Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Received 9 April 2013
Accepted 29 July 2013

Iterative projection algorithms in protein crystallography. I. Theory

Rick P. Millane* and Victor L. Lo

Computational Imaging Group, Department of Electrical and Computer Engineering, University of Canterbury, Christchurch, New Zealand. Correspondence e-mail: rick.millane@canterbury.ac.nz

A general class of iterative projection algorithms is described and proposed as a tool for phasing in protein crystallography in order to improve the radius of convergence over that of conventional density-modification algorithms. Their relationship to conventional density modification is described. The common iterative projection algorithms, their convergence properties and their application to protein crystallography are described. These algorithms offer the possibility of protein structure determination starting with only information on the molecular envelope and low-order non-crystallographic symmetry.

1. Introduction

Despite the enormous advances in protein crystallography, determination of the structures of large and complex macromolecules from crystal X-ray diffraction data can sometimes be problematic as a result of the experimental difficulties of obtaining sufficiently accurate initial phases. Initial phase information is typically obtained using molecular replacement, isomorphous replacement, single- or multiple-wavelength anomalous dispersion, electron microscopy, or a combination of these (Drenth, 1999). These techniques require either a solved structure that is sufficiently homologous to the target molecule, preparation of isomorphous heavy-atom derivatives that diffract to sufficient resolution, or incorporation of anomalous scatterers into the native protein and collection of sufficiently accurate anomalous-dispersion signals. The initial phase information is sometimes obtained only at low resolution and phase extension may be required to obtain a high-resolution map suitable for model building. Although these techniques are often successful, sometimes the experimental phases may be of insufficient quality to ultimately produce an interpretable high-resolution map. Therefore, despite the power of modern methods for macromolecular crystallography, numerical algorithms that are able to converge to the correct electron density with less, or no, experimental phase information would be useful.

Quite early on, using structural redundancy for phase determination by what has evolved into modern density-modification algorithms was identified as the application of successive projections (Crowther, 1969; Bricogne, 1974). The objective of this paper is to introduce a wider class of iterative projection algorithms that have better global convergence, and describe their potential application in protein crystallography. Application to the determination of protein structures will be described in a subsequent paper.

The paper is structured as follows. The important question of uniqueness of the solution in the absence of phase information is addressed in §2. In §3 some background to the approach we propose is described in the context of current methods of density modification. The concepts of constraint sets and projections are described in §4. In §5 a number of the more effective iterative projection algorithms and their properties are described. A summary and possible implications of this work are discussed in the final section.

2. Uniqueness

It is important when contemplating structure determination with minimal initial phase information to first consider whether the given data and constraints are sufficient to provide a unique solution. If they do not, then there is little point in pursuing algorithms for structure determination since an effective algorithm may find one of a multitude of incorrect solutions. We consider here the effect of a known molecular envelope (solvent boundary) and non-crystallographic symmetry (NCS) on uniqueness of the phase problem in protein crystallography.

Crowther (1969) and Bricogne (1974) considered this question and showed that there is redundancy in the structure amplitude data that is determined by the ratio of the number of observations and the number of free parameters in the subunit from which the electron density is built. Millane (1993) studied uniqueness properties for macromolecular crystallography that considered the shape of the support region (molecular envelope) and the order of the NCS, and gave a parameter that could be related to uniqueness of the solution (although interpretation of this factor was out by a factor two). The results of Miao *et al.* (1998) indicated that, at least for rectangular supports, the Fourier amplitudes at the Bragg density underdetermines the phase problem by a factor of two. Elser & Millane (2008) considered uniqueness of the phase problem for continuous diffraction data from isolated molecules. They defined the ‘constraint ratio’, denoted Ω, as the ratio of the number of independent data (diffraction

doi:10.1107/S0108767313015249

amplitudes) divided by the number of independent object (electron density) parameters. Note that although the diffraction is continuous in this case, as a result of the sampling theorem it contains only a finite number of *independent* data. They also showed that the number of independent data depends on the size (volume) of the autocorrelation of the molecular support region. The constraint ratio is then given by

$$\Omega = |A|\,/\,2|U|, \tag{1}$$

where $|U|$ is the volume of the object (molecule) support U (*i.e.* the region occupied by the molecule) and $|A|$ is the volume of the autocorrelation support A of U. The constraint ratio therefore depends only on the shape of the molecular support region (from which $|U|$ and $|A|$ can be calculated). A unique solution requires $\Omega > 1$. In practice, an additional margin will be required to account for noise in the data. In three dimensions, for support regions that are convex and centrosymmetric (approximately true for most macromolecular envelopes), $\Omega = 4$. The problem is therefore highly overconstrained for continuous (non-crystalline) diffraction data. This analysis can be adapted to the case of crystal diffraction data as follows.

For the crystalline case, the object is periodic and a similar analysis needs to consider the support of the molecules forming the whole crystal and its autocorrelation, *i.e.* the Patterson function. Because of overlap of the autocorrelations in the Patterson function, the support $|A|$ in equation (1) is replaced by V, the volume of the unit cell. Furthermore, the number of independent diffraction data and the number of electron-density parameters are both reduced by a factor equal to the order of the space group. For crystal diffraction data then, equation (1) reduces to $\Omega = 1/(2f)$, where $f = |U|/V$ is the fraction of the unit cell occupied by the molecular envelope(s). Therefore, unless the unit cell contains more than 50% solvent (*i.e.* $f < 1/2$), $\Omega < 1$, and the phase problem for crystal diffraction data is highly non-unique. If the molecule has R-fold NCS then the number of object parameters is reduced by a factor R and

$$\Omega = R\,/\,2f. \tag{2}$$

One can come to the same conclusion by considering the equations of Crowther (1969) and Bricogne (1974). The problem is more constrained in the presence of other real-space information such as the characteristics of typical macromolecular electron densities. The difficulty is incorporating such information into a solution to the problem, although the use of pattern matching in statistical density modification is an effective approach (Terwilliger, 2003).

In theory we require only that $\Omega > 1$ for uniqueness, but in practice the effects of noise, missing data and other uncertainties will require a somewhat larger value. The recent results of Liu *et al.* (2012) for phasing based on the molecular envelope only give some guidance as to the values of Ω that might be required in practice (although of course this will vary significantly depending on various factors such as the resolution and errors in the data in particular cases). Their results indicate that, at least for reasonably high-resolution (2 Å) data, a solvent content greater than about 65% (*i.e.* $f < 0.35$) is necessary to obtain a good solution. This indicates, using equation (2), that at least $\Omega > 1.5$ is probably required in practice. Therefore, we estimate that NCS of order $R > 3f$ is required in practice to ensure uniqueness in the absence of any phase information. This corresponds to at least fourfold NCS for low solvent content crystals, and at least threefold NCS for crystals with solvent content less than about 30%. In summary then, rather modest NCS should be sufficient in principle to ensure a unique solution to the macromolecular crystallographic phase problem in the absence of any initial phase information.

3. Context

Here we describe the key concepts of iterative projection algorithms in the context of current electron-density modification algorithms. It is instructive to first briefly review electron density modification algorithms in protein crystallography. In 'classical density modification', these involve, in summary, calculating an electron-density map using experimentally determined phases, imposing structural constraints such as solvent boundaries, electron-density histograms and NCS, transforming the modified map to obtain new phase estimates, and combining these with the experimental phases which are then used to calculate a new map, and the cycle repeated until convergence (Zhang *et al.*, 2006; Cowtan, 2010). In practice, phase distributions are estimated and centroid phases are used to calculate maps, and a new phase distribution is calculated from the modified map and combined with the experimental phase distribution (by multiplication, assuming that the two sources of phase information are independent) to obtain a new phase distribution from which the centroid phase is calculated for the next iteration. Maps may be calculated using $2mF_o - DF_c$ coefficients based on an error model (Main, 1979; Read, 1986), or a γ-correction applied to reduce bias (Abrahams, 1997).

Modern macromolecular phasing uses a probabilistic setting of density modification referred to as 'statistical density modification', which has its roots in maximum-likelihood approaches to phasing (Bricogne, 1984, 1988; Lunin, 1993; Xiang *et al.*, 1993). Statistical density modification involves calculating a likelihood function for the structure factors that involves two likelihood functions, one for the diffraction data and one based on characteristics of an expected macromolecular electron-density map (Terwilliger, 1999, 2000). Probability density functions for the electron density are developed that incorporate known characteristics of macromolecular densities (*e.g.* flat solvent regions, non-crystallographic symmetry, patterns related to likely secondary structures *etc.*). Derivatives of the likelihood with respect to the structure factors are calculated and a steepest-ascent method is used to optimize the total likelihood. This gives a new probability distribution for each phase, from which a centroid electron-density map is calculated and used in the next cycle. Statistical density modification can be applied if no experimental phase information is available by using the map likelihood term alone, so-called map-likelihood phasing (Terwilliger, 2001). This is a potentially powerful approach, but it still depends on there being sufficient real-space information for the solution to be found by a gradient optimization method, which can still present difficulties. Molecular replacement phases, for example, can be used to provide only the initial map, and then subsequent cycles performed using only the map likelihood, and not the initial phases. This so-called prime-and-switch method (Terwilliger, 2001) reduces model bias, but reasonably accurate initial phases are still required for convergence to the correct solution. Although statistical density modification is one of the most effective methods for macromolecular phasing, it is still generally most successful when some experimental phase information (*e.g.* from anomalous dispersion or molecular replacement) is available. The iterative projection algorithms described in this paper are discussed in the context of classical density modification, and their potential for supplementing statistical density modification are discussed in §6.

We note for completeness the 'holographic method' for phase refinement (Béran & Szöke, 1995; Szöke *et al.*, 1997). In this scheme, the electron density is expanded in Gaussian basis functions that represent it at the resolution of the diffraction data. Real-space (solvent region, partial structures *etc.*) and reciprocal-space (diffraction data, isomorphous replacement data *etc.*) information is incorporated and the phase problem reduces to solving an optimization problem for the basis function coefficients, which is solved by simulated annealing. This approach can potentially be used to determine the electron density with minimal experimental phase information if sufficient real-space constraints (*e.g.* greater than 50% solvent content) are available. This approach is distinct from the usual density-modification approach, however, and is not discussed further here.

Consider now the application of classical density modification in the case where there are no experimental phases available. Since there are no experimental phases there is no phase combination step and the essential steps are to adjust the current map to satisfy the real-space constraints, transform to reciprocal space, set the structure amplitudes to their measured values, transform to real space, and iterate until the map does not change. The other steps outlined above are attempts to reduce model bias (to the modified map). However, experience shows that this is not effective with poor initial phases. The procedure is essentially a local minimization, whereas what is required is a global search procedure that avoids trapping at local minima. The problem can be treated purely as a constraint satisfaction problem; the objective being to find an electron density that satisfies the real-space and reciprocal-space constraints, and not be trapped at local minima where not all of the constraints are satisfied. In this setting, the real-space and reciprocal-space constraints are treated on an equal footing. One (diffraction amplitudes) is not treated as 'data' and the other (real-space constraints) as a 'constraint'. Model bias does not arise since 'bias' towards the model would be no different to 'bias' towards the structure-factor amplitudes.

In classical density modification without bias correction, the true density is sought by alternately adjusting it to satisfy each of the constraints. Bricogne (1974) showed these adjustments are *projection* operations (this will be formalized below) in a vector space representing the electron density. If we write the samples of the electron density as a vector $\mathbf{x}$ (also formalized below), then one iteration, or cycle, of conventional density modification can be written as the update rule

$$\mathbf{x}_{n+1} = P_A P_B \, \mathbf{x}_n, \tag{3}$$

where $\mathbf{x}_n$ denotes the electron density at iteration n, P_A denotes the real-space projection operation and $P_A\mathbf{x}$ means 'adjust $\mathbf{x}$ to satisfy the real-space constraints', and P_B is the reciprocal-space projection operator and $P_B\mathbf{x}$ means 'adjust $\mathbf{x}$ to satisfy the reciprocal-space constraints'. Note that the latter means 'take the Fourier transform of the electron density, adjust the transform such that its amplitudes are equal to the measured amplitudes, and transform back to real space'. Equation (3) corresponds to the 'method of successive projections' described by Bricogne (1974).

Now equation (3) is in a sense the 'obvious' update rule. What could be more effective, or simpler, than modifying the current estimate of the electron density at each step to conform to the known constraints? If the estimate $\mathbf{x}_n$ is close to the correct solution (*e.g.* if one had good initial phase estimates) then this update is likely to be effective. However, if there are no experimental phase estimates, it is likely that one will start with an estimate $\mathbf{x}_0$ that is far from the correct solution. It is then possible (in fact very likely) that at some iteration the projection P_A will 'undo' the projection P_B, *i.e.* $P_A = P_B^{-1}$, or P_A is the inverse of P_B. In that case, equation (3) becomes

$$\mathbf{x}_{n+1} = P_A P_B \, \mathbf{x}_n = \mathbf{x}_n, \tag{4}$$

and the algorithm 'stagnates', or becomes stuck, at the density $\mathbf{x}_n$. A density $\mathbf{x}_n$ that satisfies equation (4) is called a *fixed point* of the algorithm. An important point with the update equation (3) is that $\mathbf{x}_n$ can be a fixed point of the algorithm without being a solution to the problem, *i.e.* without satisfying both the real- and reciprocal-space constraints, *i.e.* equation (4) does not require that $\mathbf{x}_n = P_A \, \mathbf{x}_n = P_B \, \mathbf{x}_n$. This is the familiar problem with density modification when starting with poor initial phases, of local convergence to an incorrect density. Equation (3) is a simple example of an iterative projection algorithm. However, there is a larger class of more sophisticated iterative projection algorithms that are effective global search procedures for solving large constraint satisfaction problems and are therefore potentially useful in solving the problem at hand.

Crowther (1969) first described an iterative method, implemented wholly in reciprocal space, involving projections to enforce real-space redundancy and the observed structure amplitudes. Bricogne (1974) described an algorithm implemented in real space, and identified the key steps as projections, that is the forerunner of modern density-modification algorithms. It was pointed out by Millane (1990) that density modification corresponds to the error-reduction algorithm in the image reconstruction literature, that this algorithm has poor global convergence properties, and that the application of improved projection algorithms such as the hybrid input–output algorithm (Fienup, 1982) would be likely to be beneficial. Millane & Stroud (1997) and van der Plas & Millane (2000) extended the hybrid input–output algorithm to incorporate NCS constraints and used it to successfully reconstruct an icosahedral virus with fivefold NCS at 8 Å resolution using synthetic data with no initial phase information. The introduction of solvent flipping or γ-correction (Abrahams, 1997) (which is shown below to be a form of iterative projection algorithm) was a significant step in density modification that increases the radius of convergence. The hybrid input–output algorithm was applied to phasing X-ray data from single particles (Miao *et al.*, 1999; Chapman *et al.*, 2006). A projection algorithm called the 'difference map algorithm' (Elser, 2003*a*) has been applied *ab initio* to small-molecule crystallography (Elser, 2003*b*), single-particle imaging (Thibault *et al.*, 2006), determination of molecular envelopes from solvent contrast variation data (Lo *et al.*, 2009), and virus crystallography (Lo & Millane, 2010). Most recently, Liu *et al.* (2012) applied the hybrid input–output algorithm to a number of proteins with high (>65%) solvent content. However, there has been little systematic exploration of the potential of these algorithms in protein crystallography.

4. Constraint sets and projections

In this section we introduce some mathematical formalisms to describe constraints and projections. This formalism is kept to a minimum but is necessary in order to describe iterative projection algorithms in a clear and concise way.

Consider an electron density that is sampled at N grid points in the unit cell. The electron density is then represented by N real numbers that are denoted x_j, for $j = 1, 2, \ldots, N$. These numbers are collected together in a vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$. This vector sits in an N-dimensional Euclidean space, or vector space, denoted $\mathbb{R}^N$. A particular electron density is then represented by a particular vector that corresponds to a single point in this space. The whole space (all points or all vectors) represents all possible electron-density functions. The set of all electron densities that satisfy some constraint therefore corresponds to a set of points in $\mathbb{R}^N$, or a subset of $\mathbb{R}^N$, that is called the constraint set.

As will be seen later, a special place is taken by constraint sets that are *convex*. A convex set is a set for which the line segment joining any two points in the set is wholly within the set, as illustrated in Fig. 1. Formally, a constraint set A is convex if, for any two points $\mathbf{x}_1, \mathbf{x}_2$ in A, any point $\mathbf{x}' = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ for which $0 \leq \alpha \leq 1$, is also in A. A set that is not convex is called a non-convex set (Fig. 1). Since constraints are defined as sets, we refer to convex and non-convex constraints.

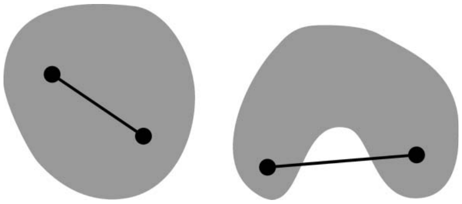

Figure 1
Convex (left) and non-convex (right) constraint sets.

We now define a projection. The projection of a point $\mathbf{x}$ onto a constraint set A, denoted $\mathbf{y} = P_A\mathbf{x}$, is the point $\mathbf{y} \in A$ that is closest to $\mathbf{x}$ (in terms of Euclidean distance). A projection can therefore be defined formally as

$$\mathbf{y} = P_A\mathbf{x} = \underset{\mathbf{y}\in A}{\operatorname{argmin}}\, ||\mathbf{y}-\mathbf{x}||, \tag{5}$$

where $||\mathbf{x}||$ is the Euclidean norm (or length) of $\mathbf{x}$, *i.e.* $||\mathbf{x}|| = (\sum_i x_i^2)^{1/2}$, and $\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$ denotes the value of $\mathbf{x}$ that minimizes $f(\mathbf{x})$. Therefore, the projection of an electron density onto a constraint set involves making the smallest change to the density such that it satisfies the constraint.

The real-space and reciprocal-space constraint sets are denoted here by A and B, respectively, so that projections onto the real-space and reciprocal-space constraints are denoted $P_A\mathbf{x}$ and $P_B\mathbf{x}$, respectively. Iterative projection algorithms are generally formulated with two constraint sets, thus the separation into real-space and reciprocal-space constraints.

In the following subsections we define some of the simple constraint sets and projections in protein crystallography. Some of these will be familiar but are worthwhile recalling in terms of constraints and projections. We emphasize that any other constraint and its associated projection can be similarly defined.

4.1. Real-space constraints

The real-space constraint A is the set of all electron densities that satisfy the given real-space constraints. These constraints might include solvent boundaries, NCS, histograms, known fragments or any other structural constraint. Here we consider only solvent boundary and NCS constraints.

4.1.1. Solvent flatness constraint. The solvent flatness, or support, constraint refers to knowledge of the molecular envelope. Depending on the problem at hand, regions outside the molecular envelope are known to contain solvent or other disordered material and hence the electron density in these regions is equal to a fixed constant value, denoted here by σ. The support constraint set, denoted A_1, is then the set of all electron densities that have this constant value outside the support region, *i.e.*

$$A_1 = \{\mathbf{x} : x_j = \sigma, \quad \forall\, j \notin U\}, \tag{6}$$

where, as previously, U denotes the set of grid points inside the molecular envelope. Referring to (6), the coordinates $j \notin U$ in $\mathbb{R}^N$ have a fixed value and the other coordinates can take on any value, so that the set A_1 is a $|U|$-dimensional hyperplane in $\mathbb{R}^N$, where $|U|$ is the number of elements in U. Since a hyperplane is a convex set, the support constraint is a convex constraint.

It is easily seen that the minimum change that can be made to an electron density to satisfy this constraint is to set it equal to σ at the grid points outside the envelope and to leave it unchanged at grid points inside the envelope. The projection is therefore defined by (Bricogne, 1974; Elser, 2003*a*)

$$P_{A_1}x_j = \begin{cases} x_j & \text{if } j \in U, \\ \sigma & \text{if } j \notin U. \end{cases} \tag{7}$$

If the value σ is not known, then the minimum change results if σ is calculated as

$$\sigma = \frac{1}{N-|U|} \sum_{j \notin U} x_j, \tag{8}$$

i.e. the average density outside the envelope.

4.1.2. NCS constraint. The NCS constraint expresses the fact that the electron density is equal at symmetry-related points. Since symmetry is a geometric concept, it is convenient to first consider it in real-space coordinates, denoted ξ, and then transfer to the vector space $\mathbb{R}^N$. Consider first the case of a continuum (*i.e.* not sampled) electron density $\rho(\xi)$ in the unit cell. Consider a general NCS of order M that is defined in the region T (T would usually correspond to the molecular envelope) by the M symmetery operators S_m for $m = 1, \ldots, M$, that map the point ξ to the M points $S_m\xi$, with S_1 being the identity operator. The NCS constraint set, denoted A_2, is then the set of all electron densities $\rho(\xi)$ that satisfy

$$\rho(\xi) = \rho(S_m\xi), \quad \forall\, m \in \{1, \ldots, M\},\ \forall\, \xi \in \mathrm{T}. \tag{9}$$

Consider two densities $\rho_1(\xi)$ and $\rho_2(\xi)$ and the density $\rho_3(\xi)$ given by

$$\rho_3(\xi) = \alpha\rho_1(\xi) + (1-\alpha)\rho_2(\xi), \tag{10}$$

with $0 < \alpha < 1$. It is easy to show that if $\rho_1(\xi)$ and $\rho_2(\xi)$ each satisfy equation (9), then $\rho_3(\xi)$ also satisfies (9). Therefore, for a continuum electron density, by the definition of a convex set, the NCS constraint is a convex constraint.

In the case at hand, however, the electron density is sampled. The difficulty here is that if ξ is a grid point in the unit cell then $S_m\xi$ will not generally be a grid point so that $\rho(S_m\xi)$ is not defined and (9) cannot be used as a definition of the NCS constraint. In view of this difficulty, we define a sampled electron density as satisfying the NCS constraint if the difference between the value at a grid point and the value at any symmetry-related position, calculated by interpolation from the values at neighbouring grid points, does not exceed an upper bound denoted ε. Such densities then satisfy

$$|\rho(\xi) - \rho'(S_m\xi)| < \varepsilon, \quad \forall\, m \in \{1, \ldots, M\},\ \forall\, \xi \in \mathrm{T}, \tag{11}$$

where $\rho'(S_m\xi)$ denotes the value calculated by interpolation from a set of grid points in the neighbourhood of $S_m\xi$, and ε is a small parameter. This definition depends, of course, on the interpolation scheme used and on ε. Consider an interpolation scheme which is a linear combination of the values at a set of neighbouring grid points, *i.e.*

$$\rho'(S_m\xi) = \sum_k \beta_{mk}\, \rho(\xi_{mk}), \tag{12}$$

where the ξ_{mk} are the grid points in the neighbourhood of $S_m\xi$, indexed by k, and β_{mk} are constants (that will generally depend on $S_m\xi$ and ξ_{mk}). Equation (11) then becomes

$$\left|\rho(\xi) - \sum_k \beta_{mk}\, \rho(\xi_{mk})\right| < \varepsilon, \quad \forall\, m \in \{1, \ldots, M\},\ \forall\, \xi \in \mathrm{T}. \tag{13}$$

Now consider the density $\rho_3(\xi)$ given by (10), where $\rho_1(\xi)$ and $\rho_2(\xi)$ now both satisfy (13). Substitution shows that $\rho_3(\xi)$ also satisfies (13), so the NCS constraint for sampled densities defined by (13) is a convex constraint.

Transforming now to the vector-space formalism, the NCS constraint set A_2 for sampled densities is defined by

$$A_2 = \{\mathbf{x} : |x_j - x'_{mj}| < \varepsilon, \quad \forall\, m,\ \forall\, j \in T\}, \tag{14}$$

for some ε, where x'_{mj} denote the values of the interpolated electron density at the M points symmetry-related to the point j, and T indexes the grid points in T. From the above, A_2 is convex. It is necessary to include the parameter ε for two reasons. First, for $\varepsilon = 0$, for a particular interpolation scheme A_2 may admit only constant densities. Second, a finite ε is needed in order to define a simple and effective projection onto A_2.

Consider now the projection P_{A_2} onto A_2. This operation makes the minimum change to the electron density such that it satisfies the NCS constraint. It is easily seen that in the continuum case this corresponds to setting the electron density at any position to the average value of the density over the symmetry-related positions (Bricogne, 1974). In the sampled case, however, this cannot be done directly since the symmetry-related positions are not grid points, and the density values at each grid point need to be adjusted such that the difference between the new values and the interpolated values at symmetry-related points does not exceed ε. However, if the density at each grid point is set to the average of the interpolated values at the symmetry-related points, then the resulting density satisfies equation (14) with the minimum possible value of ε obtainable in a single step. This is therefore the optimum choice and the projection is defined by

$$P_{A_2}x_j = \begin{cases} (1/M)\sum\limits_{m=1}^{M} x'_{mj} & \text{for } j \in T, \\ x_j & \text{for } j \notin T. \end{cases} \tag{15}$$

This is seen to correspond to the usual symmetry-averaging operation applied in conventional density modification. Using this projection, the value of ε does not need to be considered as it is an automatic outcome related to the grid spacing, resolution and interpolation scheme used.

4.1.3. Combining real-space constraints and projections. Application of an iterative projection algorithm requires that a single real-space constraint A, and a single projection P_A, be derived from the individual constraints, A_1 and A_2 in this case, and the individual projections P_{A_1} and P_{A_2}. An electron density that satisfies both the constraints A_1 and A_2 must lie in their intersection, *i.e.* in the set $A = A_1 \cap A_2$. In the case at hand, A_1 and A_2 are both convex so that their intersection is convex, and thus the full real-space constraint A is convex. Also in the case at hand, the operation P_{A_1} changes the values x_j only for $j \notin U$, and P_{A_2} changes the values x_j only for $j \in T$, and $U = T$, *i.e.* P_{A_1} and P_{A_2} act on disjoint subsets of $\mathbb{R}^N$. It is then easily seen that projection onto A is identical to sequential projections onto A_1 and A_2. Therefore, the full real-space projection P_A is obtained by composition, *i.e.*

$$P_A\mathbf{x} = P_{A_2}P_{A_1}\mathbf{x}. \tag{16}$$

For other kinds of constraints, it is likely that the individual projections cannot be rigorously combined into a single projection. In such cases a pragmatic approach may need to be taken, which may involve simply using the composition of the individual projections without concern for the resulting operation being a projection.

4.2. Reciprocal-space constraint

The reciprocal-space (Fourier-amplitude) constraint set B is the set of all electron densities whose structure-factor amplitudes are equal to the measured amplitudes, *i.e.*

$$B = \{\mathbf{x} : |\mathcal{F}[\mathbf{x}]| = \mathbf{M}\}, \tag{17}$$

where $\mathcal{F}[\cdots]$ denotes the Fourier transform and $\mathbf{M}$ is the vector of the measured structure-factor amplitudes. If we let $\tilde{B}$ denote the set of all structure factors whose amplitudes are equal to the measured amplitudes, then the set B can also be defined as the set of all electron densities that are the inverse Fourier transform of the structure factors in $\tilde{B}$, *i.e.* B can be defined as

$$B = \{\mathbf{x} : \mathbf{x} = \mathcal{F}^{-1}[\mathbf{y}], \quad \forall\ \mathbf{y} \in \tilde{B}\}, \tag{18}$$

where $\mathcal{F}^{-1}[\cdots]$ denotes the inverse Fourier transform. At a particular reciprocal-lattice point, the set $\tilde{B}$ of all structure factors with a given amplitude lies on a circle in the complex plane (Fig. 2). Since the line segment joining any two points on a circle is not on the circle, the set $\tilde{B}$ at a single reciprocal-lattice point is non-convex, and, by the distance-preserving property of the Fourier transform, the set B at a single reciprocal-lattice point is also non-convex. The constraint set $\tilde{B}$ for all reciprocal-lattice points is the intersection of a set of hypercylinders, which is non-convex, and so the full constraint set B is non-convex. The Fourier-amplitude constraint is therefore a non-convex constraint. This is important as we shall see in §5.

The reciprocal-space or Fourier-amplitude projection involves making the minimum change to the electron density such that it is consistent with the structure-factor amplitude data. Because the Fourier transform is distance preserving in $\mathbb{R}^N$, the projection operation can conveniently be applied in reciprocal space, *i.e.* the projection can be written in the form

$$P_B\mathbf{x} = \mathcal{F}^{-1}[P_{\tilde{B}}\mathcal{F}[\mathbf{x}]]. \tag{19}$$

The projection $P_{\tilde{B}}$ denotes projection of the structure factors of $\mathbf{x}$ onto $\tilde{B}$ (*i.e.* moving a point in the complex plane in Fig. 2 to the closest point on the circle), and it is easily seen that this involves setting the Fourier amplitudes to their measured values and leaving the phases unchanged, *i.e.*

$$P_{\tilde{B}}X_{\mathbf{h}} = \begin{cases} sM_{\mathbf{h}}\exp(i\varphi[X_{\mathbf{h}}]) & \text{if } \mathbf{h} \in Q, \\ X_{\mathbf{h}} & \text{if } \mathbf{h} \notin Q, \end{cases} \tag{20}$$

where $X_{\mathbf{h}}$ denotes the structure factor at reciprocal-lattice vector $\mathbf{h}$, *i.e.* $\mathcal{F}[\mathbf{x}] = (X_{\mathbf{h}_1}, X_{\mathbf{h}_2}, \ldots)$, s denotes the scale factor between the measured and calculated structure-factor amplitudes, $\varphi[\cdots]$ denotes the phase, $M_{\mathbf{h}}$ denotes the measured structure-factor amplitudes, and Q denotes the set of reciprocal-lattice points where the data are measured (*i.e.* between the minimum and maximum resolutions and excluding any missing data). Note that in (20), by the notation P_Ax_j we mean $P_A\mathbf{x} = (P_Ax_1, P_Ax_2, \ldots, P_Ax_N)$. Equations (19) and (20) therefore together define the projection P_B, which is seen to correspond to the simplest reciprocal-space step in conventional density modification.

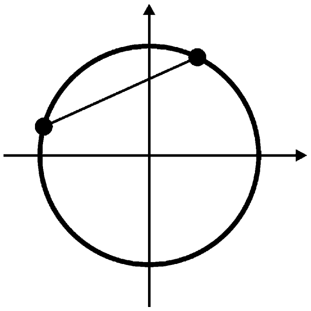

Figure 2
The circle shows the Fourier-amplitude constraint set $\tilde{B}$ for a single structure factor in the complex plane. The constraint set is non-convex.

5. Iterative projection algorithms

With the above background and definitions, we are now in a position to look at iterative projection algorithms. An iterative projection algorithm is an algorithm for finding a point in the intersection of constraint sets in $\mathbb{R}^N$. We consider the case where there are two constraint sets A and B. The algorithm generates a sequence of points in $\mathbb{R}^N$, denoted $\mathbf{x}_n$, beginning with an, often random, point $\mathbf{x}_0$. At each iteration, $\mathbf{x}_n$ is updated, using an update rule, to produce $\mathbf{x}_{n+1}$. The point $\mathbf{x}_n$ is referred to as the 'iterate'. It is important to note that, although the iterate is used to find the solution to the problem (a point in $A \cap B$), it is not usually a solution itself. The update rule is a combination of projections P_A and P_B applied to $\mathbf{x}_n$, along with $\mathbf{x}_n$ itself. A particular iterative projection algorithm is defined by its update rule. In the following subsections, the more popular and effective iterative projection algorithms are described as well as their relationships to conventional density modification.

5.1. Error-reduction algorithm

The simplest iterative projection algorithm is that in which P_A and P_B are sequentially applied to $\mathbf{x}_n$, *i.e.* as described in §3 and repeated here,

$$\mathbf{x}_{n+1} = P_AP_B\mathbf{x}_n. \tag{21}$$

This algorithm alternately adjusts $\mathbf{x}_n$ to conform to constraints A and B. The problem, of course, is that in adjusting the iterate to conform to the constraint A it is likely to no longer conform to constraint B. As described in §3, this algorithm corresponds to conventional electron-density modification, and the problem is that it can converge to a density that does not satisfy both the real-space and reciprocal-space constraints. The algorithm in (21) is commonly referred to as the 'error-reduction' (ER) algorithm in the image-reconstruction literature (Fienup, 1982). An important property of the ER algorithm is that it converges (although possibly slowly) to a point in $A \cap B$ if both the constraint sets A and B are convex. In this case the ER algorithm is also known as the 'projection onto convex sets' (POCS) algorithm. However, if one of the constraint sets is non-convex, as is the case at hand since the reciprocal-space constraint B is non-convex, then the algorithm will usually stagnate at a non-solution unless it is started close to the true solution. This is the key difficulty with this algorithm that makes it unsuitable in cases where there is little initial phase information available.

5.2. Relaxed-projection algorithm

The next iterative projection algorithm we describe uses 'relaxed projections'. Unfortunately, two different conventions are used to describe a relaxed projection. The convention we use here defines a relaxed projection, denoted $F_A(\gamma)$, onto the set A, with a constant parameter γ called the relaxation parameter, by

$$F_A(\gamma)\,\mathbf{x} = P_A\mathbf{x} + \gamma(P_A\mathbf{x} - \mathbf{x}). \qquad (22)$$

Inspection of (22) shows that the relaxed projection is equivalent to taking the regular projection and adding an additional change to $\mathbf{x}$ that is the difference between $P_A\mathbf{x}$ and $\mathbf{x}$, scaled by γ. This is illustrated in Fig. 3. For $\gamma = 0$ the relaxed projection is equivalent to the regular projection, *i.e.* $F_A(0)\mathbf{x} = P_A\mathbf{x}$, for $\gamma < 0$ the relaxed projection 'underprojects' before $P_A\mathbf{x}$, and for $\gamma > 0$ it 'overprojects' beyond $P_A\mathbf{x}$. For $\gamma = -1$, $\mathbf{x}$ remains unchanged, *i.e.* $F_A(-1)\mathbf{x} = \mathbf{x}$. For $\gamma = 1$, $F_A(1)\mathbf{x}$ projects twice as far as $P_A\mathbf{x}$ (Fig. 3) and the operator is called a 'reflector' and is denoted R_A, *i.e.*

$$R_A\mathbf{x} = F_A(1)\mathbf{x} = 2P_A\mathbf{x} - \mathbf{x}. \qquad (23)$$

The alternative convention for the relaxed projection is identical except that the relaxation parameter is replaced by $\lambda = 1 + \gamma$, and (22) becomes

$$F_A(\lambda)\mathbf{x} = \mathbf{x} + \lambda(P_A\mathbf{x} - \mathbf{x}). \qquad (24)$$

The cases $\gamma = -1, 0, 1$ are equivalent to $\lambda = 0, 1, 2$. We use the former definition (using γ) in this paper.

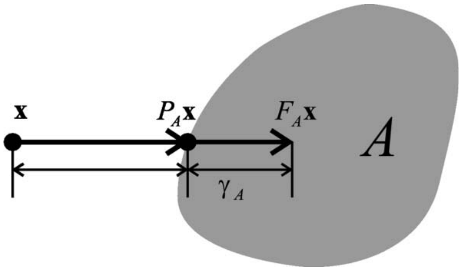

Figure 3
The relaxed projection $F_A\mathbf{x}$.

The relaxed-projection algorithm replaces the projections in the ER algorithm [equation (21)] by relaxed projections, *i.e.* the update rule for the relaxed-projection algorithm is

$$\mathbf{x}_{n+1} = F_A(\gamma_A)\,F_B(\gamma_B)\mathbf{x}_n, \qquad (25)$$

where γ_A and γ_B denote the relaxation parameters for the projections onto the sets A and B, respectively. It can be shown that for convex constraints the relaxed-projection algorithm is stable for $-1 < \gamma < 1$ and is generally unstable if γ_A or $\gamma_B < -1$, or γ_A or $\gamma_B > 1$. The relaxed-projection algorithm tends to speed up convergence relative to the ER algorithm if $0 < \gamma < 1$, particularly for convex constraints. Acceleration of convergence increases as γ approaches 1, but the algorithm can become unstable (diverge) as γ becomes close to 1. Although convergence can be improved by use of the relaxed-projection algorithm if it is started near the solution, in the case of non-convex constraints it is still prone to stagnation if not started near the solution. It is therefore potentially useful when one has good initial phase information, but not where there is minimal initial phase information.

The trade-off between convergence and stability of the relaxed projection algorithm can be illustrated as follows. Using equation (22) shows that equation (25) can be written as

$$\mathbf{x}_{n+1} = (1+\gamma_A)P_AF_B\mathbf{x}_n - \gamma_AP_B\mathbf{x}_n - \gamma_A\gamma_BP_B\mathbf{x}_n + \gamma_A\gamma_B\mathbf{x}_n. \qquad (26)$$

If now γ_A and γ_B are chosen such that $\gamma_A\gamma_B = 1$, then (26) reduces to

$$\mathbf{x}_{n+1} = (1+\gamma_A)P_AF_B\mathbf{x}_n - (1+\gamma_A)\,P_B\mathbf{x}_n + \mathbf{x}_n. \qquad (27)$$

If the algorithm reaches a fixed point, *i.e.* $\mathbf{x}_{n+1} = \mathbf{x}_n$, then (27) shows that

$$P_AF_B\mathbf{x}_n = P_B\mathbf{x}_n = \mathbf{x}^*. \qquad (28)$$

Inspection of (28) now shows an interesting property. If the algorithm reaches a fixed point $\mathbf{x}_n$, *i.e.* converges, then $\mathbf{x}^*$ satisfies both constraints A and B [since from (28) it is formed by both a projection onto A and a projection onto B], and so is a solution to the problem. This is a highly desirable property since it allows the solution to be immediately calculated once the algorithm has converged. There is a difficulty with the relaxed-projection algorithm however. The requirement $\gamma_A\gamma_B = 1$ requires that either $\gamma_A \geq 1$ or $\gamma_B \geq 1$, and in either case the algorithm is divergent. In practice one needs to use $\gamma_A, \gamma_B < 1$ and this desirable property is not attainable. However, other iterative projection algorithms have this same desirable property at their fixed points with weaker divergence problems.

We now show that the technique of γ-correction (or solvent flipping) in conventional electron-density modification (Abrahams, 1997) is a relaxed projection. (Note the unfortunate use of 'γ' in this context.) The γ-correction is an effective method for reducing model bias when applied to solvent levelling. In this case the updated value of the density in the solvent region ρ' is given by (Abrahams, 1997)

$$\rho' = \rho_{\rm solv} + k_{\rm flip}(\rho - \rho_{\rm solv}), \qquad (29)$$

where $\rho_{\rm solv}$ denotes the solvent value, ρ is the original density and $k_{\rm flip}$ is a parameter called the flipping factor. Writing (29) in our terminology gives

$$\mathbf{x}_{n+1} = P_{A_1}\mathbf{x}_n + k_{\rm flip}(\mathbf{x}_n - P_{A_1}\mathbf{x}_n), \tag{30}$$

and comparison with (22) shows that this is equivalent to a relaxed projection with relaxation parameter $\gamma_{A_1} = -k_{\rm flip}$. The flipping factor often used for solvent levelling is (Abrahams, 1997) $k_{\rm flip} = f/(f-1)$, where f is the fraction of the unit cell containing protein as defined in §2, so that

$$\gamma_{A_1} = f/(1-f). \tag{31}$$

(Note that, for even more confusion, f used here is usually denoted γ in the solvent-flipping literature!). For solvent contents between 25% and 75%, this gives values of γ_{A_1} between 3 and 0.33, respectively. The idea here is that smaller solvent contents represent a weaker constraint and therefore require more overrelaxation. Note, however, that values $\gamma_{A_1} > 1$ ($k_{\rm flip} < -1$) can cause convergence difficulties. The identification of γ-correction with relaxed projections supports experience that the former is useful in improving phases, but it does not introduce a sufficiently increased radius of convergence for the case of very little initial phase information.

An algorithm that is related to solvent flipping is that of 'charge flipping' (Oszlányi & Sütő, 2008). This algorithm was developed for small-molecule crystallography but is discussed here because of its relationship to projection algorithms and its similarity to solvent flipping. The algorithm is based on the constraint that at high resolution much of the unit cell is empty (zero electron density). The idea is to invert the electron density in regions where the current estimate is small, to encourage either small (close to zero) or large values. It is therefore similar to solvent flipping except that the electron density is flipped where the value is small (since the positions of the 'atomic envelopes' are unknown), rather than outside the molecular envelope. The algorithm is effective for small-molecule crystallography and has also been applied with success in protein crystallography if high resolution (>1 Å) data are available (Oszlányi & Sütő, 2008; Dumas & van der Lee, 2008). The real-space step at each iteration of the charge-flipping algorithm is given by

$$(\mathbf{x}_{n+1})_j = \begin{cases} (\mathbf{x}_n)_j & \text{for } (\mathbf{x}_n)_j > \delta, \\ -(\mathbf{x}_n)_j & \text{for } (\mathbf{x}_n)_j < \delta, \end{cases} \tag{32}$$

where δ is a threshold and we use the notation $(\mathbf{x})_j = x_j$. Inspection of (32) shows that the corresponding constraint set, denoted C, is

$$C = \{\mathbf{x} : x_j = 0 \quad \text{or} \quad x_j > \delta, \quad \forall\, j\}, \tag{33}$$

but that the operation (32) is not a combination of projection operators. Therefore, charge flipping is not strictly a projection algorithm. Various minor modifications can be made to the step in equation (32) to make it a combination of projections, the simplest being to replace it by

$$(\mathbf{x}_{n+1})_j = \begin{cases} (\mathbf{x}_n)_j & \text{for } (\mathbf{x}_n)_j > \delta, \\ 2\delta - (\mathbf{x}_n)_j & \text{for } \delta/2 < (\mathbf{x}_n)_j < \delta, \\ -(\mathbf{x}_n)_j & \text{for } (\mathbf{x}_n)_j < \delta/2, \end{cases} \tag{34}$$

which is equivalent to

$$\mathbf{x}_{n+1} = R_C\,\mathbf{x}_n. \tag{35}$$

This and more sophisticated options (following the ideas described in the following subsections) may be worthy of investigation. The full charge-flipping algorithm involves a number of additional steps in addition to the basic step described above, in both real space and reciprocal space (Oszlányi & Sütő, 2008). Note that an 'atomicity constraint' and the associated projection defined by Elser (2003*b*) has a similar objective to charge flipping.

As alluded to above, desirable properties of an iterative projection algorithm for reconstruction with little or no initial phase information are that it explores a large region of the parameter space $\mathbb{R}^N$ (since the starting point may be far from the solution), that it does not stagnate in the vicinity of non-solutions, and that it converges when it is in the vicinity of the true solution. A number of more sophisticated iterative projection algorithms have been developed that have these general properties. They have been used in a number of areas of image reconstruction such as in astronomy and coherent diffraction imaging, but their potential utility in protein crystallography has been little explored. There are a number of such algorithms in use, but three algorithms that have found significant application in other areas are the hybrid input–output algorithm, the difference-map algorithm, and the relaxed alternating averaged reflection algorithm. These three algorithms are outlined in the next three subsections.

5.3. Hybrid input–output algorithm

The hybrid input–output (HIO) algorithm is one of the oldest and most popular algorithms for phase retrieval, and has found wide use in image reconstruction. It was originally developed by Fienup (1982) for astronomy, in which the image is subject to support and positivity constraints. The advantage of this algorithm is that it is particularly adept at avoiding stagnation in the presence of the non-convex Fourier amplitude constraint, and with enough iterations will usually converge to the global solution. Considering here only the case of a support (solvent level) constraint in real space [equation (6) with $\sigma = 0$], the HIO algorithm update rule is

$$(\mathbf{x}_{n+1})_j = \begin{cases} (P_B\mathbf{x}_n)_j & \text{for } j \in U, \\ (\mathbf{x}_n)_j - \beta(P_B\mathbf{x}_n)_j & \text{for } j \notin U, \end{cases} \tag{36}$$

where β is a parameter. Inspection of (36) shows that for grid points inside the envelope the iterate $\mathbf{x}_n$ is left unchanged after the Fourier amplitude constraint has been applied, but outside the envelope a change is made that is different to setting the iterate to zero (*i.e.* to satisfying the constraint). It is this difference that prevents the algorithm from stagnating at a non-solution. The value of the parameter β used is variable but a value $\beta \simeq 0.7$ is often effective.

The algorithmic description [equation (36)] is useful for implementing the algorithm, but is not in the form of a single update rule [such as equation (21)]. Such a form is not straightforward to derive, but Bauschke *et al.* (2003) show that, for the case of a support constraint, equation (36) can be written as

$$\mathbf{x}_{n+1} = (1+\beta)P_A P_B \mathbf{x}_n - P_A \mathbf{x}_n - \beta P_B \mathbf{x}_n + \mathbf{x}_n. \qquad (37)$$

Equation (37) is useful for analysis of the algorithm. For example, at a fixed point of the algorithm, it shows that the iterate satisfies

$$P_A[\mathbf{x}_n - (1+\beta)P_B\mathbf{x}_n] = P_B(\beta\mathbf{x}_n) = \mathbf{x}^*. \qquad (38)$$

Inspection of (38) shows that $\mathbf{x}^*$ satisfies both constraints and so is a solution to the problem. The algorithm therefore has the desirable property mentioned above that once it converges a solution can immediately be found.

The HIO algorithm as described above is suitable only for constraints, such as support or positivity, for which the constraint value is zero. Millane & Stroud (1997) extended the idea of the HIO algorithm to accommodate more general real-space constraints. For convenience, we refer to this algorithm here as the generalized HIO (GHIO) algorithm, and (36) is replaced by (Millane & Stroud, 1997)

$$(\mathbf{x}_{n+1})_j = \begin{cases} (P_B\mathbf{x}_n)j & \text{for } (P_B\mathbf{x}_n)_j \in A, \\ (\mathbf{x}_n)_j + \beta[(P_A P_B \mathbf{x}_n)_j - (P_B\mathbf{x}_n)_j] & \text{for } (P_B\mathbf{x}_n)_j \notin A. \end{cases} \qquad (39)$$

Inspection of (39) shows that the constraint A in this case must be such that it can be evaluated on a sample-by-sample basis, independently of the values of the other samples. This means that the GHIO algorithm can be applied with only a restricted kind of real-space constraint, although this is not particularly restrictive in practice and includes, for example, the NCS constraint (Millane & Stroud, 1997). Millane & Stroud (1997) also incorporated the idea of a tolerance with which the constraints need to be satisfied into the GHIO algorithm. This is an option which removes the requirement for the constraints to be satisfied exactly. This allows softening of the constraints, similar to that with statistical density modification. This can also be easily implemented by putting a margin on the definition of the constraint sets, and be used, for example, to allow inexact satisfaction of the NCS or the diffraction amplitude data. The reader is referred to Millane & Stroud (1997) for more information. As noted above, an application of the GHIO algorithm to simulated data from a crystalline icosahedral virus, incorporating support and fivefold NCS constraints, showed considerable promise (Millane & Stroud, 1997; van der Plas & Millane, 2000).

5.4. Difference-map algorithm

The difference-map (DM) algorithm was developed by Elser (2003*a*). (Note that this unfortunate term is unrelated to the difference Fourier map or to the program *DM*.) The algorithm is designed such that the iterate is attracted to fixed points. The update rule for the DM algorithm uses projections and relaxed projections, and takes the form (Elser, 2003*a*)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \beta[P_A F_B(\gamma_B)\mathbf{x}_n - P_B F_A(\gamma_A)\mathbf{x}_n], \qquad (40)$$

where β, γ_A and γ_B are parameters of the algorithm. Convergence at the fixed points is optimized if the values

$$\gamma_A = -1/\beta \qquad \text{and} \qquad \gamma_B = 1/\beta \qquad (41)$$

are used (Elser, 2003*a*). These values are assumed here and the algorithm then has a single parameter β. Although β may need to be optimized empirically, values $\beta \simeq 0.7$ are usually effective. Note that β can be negative, which simply corresponds to interchanging the constraints A and B.

If the DM algorithm reaches a fixed point $\mathbf{x}_{n+1} = \mathbf{x}_n$, then reference to (40) shows that

$$P_A F_B \mathbf{x}_n = P_B F_A \mathbf{x}_n = \mathbf{x}^*, \qquad (42)$$

and $\mathbf{x}^*$ is a solution to the problem since it satisfies both constraints A and B. The DM algorithm therefore has the desirable property described above that the solution can be obtained immediately once the algorithm has converged to a fixed point. The DM algorithm has good search properties in the sense that if the iterate approaches a 'near-solution', *i.e.* a region of $\mathbb{R}^N$ where the sets A and B are close but do not intersect, it will subsequently move away from this region and continue to explore the parameter space. It is therefore not prone to stagnation. As noted above, the difference-map algorithm has been used in a number of crystallographic applications (Elser, 2003*b*; Lo *et al.*, 2009; Lo & Millane, 2010).

5.5. Relaxed alternating averaged reflections algorithm

The final iterative projection algorithm we describe is the relaxed alternating averaged reflections (RAAR) algorithm (Luke, 2005), which is defined by the update rule

$$\mathbf{x}_{n+1} = \beta_n(2P_A P_B \mathbf{x}_n - P_A\mathbf{x}_n - \mathbf{x}_n) + (1-2\beta_n)P_B\mathbf{x}_n. \qquad (43)$$

Note that in (43) the parameter β_n is written as a function of iteration n. Of course the parameters of any iterative projection algorithm can be changed as the iterations proceed, but this is particularly useful with the RAAR algorithm as described below. Algorithmic equations for the RAAR algorithm for the case of support and positivity constraints are given by Luke (2005). The RAAR algorithm tends to be more stable near a solution than the HIO algorithm, and good performance is obtained if β_n is started at about 0.7 and gradually increased towards (but less than) 1 as the iterations proceed (Luke, 2005). If the RAAR algorithm reaches a fixed point, then reference to (43) shows that the iterate satisfies

$$2\beta_n P_A P_B \mathbf{x}_n - \beta_n P_A \mathbf{x}_n + (1-2\beta_n)P_B\mathbf{x}_n = (1-\beta_n)\mathbf{x}_n. \qquad (44)$$

If, now, $\beta_n = 1$, then (44) reduces to

$$P_A(2P_B\mathbf{x}_n - \mathbf{x}_n) = P_B\mathbf{x}_n = \mathbf{x}^*, \qquad (45)$$

so that $\mathbf{x}^*$ is a solution (since it satisfies both constraints). Therefore, the strategy described above of moving β_n towards unity as the iterations proceed allows the solution to be calculated at convergence of the algorithm.

5.6. Error metrics

As described above, it is important to note that the iterate $\mathbf{x}_n$ of an iterative projection algorithm is usually not an estimate of the solution. One therefore has to be careful when calculating error metrics in order to monitor convergence of these algorithms. It is generally not useful to calculate an error using the iterate directly. For example, an R factor should not be calculated in the usual way by comparing the structure-factor amplitudes of the iterate $\mathbf{x}_n$ with the amplitude data, since it is likely that the iterate satisfies neither constraint. A number of options are available for error metrics. For example, an R factor can be calculated by comparing the Fourier amplitude of the iterate after it has been projected onto the real-space constraints, *i.e.* as

$$R_n = \frac{\| |\mathcal{F}[P_A \mathbf{x}_n]| - \mathbf{M} \|_1}{\|\mathbf{M}\|_1}, \tag{46}$$

where $\mathbf{M}$ is the vector of the Fourier amplitude data and $\| \cdots \|_1$ denotes the 1-norm,

$$\|\mathbf{x}\|_1 = \sum_j |x_j|. \tag{47}$$

Alternatively, one can calculate an R factor that compares the Fourier amplitude of an estimate $\mathbf{x}^*$ of the solution with the amplitude data, for example for the DM algorithm as

$$R_n = \frac{\| |\mathcal{F}[P_A F_B \mathbf{x}_n]| - \mathbf{M} \|_1}{\|\mathbf{M}\|_1}. \tag{48}$$

Another alternative is to calculate the difference between the two estimates of $\mathbf{x}^*$, which for the DM algorithm is

$$\Delta_n = \|P_A F_B \mathbf{x}_n - P_B F_A \mathbf{x}_n\|, \tag{49}$$

which would equal zero at a solution where the two estimates coincide. Any of these error metrics is effective in practice.

6. Discussion

Analysis of uniqueness properties of the macromolecular crystallographic phase problem shows that with a low-resolution envelope, the positions of any NCS axes, and rather modest NCS, protein electron densities should be uniquely defined by their diffraction data alone. Other structural information will strengthen uniqueness. An NCS constraint set is formally defined for sampled densities and is shown to be convex, and the usual electron density averaging is shown to be the projection onto this constraint set.

Classical density modification is an example of a simple iterative projection algorithm, but it has poor global convergence for non-convex constraints such as the Fourier amplitude constraint. It is therefore suitable for phase determination when started with reasonably good experimental phases, but is not effective when little or no initial phase information is available. Solvent flipping (or γ-correction) is identified as a relaxed projection that improves convergence, but does not have the global searching abilities required when little phase information is available. More sophisticated iterative projection algorithms exist that have good global convergence properties, and are a viable tool for phasing in protein crystallography where minimal phase information is available. Although only the solvent level and NCS constraints have been discussed in this paper, any real-space constraint is easily incorporated into these algorithms, and soft constraints that do not require exact satisfaction of constraints (*e.g.* the diffraction data or NCS) are also easily incorporated.

Statistical density modification puts classical density modification on a sound statistical footing and has proved to be effective for macromolecular phasing. However, it still depends on a gradient-based optimization of the overall likelihood function (or posterior density) that finds the local maximum closest to the starting point defined by the aggregate of the initial experimental and map phases. The more real-space information that can be included (such as expected patterns in a macromolecular electron density), the more the local minima are suppressed, but in many cases there will be insufficient prior information to produce a likelihood function with a single global minimum, which would be required for true *ab initio* phasing using a gradient-based approach. Therefore, a potentially useful application of the methods described in this paper might be to conduct an initial reconstruction which finds the region of the global maximum, and then use this as a starting point for statistical density modification based on the map-probability function.

This work was supported by a James Cook Research Fellowship to RPM and a New Zealand Tertiary Education Commission Top Achiever Doctoral Scholarship to VLL. We are grateful to Richard Kingston for discussion and to an anonymous referee whose comments improved the manuscript.

References

Abrahams, J. P. (1997). *Acta Cryst.* D**53**, 371–376.
Bauschke, H. H., Combettes, P. L. & Luke, D. R. (2003). *J. Opt. Soc. Am. A*, **20**, 1025–1034.
Béran, P. & Szöke, A. (1995). *Acta Cryst.* A**51**, 20–27.
Bricogne, G. (1974). *Acta Cryst.* A**30**, 395–405.
Bricogne, G. (1984). *Acta Cryst.* A**40**, 410–445.
Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.
Chapman, H. N. *et al.* (2006). *Nat. Phys.* **2**, 839–843.
Cowtan, K. (2010). *Acta Cryst.* D**66**, 470–478.
Crowther, R. A. (1969). *Acta Cryst.* B**25**, 2571–2580.
Drenth, J. (1999). *Principles of Protein X-ray Crystallography*, 2nd ed. New York: Springer-Verlag.
Dumas, C. & van der Lee, A. (2008). *Acta Cryst.* D**64**, 864–873.
Elser, V. (2003*a*). *J. Opt. Soc. Am. A*, **20**, 40–55.
Elser, V. (2003*b*). *Acta Cryst.* A**59**, 201–209.
Elser, V. & Millane, R. P. (2008). *Acta Cryst.* A**64**, 273–279.
Fienup, J. R. (1982). *Appl. Opt.* **21**, 2758–2769.
Liu, Z.-C., Xu, R. & Dong, Y.-H. (2012). *Acta Cryst.* A**68**, 256–265.
Lo, V., Kingston, R. L. & Millane, R. P. (2009). *Acta Cryst.* A**65**, 312–318.
Lo, V. & Millane, R. P. (2010). *Proc. SPIE*, **7800**, 78000N.
Luke, D. R. (2005). *Inverse Probl.* **21**, 37–50.
Lunin, V. Yu. (1993). *Acta Cryst.* D**49**, 90–99.

Main, P. (1979). *Acta Cryst.* A**35**, 779–785.

Miao, J., Charalambous, P., Kirz, J. & Sayre, D. (1999). *Nature (London)*, **400**, 342–344.

Miao, J., Sayre, D. & Chapman, H. N. (1998). *J. Opt. Soc. Am. A*, **15**, 1662–1669.

Millane, R. P. (1990). *J. Opt. Soc. Am. A*, **7**, 394–411.

Millane, R. P. (1993). *J. Opt. Soc. Am. A*, **10**, 1037–1045.

Millane, R. P. & Stroud, W. J. (1997). *J. Opt. Soc. Am. A*, **14**, 568–579.

Oszlányi, G. & Sütő, A. (2008). *Acta Cryst.* A**64**, 123–134.

Plas, J. L. van der & Millane, R. P. (2000). *Proc. SPIE*, **4123**, 249–260.

Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.

Szöke, A., Szöke, H. & Somoza, J. R. (1997). *Acta Cryst.* A**53**, 291–313.

Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.

Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.

Terwilliger, T. C. (2001). *Acta Cryst.* D**57**, 1763–1775.

Terwilliger, T. C. (2003). *Acta Cryst.* D**59**, 1688–1701.

Thibault, P., Elser, V., Jacobsen, C., Shapiro, D. & Sayre, D. (2006). *Acta Cryst.* A**62**, 248–261.

Xiang, S., Carter, C. W., Bricogne, G. & Gilmore, C. J. (1993). *Acta Cryst.* D**49**, 193–212.

Zhang, K. Y. J., Cowtan, K. D. & Main, P. (2006). *International Tables for Crystallography*, Vol. F, ch. 15.1, pp. 311–324. Chester: International Union of Crystallography.