of Arizona provided the data on 6-acetyldolatriol.

*Acta Cryst.* (1978). A34, 809–811

## Correlation between Third Cumulants in the Refinement of Noncentrosymmetric Structures

By R. G. HAZELL AND B. T. M. WILLIS\*

*Department of Inorganic Chemistry, Aarhus University, DK-8000 Aarhus C, Denmark*

(Received 16 August 1977; accepted 5 April 1978)

It is shown that correlation between the third-cumulant coefficients in noncentrosymmetric structures restricts the number of coefficients which can be refined. In the space group  $P1$  all ten coefficients of one atom have to be kept fixed.

With the accurate neutron diffraction data that is now becoming available, it is possible to refine third- (and even fourth) cumulant coefficients. This note is concerned with the restriction on the number of refinable third-cumulant coefficients, arising from correlation between those coefficients in noncentrosymmetric structures. The symmetry restrictions on cumulants for atoms at special positions are tabulated in

\*Permanent address: Materials Physics Division, AERE Harwell, Didcot, Oxfordshire OX11 0RA, England.

## References

- AGARWAL, R. C. & ISAACS, N. W. (1977). *Proc. Natl Acad. Sci. USA*, **74**, 2835–2839.
- COOLEY, J. W. & TUKEY, J. W. (1965). *Math. Comput.* **19**, 297–301.
- DIAMOND, R. (1966). *Acta Cryst.* **21**, 253–266.
- DIAMOND, R. (1971). *Acta Cryst.* A**27**, 436–452.
- DIAMOND, R. (1974). *J. Mol. Biol.* **82**, 371.
- FORSYTH, J. B. & WELLS, M. (1959). *Acta Cryst.* **12**, 412–415.
- International Tables for X-ray Crystallography* (1959). Vol. II. Birmingham: Kynoch Press.
- International Tables for X-ray Crystallography* (1974). Vol. IV. Birmingham: Kynoch Press.
- ISAACS, N. W. & AGARWAL, R. C. (1978). *Acta Cryst.* A**34**, 782–791.
- LIPSON, H. & COCHRAN, W. (1966). *The Determination of Crystal Structure*, 3rd ed. London: Bell.
- LUENBERGER, D. G. (1973). *Introduction to Linear and Non-linear Programming*. Reading, Mass.: Addison-Wesley.
- ROLLETT, J. S. (1965). *Computing Methods in Crystallography*. Oxford: Pergamon Press.
- SAYRE, D. (1951). *Acta Cryst.* **4**, 362–367.
- SAYRE, D. (1972). *Acta Cryst.* A**28**, 210–212.
- TAKANO, T. (1977). *J. Mol. Biol.* **110**, 537–568.
- TEN EYCK, L. F. (1973). *Acta Cryst.* A**29**, 183–191.
- TEN EYCK, L. F. (1977). *Acta Cryst.* A**33**, 486–492.
- VAND, V., EILAND, P. F. & PEPINSKY, R. (1957). *Acta Cryst.* **10**, 303–306.
- VON DREELE, R. B. (1977). *Acta Cryst.* B**33**, 1047–1052.

*International Tables for X-ray Crystallography* (1974) (*IT*) and by Birss (1964) (who works in a Cartesian coordinate system).

The structure factor equation including third cumulants is of the form

$$F(\mathbf{H}) = \sum_{\kappa=1}^n f_{\kappa} \exp(2\pi i x_{\kappa} h_i - b_{ij} h_i h_j - ic_{ijk} h_i h_j h_k), \quad (1)$$

where  $\mathbf{H}$  is the diffraction vector, and  $h_i$  ( $i = 1, 2, 3$ ) are the Miller indices. If the third-cumulant coefficients  $c_{ijk}$