Acta Crystallographica Section A
Foundations of Crystallography
ISSN 0108-7673

Received 4 February 2011
Accepted 3 March 2011

A charge-flipping algorithm to handle incomplete data

Gábor Oszlányi* and András Sütő

Research Institute for Solid State Physics and Optics of the Hungarian Academy of Sciences, POB 49, H-1525 Budapest, Hungary. Correspondence e-mail: go@szfki.hu

Missing data are a general hindrance for all iterative, dual-space methods of structure determination. Charge flipping is no exception; its real-space perturbation may turn out to be too strong if the amount of diffraction data is not sufficient. To handle this situation, we introduce a variant of the basic algorithm which combines the original charge-flipping density modification in real space, the reflector of the Fourier-modulus projection in reciprocal space and the parameterless iteration scheme of averaged alternating reflections (AAR). This simple algorithm is a balance of increased perturbations and full negative feedback, with the extra freedom that it can be fine-tuned by a different treatment of different unobserved reflections. The efficiency of the method was tested using several single-crystal data sets and varying the amount of missing data at both high and low resolution. The results prove that many small-molecule structures can be solved by utilizing significantly less data than is standard in current crystallographic practice.

1. Introduction

Charge flipping is a simple, dual-space algorithm of structure determination (Oszlányi & Sütő, 2004, 2008) which has enjoyed increasing acceptance in the last few years. Its many variants have been successfully applied to the *ab initio* solution of single crystals (Piao *et al.*, 2008; Lister *et al.*, 2009), powders (Wu *et al.*, 2006; Baerlocher, McCusker & Palatinus, 2007), periodic and aperiodic structures (Palatinus, 2004), quasicrystals (Katrych *et al.*, 2007; Weber *et al.*, 2009), zeolites (Baerlocher, Gramm *et al.*, 2007; O'Keeffe, 2009), protein heavy-atom substructures (Dumas & van der Lee, 2008), to the determination of two-dimensional exit wavefunctions (Wu & Spence, 2005; Eggeman *et al.*, 2009) and to the reconstruction of non-periodic objects (Shneerson *et al.*, 2008; Fung *et al.*, 2008). Our own research was mainly focused on selecting and solving hard-to-solve crystal structures, cases which can be problematic, even if the diffraction data are complete and the resolution is high. This required various modifications of the real- and reciprocal-space parts of the iterative Fourier cycle, and may have given the impression that because charge flipping is essentially a Fourier method, high-resolution data are inevitable for its success.

In this paper we show that this is not the case, and at least for many small-molecule structures the completeness and resolution requirements can be significantly relaxed. In early tests of the original charge-flipping algorithm it was noticed that for some crystal structures the convergence of the iteration process occurs much faster than for others. The qualitative description of these 'easy-to-solve' cases was given in Oszlányi & Sütő (2005); favourable characteristics are the presence of centrosymmetry, heavy atoms, separate molecular fragments, high volume per non-hydrogen atom *etc*. It was also observed that the same kind of structures could be solved with a data resolution significantly lower than the usual $d_{\min} \sim 0.8$ Å value, although the resulting electron density was often of poor quality. Here we pick a few such examples and investigate how much the resolution and completeness requirements can be reduced so that the iteration still converges and provides a well recognizable electron density.

Two previous publications have already addressed this issue. The first one is by Palatinus *et al.* (2007), which describes a two-stage procedure. Before the iteration process it estimates the missing structure-factor moduli by the maximum-entropy method using the condition that the Patterson function must be positive and smooth, and then runs the charge-flipping algorithm for structure solution. Tests of the method were impressive and, beyond the low-resolution case, several other geometries of missing data were well treated. A small hindrance is that the two-stage procedure requires the use of two separate programs: *BayMEM* and *SUPERFLIP* (van Smaalen *et al.*, 2003; Palatinus & Chapuis, 2007), and it seems that users rather avoid this complication in practical work. Note that the current version of *SUPERFLIP* (Palatinus, 2010) includes several simpler options for the treatment of unobserved reflections (*e.g.* freely changing reflections or upper bounds for unobserved moduli) which are still efficient, and then no external program is required.

The second main approach was described by Coelho (2007) and is implemented in the commercial *Topas* program (Bruker, 2007). It incorporates the tangent formula of classical direct methods in a modified iteration scheme. The tangent

formula acts as a corrective influence which decreases the perturbation of charge flipping; therefore, the real-space perturbation must be increased differently and in a drastic way. While this paper only discusses examples of low-resolution data, it is likely that the procedure can also be used with other geometries of missing data. The self-contained implementation in *Topas* is convenient for the user, but as the source code is not open, the method cannot be easily reproduced or modified.

The present paper describes a third approach to handling missing data which is both effective and easy to implement. It is essentially a negative-feedback variant of the iteration scheme, and keeps as much of the advantages of the original charge-flipping algorithm as possible. It is completely *ab initio* and deterministic, there is no application of the tangent formula or any target function of optimization, it still works in space group $P1$ and without using the chemical composition, and it is still based on the name-giving density modification using a single threshold parameter.

2. Elements of the algorithm

To construct a dual-space algorithm for handling incomplete data we must: (i) treat observed and unobserved structure factors separately, (ii) define the real- and reciprocal-space modifications, (iii) combine them into a suitable iteration scheme, (iv) select parameters of the algorithm, (v) choose appropriate figures of merit to be checked for convergence, and (vi) improve the quality of the final electron density.

Measured diffraction intensities provide the real moduli of the complex structure factors: $F_{\rm obs}(\mathbf{h}) = |F(\mathbf{h})|$. Any structure-solution method that relies on the fast Fourier transform must work with a finite grid of the unit cell and a reciprocal-space parallelepiped which contains all observed and unobserved structure factors to be used in the iteration process. The complete set of observed data occupies a spherical region of the reciprocal-space parallelepiped, while unobserved data can be categorized as: (i) the single point of forward scattering $F(0)$ that can never be measured; (ii) reflections outside the resolution sphere of radius H but within the reciprocal parallelepiped that are not measured even if the observed data are complete; (iii) there may be additional missing reflections within the resolution sphere because of instrumental reasons, *e.g.* regions blocked by the beamstop, blocked by a high-pressure cell *etc*.

The algorithm must handle all these structure factors, partly by an initial normalization, and partly by deciding their individual treatment in the iterations process. In this paper the moduli of observed reflections are normalized using the scattering-factor function of the heaviest atom (Oszlányi & Sütő, 2008):

$$E_{\rm obs}(\mathbf{h}) = F_{\rm obs}(\mathbf{h})/f_{\rm heavy}(\mathbf{h}). \quad (1)$$

This sharpens the target electron density as well as usual E's, and utilizing only the type of the heaviest atom preserves much of the method's *ab initio* character. The treatment of unobserved structure factors is a different issue. Although their $F_{\rm obs}$ data are not known, they must be included in the reciprocal-space part of the iteration process. As in earlier studies, they are either allowed to change freely or are reset to zero in each cycle depending on the initial decision. What these modifications really mean depends on the selected iteration scheme.

To describe the real- and reciprocal-space modifications and the iteration scheme we borrow the terminology and notation of the convex feasibility problem (Stark & Yang, 1998). Here the electron-density function sampled at the grid points $\mathbf{r}_i$ forms a vector ρ of Euclidean space. Imposing a constraint on ρ corresponds to finding the closest point of a vector set C that is achieved by an orthogonal projection P_C. There are usually two such constraints (both expressed in real space) and the task is to find a solution that simultaneously fulfils both. The simplest iterative phase-retrieval algorithm expressed this way is the Gerchberg–Saxton scheme (Gerchberg & Saxton, 1972):

$$\rho^{n+1} = P_{\rm M} P_{\rm S} \rho^n \quad (2)$$

where ρ^n and ρ^{n+1} are the electron densities before and after the nth cycle, and $P_{\rm S}$ and $P_{\rm M}$ are the support and modulus projections, respectively. Unfortunately, the modulus constraint is inherently non-convex, so while the convex feasibility problem is a useful and widespread language, we are left with its concise notation but without the convergence proofs of the original mathematical theory.

For crystals there is another problem: the support of atomic volumes is not known, so this constraint must be replaced by a procedure that promotes only the sparseness of the electron density. Charge flipping is such a real-space density modification. It is defined as

$$R_\delta = 2L_\delta - I. \quad (3)$$

Here R refers to its reflector-like construction, L_δ is a variant of low-density elimination (Shiono & Woolfson, 1992), δ is a small positive threshold and I is the identity operation. The threshold parameter is dynamic; it is updated before each iteration cycle as $\delta = k\sigma$, where k is a constant value and $\sigma = (\langle\rho^2\rangle - \langle\rho\rangle^2)^{1/2}$ is the standard deviation of the current electron-density map (Oszlányi & Sütő, 2008). L_δ and R_δ act on the electron density as

$$L_\delta[\rho(\mathbf{r})] = \begin{cases} \rho(\mathbf{r}) & \text{if } \rho(\mathbf{r}) \geq \delta \\ 0 & \text{otherwise} \end{cases}$$
$$R_\delta[\rho(\mathbf{r})] = \begin{cases} \rho(\mathbf{r}) & \text{if } \rho(\mathbf{r}) \geq \delta \\ -\rho(\mathbf{r}) & \text{otherwise.} \end{cases} \quad (4)$$

We consciously use the notation L_δ instead of P_δ, to emphasize that even though $L_\delta^2 = L_\delta$, this operation is not distance minimizing (Oszlányi & Sütő, 2008). As a consequence, the charge-flipping operation is not a true reflector, but we keep its R_δ symbol.

For completeness, we also recall the steps of the Fourier-modulus projection. $\tilde{P}_{\rm M}$ is naturally calculated in reciprocal space, so its real-space form $P_{\rm M}$ requires two additional Fourier transforms:

$$P_{\rm M} = \mathcal{F}^{-1}\tilde{P}_{\rm M}\mathcal{F}. \quad (5)$$

$\tilde{P}_{\rm M}$ acts differently on reflections which are (i) observed, (ii) unobserved and forced to be zero, or (iii) unobserved and allowed to change freely. From now on we shall use the codes 1, 0 and 2 for these three types of reflections. In the case of a complete high-resolution data set

$$(\tilde{P}_{\rm M}\mathcal{F}\rho)(\mathbf{h}) = \begin{cases} E_{\rm obs}(\mathbf{h})\exp[i\varphi(\mathbf{h})] & \text{if } 0 < h \leq H \quad \text{(code 1)} \\ 0 & \text{if } h > H \quad \text{(code 0)} \\ G(\mathbf{h}) & \text{if } \mathbf{h} = \mathbf{0} \quad \text{(code 2)} \end{cases} \quad (6)$$

where $G(\mathbf{h}) = (\mathcal{F}\rho)(\mathbf{h})$ are temporary structure factors with phases $\varphi(\mathbf{h})$. With missing data, a further decision must be made to assign them the codes 0 or 2. As an example, Fig. 1 shows a typical situation of low-resolution data, when a shell of freely floating reflections (code 2) is added in the manner of the 'free-lunch method' (Caliandro *et al.*, 2005, 2007). Special attention must be given to those reflections that are unobserved but must be extinct due to lattice centring, glide planes or screw axes. The choice of this paper is to assign them the code 0 if they are unobserved. This turns out to be very important, and assumes the knowledge of the reflection conditions but not the full space-group symmetry.

At this point we are equipped to give a coherent description of iteration schemes. The basic low-density elimination (LDE) and charge-flipping (CF) algorithms can be viewed as the alternating application of their name-giving density modifications and the Fourier-modulus projection:

$$\text{LDE} = P_{\rm M}L_{\delta}, \quad (7)$$

$$\text{CF} = P_{\rm M}R_{\delta}. \quad (8)$$

In this pair, charge flipping uses significantly more perturbation to explore the phase space, and is capable of solving more complex structures. At the same time, such an algorithm leads to more noisy solutions, so it is useful to apply LDE as a 'cleanup' procedure to improve the quality of the final electron density. Also note that fine details may be hidden by this notation, and the treatment of $F(0)$ is such an issue. For charge flipping and its variants it is always better to let $F(0)$ change freely (Oszlányi & Sütő, 2008). The same applies for LDE when used for cleanup, but when used for phasing it is more appropriate to set $F(0)$ to zero (Fleischer *et al.*, 2010).

Some additional remarks on LDE. The current variant differs from the 1992 original (Shiono & Woolfson, 1992) in two ways: (i) to see clearly the relation of the two algorithms, we keep only the name-giving real-space part, and replace the reciprocal-space weighting by the simplest Fourier-modulus projection; (ii) the real-space density modification now uses a parameter that is dynamic and scales with σ, which is a significant improvement. It is also interesting that later versions of LDE abandoned the sharp cutoff of the original real-space modification and switched to continuous damping. We think that it was a mistake: sharp cutoffs, especially reflection-like charge flipping, are more efficient for phasing.

A general strategy to improve further the efficiency of iteration schemes is to add an independent perturbation in reciprocal space. A particularly simple choice is to replace the Fourier-modulus projection by its reflector:

$$\tilde{R}_{\rm M} = 2\tilde{P}_{\rm M} - I, \quad (9)$$

$$R_{\rm M} = 2P_{\rm M} - I. \quad (10)$$

Unfortunately, the following variant of charge flipping does not work:

$$T = R_{\rm M}R_{\delta}. \quad (11)$$

We reached the point where even high-resolution data do not tolerate this amount of perturbation. There is a danger in this popular notation that one can easily overlook what happens with unobserved reflections. In this case, a large amount of unexpected perturbation came from the unobserved reflections outside the resolution sphere, which changed their sign under $\tilde{R}_{\rm M}$, according to the definition of $\tilde{P}_{\rm M}$ for code 0.

Our previous solution [also called $F_{\rm obs} + \Delta F$; see Oszlányi & Sütő (2008)] was to step back from this variant:

$$T' = R'_{\rm M}R_{\delta}. \quad (12)$$

The prime in T' means that unobserved reflections outside the resolution sphere were treated differently from T. They were explicitly reset to zero, instead of allowing the $\tilde{R}_{\rm M}$ operator to flip their sign.

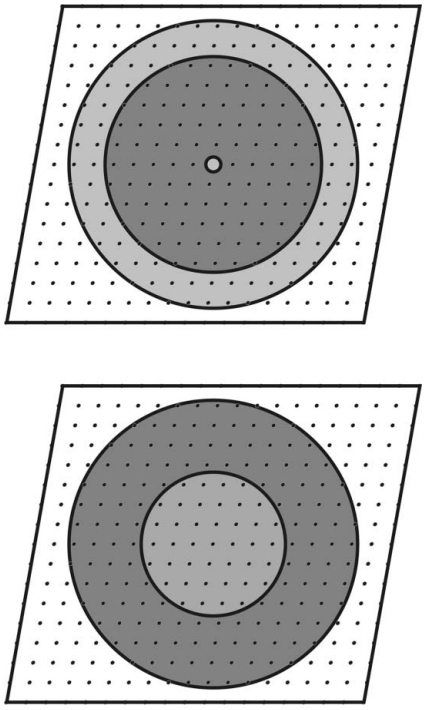

Figure 1
Parallelepipeds of reciprocal space used in the case of missing high-resolution data (top panel) and missing low-resolution data (bottom panel). Dark grey regions contain observed reflections (code 1), light grey regions contain unobserved reflections that are allowed to float freely (code 2) and white regions contain unobserved reflections that are set to zero (code 0).

Table 1
List of test structures.

Columns: journal code and reference, chemical composition (without H atoms), unit-cell volume, number of formula units, space group, data-collection temperature, data resolution.

Code	Reference	Composition	V (Å^3)	Z	Space group	T (K)	$d_{\min}$(Å)
ln1194	Yue *et al.* (2006)	$C_{30}O_9$	1496	2	$P2_1$	295	0.77
sk3023	Kikolski *et al.* (2006)	$C_{20}O_4$	1682	4	$P2_12_12_1$	100	0.85
sk1293	Wheatley *et al.* (1999)	$C_{44}N_6O_8$	1809	2	$P\bar{1}$	100	0.84
bm3037	Gliński *et al.* (2007)	$C_{12}O_2$	2175	8	$Pca2_1$	173	0.74
ci6275	Karakurt *et al.* (2003)	$C_{17}N_4O_2$	3057	8	$C2/c$	295	0.84
gd3109	Lipstman *et al.* (2007)	$C_{48}N_4O_8$	4048	4	$Cmca$	110	0.81
sk3179	Gelbrich *et al.* (2007)	$C_{10}N_2O_3$	4502	16	$P2_1/c$	120	0.84
bg3066	Machado *et al.* (2008)	$C_{54}ClCuP_3$	17597	16	$C2/c$	150	0.71

Now, we define the iteration scheme used in the present study. It is based on charge flipping in real space, the Fourier-modulus reflector in reciprocal space, and relies on full negative feedback:

$$\tfrac{1}{2}(R_{\rm M}R_\delta + I). \qquad (13)$$

The same scheme but with convex constraints in both spaces was named averaged alternating reflections (AAR) by Bauschke *et al.* (2002, 2004), who analysed its relation to earlier algorithms (Douglas & Rachford, 1956; Lions & Mercier, 1979; Fienup, 1982). Several other algorithms are also related, and can be considered as generalizations of the basic scheme while introducing different intuitions and extra parameters (Elser, 2003; Bauschke *et al.*, 2003; Luke, 2005).

We shall use AAR in its simple, parameterless form. In the case of complete data, its behaviour is entirely determined by the single δ parameter of charge flipping and the amount of available Fourier moduli. In the case of incomplete data, assigning the codes 0 or 2 to missing structure factors gives us more freedom in tuning the algorithm. The $\tilde{R}_{\rm M}$ operator flips the sign of zeroed reflections $[G(\mathbf{h}) \rightarrow 2 \times 0 - G(\mathbf{h})]$, and thus introduces a strong perturbation, while freely floating reflections remain unchanged $[G(\mathbf{h}) \rightarrow 2 \times G(\mathbf{h}) - G(\mathbf{h})]$, and act as a kind of damping. There is no 'silver bullet' of algorithms in this class; the right amount of perturbation required always depends on the available data. The sufficiently strong and tunable perturbations in the dual spaces, coupled with the negative feedback of the above iteration scheme, make this algorithm particularly useful for handling incomplete data. Although we experimented with other iteration schemes, and also obtained solutions, AAR was usually faster, gave a broader useful δ range and, most importantly, led to electron-density maps of better quality.

To completely specify the algorithm we must also answer the questions: how is the iteration process started? What figure of merit is followed to identify the convergence? What procedure is used to improve the final electron-density map?

Initialization is conventional. The starting point is in reciprocal space, where random phases are combined with $E_{\rm obs}$ for observed reflections, while all unobserved reflections are set to zero. Known phases or known structural fragments could help a lot, but we avoid their use in the present *ab initio* study. The choice of figures of merit was discussed in detail in Oszlányi & Sütő (2008). Here we skip this problem, because by working with known structures we can always compare the current electron density with the ideal one. Before calculating the correlation coefficient of electron densities we also cut the noise by applying L_δ. This is an auxiliary step that does not change the ρ of the main iteration process. The convergence is signalled by a sharp increase of the correlation coefficient, and typical values in the 0.85–0.95 range are obtained. These correlations correspond to very clean electron-density maps, excluding any doubt about the correctness of a solution. After the convergence we apply an LDE cycle as a final cleanup procedure, and we also switch back from $E_{\rm obs}$ to $F_{\rm obs}$, which decreases the series-termination effect of finite resolution.

3. Tests of the algorithm

For tests of the algorithm we selected eight small-molecule structures, mostly published in *Acta Cryst. C*. Table 1 gives a brief summary of their chemical composition, size, symmetry and the resolution of their diffraction data. The first seven items of the list are organic structures that contain only light atoms because these more often pose problems for *ab initio* phasing than crystals that also contain some heavy atoms. Inorganic or organometallic structures are not included in the present study because they are usually easy to solve, but in cases when very heavy and very light atoms are in close proximity, it is extra work to get clean electron-density maps in those regions. With this list of test structures we tried to cover a variety of space groups and a range of unit-cell sizes. Only the last item deviates from our all-light-atom standard – it is here to demonstrate how well the algorithm performs if both inversion symmetry and heavy atoms are present.

The original data sets are nearly complete and extend to high resolution, so their structure solution is not a problem for any variant of charge flipping. To study the behaviour of the previously described algorithm with incomplete data, we performed two complementary series of tests in which the original data sets were truncated at high or at low resolution.

For missing high-resolution data (see Fig. 1) the original data set is used only for $h_1 \leq h \leq h_2$ where $h_1 = 0$ is fixed and $h_2 = 1/d_2$ is a parameter of our tests which simulates an upper resolution limit that is worse than what we really have. The measured data are deleted above the radius h_2, and extended

Table 2
Structure-solution statistics of missing high-resolution data using charge flipping in the AAR scheme (*a*) and in the simple Fourier recycling scheme (*b*).

Columns: journal code of structure, upper resolution limits d_2 from 1.0 to 1.6 Å. The values in the table are the per cent success rate followed by the number of iteration cycles per solution in parentheses as δ-averaged indicators of efficiency.

Code	1.0	1.1	1.2	1.3	1.4	1.5	1.6
(*a*)							
ln1194	100.0 (219)	99.6 (428)	77.8 (2720)	36.6 (10700)	5.9 (81600)	0.1 (6×10^6)	0.0 (∞)
sk3023	99.5 (326)	99.8 (444)	97.7 (1060)	70.8 (3760)	45.5 (8170)	9.1 (52400)	0.0 (∞)
sk1293	100.0 (60)	100.0 (54)	100.0 (68)	100.0 (90)	100.0 (136)	99.9 (221)	96.8 (716)
bm3037	89.2 (1010)	80.0 (1720)	55.1 (5420)	12.6 (36900)	5.3 (92100)	0.8 (6×10^5)	0.0 (∞)
ci6275	100.0 (47)	100.0 (59)	100.0 (70)	99.9 (110)	99.6 (164)	97.6 (402)	82.3 (1600)
gd3109	84.0 (1020)	86.3 (903)	85.0 (1120)	79.2 (1930)	53.8 (4800)	48.3 (6070)	33.1 (10800)
sk3179	91.3 (800)	92.3 (744)	89.2 (1230)	79.8 (2140)	72.3 (3100)	56.2 (5520)	14.3 (32500)
bg3066	100.0 (49)	100.0 (53)	100.0 (64)	100.0 (81)	99.9 (109)	99.8 (151)	99.3 (241)
(*b*)							
ln1194	43.2 (7820)	16.1 (27900)	1.3 (3.7×10^5)	0.0 (∞)	0.0 (∞)	0.0 (∞)	0.0 (∞)
sk3023	58.8 (4720)	28.8 (14100)	1.4 (3.5×10^5)	0.0 (∞)	0.0 (∞)	0.0 (∞)	0.0 (∞)
sk1293	100.0 (125)	99.0 (395)	79.7 (1820)	56.1 (4960)	30.7 (12800)	6.2 (77600)	2.1 (2.4×10^5)
bm3037	48.8 (6350)	28.8 (13800)	2.8 (1.7×10^5)	0.0 (∞)	0.0 (∞)	0.0 (∞)	0.0 (∞)
ci6275	83.3 (1150)	83.1 (1350)	63.8 (3390)	43.7 (7500)	40.7 (8380)	17.3 (25400)	3.1 (1.6×10^5)
gd3109	97.3 (520)	81.1 (1930)	46.6 (6970)	23.3 (18100)	10.5 (44600)	4.2 (1.2×10^5)	2.0 (2.5×10^5)
sk3179	83.5 (1730)	64.5 (3820)	24.3 (17500)	9.0 (52700)	2.3 (2.1×10^5)	0.8 (6.7×10^5)	0.0 (∞)
bg3066	99.0 (420)	98.0 (560)	96.5 (720)	94.1 (891)	91.8 (1070)	88.7 (1350)	85.9 (1710)

by a spherical shell of unobserved, freely floating reflections up to $h_3 = 1/d_3$. Naturally, $h_2 \leq h_3$. This means that all observed reflections in the $h_1 \leq h \leq h_2$ range are given the code 1, while all unobserved reflections in the $h_1 \leq h \leq h_3$ range (including any such reflection with $h \leq h_2$) are given the code 2. The rest of the structure factors with $h > h_3$ are given the code 0. Finally, special attention must be paid to those reflections that are unobserved but should be extinct because of the space-group symmetry. It is very important not to let them change freely, otherwise they may run away, and drive the iteration process towards complete failure. Therefore, independent of all previous assignments, unobserved extinctions must be given the code 0.

Usually $d_3 = 1.0$ Å is a good choice for the extended resolution limit and, while it is a parameter of the algorithm, here we fix its value. The upper resolution limit d_2 is only a parameter of our tests; we shall investigate its 1.0–1.6 Å range in 0.1 Å steps. At a given d_2 we also cover a range of threshold parameters: in $\delta = k\sigma$ the k values are increased from 1.0 to 1.5 in 0.1 steps. So we have a two-dimensional parameter space where each point is a slightly different combination of 'data + algorithm' that must be characterized by sufficient statistics. The protocol is as follows. For each test structure and each (d_2, δ) pair we perform N structure-solution attempts starting from different random phase sets, and run the iteration process until convergence is detected or until we reach the maximum number of cycles M. Our choice is $N = 200$, $M = 5000$, so in the worst, no-solution case we spend $NM = 10^6$ iteration cycles on a single (d_2, δ) parameter pair.

The success rate is naturally defined as

$$\eta = \frac{n}{N}, \tag{14}$$

where $n \leq N$ is the number of converged runs.

The average number of iteration cycles spent on a successful structure solution is another useful indicator, and is defined as

$$\langle m \rangle = \frac{\sum_{i=1}^{N} m_i}{n}, \tag{15}$$

where m_i is the number of cycles needed for convergence in the ith run or $m_i = M$ if there was no convergence. The resulting tables are informative but large; therefore, these are published as supplementary material.[1] Here, we characterize only the algorithm's overall performance by averaging over the δ parameter at a fixed d_2. The expressions of η and $\langle m \rangle$ can be easily generalized to include the contribution of all runs at several δ values, in our case a total of 1200 structure-solution attempts. Table 2 summarizes the results. We emphasize that the k range of 1.0–1.5 in 0.1 steps is rather broad and rather coarse, especially when compared to our previous studies when algorithm variants were optimized by fine-tuning the threshold parameter. Still, the present rough protocol of obtaining overall indicators better approximates one's first encounter with a given problem. If the aim is to solve the structure fast, then there is no time for finding optimal parameters.

The performance of the algorithm was also studied at the other extreme, the case when all pieces of missing data are missing at low resolution (see Fig. 1). For these tests a lower-resolution limit $h_1 = 1/d_1$ is set and considered as a parameter of the data. The original data set is deleted within a sphere of radius h_1 and inside reflections $h \leq h_1$ are treated as unobserved with freely floating code 2. In contrast, measured reflections are kept up to the highest available resolution, so $h_2 = 1/d_2$ is fixed and is determined by the original data ($d_2 \sim 0.8$ Å). In the $h_1 < h \leq h_2$ shell of reciprocal space observed reflections are given the code 1, and if any unobserved reflections occur, they are given the code 2. Reflections

[1] Supplementary data for this paper, including detailed tests and comparisons of the AAR iteration scheme *versus* Fourier recycling, are available from the IUCr electronic archives (Reference: SH5127). Services for accessing these data are described at the back of the journal.

Table 3
Structure-solution statistics of missing low-resolution data using charge flipping in the AAR scheme (*a*) and in the simple Fourier recycling scheme (*b*).

Columns: journal code of structure, lower resolution limits d_1 from 3.0 to 1.1 Å. The values in the table are the per cent success rate followed by the number of iteration cycles per solution in parentheses as δ-averaged indicators of efficiency.

Code	3.0	2.0	1.5	1.4	1.3	1.2	1.1
(*a*)							
ln1194	100.0 (72)	100.0 (100)	88.7 (939)	66.3 (2900)	47.8 (5940)	18.0 (23600)	0.0 (∞)
sk3023	99.9 (199)	83.2 (1310)	48.2 (6400)	21.8 (19200)	1.4 (3.5×10^5)	0.0 (∞)	0.0 (∞)
sk1293	100.0 (95)	83.7 (1030)	83.3 (1060)	83.3 (1090)	74.7 (2030)	46.2 (5980)	0.0 (∞)
bm3037	100.0 (128)	100.0 (349)	78.0 (1640)	68.8 (2530)	50.8 (5220)	17.3 (24700)	1.3 (3.7×10^5)
ci6275	100.0 (35)	100.0 (54)	82.1 (1210)	72.0 (2010)	67.1 (2520)	52.8 (4540)	0.0 (∞)
gd3109	100.0 (80)	100.0 (853)	75.4 (1770)	66.8 (2570)	59.3 (3530)	31.3 (11200)	0.8 (6×10^5)
sk3179	96.7 (482)	83.3 (1080)	83.3 (1110)	82.2 (1300)	77.4 (1650)	45.5 (6300)	0.0 (∞)
bg3066	100.0 (41)	100.0 (48)	100.0 (116)	98.2 (355)	84.3 (1020)	83.3 (1060)	83.3 (1080)
(*b*)							
ln1194	94.2 (1040)	87.6 (3140)	31.2 (12500)	15.3 (29400)	4.4 (1.1×10^5)	0.3 (2×10^6)	0.0 (∞)
sk3023	62.3 (4160)	20.4 (21100)	2.8 (1.7×10^5)	1.3 (4×10^5)	0.0 (∞)	0.0 (∞)	0.0 (∞)
sk1293	100.0 (120)	100.0 (134)	66.7 (2720)	66.7 (2780)	49.3 (5690)	24.3 (16700)	0.0 (∞)
bm3037	81.4 (1920)	76.9 (2440)	55.7 (5260)	46.0 (7260)	24.1 (17300)	4.8 (1×10^5)	0.0 (∞)
ci6275	100.0 (95)	100.0 (108)	82.3 (1240)	66.5 (2700)	57.7 (3980)	28.9 (12700)	0.0 (∞)
gd3109	100.0 (359)	99.9 (362)	82.1 (1620)	65.3 (3300)	46.3 (6670)	16.3 (27300)	0.0 (∞)
sk3179	95.0 (857)	95.9 (835)	95.3 (1160)	78.6 (2270)	62.5 (4150)	22.9 (18700)	0.0 (∞)
bg3066	100.0 (215)	99.9 (188)	99.5 (188)	98.9 (213)	99.1 (203)	98.4 (242)	82.8 (1240)

are not extended beyond the measured resolution, and all $h > h_2$ reflections are given the code 0. As earlier, independent of all previous assignments, unobserved extinctions must be given the code 0. For each test structure we investigate the lower-resolution limit d_1 in the 3.0–1.1 Å range, and the previously defined protocol of structure-solution statistics is performed. For each d_1 a k range of 1.0–1.5 in 0.1 steps is covered, that again means 1200 runs in total and 6×10^6 iteration cycles in the worst case. Table 3 summarizes the results.

For comparison and completeness, Tables 2 and 3 also contain the double-averaged structure-solution statistics using charge flipping in the simple Fourier recycling scheme. Data treatment, reflection codes and calculation protocols are identical to the above specifications, the only difference is the k range of 0.8–1.3 that is slightly shifted to obtain better results. It is quite obvious that as all other elements/improvements of the algorithm are the same, it is really the AAR iteration scheme that performs better than Fourier recycling. Even more detailed tests and comparisons can be found in the supplementary material.

Although such tables are never complete, we can make some general observations about the performance of the present algorithm. First, if a structure is easy to solve in the case of complete high-resolution data, then there is a fair chance that it can also be solved using incomplete data. There are no absolute resolution limits, but we can easily find cases in Table 2 when no data of resolution higher than $d_2 = 1.6$ Å or in Table 3 when no data of resolution lower than $d_1 = 1.2$ Å are needed for a successful structure solution. These numbers mean that only a 12.5% or 30% fraction of the complete 0.8 Å resolution sphere is needed for a successful structure solution. In favourable cases we can do even better (*e.g.* reach $d_2 = 2.0$ Å), so these requirements may be considered as conservative estimates. On the other hand, there is no guarantee that such resolutions will always suffice. Some non-centrosymmetric structures still pose a problem for all variants of charge flipping, and this just gets worse with incomplete data. A further observation is regarding the speed of convergence. For a given structure, increasing the amount of missing data usually means that the number of iteration cycles needed for a solution increases by several orders of magnitude before we run out of computational resources. Although the η and $\langle m \rangle$ indicators included in the tables are double averages, simple averages at a given δ behave similarly, while at the best δ and the best run the computational cost of a solution may be 10–100 times lower. Therefore, we hope to gain further experience with the modified algorithm and improve its choice of parameters.

The quality of the electron density was discussed earlier. Here we only stress that it must be evaluated in two different situations: first when the figures of merit are checked at regular intervals, and second after the convergence when the final electron density is improved to facilitate its atomic interpretation and refinement. As mentioned above, the convergence test is coded as an auxiliary L_δ step that does not change the ρ of the main iteration process. After convergence one cycle of the LDE cleanup procedure is applied on ρ that is further modified by an $E_{\rm obs} \rightarrow F_{\rm obs}$ transformation. More powerful techniques like averaging multiple runs or switching to a finer real-space grid were not applied here. Fig. 2 shows a comparison of the reconstructed electron densities obtained by using the original complete and the truncated incomplete data. We judge that such figures speak for themselves.

4. Summary

While in our earlier works we confined ourselves to structure solution from complete high-resolution data and tried to push the limit of solvability towards larger structures, in the present paper we proposed a variant of the charge-flipping algorithm which is efficient when the structures are somewhat simpler

but the available data sets are incomplete. This variant preserves the simplicity of the original algorithm, its key component is the name-giving density modification, and its key parameter is the dynamically determined threshold δ.

Specifically, we studied the two extreme cases of missing data that were generated from high-resolution data sets by truncating them at high or low wavenumbers. In the case of missing high-resolution data measured reflections are available up to a wavenumber h_2. A parameter $h_3 > h_2$ defines an extended spherical layer from h_2 to h_3 within which the unknown Fourier amplitudes are allowed to evolve freely, and only outside h_3 are they reset to zero. In the case of missing low-resolution data there are no observed reflections below a wavenumber h_1, and there is no other parameter than the threshold δ: the unknown Fourier amplitudes below h_1 are allowed to evolve freely, and those outside the resolution sphere are reset to zero. In both cases, for symmetry-induced extinctions the structure factors are also reset to zero.

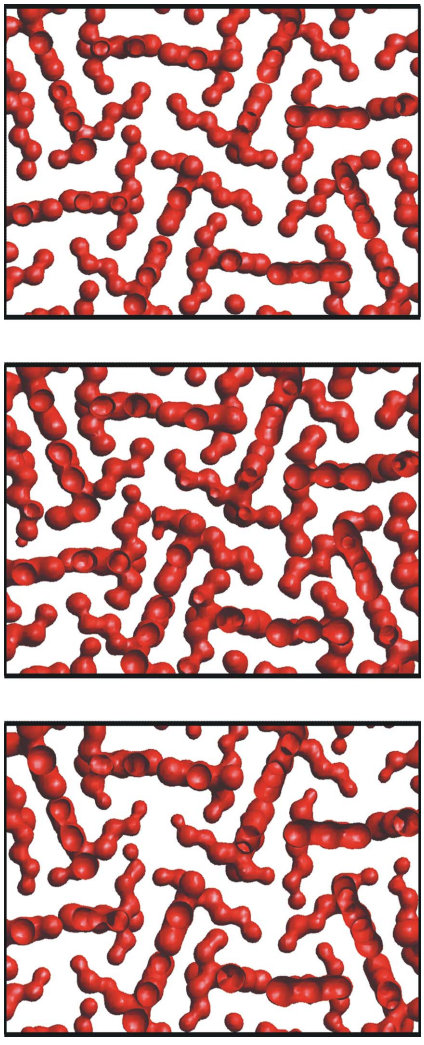

Figure 2
Electron densities of butobarbital (code sk3179). The original structure (top), structure solution of missing high-resolution data (middle, $d_2 = 1.6$ Å) and of missing low-resolution data (bottom, $d_1 = 1.3$ Å). Converged electron densities were obtained using the charge-flipping threshold $\delta = 1.2\sigma$, and were improved by one cycle of LDE and by switching back from $E_{\rm obs}$ to $F_{\rm obs}$. All plots are shown at the 1.0σ isosurface level.

The original charge-flipping algorithm with the above specifications still works. However, the success rate is often low, the useful δ range is narrow and the detection of the convergence is difficult. It is apparent that the perturbation introduced by charge flipping is too strong in the case of missing data, and some damping is necessary. As usual, damping in the form of a negative feedback slows down the convergence, and its unwanted effect must be counterbalanced by some additional perturbation. After evaluating different possibilities we opted for the iteration scheme AAR (Bauschke *et al.*, 2002, 2004), which adds a negative feedback and increases the perturbation through the replacement of the Fourier-modulus projection by its reflector; meanwhile, no new parameter is introduced. Contrary to the original algorithm, in this scheme the standard deviation σ of the electron density is not conserved, and the dynamical choice of δ is compulsory. Fortunately, by associating δ to σ *via* $\delta = k\sigma$, the success of the iteration is not very sensitive to the choice of k, and the same can be said about the choice of h_3.

Although the present charge-flipping variant performs quite well, it may fail, as any other method, and the cause of its occasional failure is not easy to identify. In the tables the test structures are listed in the order of increasing unit-cell volumes. The order of the arising difficulty does not agree with this. For instance, symmetry is a much more important factor: non-centrosymmetric structures with only screw axes are a greater challenge, irrespective of their size. The absence of clear criteria of convergence is notorious not only for charge flipping but for all iterative phase-retrieval methods. It is the synonym of the absence of a proof of convergence, since a theorem would necessarily include the conditions of its validity. Iteration methods based on alternating projections to two or more constraint sets have long been used in mathematics; their first appearance is perhaps in von Neumann (1949). In the numerical solution of partial differential equations by the operator splitting method (Douglas & Rachford, 1956; Lions & Mercier, 1979) convergence could be proven. The apparently insurmountable difficulty in applications to phase retrieval is the non-convexity of the modulus constraint. A careful analysis of the situation can be found in papers by Combettes & Trussell (1990), Bauschke *et al.* (2002) and Luke (2005). Even in the case of a complete, perfect data set and known feasibility (*i.e.* knowing that all the constraints can simultaneously be satisfied) only local convergence and escape from 'near solutions' could be shown (Elser, 2003). However, in any real-life problem the resolution is finite and the data are imperfect. Therefore, either at least one of the constraints should be loosely defined (charge flipping works in this way) or they cannot be perfectly satisfied, and it is under such conditions that the iteration must get close to a solution by starting far from it. The theorem we would need should provide a lower bound on the limiting success rate $\lim_{N\to\infty} \eta(N)$ of finding a solution with a prescribed error within a given 'time' M by starting from a random initial condition, provided that the data satisfy some conditions. To

our knowledge, no existing iterative algorithm for phase retrieval is supported by such a theorem.

In the absence of theorems and proofs, experience and intuition are primordial. Based on them, we are confident that the charge-flipping variant presented in this paper will find its applications. Apart from the single-crystal cases discussed above, other types of lacunary data could be dealt with. A natural option is powder diffraction, where peak overlap quickly increases with 2θ and seriously limits the useful resolution range, while the structures to be solved are not necessarily complex. Resolutions are often similar to those considered in the present paper, and it is likely that an extended shell of freely floating reflections combined with the present iteration scheme will allow structure solution at the usual resolution limits. Another possible application is high-pressure single-crystal diffraction. This is a more serious problem because only reflections within a torus-shaped volume can be measured, while certain directions of the reciprocal space are completely blocked by the pressure cell. The treatment of such incomplete data may need the combination of several techniques, but we believe that the algorithm presented here can be one of them. The third possible application is macromolecular crystallography, where very large structures and low-resolution data occur simultaneously. In this case it may be more appropriate to apply the modified algorithm not *ab initio* but as an alternative density-modification technique after some phase information is already available. Here the advantage of charge flipping could be that it works uniformly in real space, and it does not need (or need to work out) any positional information about the molecular envelope.

This research was supported by OTKA grant No. 67980K.

References

Baerlocher, Ch., Gramm, F., Massüger, L., McCusker, L. B., He, Z., Hovmöller, S. & Zou, X. (2007). *Science*, **315**, 1113–1116.

Baerlocher, Ch., McCusker, L. B. & Palatinus, L. (2007). *Z. Kristallogr.* **222**, 47–53.

Bauschke, H. H., Combettes, P. L. & Luke, D. R. (2002). *J. Opt. Soc. Am. A*, **19**, 1334–1345.

Bauschke, H. H., Combettes, P. L. & Luke, D. R. (2003). *J. Opt. Soc. Am. A*, **20**, 1025–1034.

Bauschke, H. H., Combettes, P. L. & Luke, D. R. (2004). *J. Approx. Theory*, **127**, 178–192.

Bruker (2007). *Topas 4.1 User's Manual.* Bruker AXS Inc., Madison, Wisconsin, USA.

Caliandro, R., Carrozzini, B., Cascarano, G. L., De Caro, L., Giacovazzo, C. & Siliqi, D. (2005). *Acta Cryst.* D**61**, 556–565.

Caliandro, R., Carrozzini, B., Cascarano, G. L., De Caro, L., Giacovazzo, C. & Siliqi, D. (2007). *J. Appl. Cryst.* **40**, 931–937.

Coelho, A. A. (2007). *Acta Cryst.* A**63**, 400–406.

Combettes, P. L. & Trussell, H. J. (1990). *J. Optimiz. Theory App.* **67**, 487–507.

Douglas, J. & Rachford, A. (1956). *Trans. Am. Math. Soc.* **82**, 421–439.

Dumas, C. & van der Lee, A. (2008). *Acta Cryst.* D**64**, 864–873.

Eggeman, A., White, T. & Midgley, P. (2009). *Acta Cryst.* A**65**, 120–127.

Elser, V. (2003). *J. Opt. Soc. Am. A*, **20**, 40–55.

Fienup, J. R. (1982). *Appl. Opt.* **21**, 2758–2769.

Fleischer, F., Weber, T., Deloudi, S., Palatinus, L. & Steurer, W. (2010). *J. Appl. Cryst.* **43**, 89–100.

Fung, R., Shneerson, V., Saldin, D. K. & Ourmazd, A. (2008). *Nat. Phys.* **5**, 64–67.

Gelbrich, T., Zencirci, N. & Griesser, U. J. (2007). *Acta Cryst.* C**63**, o751–o753.

Gerchberg, R. W. & Saxton, W. O. (1972). *Optik*, **35**, 237–246.

Gliński, M., Wilczkowska, E., Madura, I. D. & Zachara, J. (2007). *Acta Cryst.* C**63**, o720–o722.

Karakurt, T., Dinçer, M., Kahveci, B., Ağar, E., Ağar, A. & Şaşmaz, S. (2003). *Acta Cryst.* E**59**, o1616–o1617.

Katrych, S., Weber, Th., Kobas, M., Massüger, L., Palatinus, L., Chapuis, G. & Steurer, W. (2007). *J. Alloys Compd.* **428**, 164–172.

Kikolski, E. M., Thompson, H. W. & Lalancette, R. A. (2006). *Acta Cryst.* C**62**, o397–o398.

Lions, P. L. & Mercier, B. (1979). *SIAM J. Numer. Anal.* **16**, 964–979.

Lipstman, S., Muniappan, S. & Goldberg, I. (2007). *Acta Cryst.* C**63**, o371–o373.

Lister, S. E., Evans, I. R. & Evans, J. S. O. (2009). *Inorg. Chem.* **48**, 9271–9281.

Luke, D. R. (2005). *Inverse Probl.* **21**, 37–50.

Machado, A., Manzoni de Oliveira, G. N., Fenner, H. & Burrow, R. A. (2008). *Acta Cryst.* C**64**, m233–m236.

Neumann, J. von (1949). *Ann. Math.* **50**, 401–485.

O'Keeffe, M. (2009). *Angew. Chem. Int. Ed.* **48**, 8182–8184.

Oszlányi, G. & Sütő, A. (2004). *Acta Cryst.* A**60**, 134–141.

Oszlányi, G. & Sütő, A. (2005). *Acta Cryst.* A**61**, 147–152.

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

Palatinus, L. (2004). *Acta Cryst.* A**60**, 604–610.

Palatinus, L. (2010). *Acta Cryst.* A**66**, s106.

Palatinus, L. & Chapuis, G. (2007). *J. Appl. Cryst.* **40**, 786–790.

Palatinus, L., Steurer, W. & Chapuis, G. (2007). *J. Appl. Cryst.* **40**, 456–462.

Piao, S. Y., Palatinus, L. & Lidin, S. (2008). *Inorg. Chem.* **47**, 1079–1086.

Shiono, M. & Woolfson, M. M. (1992). *Acta Cryst.* A**48**, 451–456.

Shneerson, V. L., Ourmazd, A. & Saldin, D. K. (2008). *Acta Cryst.* A**64**, 303–315.

Smaalen, S. van, Palatinus, L. & Schneider, M. (2003). *Acta Cryst.* A**59**, 459–469.

Stark, H. & Yang, Y. (1998). *Vector Space Projections*. New York: Wiley.

Weber, T., Dshemuchadse, J., Kobas, M., Conrad, M., Harbrecht, B. & Steurer, W. (2009). *Acta Cryst.* B**65**, 308–317.

Wheatley, P. S., Lough, A. J., Ferguson, G. & Glidewell, C. (1999). *Acta Cryst.* C**55**, 1489–1492.

Wu, J. S., Leinenweber, K., Spence, J. C. H. & O'Keeffe, M. (2006). *Nat. Mater.* **5**, 647–652.

Wu, J. S. & Spence, J. C. H. (2005). *Acta Cryst.* A**61**, 194–200.

Yue, Z.-Y., Li, S.-H., Gao, P., Zhang, J.-H. & Yan, P.-F. (2006). *Acta Cryst.* C**62**, o281–o282.