



Programming new geometry restraints: parallelity of atomic groups

Oleg V. Sobolev,^{a,*} Pavel V. Afonine,^a Paul D. Adams^{a,b} and Alexandre Urzhumtsev^{c,d}

^aLawrence Berkeley National Laboratory, One Cyclotron Road, MS64R0121, Berkeley, CA 94720, USA, ^bDepartment of Bioengineering, University of California Berkeley, Berkeley, CA 94720, USA, ^cCentre for Integrative Biology, IGBMC, CNRS-INSERM-UdS, 1 rue Laurent Fries, BP 10142, 67404 Illkirch, France, and ^dDépartement de Physique, Faculté des Sciences et des Technologies, Université de Lorraine, 54506 Vandoeuvre-lès-Nancy, France. *Correspondence e-mail: osobolev@lbl.gov

Received 24 February 2015

Accepted 31 May 2015

Edited by D. I. Svergun, European Molecular Biology Laboratory, Hamburg, Germany

Keywords: restraints; atomic model refinement; parallel planes; *cctbx*; PHENIX; gradient calculation.

Improvements in structural biology methods, in particular crystallography and cryo-electron microscopy, have created an increased demand for the refinement of atomic models against low-resolution experimental data. One way to compensate for the lack of high-resolution experimental data is to use *a priori* information about model geometry that can be utilized in refinement in the form of stereochemical restraints or constraints. Here, the definition and calculation of the restraints that can be imposed on planar atomic groups, in particular the angle between such groups, are described. Detailed derivations of the restraint targets and their gradients are provided so that they can be readily implemented in other contexts. Practical implementations of the restraints, and of associated data structures, in the *Computational Crystallography Toolbox (cctbx)* are presented.

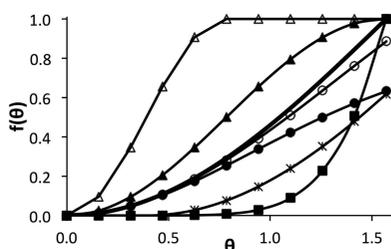
1. Introduction

The restrained refinement of atomic models is performed through minimization of a target function. First introduced in the macromolecular crystallography field by Konnert (1976) and Jack & Levitt (1978), this target is a weighted sum of several terms including R_X describing the model fit to the experimental data:

$$R = w_X R_X + \sum_{n=1}^N R_n. \quad (1)$$

In the case of refinement of atomic coordinates, the terms R_n describe the model geometry and are known as geometric or stereochemical restraints.

At low resolution there may be many models that may fit the experimental data equally well, but not all of these models may be chemically reasonable. This is why additional information needs to be used to restrict the models to those that conform to expectations about the correct chemical structure. These expectations may originate from various sources, such as small-molecule crystallographic studies, high-quality macromolecular structures or quantum chemistry calculations, forming the basis for libraries of *a priori* knowledge. At lower resolution, there are fewer experimental data available and therefore a larger number of restraints are required to ensure a high-quality refined model. Therefore, low-resolution structural studies, which have become more and more frequent, require additional restraints (see, for example, Smart *et al.*, 2012; Headd *et al.*, 2012; Brown *et al.*, 2015) to complement the traditionally used restraints, such as bond



OPEN ACCESS

lengths, bond angles, dihedral angles, non-bonded interactions, and planarity and chirality restraints.

As an example, nucleic acids have a number of base pairs, some of which are parallel or nearly parallel to each other. This knowledge may serve as a restraint that can be applied to the refinement of such models. Adding a new restraint into the refinement target means developing a corresponding algorithm to calculate its value, as well as an algorithm to calculate the gradient of this restraint function. We use the example of the parallelity restraint, implemented in *cctbx* (Grosse-Kunstleve *et al.*, 2002), to illustrate the general approach of adding new restraints in *cctbx* and *PHENIX* (Adams *et al.*, 2010) and also to reinforce the general principles of crystallographic refinement programs. This article describes the algorithms in a ready-to-program manner, with sufficient mathematical and computational details to serve a didactic purpose as well. Practical examples of the application of this new restraint function to model refinement will be presented elsewhere.

Each individual target in equation (1) is the sum of a large number of identical terms, each depending on a small number of parameters. For example, a diffraction target usually depends on a structure factor amplitude or the parameters of its phase distribution, a density target depends on the density values in a few grid points around a particular atom (see, for example, Afonine & Urzhumtsev, 2004, and references therein), a geometric restraint depends on the coordinates of a few atoms involved in particular restraint, *etc.* Such a composition of the target function (1) is very important for several reasons. First, it allows the manipulation of each kind of restraint, *e.g.* selection of the atoms involved or an update of the function. Second, each restraint is relatively simple and easy to program. Third, each restraint can be weighted individually. Finally, the gradient of the crystallographic target used for the minimization can be calculated as a sum of gradients from the individual terms.

Several general points are important to note when designing refinement programs (Lunin & Urzhumtsev, 1985):

(i) Refinement targets may be functions of the atomic parameters and values of electron density (or structure factors), in combination with the restraints and/or constraints imposed on the atomic model. Each component of the overall target is a relatively simple function when it is expressed through the parameters of the appropriate type.

(ii) These parameters are calculated from each other as a chain of consecutive transitions.

(iii) The gradient calculations are performed step by step using the chain rule, inverting the chain of steps used to calculate the target (Baur & Strassen, 1983; Kim *et al.*, 1984). This procedure results in exact gradient values that require approximately the same amount of time to calculate as a single value of the target, and this is independent of the number of parameters to be refined.

(iv) The gradient calculation does not require an analytical function; it may be any algorithm that provides a value for the corresponding target. At each refinement step, the target or gradient calculation is a local operation. It takes the numerical values obtained at the previous step, calculates the result of

the step and passes the numerical values to the next step. Such a scheme allows one to avoid complicated mathematical equations connecting the final target or gradient value with the initial model parameters.

In order to introduce a parallelity restraint into refinement, the considerations stated above require that we propose an algorithm to calculate a numerical value describing by how much two atomic groups that form pseudo-planes (approximately belong to two planes) are not parallel to each other. This algorithm should be numerically robust enough to handle a broad range of model parameters, from good (towards the end of refinement) to distorted (at the beginning of refinement).

2. Possible parallelity restraints and their gradient

2.1. Definition of the 'best plane'

Let us consider two pseudo-planar atomic groups (groups that should be planar but may not currently be so), $m = 1, 2$, containing atoms defined by their Cartesian coordinates, $\mathbf{r}_{k,m} = (x_{k,m}, y_{k,m}, z_{k,m})$, $k = 1, \dots, K_m$, in the same coordinate system. For each such group we need to define the corresponding 'best plane'. A plane is defined by a vector \mathbf{N}_m normal to it and by a point \mathbf{C}_m that belongs to it. For a given atomic group, the plane can be characterized by the sum of the squared deviations of the atoms from it (Urzhumtsev, 1991; Blanc & Paciorek, 2001; Brown *et al.*, 2015), and these deviations may be weighted if desired. The 'best plane' is the one that corresponds to the minimum of this value (\mathbf{N}_m and \mathbf{C}_m are variables),

$$\sum_{k=1}^{K_m} w_{k,m} [(\mathbf{r}_{k,m} - \mathbf{C}_m) \cdot \mathbf{n}_m]^2 = \sum_{k=1}^{K_m} w_{k,m} \frac{[(\mathbf{r}_{k,m} - \mathbf{C}_m) \cdot \mathbf{N}_m]^2}{\mathbf{N}_m \cdot \mathbf{N}_m} \rightarrow \min_{\mathbf{N}_m, \mathbf{C}_m}, \quad (2)$$

where $w_{k,m}$ is a weight (*e.g.* a unit value or an atomic mass) for the contribution of an atom with coordinates $\mathbf{r}_{k,m}$ to the target. Here and below, a centred dot ' \cdot ' represents the scalar (dot) product of two vectors, and $\mathbf{n}_m = \mathbf{N}_m / (\mathbf{N}_m \cdot \mathbf{N}_m)^{1/2}$ is the normalized vector \mathbf{N}_m .

Two facts are useful (see, for example, Urzhumtsev, 1991). First, the minimum of equation (2) is reached when \mathbf{C}_m is the weighted geometric center of the group

$$\mathbf{C}_m = \left(\sum_{k=1}^{K_m} w_{k,m} \right)^{-1} \sum_{k=1}^{K_m} w_{k,m} \mathbf{r}_{k,m}. \quad (3)$$

Second, if for a given atomic group we calculate the minimal eigenvalue of its inertia matrix (see §3.2), the corresponding eigenvector is normal to the best plane defined by condition (2) and therefore it can be taken as \mathbf{N}_m .

The angle between two atomic groups can be described by the angle θ between two vectors normal to the best planes for the corresponding groups. This angle varies between the limits $0 \leq \theta \leq \pi/2$. This angle is expected to be zero if we require the planes to be parallel, which is the most frequent restraint of this kind. We therefore need to design a function which reaches its minimum when $\theta = 0$. Note that there are multiple

functions which could be constructed such that $\theta_0 = 0$ is a minimum. Also, the particular restraint function needs to describe the stereochemical knowledge near the point of interest (minimum) and have a desired behavior far from it. Additionally, the computation of the restraint and its derivatives must have no numerical irregularities. Below, we analyze some possible restraints that describe the angle between two planar groups, noting that different restraints can be used in different circumstances.

2.2. Analysis of various possible parallelity targets

The computationally simplest target to restrain the angle between normal vectors is

$$R_{\text{parallelity}} = w[1 - \cos(\theta - \theta_0)] \\ = w[1 - (\cos \theta \cos \theta_0 + \sin \theta \sin \theta_0)], \quad (4)$$

where $\theta = \theta(\mathbf{n}_1, \mathbf{n}_2)$ is the angle between two unit vectors \mathbf{n}_1 and \mathbf{n}_2 , θ_0 is the target angle, and w is the individual weight factor for a particular pair of planes. By default, *phenix.refine* (Afonine *et al.*, 2012) uses the form $w = 1/\sigma^2$, where σ is the root mean-square deviation from the ideal value obtained by some means. Introducing the angle θ_0 in equation (4) will force the planes to form the prescribed angle between them. For example, as suggested by Brown *et al.* (2015), a restraint with $\theta_0 = \pi/2$ is important when working with aromatic groups. For simplicity, we continue to call this restraint ‘parallelity’, even though this is not completely appropriate. We note that $\cos \theta = \mathbf{n}_1 \cdot \mathbf{n}_2$ and $\sin \theta = \|\mathbf{n}_1 \times \mathbf{n}_2\|$.

For small deviations from θ_0 , restraint (4) behaves as the quadratic function $\frac{1}{2}w(\theta - \theta_0)^2$. However, using the quadratic function of the angle itself [see in particular equation (14) of Brown *et al.* (2015)] requires its derivatives, *i.e.* the derivative of $\arccos(\mathbf{n}_1 \cdot \mathbf{n}_2)$, which is undefined for $\theta = \theta_0$. Also, its calculation is computationally unstable for $\theta \simeq \theta_0$. For this reason, we have excluded this kind of target from our consideration.

Also, a seemingly plausible restraint function that one may wish to avoid is

$$R_{\text{parallelity}} = w(\cos \theta - \cos \theta_0)^2. \quad (5)$$

This function is easy to calculate but it has very different behavior across different ranges of the parameter θ_0 . For example, it is similar to a second-order polynomial near $\theta_0 = \pi/2$ and to a fourth-order polynomial near $\theta_0 = 0$. Such heterogeneity is generally disadvantageous for minimization.

Another known target for the parallelity restraint is

$$R_{\text{parallelity}} = \sum_{k=1}^{K_1} w_{k,1}(\mathbf{r}_{k,1} \cdot \mathbf{n}_1 - d_1)^2 + \sum_{k=1}^{K_2} w_{k,2}(\mathbf{r}_{k,2} \cdot \mathbf{n}_1 - d_2)^2, \quad (6)$$

[equation (13) of Brown *et al.* (2015); the notation of the present article is used]. It imposes a rigid condition on the parallelity of two planes, $\mathbf{n}_2 = \mathbf{n}_1$, simultaneously with planarity of the corresponding groups. This constraint-type condition may be useful, for example, to simplify the parameterization of the model, but in practice nucleic acid bases may bend from

one base pair to another, or they may be twisted inside a given base pair. As a consequence, it is more convenient to have planarity and parallelity restraints defined separately, allowing for more flexibility if desired, in particular allowing the interplanar angle to vary from one planar group to another.

When the function is far from the minimum (*e.g.* when the current structure is distorted with respect to the restraints), one may wish that the target be relatively insensitive to the variation in the angle θ . This can be achieved by using various kinds of ‘top-out’ functions (Dennis & Welsch, 1987; Murshudov *et al.*, 2011; Smart *et al.*, 2012; Headd *et al.*, 2012). A direct application of this idea results in

$$R_{\text{parallelity}} = w\Omega^2 \left\{ 1 - \exp\left[\frac{\cos(\theta - \theta_0) - 1}{\Omega^2}\right] \right\}, \quad (7)$$

where Ω is a constant (Fig. 1). Using $(\theta - \theta_0)^2$ or $(\cos \theta - \cos \theta_0)^2$ in equation (7) instead of $\cos(\theta - \theta_0) - 1$ is not desirable for the reasons discussed above. In fact, we can use directly a sigmoid form of the cosine function and introduce the ‘top-out type’ targets in a simpler form:

$$R_{\text{parallelity}} = w[1 - \cos 2(\theta - \theta_0)] = 2w[1 - \cos^2(\theta - \theta_0)] \quad (8)$$

or

$$R_{\text{parallelity}} = \begin{cases} w[1 - \cos n(\theta - \theta_0)] & \text{if } |\theta - \theta_0| \leq \pi/n, \\ 2w & \text{if } |\theta - \theta_0| > \pi/n, \end{cases} \quad (9)$$

where $n > 2$. These functions (Fig. 1) have harmonic behavior when $\theta \simeq \theta_0$, while for large values of the argument they are constant [equation (9)] or nearly constant [equation (8)].

A further useful modification of target function (4) can be achieved by introducing the ‘slack’ parameter (square well-like potential), which scores small deviations from the target value equally. This can be achieved in a ‘soft’ way (Fig. 1) as

$$R_{\text{parallelity}} = w[1 - \cos(\theta - \theta_0)]^n, \quad n \geq 2, \quad (10)$$

or following explicitly the slack definition (Fig. 1) as

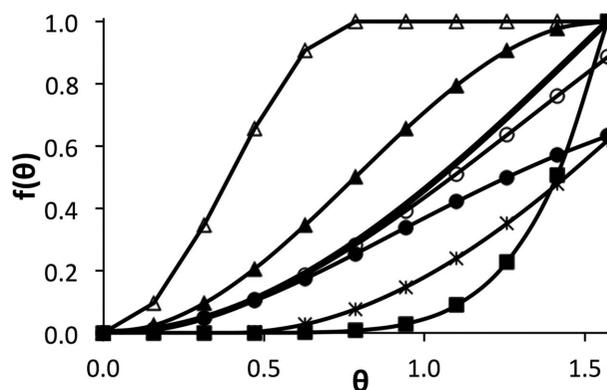


Figure 1
Examples of various parallelity targets, $\theta_0 = 0$. Heavy line with no markers: $f(\theta) = 1 - \cos \theta$, equation (4). Line with filled triangles: $f(\theta) = 1 - \cos 2\theta$, equation (8). Line with open triangles: $f(\theta) = 1 - \cos 4\theta$ for $\theta \leq \pi/4$, equation (9). Lines with black ($\Omega = 1$) and open ($\Omega = 2$) circles: $f(\theta) = \Omega^2[1 - \exp[(\cos \theta - 1)/\Omega^2]]$, equation (7). Line with filled squares: $f(\theta) = (1 - \cos \theta)^4$, equation (10). Line with asterisks: $f(\theta)$, slack function, equation (11).

$$R_{\text{parallelity}} = \begin{cases} 0 & \text{if } |\theta - \theta_0| \leq \text{slack}, \\ w[1 - \cos(\theta - \theta_0 - \text{slack})] & \text{if } \text{slack} < \theta - \theta_0, \\ w[1 - \cos(\theta - \theta_0 + \text{slack})] & \text{if } \theta - \theta_0 < -\text{slack}. \end{cases} \quad (11)$$

2.3. Practical calculation of the parallelity targets and their gradient

All the restraint functions mentioned above are similar in the sense that they require calculation of the normal vectors to the optimal planes of the atomic groups; then their dot product and finally a vector product of these vectors are calculated to obtain the target value.

A number of other restraints, such as group planarity or the distance between parallel groups, are naturally expressed through the same types of parameters used for the parallelity restraint (Appendix A). These parameters are the center of the atomic group, and the eigenvalues and eigenvectors of its inertia matrix (generally speaking, all three eigenvalues and all three eigenvectors may be required); a particular target may use only some of the parameters mentioned. Thus, the calculation of these parameters from the initial set of atomic coordinates is common to all such restraints on planar groups (Fig. 2). For this reason, in §3 we provide details of these common steps.

Assuming we have a function that we can calculate numerically, the principle of efficient gradient calculation (Baur & Strassen, 1983; Kim *et al.*, 1984) suggests inverting the steps of the function's calculation and obtaining the gradient step by step using the chain rule. This would result in the exact value of the gradient with respect to the original parameters, and this calculation will take no more than four times the CPU time required to calculate a single value of the function. This factor of four is invariant with respect to the number of parameters and the function type. In crystallographic practice, this factor of four is typically closer to one than to four (Lunin & Urzhumtsev, 1985). §4 illustrates this principle by calculating the gradient of the parallelity restraints. It is easy to see that the calculation of the gradient of other restraints for pseudo-plane groups differs only in the first step.

3. Planarity targets calculation

This section describes the calculation of the target value for the suggested restraint. Fig. 2 shows the overall scheme, where the letters *a* to *f* refer to the corresponding steps described below.

3.1. Centered coordinate system (Step a)

We start by calculating the atomic positions in the coordinate system with the origin shifted into the geometric center \mathbf{C}_m of the group, equation (3). This operation is similar for all pseudo-plane groups, and in what follows we omit the group index *m* to simplify the formulae. The new parameters are the Cartesian coordinates (X_k, Y_k, Z_k) of the atomic positions:

$$\mathbf{q}_k = \mathbf{r}_k - \left(\sum_{k=1}^K w_k \right)^{-1} \sum_{k=1}^K w_k \mathbf{r}_k, \quad k = 1, \dots, K, \quad (12)$$

and the Cartesian coordinates (C_x, C_y, C_z) of the weighted geometric center of the group.

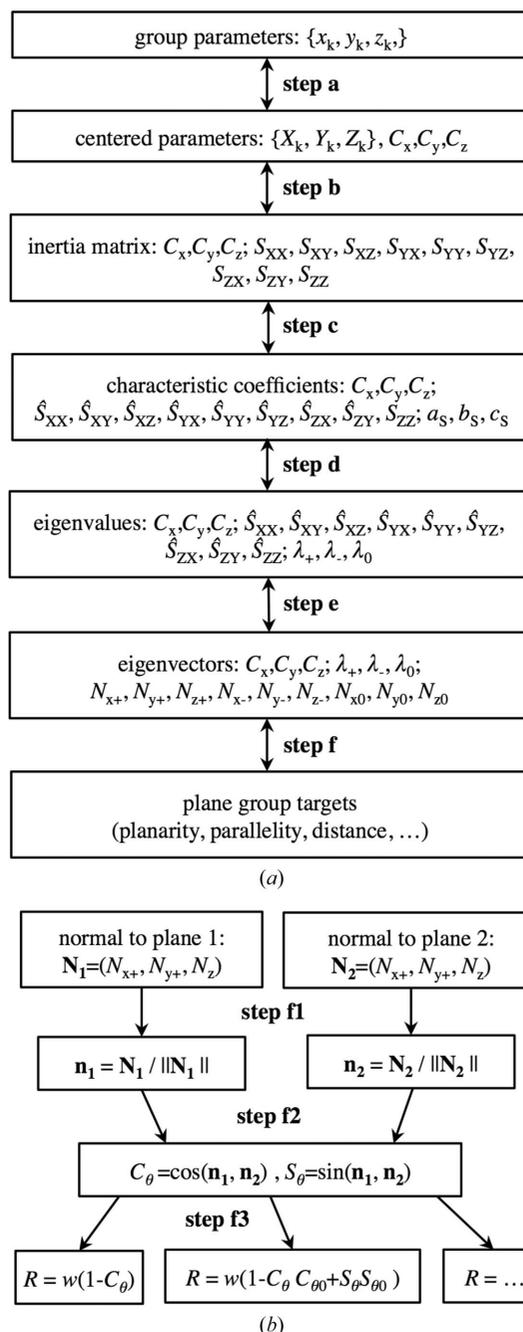


Figure 2 The overall calculation scheme. (a) The steps used to calculate parameters common to all targets. The letters *a–f* from top to bottom refer to the steps for calculating the target described in §3. The same steps in the direction from bottom to top refer to the steps for calculating the derivatives described in §4. The steps are applied independently to each atomic group required to be planar. (b) The particular parallelity targets that can be constructed on the basis of the calculated parameters of an atomic group.

3.2. Inertia matrix (Step b)

For a given atomic group, its inertia matrix is

$$\mathbf{S} = \begin{pmatrix} \sum_{k=1}^K w_k X_k X_k & \sum_{k=1}^K w_k X_k Y_k & \sum_{k=1}^K w_k X_k Z_k \\ \sum_{k=1}^K w_k Y_k X_k & \sum_{k=1}^K w_k Y_k Y_k & \sum_{k=1}^K w_k Y_k Z_k \\ \sum_{k=1}^K w_k Z_k X_k & \sum_{k=1}^K w_k Z_k Y_k & \sum_{k=1}^K w_k Z_k Z_k \end{pmatrix} = \begin{pmatrix} S_{XX} & S_{XY} & S_{XZ} \\ S_{YX} & S_{YY} & S_{YZ} \\ S_{ZX} & S_{ZY} & S_{ZZ} \end{pmatrix}. \quad (13)$$

This matrix is symmetric, $S_{XY} = S_{YX}$, $S_{YZ} = S_{ZY}$, $S_{ZX} = S_{XZ}$, but for convenience in subsequent calculations we refer to all nine of its elements. Matrix (13) is positive semidefinite, having always three non-negative eigenvalues, $0 \leq \lambda_+ \leq \lambda_- \leq \lambda_0$, with the corresponding eigenvectors orthogonal to each other. The result of this step contains nine elements of matrix (13) and three coordinates of the center [equation (12)] taken from the previous step.

It is important to remember that at each step we calculate the values of the new parameters and do not keep their analytical expression.

3.3. Characteristic equation (Step c)

The eigenvalues $0 \leq \lambda_+ \leq \lambda_- \leq \lambda_0$ of the inertia matrix (13) are the roots of the cubic characteristic equation

$$\lambda^3 + a_S \lambda^2 + b_S \lambda + c_S = 0, \quad (14)$$

with the coefficients

$$a_S = -S_{XX} - S_{YY} - S_{ZZ}, \quad (15a)$$

$$b_S = \det \begin{pmatrix} S_{XX} & S_{XY} \\ S_{YX} & S_{YY} \end{pmatrix} + \det \begin{pmatrix} S_{YY} & S_{YZ} \\ S_{ZY} & S_{ZZ} \end{pmatrix} + \det \begin{pmatrix} S_{ZZ} & S_{ZX} \\ S_{XZ} & S_{XX} \end{pmatrix}, \quad (15b)$$

$$c_S = -\det \mathbf{S}. \quad (15c)$$

Further steps of the procedure use both the coefficients of the cubic equation (14) and the elements of the inertia matrix (13). To avoid confusion when calculating the gradients [see, for example, equation (18) below], it is more convenient to introduce new variables formally:

$$\begin{aligned} \hat{S}_{XX} &= S_{XX}, & \hat{S}_{XY} &= S_{XY}, & \hat{S}_{XZ} &= S_{XZ}, \\ \hat{S}_{YX} &= S_{YX}, & \hat{S}_{YY} &= S_{YY}, & \hat{S}_{YZ} &= S_{YZ}, \\ \hat{S}_{ZX} &= S_{ZX}, & \hat{S}_{ZY} &= S_{ZY}, & \hat{S}_{ZZ} &= S_{ZZ}. \end{aligned} \quad (16)$$

Thus, the new set of parameters contains 15 values: three coordinates (C_x, C_y, C_z) of the center of the atomic group that are simply taken from the previous step, three equation coefficients a_S, b_S, c_S , and nine elements of the inertia matrix $\hat{S}_{XX}, \hat{S}_{XY}, \dots, \hat{S}_{ZZ}$.

3.4. Eigenvalues of the inertia matrix (Step d)

Different algorithms may be applied to calculate the eigenvalues and choose the minimum of them; we use the one described by Urzhumtsev (1991), as it gives analytically the minimal real eigenvalue that is required for the planarity and parallelity restraints. Appendix B gives an extended description in the notation of the present paper. The coordinates of the center of the atomic group [equation (12)] and the matrix elements [equation (16)] are not used at this step and pass to the next steps without change.

3.5. Eigenvectors of the inertia matrix (Step e)

Different algorithms may be applied to calculate the eigenvectors \mathbf{N} of the inertia matrix that are solutions of the matrix equation

$$(\hat{\mathbf{S}} - \lambda \mathbf{I})\mathbf{N} = 0, \quad (17)$$

with the eigenvalues λ obtained in the previous step. Appendix B gives technical details of the method used in our program. For an atomic group that is approximately planar, $\lambda_+ < \lambda_-$, the solution of equation (17) with $\lambda = \lambda_+$ defines unambiguously a single direction \mathbf{N}_+ of the normal to the best plane.

Note that other restraints on the plane groups may require knowledge of other eigenvalues and/or eigenvectors (Appendix A). Therefore, in the general case, the result of this step is a set of parameters fully describing a pseudo-planar atomic group, namely the three coordinates (C_x, C_y, C_z) of its center, all three eigenvalues $0 \leq \lambda_+ \leq \lambda_- \leq \lambda_0$ and all three eigenvectors $\mathbf{N}_+, \mathbf{N}_-, \mathbf{N}_0$ of the inertia matrix.

3.6. Parallelity targets (Step f)

To calculate a planarity restraint for two pseudo-planar atomic groups, each described by the Cartesian coordinates of the vector $\mathbf{N}_1 = \mathbf{N}_{1+}$ and $\mathbf{N}_2 = \mathbf{N}_{2+}$ normal to their respective best planes, we may need to follow several intermediate steps (Appendix D). We start (Step f1) from normalization of the normal vectors, $\{\mathbf{N}_1, \mathbf{N}_2\} \rightarrow \{\mathbf{n}_1, \mathbf{n}_2\}$. This normalization includes choosing the correct direction of the normalized vectors so that the value of the angle θ between the normal vectors is not larger than $\pi/2$. Once the normalized vectors are known, we calculate (Step f2) the cosine and sine of the angle θ between them, which we denote C_θ and S_θ , respectively. Finally (Step f3), we calculate the value of the chosen target.

4. Gradient calculation

4.1. Gradient with respect to the plane parameters

We start by inverting the formulae to calculate the chosen target (Step f3 in Fig. 2) and obtain $\partial R_{\text{parallelity}}/\partial C_\theta$ and $\partial R_{\text{parallelity}}/\partial S_\theta$ (Appendix D).

Inverting the previous step (Step f2), we take as input the values of $\partial R_{\text{parallelity}}/\partial C_\theta$ and $\partial R_{\text{parallelity}}/\partial S_\theta$ (and not their analytical expressions), whatever the target $R_{\text{parallelity}}$ is. As a result, we obtain $(\partial R_{\text{parallelity}}/\partial n_{1X}, \partial R_{\text{parallelity}}/\partial n_{1Y}, \partial R_{\text{parallelity}}/\partial n_{1Z})$

and $(\partial R_{\text{parallelity}}/\partial n_{2X}, \partial R_{\text{parallelity}}/\partial n_{2Y}, \partial R_{\text{parallelity}}/\partial n_{2Z})$. Finally, inverting the normalization (Step *f1*) we obtain $(\partial R_{\text{parallelity}}/\partial N_{1X}, \partial R_{\text{parallelity}}/\partial N_{1Y}, \partial R_{\text{parallelity}}/\partial N_{1Z})$ and $(\partial R_{\text{parallelity}}/\partial N_{2X}, \partial R_{\text{parallelity}}/\partial N_{2Y}, \partial R_{\text{parallelity}}/\partial N_{2Z})$.

Working with other restraints on plane groups, we can apply the same kind of procedure as Steps *f3* to *f1* above. For all these targets, we arrive at a set of partial derivatives of a target with respect to the following:

- (i) The three eigenvalues for each pseudo-planar group.
- (ii) The coordinates of the three eigenvectors for each pseudo-planar group.
- (iii) The coordinates of the center of each pseudo-planar group.

For the parallelity restraints discussed above, all these derivatives are equal to zero except those for the coordinates of \mathbf{N}_{1+} and \mathbf{N}_{2+} . Further steps of the gradient calculation use these partial derivatives as input data whatever the target is. These steps are common to all restraints and all atomic groups. Therefore, starting with the next section we omit the index used to indicate an atomic group and the index of a particular plane-group restraint for which all the partial derivatives above have been calculated. Partial derivatives with respect to the coordinates (C_x, C_y, C_z) will not change until the last step.

4.2. Gradient with respect to the coefficients of the characteristic equation (Steps *e* and *d*)

As mentioned in §§3.4 and 3.5, different algorithms exist for calculating the eigenvalues and eigenvectors of the inertia matrix from the coefficients of the characteristic equation (14). According to the main scheme, we invert these algorithms to calculate the corresponding derivatives. Appendix C provides these for the particular method of calculation chosen in our program. The output of this step contains the partial derivatives of the target R with respect to the three coefficients a_s, b_s, c_s of equation (14), to the nine elements $\hat{S}_{XX}, \hat{S}_{XY}, \dots, \hat{S}_{ZZ}$ of equation (16), and to the three coordinates (C_x, C_y, C_z) of the center of the atomic group that are taken, with no change, from the previous step.

4.3. Gradient with respect to the elements of the inertia matrix (Step *c*)

Using the partial derivatives of a_s, b_s, c_s [equation (50) in Appendix C] and of the $\hat{\mathbf{S}}$ elements [equation (43) in Appendix C] with respect to the elements of the matrix \mathbf{S} [equation (13)], we obtain

$$\begin{aligned} \frac{\partial R}{\partial S_{XX}} &= -\frac{\partial R}{\partial a_s} + (S_{YY} + S_{ZZ})\frac{\partial R}{\partial b_s} + (S_{YZ}S_{ZY} - S_{YY}S_{ZZ})\frac{\partial R}{\partial c_s} \\ &\quad + \frac{\partial R}{\partial \hat{S}_{XX}}, \\ \frac{\partial R}{\partial S_{XY}} &= -S_{YX}\frac{\partial R}{\partial b_s} + (S_{YX}S_{ZZ} - S_{ZX}S_{YZ})\frac{\partial R}{\partial c_s} + \frac{\partial R}{\partial \hat{S}_{XY}}, \\ \frac{\partial R}{\partial S_{XZ}} &= -S_{ZX}\frac{\partial R}{\partial b_s} + (S_{ZX}S_{YY} - S_{YX}S_{ZY})\frac{\partial R}{\partial c_s} + \frac{\partial R}{\partial \hat{S}_{XZ}}, \end{aligned} \tag{18}$$

and similar expressions for the other six derivatives easily obtained from equation (18) by the corresponding cyclic substitution $X \rightarrow Y \rightarrow Z \rightarrow X$ of the indices, e.g. to obtain $\partial R/\partial S_{YZ}$ we make the following substitutions in the second equation: $XY \rightarrow YZ, YX \rightarrow ZY, ZZ \rightarrow XX, ZX \rightarrow XY, YZ \rightarrow ZX$.

4.4. Gradient with respect to the coordinates in the centered system (Step *b*)

We obtain the derivatives with respect to the atomic coordinates, $k = 1, \dots, K$,

$$\begin{aligned} \frac{\partial R}{\partial X_k} &= 2X_k \frac{\partial R}{\partial S_{XX}} + Y_k \left(\frac{\partial R}{\partial S_{XY}} + \frac{\partial R}{\partial S_{YX}} \right) + Z_k \left(\frac{\partial R}{\partial S_{XZ}} + \frac{\partial R}{\partial S_{ZX}} \right), \\ \frac{\partial R}{\partial Y_k} &= 2Y_k \frac{\partial R}{\partial S_{YY}} + Z_k \left(\frac{\partial R}{\partial S_{YZ}} + \frac{\partial R}{\partial S_{ZY}} \right) + X_k \left(\frac{\partial R}{\partial S_{YX}} + \frac{\partial R}{\partial S_{XY}} \right), \\ \frac{\partial R}{\partial Z_k} &= 2Z_k \frac{\partial R}{\partial S_{ZZ}} + X_k \left(\frac{\partial R}{\partial S_{ZX}} + \frac{\partial R}{\partial S_{XZ}} \right) + Y_k \left(\frac{\partial R}{\partial S_{ZY}} + \frac{\partial R}{\partial S_{YZ}} \right). \end{aligned} \tag{19}$$

4.5. Gradient of restraint with respect to the original coordinates (Step *a*)

Using the partial derivatives of equations (3) and (12) and the notation $W_j = w_j(\sum_{k=1}^K w_k)^{-1}$,

$$\begin{aligned} \frac{\partial C_x}{\partial x_j} = \frac{\partial C_y}{\partial y_j} = \frac{\partial C_z}{\partial z_j} &= W_j, \\ \frac{\partial X_j}{\partial x_j} = \frac{\partial Y_j}{\partial y_j} = \frac{\partial Z_j}{\partial z_j} &= 1 - W_j, \\ \frac{\partial X_k}{\partial x_j} = \frac{\partial Y_k}{\partial y_j} = \frac{\partial Z_k}{\partial z_j} &= -W_j \quad \text{if } j \neq k, \end{aligned} \tag{20}$$

and knowing that all cross-term derivatives are equal to 0, we obtain the required gradient with respect to the original atomic coordinates

$$\begin{aligned} \frac{\partial R}{\partial x_j} &= \frac{\partial C_x}{\partial x_j} \frac{\partial R}{\partial C_x} + \sum_{k=1}^K \frac{\partial X_k}{\partial x_j} \frac{\partial R}{\partial X_k} = \frac{\partial R}{\partial X_j} \\ &\quad + W_j \left(\frac{\partial R}{\partial C_x} - \sum_{k=1}^K \frac{\partial R}{\partial X_k} \right), \\ \frac{\partial R}{\partial y_j} &= \frac{\partial R}{\partial Y_j} + W_j \left(\frac{\partial R}{\partial C_y} - \sum_{k=1}^K \frac{\partial R}{\partial Y_k} \right), \\ \frac{\partial R}{\partial z_j} &= \frac{\partial R}{\partial Z_j} + W_j \left(\frac{\partial R}{\partial C_z} - \sum_{k=1}^K \frac{\partial R}{\partial Z_k} \right). \end{aligned} \tag{21}$$

5. Implementation of parallelity restraints in *cctbx*

5.1. Computational Crystallography Toolbox

The *Computational Crystallography Toolbox* (*cctbx*) is an open-source library with numerous functionalities necessary

to construct crystallography-oriented programs. In fact, it finds applications beyond crystallography, such as small-angle scattering (*sastbx*; Liu *et al.*, 2012) or cryo-EM (Afonine *et al.*, 2013). *cctbx* contains tools to handle various objects such as crystal symmetry, diffraction data, maps (crystallographic or cryo-EM) and atomic models, and provides an array of building blocks necessary to construct complex procedures such as model refinement (Afonine *et al.*, 2012) or diffraction data analysis (Sauter *et al.*, 2013; Parkhurst *et al.*, 2014). In particular, to support macromolecular refinement algorithms *cctbx* includes tools to handle various geometry restraints (Grosse-Kunstleve & Adams, 2004). The main set of geometry restraints includes those on covalent bonds and angles, chirality, dihedral (torsion) angles, non-bonded interactions, planar groups, and now parallelity. The overall restraint target, the sum in equation (1), is the sum of contributions from all restraint types

$$\sum_{n=1}^N w_n R_n = \sum_{i=1}^{N_{\text{bonds}}} w_i R_{\text{bonds},i} + \sum_{i=1}^{N_{\text{angles}}} w_i R_{\text{angles},i} + \dots + \sum_{i=1}^{N_{\text{parallelities}}} w_i R_{\text{parallelities},i} \quad (22)$$

If additional information about the molecule or its environment is available and deemed necessary to use, additional terms can be added to equation (22) (Echols *et al.*, 2010; Headd *et al.*, 2012, 2014). Such additional information may be the regular conformations of a protein's secondary structure, non-crystallographic symmetry or similarity to a previously solved structure. Technically, adding additional restraints means adding additional terms to equation (22), which in turn means that an algorithm for calculating the additional restraint target and its gradient with respect to atomic coordinates, and the data structures for holding information about the atoms involved, are required.

The implementation of the parallelity restraint in *cctbx* is very similar to other restraints (angle, dihedral angle, planarity *etc.*) and therefore may serve as an example of the common framework for restraints.

5.2. General approach to defining stereochemical restraints in *cctbx*

The source code for commonly used restraints is located at http://sourceforge.net/p/cctbx/code/HEAD/tree/trunk/cctbx/geometry_restraints in the files `bond.h`, `bond_bpl.cpp`, `angle.h`, `angle_bpl.cpp`, `dihedral.h`, `dihedral_bpl.cpp`, `chirality.h`, `chirality_bpl.cpp`, `nonbonded.h`, `nonbonded_bpl.cpp`, `planarity.h`, `planarity_bpl.cpp`, `parallelity.h` and `parallelity_bpl.cpp`.

Files with the name `<type of restraint>_bpl.cpp` contain `boost.python` (Abrahams & Grosse-Kunstleve, 2003) descriptions of classes and functions for the restraint, and the `bpl` in the file names stands for 'boost.python library'. Generally, these files contain `boost.python` wrappers for classes describing the restraint and the associated data structure (termed proxy; see below) holding the information

necessary to calculate the restraint. Files with the name `<type of restraint>.h` contain the actual implementation of classes and functions for the particular type of restraint.

5.3. Proxy class

The calculation of a particular restraint requires information about the atoms involved, such as the atomic coordinates, and the restraint target and weight. Since the restraints are built once at the beginning of refinement, while the atomic coordinates change during refinement, information about the atoms that participate in a restraint and the actual atomic coordinates is decoupled. The data structure that holds the restraint information is called a proxy, and it contains integer indices that can be used to access a particular atom in an array of atoms and therefore individual coordinates, along with some other restraint-specific information.

For example, the bond proxy contains a pair of integer indices (atom numbers in an atom list), the restraint target value and weight, the slack and limit values, a flag that specifies which function to use (least-squares or top-out) and a rotation matrix operator in case bonded atoms belong to different crystallographic symmetry copies, and an indicator to distinguish the origin of a particular restraint (*e.g.* covalent or hydrogen bond). Since each proxy object contains all the parameters necessary to define a particular restraint, each restraint can be parameterized individually.

The parallelity proxy is organized in a very similar manner to the planarity proxy, the difference being that the atom numbers are split and contained in two groups because they define two separate planes with an arbitrary number of atoms in each one.

Usually, one constructs multiple restraints of one type for a structure, so it is convenient to have a structure to store an array of proxies of one type and perform various operations on them. Similarly to other restraint types, for the parallelity restraints there is an array that can contain multiple proxy objects, called the `shared_parallelity_proxy`. Operations on proxy arrays include selection (this allows selection of a subset of restraints corresponding to a part of the structure), deletion (restraints can be deleted for a part of structure), and calculation of root mean-square deviations of the model from target values, absolute differences between the restraint target and actual model values, the restraint target value, gradients and so on.

5.4. Class to handle parallelity restraint

The parallelity restraint itself [equation (4)] and its modifications [equations (7) and (11)] are implemented as a C++ class `parallelity` available in Python *via* `boost.python` from `cctbx.geometry_restraints.parallelity`. It has two constructors. One requires providing all parameters to define a parallelity proxy (such as target value, type of function *etc.*) and two sets of Cartesian atomic coordinates corresponding to the two planes. The other constructor takes an array with Cartesian coordinates of atoms and an instance of the paral-

lelity proxy class that contains the rest of the required information.

All constructors contain a call to the `init_deltas()` function which calculates all necessary values, such as the overall restraint value and the gradients. This means that instantiation of a parallelity object leads to the immediate calculation of the target value and gradients for the restraint.

To simplify the handling of the slack functionality and to make it universal for harmonic and top-out potentials, the conditions of equation (11) are evaluated first and θ_0 adjusted accordingly before any other calculations. This makes it possible to avoid the gradient calculation if it is not necessary, for example when $|\theta - \theta_0| \leq \text{slack}$, and also to use all of the derivative equations for harmonic and top-out potentials without change.

In addition to the two constructors, the parallelity class has member functions `residual()` and `gradients()` that return the value of the restraint function and gradients with respect to the atomic coordinates. Since all necessary calculations are already performed in the constructor, these functions return precalculated values.

6. Discussion

Applying the general principles of refinement program construction, it is straightforward to build various stereochemical restraints. However, a careful analysis is required to choose an appropriate function for the restraint target. This choice is based on a function's ability to provide the desired behavior, both near the minimum and far from it. Possible singularities when calculating the target and its derivatives must also be taken into account. With a view to improving the refinement of atomic models, especially at low resolution, we have developed new restraints to control the angle between two approximately planar atomic groups. As a particular and more frequent case, this restraint imposes the groups to be parallel or near parallel. We have proposed a practical algorithm to calculate this target and its gradient. This algorithm has been implemented in the *cctbx* open-source library, making it available in *PHENIX* and to the broader crystallographic and macromolecular structure community. All calculation details are provided, making it easy to add other types of restraint in the future.

APPENDIX A

Other restraints on plane groups and their gradients

A1. Planarity targets

This restraint requires an atomic group to be as planar as possible. Calculation of the target uses exactly the same Steps *a–e* described in §3 and Fig. 2 and differs only in Step *f*. As for the other steps, in order to avoid confusion below when calculating the gradients we introduce new parameters $\hat{\lambda}_+ = \lambda_+$, $\hat{\lambda}_- = \lambda_-$, $\hat{\lambda}_0 = \lambda_0$ instead of the eigenvalues of the inertia matrix (Appendix C).

(Step *f*) The simplest target can be expressed as (Urzhumtsev, 1991)

$$R_{\text{planarity}}(\hat{\lambda}_+) = w\hat{\lambda}_+, \quad (23)$$

which takes the value of the minimal eigenvalue already calculated at Step *d*. Another approach used currently (Blanc & Paciorek, 2001) uses a different method of calculation but is essentially the same.

The restraint of equation (23) does not take into account the number of atoms in the restrained group. This leads to the effect that the more populated group (*i.e.* the group with more atoms) will be forced to be more planar than the less populated group given the same weight. This may be corrected by normalizing by the number of atoms in the group:

$$R_{\text{planarity}}(\hat{\lambda}_+) = \frac{w}{K_{\text{atoms}}} \hat{\lambda}_+. \quad (24)$$

While equation (23) expresses a type of absolute value of the ‘flatness’ of the ellipsoid of inertia, other targets may estimate a relative value of such a flatness using several eigenvalues, for example

$$R_{\text{planarity}}(\hat{\lambda}_+, \hat{\lambda}_0) = \frac{w\hat{\lambda}_+}{\hat{\lambda}_0}. \quad (25)$$

For this target the normalization with respect to the number of atoms is performed automatically.

A2. Parallel-distance target

Another geometric restraint that can be imposed on atomic models, especially those of nucleic acids, is requiring the distance between two parallel planes to be equal to a prescribed value l_{target} . Its calculation follows the same algorithm presented above. We first pass through the same Steps *a–e*, finishing as follows.

(Step *f1*) We repeat Step *f1* of the parallelity restraint (§3.6) and calculate normalized eigenvectors.

(Step *f2*) We then calculate the median (subscript *med*) direction between two normal vectors,

$$\mathbf{N}_{\text{med}} = \frac{\mathbf{n}_1 + \mathbf{n}_2}{2}, \quad (26)$$

(Step *f3*) and normalize it,

$$\mathbf{n}_{\text{med}} = \frac{\mathbf{N}_{\text{med}}}{(\mathbf{N}_{\text{med}} \cdot \mathbf{N}_{\text{med}})^{1/2}}. \quad (27)$$

(Step *f4*) Finally, we calculate the ‘oriented’ distance between the best planes that pass through the two atomic groups as

$$l = (\mathbf{C}_2 - \mathbf{C}_1) \cdot \mathbf{n}_{\text{med}}, \quad (28)$$

(Step *f5*) and the target itself,

$$R_{\text{par-dist}}(\mathbf{N}_1, \mathbf{N}_2, \mathbf{C}_1, \mathbf{C}_2) = w(l^2 - l_{\text{target}}^2)^2. \quad (29)$$

Target (29) uses the square of the parameter from equation (28) to remove the influence of the sign of the projection of $(\mathbf{C}_2 - \mathbf{C}_1)$ on the unit vector \mathbf{n}_{med} .

APPENDIX B

Eigenvalues and eigenvectors of the inertia matrix

Below, when describing the intermediate transformations of parameters at Steps *d* and *e* of the principal algorithm used in our program, the output parameters of each transformation are the input parameters for the next one. We start from the following input parameters:

$$C_x, C_y, C_z, \hat{S}_{XX}, \hat{S}_{XY}, \hat{S}_{XZ}, \hat{S}_{YX}, \hat{S}_{YY}, \hat{S}_{YZ}, \hat{S}_{ZX}, \hat{S}_{ZY}, \hat{S}_{ZZ}, a_S, b_S, c_S. \tag{30}$$

At Step *d*, all these parameters except a_S , b_S and c_S remain unchanged and we omit them in the explicit list of parameters in this appendix.

(Step *d1*) We replace the initial equation (14) by

$$(\lambda + \frac{1}{3}a_S)^3 + g_S(\lambda + \frac{1}{3}a_S) + h_S = 0, \tag{31}$$

with the new parameters replacing a_S , b_S and c_S

$$\begin{aligned} g_S &= -(\frac{1}{3}a_S^2 + b_S) = \frac{1}{3}a_S^2 - b_S, \\ h_S &= \frac{2}{27}a_S^3 - \frac{1}{3}a_S b_S + c_S, \\ \hat{a}_S &= \frac{1}{3}a_S, \end{aligned} \tag{32}$$

If the value of g_S in equation (32) is close to zero, or even slightly negative because of rounding errors, we assign $g_S = 0$; when an atomic group is already more or less planar this will not be the case.

(Step *d2*) We calculate the cosine ξ_S of the characteristic angle $3\gamma_S$ as

$$\begin{aligned} \xi_S &= -\frac{1}{2}h_S(\frac{1}{3}g_S)^{-3/2}, \\ \hat{g}_S &= 2(\frac{1}{3}g_S)^{1/2}, \end{aligned} \tag{33}$$

resulting in the output parameters \hat{a}_S , \hat{g}_S and ξ_S .

(Step *d3*) First, we calculate

$$\gamma_S = \frac{1}{3} \arccos \xi_S, \tag{34}$$

and then

$$\tau_{S+} = \cos(\gamma_S + \frac{2}{3}\pi), \tau_{S-} = \cos(\gamma_S - \frac{2}{3}\pi), \tau_{S0} = \cos(\gamma_S). \tag{35}$$

The output parameters are \hat{a}_S , \hat{g}_S , τ_{S+} , τ_{S-} and τ_{S0} .

(Step *d4*) The eigenvalues are

$$\lambda_+ = \hat{g}_S \tau_{S+} - \hat{a}_S, \lambda_- = \hat{g}_S \tau_{S-} - \hat{a}_S, \lambda_0 = \hat{g}_S \tau_{S0} - \hat{a}_S, \tag{36}$$

where $\lambda_+ \leq \lambda_- \leq \lambda_0$. We then calculate the corresponding eigenvectors starting from the set of parameters

$$C_x, C_y, C_z, \hat{S}_{XX}, \hat{S}_{XY}, \hat{S}_{XZ}, \hat{S}_{YX}, \hat{S}_{YY}, \hat{S}_{YZ}, \hat{S}_{ZX}, \hat{S}_{ZY}, \hat{S}_{ZZ}, \lambda_+, \lambda_-, \lambda_0. \tag{37}$$

(Step *e1*) The eigenvector \mathbf{N}_+ corresponding to the minimal eigenvalue λ_+ satisfies the condition

$$(\hat{\mathbf{S}} - \lambda_+ \mathbf{I})\mathbf{N}_+ = 0. \tag{38}$$

This means that it is orthogonal to the three vectors $\mathbf{t}_X = (\hat{S}_{XX} - \lambda_+, \hat{S}_{XY}, \hat{S}_{XZ})$, $\mathbf{t}_Y = (\hat{S}_{YX}, \hat{S}_{YY} - \lambda_+, \hat{S}_{YZ})$, $\mathbf{t}_Z = (\hat{S}_{ZX}, \hat{S}_{ZY}, \hat{S}_{ZZ} - \lambda_+)$. In other words, \mathbf{N}_+ can be taken as the vector product of any two of them. To minimize rounding

errors and avoid singular situations, for example when two chosen vectors are collinear or practically collinear to each other, we check all permutations

$$(\mathbf{t}_1 = \mathbf{t}_X, \mathbf{t}_2 = \mathbf{t}_Y), (\mathbf{t}_1 = \mathbf{t}_Y, \mathbf{t}_2 = \mathbf{t}_Z), (\mathbf{t}_1 = \mathbf{t}_Z, \mathbf{t}_2 = \mathbf{t}_X). \tag{39}$$

For each of these permutations we calculate the vector

$$\begin{aligned} \mathbf{N}_+ &= \mathbf{t}_1 \times \mathbf{t}_2 \\ &= (t_{1Y}t_{2Z} - t_{1Z}t_{2Y}, t_{1Z}t_{2X} - t_{1X}t_{2Z}, t_{1X}t_{2Y} - t_{1Y}t_{2X}), \end{aligned} \tag{40}$$

and accept the permutation for which the norm of equation (40) is maximal.

(Step *e2*) After selecting vectors \mathbf{t}_1 and \mathbf{t}_2 we take the corresponding product of equation (40) as the eigenvector to define the 'best plane'.

The calculation of other eigenvectors (if necessary) follows the same procedure. Currently known geometric restraints, including parallelity ones, do not require more than one eigenvector that corresponds to the minimal eigenvalue λ_+ . However, considering other possible new restraints, the complete transition from atomic coordinates to plane parameters in the general case should include as output all three eigenvalues and all three corresponding eigenvectors

$$\lambda_+, \lambda_-, \lambda_0, \mathbf{N}_+, \mathbf{N}_-, \mathbf{N}_0, C_x, C_y, C_z. \tag{41}$$

APPENDIX C

Gradient with respect to the coefficients of the characteristic equation

Recalculation of the gradient is applied to each pseudo-planar group, one by one, and does not depend on the particular plane-group restraint R . The input to this computational module is a set of partial derivatives of R with respect to the center of the group, to the three eigenvalues and to the coordinates of the three eigenvectors, as mentioned above. For a particular restraint, some of these derivatives, if not most of them, may be equal to zero.

(Step *e2*) Derivation of the eigenvector with respect to the \mathbf{t} vectors.

For the eigenvector $\mathbf{N} = \mathbf{N}_+$ calculated by equation (40) we have

$$\begin{aligned} \frac{\partial R}{\partial t_{1X}} &= \frac{\partial N_X}{\partial t_{1X}} \frac{\partial R}{\partial N_X} + \frac{\partial N_Y}{\partial t_{1X}} \frac{\partial R}{\partial N_Y} + \frac{\partial N_Z}{\partial t_{1X}} \frac{\partial R}{\partial N_Z} \\ &= 0 - t_{2Z} \frac{\partial R}{\partial N_Y} + t_{2Y} \frac{\partial R}{\partial N_Z}, \\ \frac{\partial R}{\partial t_{1Y}} &= -t_{2X} \frac{\partial R}{\partial N_Z} + t_{2Z} \frac{\partial R}{\partial N_X}, \\ \frac{\partial R}{\partial t_{1Z}} &= -t_{2Y} \frac{\partial R}{\partial N_X} + t_{2X} \frac{\partial R}{\partial N_Y}. \end{aligned} \tag{42}$$

The derivatives $\partial R/\partial t_{2X}$, $\partial R/\partial t_{2Y}$ and $\partial R/\partial t_{2Z}$ differ from equation (42) by swapping the indices 1 and 2 and inverting the sign.

(Step *e1*) Derivation of the eigenvector with respect to the eigenvalues and $\hat{\mathbf{S}}$ elements.

The partial derivatives with respect to the $\hat{\mathbf{S}}$ elements depend on the choice of vectors for \mathbf{t}_1 and \mathbf{t}_2 when calculating \mathbf{N} , equation (40). For example, when ($\mathbf{t}_1 = \mathbf{t}_x$, $\mathbf{t}_2 = \mathbf{t}_y$)

$$\begin{aligned} \frac{\partial R}{\partial \hat{S}_{XX}} &= \frac{\partial R}{\partial t_{1X}}, \\ \frac{\partial R}{\partial \hat{S}_{XY}} &= \frac{\partial R}{\partial t_{1Y}}, \\ \frac{\partial R}{\partial \hat{S}_{XZ}} &= \frac{\partial R}{\partial t_{1Z}}, \\ \frac{\partial R}{\partial \hat{S}_{YX}} &= \frac{\partial R}{\partial t_{2X}}, \\ \frac{\partial R}{\partial \hat{S}_{YY}} &= \frac{\partial R}{\partial t_{2Y}}, \\ \frac{\partial R}{\partial \hat{S}_{YZ}} &= \frac{\partial R}{\partial t_{2Z}}, \\ \frac{\partial R}{\partial \hat{S}_{ZX}} &= \frac{\partial R}{\partial \hat{S}_{ZY}} = \frac{\partial R}{\partial \hat{S}_{ZZ}} = 0, \\ \frac{\partial R}{\partial \lambda_+} &= -\frac{\partial R}{\partial t_{1X}} - \frac{\partial R}{\partial t_{2Y}} + \frac{\partial R}{\partial \lambda_+}. \end{aligned} \quad (43)$$

The equations are similar for other choices of \mathbf{t}_1 and \mathbf{t}_2 . It is important not to forget the direct partial derivative $\partial R/\partial \lambda_+$ if the target depends directly on the eigenvalue. The partial derivatives of (43) with respect to other eigenvalues, for example when using targets such as equation (24), are calculated similarly. In this work we do not consider the targets depending on eigenvectors other than \mathbf{N}_+ ; equations (42) and (43) may easily be generalized for more complicated targets in the future, but here we avoid giving these simple but lengthy expressions.

(Step d4) Derivation of eigenvalues with respect to \hat{a}_s , \hat{g}_s , τ_{s+} , τ_{s-} , τ_{s0} . For each eigenvalue we obtain

$$\begin{aligned} \frac{\partial R}{\partial \hat{a}_s} &= \frac{\partial \lambda_+}{\partial \hat{a}_s} \frac{\partial R}{\partial \lambda_+} + \frac{\partial \lambda_-}{\partial \hat{a}_s} \frac{\partial R}{\partial \lambda_-} + \frac{\partial \lambda_0}{\partial \hat{a}_s} \frac{\partial R}{\partial \lambda_0} \\ &= -\frac{\partial R}{\partial \lambda_+} - \frac{\partial R}{\partial \lambda_-} - \frac{\partial R}{\partial \lambda_0}, \\ \frac{\partial R}{\partial \hat{g}_s} &= \tau_{s+} \frac{\partial R}{\partial \lambda_+} + \tau_{s-} \frac{\partial R}{\partial \lambda_-} + \tau_{s0} \frac{\partial R}{\partial \lambda_0}, \\ \frac{\partial R}{\partial \tau_{s+}} &= \hat{g}_s \frac{\partial R}{\partial \lambda_+}, \\ \frac{\partial R}{\partial \tau_{s-}} &= \hat{g}_s \frac{\partial R}{\partial \lambda_-}, \\ \frac{\partial R}{\partial \tau_{s0}} &= \hat{g}_s \frac{\partial R}{\partial \lambda_0}. \end{aligned} \quad (44)$$

Partial derivatives with respect to the matrix $\hat{\mathbf{S}}$ elements were calculated above [equation (43)].

(Step d3) Derivation of eigenvalues with respect to intermediate variables. Using the property of a derivative of the inverse function, we have

$$\begin{aligned} \frac{d\tau_{s+}}{d\xi_s} &= \left(\frac{d\xi_s}{d\tau_{s+}} \right)^{-1} = \left[\frac{d \cos(3\gamma)}{d \cos(\gamma + 2\pi/3)} \right]^{-1} \\ &= \left[\frac{d(4\tau_{s+}^3 - 3\tau_{s+})}{d\tau_{s+}} \right]^{-1} = \frac{1}{12\tau_{s+}^2 - 3}, \end{aligned} \quad (45)$$

and similar equations for τ_{s-} and τ_{s0} . Therefore

$$\frac{\partial R}{\partial \xi_s} = \frac{1}{(12\tau_{s+}^2 - 3)} \frac{\partial R}{\partial \tau_{s+}} + \frac{1}{(12\tau_{s-}^2 - 3)} \frac{\partial R}{\partial \tau_{s-}} + \frac{1}{(12\tau_{s0}^2 - 3)} \frac{\partial R}{\partial \tau_{s0}}. \quad (46)$$

(Step d2) To obtain the derivatives with respect to g_s and h_s , we first calculate

$$\begin{aligned} \frac{d\hat{g}_s}{dg_s} &= \left(\frac{1}{3}g_s \right)^{-1/2}, \quad \frac{d\xi_s}{dg_s} = \frac{1}{4}h_s \left(\frac{1}{3}g_s \right)^{-5/2}, \\ \frac{d\xi_s}{dh_s} &= \left(-\frac{1}{2} \right) \left(\frac{1}{3}g_s \right)^{-3/2}, \end{aligned} \quad (47)$$

with which

$$\frac{\partial R}{\partial g_s} = \frac{\partial \xi_s}{\partial g_s} \frac{\partial R}{\partial \xi_s} + \frac{\partial \hat{g}_s}{\partial g_s} \frac{\partial R}{\partial \hat{g}_s}, \quad \frac{\partial R}{\partial h_s} = \frac{\partial \xi_s}{\partial h_s} \frac{\partial R}{\partial \xi_s}. \quad (48)$$

(Step d1) Derivation of eigenvalues with respect to the coefficients of the original characteristic equation. We can use

$$\begin{aligned} \frac{\partial h_s}{\partial a_s} &= \frac{2}{9}a_s^2 - \frac{1}{3}b_s, \quad \frac{\partial h_s}{\partial b_s} = -\frac{1}{3}a_s, \quad \frac{\partial h_s}{\partial c_s} = 1 \\ \frac{\partial g_s}{\partial a_s} &= \frac{2}{3}a_s, \quad \frac{\partial g_s}{\partial b_s} = -1, \quad \frac{\partial \hat{a}_s}{\partial a_s} = \frac{1}{3}, \end{aligned} \quad (49)$$

This results in

$$\begin{aligned} \frac{\partial R}{\partial a_s} &= \frac{\partial h_s}{\partial a_s} \frac{\partial R}{\partial h_s} + \frac{\partial g_s}{\partial a_s} \frac{\partial R}{\partial g_s} + \frac{\partial \hat{a}_s}{\partial a_s} \frac{\partial R}{\partial \hat{a}_s} \\ &= \left(\frac{2a_s^2 - 3b_s}{9} \right) \frac{\partial R}{\partial h_s} + \frac{2a_s}{3} \frac{\partial R}{\partial g_s} + \frac{1}{3} \frac{\partial R}{\partial \hat{a}_s}, \\ \frac{\partial R}{\partial b_s} &= \frac{\partial h_s}{\partial b_s} \frac{\partial R}{\partial h_s} + \frac{\partial g_s}{\partial b_s} \frac{\partial R}{\partial g_s} = -\frac{1}{3}a_s \frac{\partial R}{\partial h_s} - \frac{\partial R}{\partial g_s}, \\ \frac{\partial R}{\partial c_s} &= \frac{\partial h_s}{\partial c_s} \frac{\partial R}{\partial h_s} = \frac{\partial R}{\partial h_s}, \end{aligned} \quad (50)$$

APPENDIX D

Calculation of the parallelity targets and their gradients with respect to the normal vectors

D1. Parallelity targets

Let each of the two pseudo-planar atomic groups be described by the Cartesian coordinates of the vector normal to its respective best plane (\mathbf{N}_1 and \mathbf{N}_2), not necessarily of unit length.

(Step f1) First we calculate the unit normal vectors as

$$\mathbf{n}_1 = \frac{\mathbf{N}_1}{(\mathbf{N}_1 \cdot \mathbf{N}_1)^{1/2}}, \quad \mathbf{n}_2 = \frac{\sigma \mathbf{N}_2}{(\mathbf{N}_2 \cdot \mathbf{N}_2)^{1/2}}, \quad \sigma = \pm 1. \quad (51)$$

The correct sign for σ must be chosen so that the absolute value of the angle θ between the normal vectors is not larger than $\pi/2$.

(Step *f2*) Once the normalized vectors are known, we calculate the cosine and sine of the angle θ between them, considering that $\sin\theta \geq 0$ because $0 \leq \theta \leq \pi/2$:

$$C_\theta = \cos(\mathbf{n}_1, \mathbf{n}_2) = \mathbf{n}_1 \cdot \mathbf{n}_2 = n_{1X}n_{2X} + n_{1Y}n_{2Y} + n_{1Z}n_{2Z}, \quad (52)$$

$$\begin{aligned} \mathbf{n} &= (n_X, n_Y, n_Z) = \mathbf{n}_1 \times \mathbf{n}_2 \\ &= (n_{1Y}n_{2Z} - n_{1Z}n_{2Y}, n_{1Z}n_{2X} - n_{1X}n_{2Z}, n_{1X}n_{2Y} - n_{1Y}n_{2X}), \end{aligned} \quad (53)$$

$$S_\theta = \|\mathbf{n}\| = (n_X^2 + n_Y^2 + n_Z^2)^{1/2}. \quad (54)$$

(Step *f3*) Finally, we calculate the value of the chosen target. Generally, one needs first to calculate the intermediate parameters

$$\begin{aligned} C_{D\theta} &= C_\theta \cos\theta_0 + S_\theta \sin\theta_0, \\ S_{D\theta} &= S_\theta \cos\theta_0 - C_\theta \sin\theta_0, \end{aligned} \quad (55)$$

which lead to the targets of equations (4), (7) and (8) as

$$R_{\text{parallelity}} = w(1 - C_{D\theta}), \quad (56)$$

$$R_{\text{parallelity}} = w\Omega^2 \left[1 - \exp\left(\frac{C_{D\theta} - 1}{\Omega^2}\right) \right], \quad (57)$$

$$R_{\text{parallelity}} = 2w(1 - C_{D\theta}^2). \quad (58)$$

Equation (10) for $n > 2$ may require, for example, using the De Moivre formula (Lial *et al.*, 2008) $C_{nD\theta} = \text{Re}(C_{D\theta} + iS_{D\theta})^n$; we do not provide these expressions here.

Obviously, the particular and most frequent case $\theta_0 = 0$ can be considered as an independent target

$$R_{\text{parallelity}} = w(1 - C_\theta), \quad (59)$$

which does not require equations (53)–(55).

D2. Gradient with respect to the normal vectors

(Step *f3*) Inversion of the last step of the target calculation provides the values of the gradients of the targets of equations (4), (7) and (8) with respect to the intermediate parameters as

$$\frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} = -w, \quad \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} = 0, \quad (60)$$

$$\frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} = -w \exp\left(\frac{C_{D\theta} - 1}{\Omega^2}\right), \quad \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} = 0, \quad (61)$$

$$\frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} = -4wC_{D\theta}, \quad \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} = 0. \quad (62)$$

Then, using the chain rule,

$$\begin{aligned} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} &= \frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} \frac{\partial C_{D\theta}}{\partial C_\theta} + \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} \frac{\partial S_{D\theta}}{\partial C_\theta} \\ &= \frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} \cos\theta_0 - \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} \sin\theta_0, \\ \frac{\partial R_{\text{parallelity}}}{\partial S_\theta} &= \frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} \frac{\partial C_{D\theta}}{\partial S_\theta} + \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} \frac{\partial S_{D\theta}}{\partial S_\theta} \\ &= \frac{\partial R_{\text{parallelity}}}{\partial C_{D\theta}} \sin\theta_0 + \frac{\partial R_{\text{parallelity}}}{\partial S_{D\theta}} \cos\theta_0. \end{aligned} \quad (63)$$

(Step *f2*) At the next step, applying the chain rule, we obtain

$$\begin{aligned} \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} &= \frac{\partial C_\theta}{\partial n_{1X}} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \frac{\partial S_\theta}{\partial n_{1X}} \frac{\partial R_{\text{parallelity}}}{\partial S_\theta} \\ &= n_{2X} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \left(\frac{\partial n_X}{\partial n_{1X}} \frac{\partial S_\theta}{\partial n_X} + \frac{\partial n_Y}{\partial n_{1X}} \frac{\partial S_\theta}{\partial n_Y} + \frac{\partial n_Z}{\partial n_{1X}} \frac{\partial S_\theta}{\partial n_Z} \right) \\ &\quad \times \frac{\partial R_{\text{parallelity}}}{\partial S_\theta} \\ &= n_{2X} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \left(\frac{\partial n_X}{\partial n_{1X}} \frac{n_X}{S_\theta} + \frac{\partial n_Y}{\partial n_{1X}} \frac{n_Y}{S_\theta} + \frac{\partial n_Z}{\partial n_{1X}} \frac{n_Z}{S_\theta} \right) \\ &\quad \times \frac{\partial R_{\text{parallelity}}}{\partial S_\theta} \\ &= n_{2X} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \left(n_{2Y} \frac{n_Z}{S_\theta} - n_{2Z} \frac{n_Y}{S_\theta} \right) \frac{\partial R_{\text{parallelity}}}{\partial S_\theta}, \end{aligned} \quad (64)$$

and similarly

$$\begin{aligned} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} &= n_{2Y} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \left(n_{2Z} \frac{n_X}{S_\theta} - n_{2X} \frac{n_Z}{S_\theta} \right) \frac{\partial R_{\text{parallelity}}}{\partial S_\theta}, \\ \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} &= n_{2Z} \frac{\partial R_{\text{parallelity}}}{\partial C_\theta} + \left(n_{2X} \frac{n_Y}{S_\theta} - n_{2Y} \frac{n_X}{S_\theta} \right) \frac{\partial R_{\text{parallelity}}}{\partial S_\theta}. \end{aligned} \quad (65)$$

The expressions for $\partial R_{\text{parallelity}}/\partial n_{2X}$, $\partial R_{\text{parallelity}}/\partial n_{2Y}$ and $\partial R_{\text{parallelity}}/\partial n_{2Z}$ are similar to equations (64) and (65) with n_{1X} , n_{1Y} , n_{1Z} instead of n_{2X} , n_{2Y} , n_{2Z} in the right-hand parts and with a $-$ sign instead of a $+$ before the parentheses.

(Step *f1*) Taking the values obtained in equations (64) and (65), we calculate

$$\begin{aligned} \frac{\partial R_{\text{parallelity}}}{\partial N_{1X}} &= \frac{\partial n_{1X}}{\partial N_{1X}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} + \frac{\partial n_{1Y}}{\partial N_{1X}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} \\ &\quad + \frac{\partial n_{1Z}}{\partial N_{1X}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} \\ &= \frac{N_{1Y}^2 + N_{1Z}^2}{(\mathbf{N}_1 \cdot \mathbf{N}_1)^{3/2}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} - \frac{N_{1X}N_{1Y}}{(\mathbf{N}_1 \cdot \mathbf{N}_1)^{3/2}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} \\ &\quad - \frac{N_{1X}N_{1Z}}{(\mathbf{N}_1 \cdot \mathbf{N}_1)^{3/2}} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} \\ &= \|\mathbf{N}_1\|^{-3} \left[(N_{1Y}^2 + N_{1Z}^2) \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} \right. \\ &\quad \left. - N_{1X}N_{1Y} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} - N_{1X}N_{1Z} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} \right], \end{aligned} \quad (66)$$

where $\|\mathbf{N}_1\| = (\mathbf{N}_1 \cdot \mathbf{N}_1)^{1/2}$. In the same way, we obtain

$$\frac{\partial R_{\text{parallelity}}}{\partial N_{1Y}} = \|\mathbf{N}_1\|^{-3} \left[(N_{1Z}^2 + N_{1X}^2) \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} - N_{1Y}N_{1Z} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} - N_{1Y}N_{1X} \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} \right], \quad (67)$$

$$\frac{\partial R_{\text{parallelity}}}{\partial N_{1Z}} = \|\mathbf{N}_1\|^{-3} \left[(N_{1X}^2 + N_{1Y}^2) \frac{\partial R_{\text{parallelity}}}{\partial n_{1Z}} - N_{1Z}N_{1X} \frac{\partial R_{\text{parallelity}}}{\partial n_{1X}} - N_{1Z}N_{1Y} \frac{\partial R_{\text{parallelity}}}{\partial n_{1Y}} \right]. \quad (68)$$

Expressions for $\partial R_{\text{parallelity}}/\partial N_{2X}$, $\partial R_{\text{parallelity}}/\partial N_{2Y}$ and $\partial R_{\text{parallelity}}/\partial N_{2Z}$ are obtained from equations (66)–(68) by substituting $\sigma\|\mathbf{N}_2\|^{-3}$ for $\|\mathbf{N}_1\|^{-3}$ and replacing the index 1 by 2 for the rest.

Acknowledgements

This work was supported by the NIH (project 1P01 GM063210), by the PHENIX Industrial Consortium, and in part by the US Department of Energy under contract No. DE-AC02-05CH11231. AU thanks the French Infrastructure for Integrated Structural Biology (FRISBI) (grant No. ANR-10-INSB-05-01) and Instruct, part of the European Strategy Forum on Research Infrastructures (ESFRI).

References

Abrahams, D. & Grosse-Kunstleve, R. W. (2003). *C/C++ Users J.* **21**, 29–36.
 Adams *et al.* (2010). *Acta Cryst.* **D66**, 213–221.
 Afonine, P. V., Grosse-Kunstleve, R. W., Echols, N., Headd, J. J., Moriarty, N. W., Mustyakimov, M., Terwilliger, T. C., Urzhumtsev, A., Zwart, P. H. & Adams, P. D. (2012). *Acta Cryst.* **D68**, 352–367.
 Afonine, P. V., Headd, J. J., Terwilliger, T. C. & Adams, P. D. (2013). *Comput. Crystallogr. Newsl.* **4**, 43–44.

Afonine, P. V. & Urzhumtsev, A. (2004). *Acta Cryst.* **A60**, 19–32.
 Baur, W. & Strassen, V. (1983). *Theor. Comput. Sci.* **22**, 317–330.
 Blanc, E. & Paciorek, W. (2001). *J. Appl. Cryst.* **34**, 480–483.
 Brown, A., Long, F., Nicholls, R. A., Toots, J., Emsley, P. & Murshudov, G. (2015). *Acta Cryst.* **D71**, 136–153.
 Dennis, J. E. Jr & Welsch, R. E. (1987). *Commun. Stat. Simul. Comput.* **7**, 345–359.
 Echols, N., Headd, J. J., Afonine, P. & Adams, P. D. (2010). *Comput. Crystallogr. Newsl.* **1**, 12–17.
 Grosse-Kunstleve, R. W. & Adams, P. D. (2004). *IUCr Comput. Commission. Newsl.* **4**, 19–36.
 Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams, P. D. (2002). *J. Appl. Cryst.* **35**, 126–136.
 Headd, J. J., Echols, N., Afonine, P. V., Grosse-Kunstleve, R. W., Chen, V. B., Moriarty, N. W., Richardson, D. C., Richardson, J. S. & Adams, P. D. (2012). *Acta Cryst.* **D68**, 381–390.
 Headd, J. J., Echols, N., Afonine, P. V., Moriarty, N. W., Gildea, R. J. & Adams, P. D. (2014). *Acta Cryst.* **D70**, 1346–1356.
 Jack, A. & Levitt, M. (1978). *Acta Cryst.* **A34**, 931–935.
 Kim, K. M., Nesterov, Yu. E. & Cherkassky, B. V. (1984). *Dokl. Acad. Nauk. SSSR*, **275**, 1306–1309.
 Konnert, J. H. (1976). *Acta Cryst.* **A32**, 614–617.
 Lial, M. L., Hornsby, J., Schneider, D. I., Callie, J. & Daniels (2008). *College Algebra and Trigonometry*, 4th ed., p.792. Boston: Pearson/Addison Wesley.
 Liu, H., Hexemer, A. & Zwart, P. H. (2012). *J. Appl. Cryst.* **45**, 587–593.
 Lunin, V. Yu. & Urzhumtsev, A. G. (1985). *Acta Cryst.* **A41**, 327–333.
 Murshudov, G. N., Skubák, P., Lebedev, A. A., Pannu, N. S., Steiner, R. A., Nicholls, R. A., Winn, M. D., Long, F. & Vagin, A. A. (2011). *Acta Cryst.* **D67**, 355–367.
 Parkhurst, J. M., Brewster, A. S., Fuentes-Montero, L., Waterman, D. G., Hattne, J., Ashton, A. W., Echols, N., Evans, G., Sauter, N. K. & Winter, G. (2014). *J. Appl. Cryst.* **47**, 1459–1465.
 Sauter, N. K., Hattne, J., Grosse-Kunstleve, R. W. & Echols, N. (2013). *Acta Cryst.* **D69**, 1274–1282.
 Smart, O. S., Womack, T. O., Flensburg, C., Keller, P., Paciorek, W., Sharff, A., Vornhein, C. & Bricogne, G. (2012). *Acta Cryst.* **D68**, 368–380.
 Urzhumtsev, A. G. (1991). *Acta Cryst.* **A47**, 723–727.