Computer Programs

J. Appl. Cryst. (1989). **22**, 500-506

FROG – high-speed restraint–constraint refinement program for macromolecular structure.

By A. G. URZHUMTSEV, V. YU. LUNIN and E. A. VERNOSLOVA, *Research Computer Center, Academy of Sciences of the USSR, Pushchino, Moscow Region* 142292, *USSR*

(*Received* 15 *August* 1988; *accepted* 13 *April* 1989)

Abstract

A new program complex for macromolecular refinement is described. The kernel of the complex is the program *FROG*2 to refine proteins, RNA and other biopolymers. Its high-speed operation is due to the fast differentiation and the fast Fourier transform algorithms it uses. A model can be described along four levels: generalized parameters, atomic parameters, density distribution and diffraction parameters. The functional to be minimized is set by special reference files which make it easy to modify. The functional consists of many criteria; some of them, such as the phase criterion, are new; others, the density criterion and intermolecular interaction, are encountered, though not very often; and the rest of them are familiar, like *R* factor, bond-length or bond-angle restraints. A model may be a mixture of individual atoms and rigid groups, each group including an arbitrary number of atoms. The complex has a routine to organize atoms into rigid groups for the most common situations. The *FROG*2 program does not in practice restrict model size or space group. However, it takes account of symmetry, which gives another reduction of computing time. The program does not require special effort to adapt it to any memory. It can easily be extended by new criteria, atomic types, and more effective algorithms for calculating electron density or Fourier transform may be included. Finally, program blocks may be used in other programs, *e.g.* those for phase refinement.

Introduction

The term 'refinement of macromolecular structure' broadly means a series of investigation steps which include a number of different actions: Fourier synthesis calculation, interactive model correction and automatic modification of atomic model. In a more narrow sense 'refinement of atomic structure' is only an automatic model correction carried out by special computer programs - refinement programs. These programs may also be used at earlier stages of the analysis when the atomic model is either absent (Agarwal & Isaacs, 1977; Lunin & Urzhumtsev, 1984) or only partially constructed (Lunin *et al.*, 1985; Bhat & Blow, 1982).

Various requirements may be imposed on the atomic model during automatic correction: it should correspond to X-ray or neutron diffraction data and to NMR data, satisfy energetic and stereochemical restrictions, be close to homologues and so on. These requirements reflect the available information about the object in a quantitative form. Usually, refinement is carried out as minimization of a functional which is their weighed sum. Other approaches are, however, possible (Brünger, Kuriyan & Karplus, 1987). Possibilities of automatic correction depend largely on the type of information a refinement program takes into account. Since this information is usually complex, the convenience of the program becomes important: simple inclusion of various requirements, good service for data preparation and processing and clear control parameters are all required.

Large amounts of data and the complex character of the requirements need significant computer resources. Model refinement uses a major part of the total computer time needed for structure determination. Hence, a refinement program which is to be applied to a large object must be very effective.

Effective refinement programs are very complex, and therefore they are still rather few in number (Diamond, 1971; Dodson, Isaacs & Rollett, 1976; Sussmann, Holbrook, Church & Kim, 1977; Jack & Levitt, 1978; Agarwal, 1978; Hendrickson & Konnert, 1980). New more complicated objects and new requirements of the model stimulated creation of new powerful refinement programs (Moss & Morffew, 1982; Huber & Schneider, 1985; Tronrud, Ten Eyck & Matthews, 1987). Some of the existing programs have been effectively modified (Haneef, Moss, Stanford & Borkakoti, 1985; Cohen, 1986; Finzel, 1987).

In the early stages of the refinement one may decrease the number of free parameters by introducing atomic groups which move as rigid bodies. This decrease influences the convergence of the minimization process (Pavlovskiy *et al.*, 1987). Of the programs which are of current use only *CORELS* by Sussmann *et al.* (1977) utilizes these groups in the course of refinement. An attempt to replace rigid groups by a 'strong' penalty function makes worse the minimization properties of the summary function (its equipotential surfaces become 'ravine' shaped) and, correspondingly, slows down the process of refinement. The introduction of quasi-rigid groups (Tronrud *et al.*, 1987) does not decrease the number of free parameters and allows a certain deformation of these groups during the refinement.

An analysis of the means of increasing the efficiency of refinement programs (Sayre, 1951; Ten Eyck, 1977; Agarwal, 1978; Kim, Nesterov & Cherkasskiy, 1984) permitted us to elaborate their general conception (Lunin & Urzhumtsev, 1985). It provides the basis for our *FROG*2 program described below. The main features of *FROG*2 are:

(1) low computer time: a refinement cycle of a middle-size protein (3000 atoms) at a resolution of 2·5 Å (12 000 reflections) takes about 35 min on a computer with 300 000 operations per second; computer time is linearly dependent on molecular size;

(2) special composition of the model: it may consist of rigid groups of atoms and of individual atoms at a time; no changes to the program are needed to change the method of group formation;

0021-8898/89/050500-07$03.00

(3) method of criteria assignment: refinement criteria (requirements that are imposed on the model) are described by external files and may be changed with no intervention in the *FROG*2 text; in particular, models of various types of macromolecules, such as proteins, DNAs *etc.*, may be refined with their own stereochemical restraints;

(4) adaptability to the available memory: the program is easily adapted to any memory of more than 512Kbyte; the more memory is used, the less computer time is taken;

(5) single language: only Fortran IV has been used;

(6) optional service programs - *FROG* complex: service programs (Urzhumtsev, Lunin & Vernoslova, 1988) are added to the *FROG*2 program, which facilitates the work in basic regimes.

A short description of the *FROG*2 program is given in § 1 and its general structure is described in § 2.

1. The *FROG*2 program

1.1. *Model*

(*a*) *Atomic-blocked model.* The *FROG*2 program refines atomic models. By an atomic model we mean a model composed of a set of 'atoms', parameters of which are known during the refinement. *FROG*2 describes each atom by the Cartesian coordinates of its center, by an isotropic displacement factor B and by an occupancy T. Besides these, each atom has a fixed type which defines its scattering factor, $f(s)$, in two-Gauss form (Agarwal, 1978).

The number of free parameters of the model may be decreased by declaring several atomic groups rigid. In *FROG*2 a rigid group may include various sets of atoms. A rigid group is described by the coordinates of all its atoms in any Cartesian coordinate system and by Euler angles and translation vector to place the group, as a rigid body, at its current position in the standard Cartesian coordinate system. The other parameters of the group are displacement factor and occupancy. The displacement factor of an atom is calculated by its own fixed displacement factor inside the group and by the group displacement factor which can be refined. Similarly, the atomic occupancy is calculated. For an N-atom rigid group these eight parameters (Euler angles, translation vector, displacement factor and occupancy) can be refined instead of the $5N$ parameters required for individual atoms.

The *FROG*2 program lays no restriction on the number and composition of rigid groups. In particular, all the atoms may be treated as individual. In contrast, the whole molecule may be refined as a rigid body. The *FROG* complex (Urzhumtsev *et al.*, 1988) contains a special service routine to simplify association of atoms into rigid groups. A model that consists of both individual atoms and rigid groups is further called an 'atomic-blocked model'.

(*b*) *Variable parameters.* The atomic-blocked model builds input and output for *FROG*2. It may include the following variable parameters: (i) for individual atoms, coordinates of the atomic center, isotropic displacement factor, occupancy; (ii) for rigid groups, Euler angles, translation vector, group displacement factor, group occupancy. One may also refine the coefficients to scale calculated values up to the observed ones.

A user can fix some of these parameters either individually for selected atoms (rigid groups) or for all the atoms of the model.

(*c*) *Symmetry account.* The program can keep the symmetry relation between the parts of the model unchanged in the course of the refinement. In this case only parameters of the 'asymmetric part' of the model are varied and the rest of the parameters are generated by the symmetry relations. This, in particular, provides non-crystallographic symmetry. Another application is the refinement of the model from any space group with routines for space group $P1$. For specific features of symmetry space groups, see § 2.

1.2. *Refinement criteria*

From a mathematical point of view, the refinement program should change parameter values so as to minimize the value of a functional R. The functional is a quantitative representation of the user's requirements for a model. The *FROG*2 program fixes the general structure of R, rather than its concrete form. The concrete form of the functional is described by special input files which allow it to be varied with no change in the Fortran text of the program.

The functional R may include any requirement which can be expressed in terms of atomic-blocked model parameters, atomic parameters, electron density distribution or structure factors. In particular, we may require that: moduli of calculated structure factors correspond to the observed values; phases of calculated structure factors fit their probability distributions (Lunin & Urzhumtsev, 1985); bond lengths, bond and dihedral angles, chirality volume *etc.* minimally deviate from given values; the same for planarity of some atomic groups; non-bonded interaction energy including intermolecular interaction be minimal; discrepancies in model symmetry be minimal; the part of the model that lies outside of given boundaries be minimal; and rigid-group orientation or position minimally deviate from given values.

The list may easily be extended.

1.3. *Program restrictions and possible developments*

The only serious restriction on the program is that the buffer array which is used by the program must contain no less than two sections of the calculated electron density. An excessive number of atoms is no restriction and only somewhat increases the CPU time. Nor is the space-group symmetry of an object restricted.

The *FROG*2 program allows different modifications; in particular, one may add or replace refinement criteria, introduce new atomic types, or change the size of the buffer array to adapt the program to new computers or objects.

1.4. *Examples of FROG's application*

The *FROG* program may be useful in various macromolecular projects. Here are four examples illustrating some of the possible applications of the program and of the resources the program needs.

(*a*) *Refinement of γ-crystallin* IIIb *atomic model.* The three-dimensional structure of the protein γ-crystallin IIIb from calf lens has been investigated by researchers at the Protein Research Institute of the USSR Academy of Sciences together with the present authors (Chirgadze *et al.*, 1986). Crystals of the protein belong to the space group $P2_12_12_1$ with $a = 58{\cdot}7$, $b = 69{\cdot}5$ and $c = 116{\cdot}9$ Å. The asymmetric part of the unit cell contains two protein molecules, each with a molecular weight of 20 000 daltons (173 amino-acid residues). The starting atomic model of γ-cristallin

IIIb was obtained with the *ISOCUB* molecular graphics program (Nevskaya, Kurochkina & Chirgadze, 1986) and electron density maps which were calculated by the method of mixed atomic model (Lunin *et al.*, 1985).

Two stages of refinement were carried out with different data sets. The first data set with a resolution up to 2·5 Å was collected at the Protein Research Institute. Refinement of the model consisting of individual atoms with stereochemical restraints against this data set produced an R factor of 0·25 (Chirgadze *et al.*, 1986).

At the second stage we used a new data set with a resolution up to 1·9 Å collected at the Daresbury synchrotron. Changes in the conditions of data collection resulted in a modification of the molecular structure. Molecular packing also changed so that the unit-cell parameters became $a = 57{\cdot}4$, $b = 69{\cdot}2$, $c = 115{\cdot}5$ Å. This required a preliminary refinement of the rigid-body model followed by one of a model with rigid kernel and flexible surface side chains. Now we continue working with a high-resolution model.

A refinement cycle of γ-crystallin IIIb model at a resolution of 2·5 Å (12 000 reflections) took about 35 min of CPU time on an EC-1055M computer (300 000 op s^{-1}).

(*b*) *Model 'protein engineering' of phage T4 lysozyme.* The structure of the fragment 37–73 of phage T4 lysozyme (Weaver & Matthews, 1987) resembles the 'EF hand' (Kretsinger & Nockolds, 1973) of some proteins, such as parvalbumin (Kretsinger & Nockolds, 1973), calmodulin (Babu *et al.*, 1985) *etc.*, with the difference that it includes Gly51 instead of Asp51 and Asn55 instead of Asp55. Tufty & Kretsinger (1975) supposed that the reverse changes allow segment 51–62 of lysozyme to take the form of Ca^{2+}-connecting loop of EF proteins (Herzberg & James, 1985).

A computing experiment was undertaken to understand whether the fragment can hold the EF-hand structure without any stereochemical distortions (Murzin & Vernoslova, 1988).

An initial model, which was the model of phage T4 lysozyme (Weaver & Matthews, 1987), complemented by a side chain of Asp on the C_α atom of Gly51 and by an OD2 atom instead of an ND2 atom in the 55th residue, was gradually changed by *FROG*2 to form residues 51–62 into the conformation of residues 90–101 of carp parvalbumin.

The main part of the lysozyme model (residues 94–164) was fixed and some elements of the secondary structure (α-helixes 1–11, 40–46, 63–80, β-sheet 14–33) were moved as rigid bodies. Loops and the rest of the α-helix (residues 47–62) were modelled by individual atoms linked stereochemically. A refinement cycle of the model took 2 min on the EC-1055M.

The final model satisfied all the requirements: similarity of the loop, stereochemical and energy restrictions. The possibility of such a conformation was a good reason for biochemists to start working with phage T4 lysozyme.

(*c*) *Phase extension of pea lectin.* The structure of pea lectin has been investigated by Riskulov *et al.* (1984) at the Institute of Molecular Genetics of the Academy of Sciences of the USSR. Crystals of the protein belong to space group $P2_12_12_1$ with $a = 51{\cdot}0$, $b = 61{\cdot}7$ and $c = 137{\cdot}6$ Å. The asymmetric part of the unit cell contains one molecule with two similar subunits; their common weight is 52 000 daltons (about 480 residues).

The method of multiple isomorphouse replacement was applied to get an initial electron density map at a resolution of 3·0 Å. A mixed atomic model (Lunin *et al.*, 1985) was used to refine and extend the phases of structure factors.

This model consisted of the atoms of the partial model interconnected by stereochemical restraints, and of independent dummy atoms which interacted with real atoms only. The total number of atoms was about 4000.

At the end of the work the phase set was extended from 3·0 to 2·4 Å, which improved an electron density map. This map was used to increase the number of identified atoms from 40 to 70% of the total.

(*d*) *Carnation mottle virus refinement for phase extension.* The low-resolution structure (up to 18 Å) and the approximate position of the carnation mottle virus (CMV) have been determined by Morgunova, Mikhailov, Nekrasov, Kraftanova & Vainstein (1988) at the Institute of Crystallography of the USSR Academy of Sciences. Crystals of the CMV belong to space group $F23$ with $a = b = c = 482{\cdot}65$ Å. It is known that the molecule of CMV is similar to that of TBSV (Harrison & Jack, 1975) and that there are about two million atoms of CMV per unit cell.

The atomic model of TBSV was taken as the initial atomic model of CMV to refine its position and, accordingly, to extend the phase set up to a resolution of 6 Å. We used an EC-1045 computer of about 600 000 op s^{-1} with a memory of about 4Mbyte, and the only program which could do the work was *FROG*2.

A refinement cycle of the model took 7 h at a resolution of 10 Å (6000 independent reflections) and 11 h at a resolution of 6 Å (23 000 independent reflections).

2. Construction principles of the *FROG*2 program

2.1. *General structure*

Automatic refinement of an atomic model of a macromolecule normally resides in changing the value of model parameters χ so as to minimize the value of a functional $R(\chi)$,

$$R(\chi) \Rightarrow \min_{\chi}. \qquad (1)$$

The functional can be expressed through the model's characteristics such as atomic parameters, structure factors *etc.*, which are calculated from the parameters χ. The smaller the value of $R(\chi)$, the better the fit of the calculated characteristics to the observed ones.

The major problems in the refinement are caused by the very large dimension of the space of the variable parameters χ, laborious calculation of the model's characteristics, such as structure factors, and by the non-monotonic behaviour of $R(\chi)$. They are all responsible for the local character of minimization.

The *FROG*2 program minimizes (1) by the steepest descent method or by the conjugate gradient method, depending on what is desired. Accordingly, the central part of the program is a minimization block which is formally independent of the form of $R(\chi)$ and can cyclically calculate the direction in the χ parameter space for linear minimization and find the optimal point on it. This part of the program deals with traditional approaches; their main essentials are described in the Appendix.

The value of the functional and of its gradient are necessary for gradient minimization. This part of the program is the most extensive. Therefore, we devote the rest of the paper to the description of its organization.

2.2. *Controlled characteristics*

The *FROG*2 program works with atomic-blocked models of macromolecules (§ 1.1). We call their parameters generalized parameters and denote them by $\boldsymbol{\chi}$.

The generalized parameters may be used to calculate those characteristics of the model which may be controlled during the refinement. They are conveniently classified into several levels: atomic parameters $\{\mathbf{q}\}$, *i.e.* parameters of all the atoms composing the model (Cartesian coordinates, displacement factor, occupancy); electron density parameters $\{\boldsymbol{\rho}\}$, *i.e.* the values of the electron density distribution calculated from the atomic model at grid points $\mathbf{r}$ of the unit cell; structure factors $\{\mathbf{F}\}$ which correspond to X-ray diffraction by the electron distribution $\rho(\mathbf{r})$. These levels are naturally organized as

$$\boldsymbol{\chi} \Rightarrow \mathbf{q} \Rightarrow \boldsymbol{\rho} \Rightarrow \mathbf{F}. \tag{2}$$

Here the arrows show the direction in which the calculation does not present problems. The reverse way is usually problematic. For example, the transfer $\mathbf{q} \Leftarrow \boldsymbol{\rho}$ is an interpretation of an electron density map.

It should be noted that in a number of refinement programs structure factors are calculated directly through atomic parameters $\mathbf{q}$ ($\mathbf{q} \Rightarrow \mathbf{F}$) by analytical formulae. However, the two-step transfer which *FROG*2 uses,

$$\mathbf{q} \Rightarrow \boldsymbol{\rho} \Rightarrow \mathbf{F}, \tag{3}$$

is more effective for macromolecules (Sayre, 1951; Ten Eyck, 1977) and allows those requirements for the model to be included that are expressed in terms of electron density distribution (Diamond, 1971). That is why we classify the electron density distribution into a separate level.

2.3. *Structure of functional*

Requirements which are imposed on a model lay various restraints on its characteristics. Their general feature is that each of them is local, restricting a few parameters only. An example is the condition

$$(F_{\mathbf{s}}^{c} - F_{\mathbf{s}}^{o})^2 \Rightarrow \min \tag{4}$$

which restricts the value of a single structure factor $F_{\mathbf{s}}^{c}$. Another example is the condition

$$(|\mathbf{r}_j - \mathbf{r}_k| - d_{jk}^{0})^2 \Rightarrow \min \tag{5}$$

which connects coordinates of two atoms only. Each condition is applied only at a specific level. For example, when (4) is written in terms of atomic parameters, the requirement connects the coordinates of all the atoms.

In the *FROG*2 program the functional R is a sum of a (large) number of requirements expressed in terms of the above levels $\boldsymbol{\chi}, \mathbf{q}, \boldsymbol{\rho}, \mathbf{F}$.

In accordance with the introduced levels (2) we subdivide the functional R into four components

$$R = R_{\chi} + R_{q} + R_{\rho} + R_{F}, \tag{6}$$

which are expressed in terms of the levels $\boldsymbol{\chi}, \mathbf{q}, \boldsymbol{\rho}$ and $\mathbf{F}$.

Each of the components may include various requirements. For example, R_q may include restraints on bond lengths, non-valent interaction, planarity and so on (all expressed through atomic coordinates). Hence, it is convenient to group together one-type terms inside R_{χ}, R_q, R_{ρ}, R_F, which means separate processing of restraints on bond lengths, on bond angles and so on:

$$R_{\alpha} = \sum_{k=1}^{K_{\alpha}} \omega_{\alpha k} R_{\alpha k}. \tag{7}$$

Here the $\omega_{\alpha k}$ are weight coefficients and K_{α} is the number of requirement types of the level $\boldsymbol{\alpha}$, which is one of the levels $\boldsymbol{\chi}, \mathbf{q}, \boldsymbol{\rho}, \mathbf{F}$.

As a result, each of the $R_{\alpha k}$, called model quality criteria, is the sum of a large number of one-type requirements

$$R_{\alpha k} = \sum_{(j_1, \ldots, j_m) \in J_{\alpha k}} f_{\alpha k}(\alpha_{j_1}, \ldots, \alpha_{j_m}; \mathbf{d}(j_1, \ldots, j_m)). \tag{8}$$

Here $f_{\alpha k}(t_1, \ldots, t_m; \mathbf{d})$ is a family of functions which depend on a small number m of variables $t_1, \ldots, t_m$ (locality of the $f_{\alpha k}$) and on the parameter vector $\mathbf{d} = (d_1, \ldots, d_n)$ ('observed data', scale coefficients, weights and so on); $(\alpha_{j_1}, \ldots, \alpha_{j_m})$ is a subset of the full vector $(\alpha_1, \ldots, \alpha_M)$ of the variable (model) parameters of $\boldsymbol{\alpha}$ level, $M \gg 1$; the functional value is calculated for $t_1 = \alpha_{j_1}, \ldots, t_m = \alpha_{j_m}$ and $\mathbf{d} = \mathbf{d}(j_1, \ldots, j_m)$; and $J_{\alpha k}$ is a set of summation indexes $(j_1, \ldots, j_m)$.

For example, if

$$R_{q,1} = \sum_{j,k} \left[\frac{A_{jk}}{|\mathbf{r}_j - \mathbf{r}_k|^6} + \frac{B_{jk}}{|\mathbf{r}_j - \mathbf{r}_k|^{12}} \right],$$

then

$$f_{q,1}(\mathbf{t}_1, \mathbf{t}_2; d_1, d_2) = \frac{d_1}{|\mathbf{t}_1 - \mathbf{t}_2|^6} + \frac{d_2}{|\mathbf{t}_1 - \mathbf{t}_2|^{12}},$$

and if

$$R_{F,1} = \sum_{\mathbf{s}} W_{\mathbf{s}} (F_{\mathbf{s}}^{c} - F_{\mathbf{s}}^{o})^2,$$

then

$$f_{F,1}(\mathbf{t}; d_1, d_2) = d_1 (|\mathbf{t}| - d_2)^2.$$

2.4. *Concretization of functional*

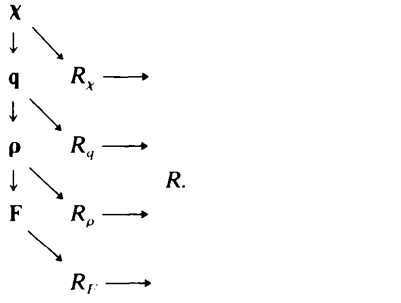
The choice of a criterion of type (6)–(8) depends on: the weight coefficients $\omega_{\alpha k}$; functions $f_{\alpha k}(t_1, \ldots, t_m; d)$; indexes $J_{\alpha k}$ for (8); and values of parameters $\mathbf{d}(j_1, \ldots, j_m)$ for each term in (8).

*FROG*2 uses different sources to define these values. First, coefficients $\omega_{\alpha k}$ are defined as input control data. Then, penalty functions $f_{\alpha k}$ are implemented as subroutines since they are relatively few, as the literature on protein structure refinement evidences. Subroutines are consecutively numbered, which facilitates their use in defining terms in (8). Now to define a term in (8) one must point out the conditional number of the criterion it belongs to, indexes $j_1, \ldots, j_m$ of the model parameters related by the term and values of parameters $\mathbf{d}(j_1, \ldots, j_m)$. The program *FROG*2 takes the above information from special input files ('reference files'). The set of file records is the set of requirements $J_{\alpha k}$.

The use of reference files (it is convenient that each reference file corresponds to the level of the model description - $\chi, \mathbf{q}, \boldsymbol{\rho}, \mathbf{F}$) has several advantages over the usual definition of the functional directly inside the program. They are: the possibility of refinement by *FROG*2 not only of protein but of other macromolecular models, for example DNA models, without programmer intervention; the possibility of introducing non-standard restraints; the utilization of various weighting schemes; the possibility of working individually with each term of the functional by ascribing it individual parameter values (this can include it in or exclude it from the criterion); the possibility of including reflections in two or more criteria simultaneously, *e.g.* into the R factor and in the phase criterion; absence of the problem of accounting for the intermolecular interaction (Vincent & Priestle, 1985; Sheriff, 1987); independence of the criteria which makes them easier to change or to be extended by new criteria - in this case one only has to change or add a short subroutine to process a single term.

2.5. *Calculation of functional and gradient*

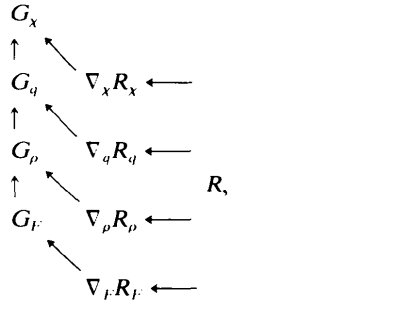
The hierarchy of levels of model description (2) and the possibility of the functional R incorporating requirements on any level type χ, $\mathbf{q}$, $\boldsymbol{\rho}$, $\mathbf{F}$ gives the most economical scheme for calculating R, which is as follows:



(9)

We have used this in *FROG*2.

As follows from the fast differentiation algorithm (Kim *et al.*, 1984; Lunin & Urzhumtsev, 1985), an efficient scheme of the gradient calculation may be obtained by 'inverting' the efficient scheme of functional calculation. Hence, the scheme (9) may lead to the algorithm



(10)

where $\nabla_\alpha R_\alpha$ is the gradient of R_α calculated with respect to variables $\boldsymbol{\alpha}$, and

$$G_F = \nabla_F R_F, \qquad G_\rho = \nabla_\rho (R_\rho + R_F)$$

$$G_q = \nabla_q (R_q + R_\rho + R_F), \qquad G_\chi = \nabla_\chi (R_\chi + R_q + R_\rho + R_F).$$

Here each transfer $\mathbf{x} \rightarrow \mathbf{y}$ from the previous level $\mathbf{x}$ to the next level $\mathbf{y}$ leads to the transfer $\nabla_y f \rightarrow \nabla_x f$ in the gradient calculation scheme, whether or not $f(\mathbf{y})$ is a complex function of $\mathbf{y}$. As has been shown earlier (Lunin & Urzhumtsev, 1985), for any transfer $\mathbf{x} \rightarrow \mathbf{y}$ the reverse transfer $\nabla_x f = \nabla_y f(\mathrm{d}y/\mathrm{d}x)$ may be calculated for less than fourfold computing cost of the direct transfer. There are two conclusions about this fact. The first is that, if the transfer $\mathbf{x} \rightarrow \mathbf{y}$ is simple enough, then the reverse transfer $\nabla_x f = \nabla_y f(\mathrm{d}y/\mathrm{d}x)$ is simple too. The second is that the optimal computing cost, T_∇, of all N components of the gradient of any function $R(x_1, \ldots, x_N)$ is less than four times the computing cost of one value of the criterion, T_R, so that $T_\nabla < 4T_R$, rather than NT_R (Lunin & Urzhumtsev, 1985). Moreover, in the *FROG*2 program $T_\nabla < T_R$.

2.6. *Structural elements of FROG*2

The scheme (9) suggests that the first set of structural elements for the refinement program may be built from blocks we use to pass over from one type of the model description to the next one ($\chi \rightarrow \mathbf{q}$, $\mathbf{q} \rightarrow \boldsymbol{\rho}$, $\boldsymbol{\rho} \rightarrow \mathbf{F}$) and blocks used to calculate criteria ($\chi \rightarrow R_\chi$ and so on). The latter may also be interpreted as transfers to new types of the model description. The second set of structural elements, as follows from the scheme (10), consists of blocks of $\nabla_\alpha R_\alpha$ calculation, where $\boldsymbol{\alpha} = \chi, \mathbf{q}, \boldsymbol{\rho}, \mathbf{F}$, and blocks of transfers $G_F \rightarrow G_\rho$, $G_\rho \rightarrow G_q$, $G_q \rightarrow G_\chi$.

The structure of the *FROG*2 program - one independent minimization block and the sets of direct and inverse transfer blocks - is advantageous in many respects. Firstly, the program may be used to produce new refinement programs, serving, for example, to refine phases by Sayre's equations (Sayre, 1974; Lunin, 1985), histograms of electron density (Lunin, 1988) and so on. Secondly, the program can easily be adapted to other models: those with anisotropic displacement factors, those with only dihedral angles variable, or those in which X-ray scattering is not produced by Gaussian spheres but by other geometric objects, *e.g.* by cylinders (Kalinin, 1980). Thirdly, it 'unhooks' calculations of different criteria, making them easy to replace or append. Fourthly, the program admits new algorithms, *e.g.* for electron density generation, or special devices, *e.g.* for Fourier transform.

The reasoned composition of the program is 'confirmed' by the fact that when a protein model is processed in the usual way, the computer costs of transfers $\mathbf{q} \rightarrow \boldsymbol{\rho}$, $\boldsymbol{\rho} \rightarrow \mathbf{F}$, $\mathbf{q} \rightarrow R_q$ and others are close in magnitude.

2.7. *Account of model symmetry*

A further increase in the efficiency of the program is caused by the growing power of the blocks we have described in the foregoing section, and is, in particular, due to the symmetry of a molecule or of a crystal.

In crystallographic refinement we consider only the asymmetric part of the model if we take account of the space-group symmetry. Accordingly, the electron density should be calculated only for the asymmetrical set of grid points, as well as the set of structure factors. These incomplete sets of data give a considerable decrease in the CPU time. The *FROG*2 program implements all these symmetry possibilities.

Reduction in the computing time of the transfer $\boldsymbol{\rho} \rightarrow \mathbf{F}$ may require special subroutines for Fourier transformation.

A number of subroutines in *FROG*2 for the most widespread space groups already exist, and new subroutines may easily be included.

According to the general theory (Lunin & Urzhumtsev, 1985) symmetry should be used to calculate the gradient. It applies constraints to structure factors, which leads to some features of the reverse transform that will be published elsewhere.

The programs and documentation are available on request from the authors.

The authors thank O. M. Liguinchenko and one of the referees for improving the text of the paper.

APPENDIX
Some problems of minimization

1. *Scaling of variables*

Diverse parameters and criteria with different weights which the minimized functional R includes result in a 'ravined' shape of equipotential surfaces of the R. Scaling of variables may improve the situation.

Usually, a block-diagonal approximation of a Hesse matrix is used for scaling. The *FROG*2 program uses an exact matrix of second derivatives of some 'approximate' functional which accounts for the type of criteria included and for the corresponding weights.

2. *Optimal point selection*

Usually, the optimal point in a direction for a non-quadratic functional is selected by testing its values at some points along it. These values are used to determine an interpolation polynomial (usually quadratic) in order that the next test point can be defined.

The fast differentiation algorithm (Kim *et al.*, 1984; Lunin & Urzhumtsev, 1985) states that computation costs of the functional and the derivative along the direction are nearly the same.

In the *FROG*2 program test points are selected by a third-order polynomial analysis. Coefficients of the polynomial are determined by the values of the functional and its derivative along the direction, which are calculated at two previously determined points. The derivative is used for a more detailed analysis of the behaviour of the minimized functional.

It is important that, in practice, simultaneous calculation of the functional and of the derivative should take not twice but only one and a half times the time needed to calculate the functional.

3. *Choice of functional*

(*a*) *R-factor-type criteria.* The commonly used criterion to refine a model against moduli of structure factors has the form

$$R_F = \sum_{s \in S} w_s (kF_s^c - F_s^o)^2. \qquad (A1)$$

The set S of structure factors consists traditionally of the reflections which are selected by a 'σ_s criterion': $F_s^o \geq n\sigma_s$, $n = 2$ or 3. Agarwal (1978) empirically has introduced a new 'selection' of reflections based on the ratio F_s^c/F_s^o. Refinement is only carried out over the reflections which satisfy the inequalities: $t_1 \leq F_s^c/F_s^o \leq t_2$, where $t_2 = 1/t_1$ may vary from 4 to 2 in the course of the refinement (Agarwal, 1978). A similar selection was used by Vijay-Kumar, Bugg & Cook (1987).

The requirement of convexity of the criterion (*A*1) leads to an exact and hardly expectable estimate for t_1 which is $t_1 = 1$. The parameter t_2 is responsible for the selection of weak reflections and may vary.

In the refinement program by Tronrud *et al.* (1987) an R-factor-type criterion has the form

$$R_F = \sum_s w_s (F_s^c - \kappa F_s^o)^2. \qquad (A2)$$

Here the relation of the scale coefficient κ to the observed values F_s^o is undoubtedly more natural. However, since the value of κ usually varies during the minimization of (*A*2), besides the real minimum $F_s^c = F_s^o$ there appears the 'false' minimum

$$B = \infty, \kappa = 0,$$

in the course of the refinement of displacement factors B, which may twist refinement results.

(*b*) *Phase criterion.* The efficiency of a phase criterion was shown by Rees & Lewis (1983). The most familiar phase criterion [for example, in *RESTRAIN* (Moss & Morffew, 1982)] has a quadratic form

$$R_F^\varphi = \sum_s (\varphi_s^c - \varphi_s^o)^2. \qquad (A3)$$

Lunin & Urzhumtsev (1985) suggested another phase criterion,

$$R_F^\varphi = -\sum_s (A_s \cos \varphi_s^c + B_s \sin \varphi_s^c + C_s \cos 2\varphi_s^c + D_s \sin 2\varphi_s^c), \qquad (A4)$$

that is included in *FROG*2. This criterion is more versatile and makes use of heavy-atom experimental information (Hendrickson & Lattman, 1970). It should be stressed that, unlike (*A*3), criterion (*A*4) accounts for the bimodality of phase distributions.

(*c*) *Planarity criterion.* A new rapid method for controlling planarity of an atomic group was suggested by Haneef *et al.* (1985). In contrast to the *PROLSQ* criterion by Hendrickson & Konnert (1980), this provides essentially minimization of the 'volume' of an atomic group, rather than planarity. We believe it to be more reasonable that a refinement program should use the *PROLSQ*-type criterion instead of the new criterion or those with dummy atoms (Dodson *et al.*, 1976; Tomlin, 1987). This criterion may be improved by analytical calculation of the minimal eigenvalue of the 3×3 inertia matrix instead of 'the best plane' determination.

To conclude, we give some words about the calculation of the planarity criterion gradient. As has been mentioned above, the effective procedure is the 'inversion' of the scheme used to calculate the functional, needing thereby no difference methods (Haneef *et al.*, 1985).

(*d*) *Rigid-group rotation.* Note that there is no need for linearity or quasi-linearity of transfers $\mathbf{x} \rightarrow \mathbf{y}$ to calculate $R(\chi)$ or $\nabla_\chi R(\chi)$. So one may describe rigid-group rotation in any possible angle system where the starting point is far from a singular one. Because such points for standard Euler angles (rotation about x, z, x axes) are $(\alpha, \beta, \gamma) = (\alpha, 0, \gamma)$

we rotated each rigid group by $\beta = -90°$ in the first version of the *FROG2* program when the input model file was prepared. But for greater user's convenience, in the current version of the program rigid-group orientation is described by quasi-Euler angles (rotation about z, y, x axes) where $(\alpha', \beta', \gamma') = (0, 0, 0)$ is not a singular point.

References

AGARWAL, R. C. (1978). *Acta Cryst.* A**34**, 791-809.

AGARWAL, R. C. & ISAACS, N. W. (1977). *Proc. Natl Acad. Sci. USA*, **74**, 2835-2839.

BABU, Y. S., SACK, J. S., GREENHOUGH, T. J., BUGG, C. E., MEANS, A. R. & COOK, W. J. (1985). *Nature (London)*, **315**, 37-40.

BHAT, T. N. & BLOW, D. M. (1982). *Acta Cryst.* A**38**, 21-29.

BRÜNGER, A. T., KURIYAN, J. & KARPLUS, M. (1987). *Science*, **235**, 458-460.

CHIRGADZE, YU. N., NEVSKAYA, N. A., FOMENKOVA, N. P., NIKONOV, S. V., SERGEEV, YU. V., BRAZHNIKOV, E. V., GARBER, M. B., LUNIN, V. YU., URZHUMTSEV, A. G. & VERNOSLOVA, E. A. (1986). *Dokl. Akad. Nauk SSSR*, **290**, 492-495.

COHEN, G. H. (1986). *J. Appl. Cryst.* **19**, 486-488.

DIAMOND, R. (1971). *Acta Cryst.* A**27**, 436-452.

DODSON, E. J., ISAACS, N. W. & ROLLETT, J. S. (1976). *Acta Cryst.* A**32**, 311-315.

FINZEL, B. C. (1987). *J. Appl. Cryst.* **20**, 53-58.

HANEEF, I., MOSS, D. S., STANFORD, M. J. & BORKAKOTI, N. (1985). *Acta Cryst.* A**41**, 426-433.

HARRISON, S. C. & JACK, A. (1975). *J. Mol. Biol.* **97**, 171-191.

HENDRICKSON, W. A. & KONNERT, J. H. (1980). In *Biomolecular Structure, Function, Conformation and Evolution*, edited by R. SRINIVASAN, Vol. 1, pp. 43-57. Oxford: Pergamon.

HENDRICKSON, W. A. & LATTMAN, E. E. (1970). *Acta Cryst.* B**26**, 136-143.

HERZBERG, O. & JAMES, M. N. G. (1985). *Biochemistry*, **24**, 5298-5302.

HUBER, R. & SCHNEIDER, M. (1985). *J. Appl. Cryst.* **18**, 165-169.

JACK, A. & LEVITT, M. (1978). *Acta Cryst.* A**34**, 931-935.

KALININ, D. I. (1980). *Kristallografiya*, **25**, 535-544.

KIM, K. V., NESTEROV, YU. E. & CHERKASSKIY, B. V. (1984). *Dokl. Akad. Nauk SSSR*, **275**, 1306-1309.

KRETSINGER, R. H. & NOCKOLDS, C. E. (1973). *J. Biol. Chem.* **248**, 3313-3326.

LUNIN, V. YU. (1985). *Acta Cryst.* A**41**, 551-556.

LUNIN, V. YU. (1988). *Acta Cryst.* A**44**, 144-150.

LUNIN, V. YU. & URZHUMTSEV, A. G. (1984). *Acta Cryst.* A**40**, 269-277.

LUNIN, V. YU. & URZHUMTSEV, A. G. (1985). *Acta Cryst.* A**41**, 327-333.

LUNIN, V. YU., URZHUMTSEV, A. G., VERNOSLOVA, E. A., CHIRGADZE, YU. N., NEVSKAYA, N. A. & FOMENKOVA, N. P. (1985). *Acta Cryst.* A**41**, 166-171.

MORGUNOVA, E. IU., MIKHAILOV, A. M., NEKRASOV, IU. V., KAFTANOVA, A. S. & VAINSTEIN, B. K. (1988). *Dokl. Akad. Nauk SSSR*, **299**, 1129-1134.

MOSS, D. S. & MORFFEW, A. (1982). *Comput. Chem.* **6**, 1-3.

MURZIN, A. G. & VERNOSLOVA, E. A. (1988). Proc 5th Conf. of Young Scientists of Socialist Countries on Bioorganic Chemistry, 21-28 August 1988, Pushchino, USSR. Abstracts, pp. 83-84.

NEVSKAYA, N. A., KUROCHKINA, N. A. & CHIRGADZE, YU. N. (1986). *Kristallografiya*, **31**, 303-311.

PAVLOVSKIY, A. G., STROKOPYTOV, B. V., BORISOVA, S. N., VAINSHTEIN, B. K., KARPEISKIY, M. YA. & YAKOVLEV, G. N. (1987). *Dokl. Akad. Nauk SSSR*, **292**, 1253-1256.

REES, D. C. & LEWIS, M. (1983). *Acta Cryst.* A**39**, 94-97.

RISKULOV, R. R., DOBROKHOTOVA, Z. D., KUZEV, S. V., LOBSANOV, YU. D., LUBIN, M. YU., MOKULSKAYA, T. D., MYSHKO, G. E., PROSKUDINA, L. T., ROGACHEVA, M. M., SAPRYKINA, L. F., KHRENOV, A. A. & MOKULSKII, M. A. (1984). *FEBS Lett.* **165**, 97-101.

SAYRE, D. (1951). *Acta Cryst.* **4**, 362-367.

SAYRE, D. (1974). *Acta Cryst.* A**30**, 180-184.

SHERIFF, S. (1987). *J. Appl. Cryst.* **20**, 55-57.

SUSSMANN, J. L., HOLBROOK, S. R., CHURCH, G. M. & KIM, S.-H. (1977). *Acta Cryst.* A**33**, 800-804.

TEN EYCK, L. F. (1977). *Acta Cryst.* A**33**, 486-492.

TOMLIN, J. L. (1987). *J. Appl. Cryst.* **20**, 48-53.

TRONRUD, D. E., TEN EYCK, L. F. & MATTHEWS, B. W. (1987). *Acta Cryst.* A**43**, 489-501.

TUFTY, R. M. & KRETSINGER, R. H. (1975). *Science*, **187**, 167-169.

URZHUMTSEV, A. G., LUNIN, V. YU. & VERNOSLOVA, E. A. (1988). *Program Complex FROG*. Software, iss. 10, ser. Fortran. Pushchino, ONTI NCBI, USSR Academy of Sciences.

VIJAY-KUMAR, S., BUGG, C. E. & COOK, W. J. (1987). *J. Mol. Biol.* **194**, 531-544.

VINCENT, M. G. & PRIESTLE, J. P. (1985). *J. Appl. Cryst.* **18**, 185-188.

WEAVER, L. H. & MATTHEWS, B. W. (1987). *J. Mol. Biol.* **193**, 189-199.

J. Appl. Cryst. (1989). **22**, 506-510

***POWD*, an interactive program for powder diffraction data interpretation and indexing.**

By E. WU, *School of Physical Sciences, Flinders University of South Australia, Bedford Park, SA* 5042, *Australia*

(*Received* 17 *March* 1989; *accepted* 20 *April* 1989)

Abstract

An interactive program, *POWD*, written in Fortran77, for powder diffraction data interpretation and indexing is described. Various functions such as the least-squares refinement and the systematic absence test are also included in this versatile program. The modified de Wolff figure of merit [Wu (1988). *J. Appl. Cryst.* **21**, 530-535] is introduced

0021-8898/89/050506-05$03.00