Ab initio structure solution by charge flipping

Gábor Oszlányi and András Sütő

Research Institute for Solid State Physics and Optics
Hungarian Academy of Sciences
P.O.B. 49, H-1525 Budapest, Hungary

Abstract

In this paper we present an extremely simple structure solution method termed *charge flipping*. It works ab initio on high resolution x-ray diffraction data in the manner of Fourier recycling. The real space modification simply changes the sign of charge density below a threshold, while in reciprocal space the modification is the F_{obs} map without any weighting. We test the algorithm using synthetic data for a wide range of structures, analyse the solution statistics and check the quality of reconstruction. Finally, we reconsider mathematical aspects of the algorithm in detail, showing that in this chaotic iteration process the solution is a limit cycle and not a fixed point.

1 Introduction

Ab initio structure solution by x-ray diffraction is a success story of the last century. Today the practicing crystallographer can rely on high quality data obtained on cryocooled crystals by area detectors and synchrotron radiation. In the structure solution process freely available software take the workload, a large part of accumulated knowledge is contained in their elaborate algorithms. The forefront is indisputably protein crystallography where the coordinates of more than a thousand atoms per asymmetric unit can be determined. The field of ab initio structure solution is mature, one has the impression that there is no room for big surprises.

Nevertheless, in this paper we present an amazingly simple structure solution algorithm – termed *charge flipping*. This algorithm was inspired by several methods described in the literature: phase retrieval in optics [1, 2, 3], dual space programs `SnB` and `ShelxD` [4, 5] and recent work on iterated projections [6, 7]. All these methods alternate between real and reciprocal space by the Fourier transform and do part of the job by imposing constraints on the real space charge density. In a sense this is a return to the era before direct methods but armed with orders of magnitude more computing power. The charge flipping algorithm described below is probably less efficient than state of the art programs today. However, it is surprising that it works at all, even in the small molecule world. Its extreme simplicity offers the prospect for exact mathematical formulation, raises hopes for further improvements and may help to understand the working and limitation of other methods.

2 On the use of data and grid

By ab initio structure solution we mean that there is no preliminary chemical or phase information and only a single wavelength diffraction data set is used in the process. Furthermore, anomalous scattering is not exploited, atomic scattering factors are taken to be strictly real. In this first presentation we use synthetic data and focus on what is feasible given ideal conditions. We assume that the diffraction data is complete up to a given resolution, it is error free, the absolute scale and global isotropic temperature factor are known. None of these conditions are required for the working of the algorithm, we tested the effect of Poisson noise, the error in absolute scale and temperature factor. However, only the ideal case is discussed here to keep the main message of the paper brief.

The real space charge density and reciprocal space structure factors are related by the Discrete Fourier Transform which is a unitary mapping between the two spaces. For practice this is coded as variants of the high speed FFT algorithm. If structure factors are limited by a given resolution then the charge density can be represented on a grid without loss of information. The necessary grid spacing is $d_{\mathrm{grid}} = d_{\mathrm{min}}/2$ where $K = 2\pi/d_{\mathrm{min}}$ is the radius of resolution sphere in reciprocal space. Charge density pixels are often calculated on a finer grid so that contour maps look better. However, this only involves a larger region of unobserved structure factors in the calculation and does not provide more information.

The importance of finite resolution is best shown with a plot. We generated the structure factors for a typical organic structure (example 1. of Table 1.). These are the true complex amplitudes $F(\mathbf{k})$ and not just their moduli $F_{\mathrm{obs}}(\mathbf{k})$ which are used as observed data later. The resolution was set to $d_{\mathrm{min}} = 0.8\,\text{Å}$, structure factors outside the resolution sphere were treated as zeros. Then we calculated the real space charge density on a $0.4\,\text{Å}$ grid using the inverse FFT. Figure 1.a shows the sorted charge pixels. The main characteristics is the small number of large positive values. Most pixels are concentrated around zero, it is exactly this real space property which allows structure solution. Small negative values are naturally present because all observed and unobserved structure factors would be needed to generate a truly positive charge density. Behind the following algorithm the simple thought is: positivity should be forced with care, small negative charge density may help the process of structure solution.

3 The charge flipping algorithm

The structure factor moduli $F_{\mathrm{obs}}(\mathbf{k})$ are known for $0 < |\mathbf{k}| \equiv k \leq K$, these are the observed data used by the algorithm. Unobserved moduli are treated as zeros throughout the iteration process, except $F(0)$ which is initialized to zero but later let to change freely. The algorithm is initiated by selecting a random phase set $\{\varphi(\mathbf{k})\}$ which satisfies Friedel's law $\varphi(-\mathbf{k}) = -\varphi(\mathbf{k})$. Structure factor amplitudes are created as $F_{\mathrm{obs}}e^{i\varphi}$ and an inverse FFT gives a real charge density $\rho(\mathbf{r})$. This is our starting point in real space.

Then one cycle of iteration goes from real space through reciprocal to real space again according to the following scheme:

$$\begin{tikzcd} \rho \arrow[r, "\text{FLIP}"] & g \arrow[d, "\text{FFT}"] \\ F \arrow[u, "\text{FFT}^{-1}"] & G \arrow[l] \end{tikzcd} \tag{1}$$

The charge density modification $\rho(\mathbf{r}) \to g(\mathbf{r})$ starts the current iteration cycle. It uses a positive threshold δ for the charge pixels. The value of δ is a fraction of a typical light atom peak and is the only parameter of the algorithm. As the height of atomic peaks depends on the grid size, data resolution and thermal parameter, these factors also affect the choice of δ. Pixels above this value are accepted unchanged on the assumption that they belong to atomic peaks. Pixels below δ are simply multiplied by -1 which is made plausible later. In the next step temporary structure factors $G(\mathbf{k})$ are calculated by an FFT. Then structure factors $F(\mathbf{k})$ are constructed by accepting phases and replacing the moduli by $F_{\text{obs}}(\mathbf{k})$. $F(0) = G(0)$ is accepted as is without fixing or limiting its value, and $F(\mathbf{k})$ for $k > K$ are reset to zero. Finally the $F(\mathbf{k})$ amplitudes are inverted to obtain the new approximation of the charge density $\rho(\mathbf{r})$. This unconditional iteration process can continue without intervention, the traditional R-factor or some other figure of merit serves only for monitoring and not as the objective function of an optimization approach.

Symmetry is an important issue. Following positive experience [8, 9], we handle all structures in the spacegroup P1 and neglect any symmetry constraints. Accordingly, nothing fixes the origin which is an advantage, the structure can emerge anywhere. The disadvantage is that the charge density of the whole unit cell must be determined and not just that of the asymmetric unit. It turned out that the first factor is more important. When we applied the symmetry constraints of a given structure and thus forced the origin to a particular pixel of the unit cell, the success rate of the algorithm became much worse.

The algorithm is local in both spaces, modification of charge pixels and structure factors occurs only in-place. In real space only the charge pixels below $+\delta$ are modified which can be further divided into two parts. Large negative values below $-\delta$ are flipped simply to force positivity. More interesting is the $[-\delta, +\delta]$ range which is not negligible, it gives a substantial contribution to the structure factors. Figure 1.b shows the sorted charge pixels of a typical solution. The $[-\delta, +\delta]$ range is roughly linear which is an approximation of the target charge density. Flipping this region does not significantly change the distribution of pixels, but at the same time sufficiently explores the phase space. In reciprocal space the modification of structure factors corresponds to the unweighted F_{obs} map. The treatment of the unobserved $F(0)$ is less standard. While its value equals the total charge and could come from the chemical composition, we do not make use of it, keeping the algorithm ab initio in the strict sense. $F(0)$ is initialized to be zero and is let to change freely in the iteration cycles. In our studies this approach worked better than fixing the total charge. Note that in this simple scheme there is no reciprocal space weighting, no tangent formula and no use of probability. The concepts of atomicity and positivity are there but in a strange, indirect way.

During prolonged tests of the algorithm we realized that it is closely related to the solvent flipping method of Abrahams and Leslie [10, 11] used as density modification in protein crystallography. However, there are important differences. Solvent flipping – as other methods of density modification – is used for improving already existing phases and does not need atomic resolution data. In contrast, the charge flipping algorithm of this work is used ab initio and high resolution data is essential for its success. Solvent flipping requires the existence of separate solvent and protein regions. It modifies only the solvent charge density as $\rho^{\mathrm{new}} = \rho_0 + k_{\mathrm{flip}} \cdot (\rho - \rho_0)$, where ρ_0 is the expected solvent level and k_{flip} depends on the solvent content. In contrast, the charge flipping algorithm does not need separate real space regions, it is applied everywhere. There is no choice of ρ_0 or k_{flip}, the modification is always the sign change of pixels below the threshold parameter δ. The low density region occupies the space between atoms, which is automatically found and perpetually adjusted by the algorithm.

In the following two sections we first give several examples of ab initio structure solution using charge flipping and then discuss mathematical aspects of the algorithm in detail.

4 Structure solution examples

We tested the charge flipping algorithm on more than 200 structures taken from the Cambridge Structural Database. For this presentation we selected ten examples in the simplest centrosymmetric and non-centrosymmetric spacegroups each with a considerable number of atoms. The structures are listed in Table 1.

Table 1: Example structures. Columns: CSD code and original reference, spacegroup, number of non-hydrogen atoms and chemical formula per unit cell

	code and ref.	spgr.	N	unit cell content
1.	`feryoq` [12]	P$\bar{1}$	172	$2 \cdot C_{80}N_1O_5$
2.	`rawtoy` [13]	P$\bar{1}$	216	$2 \cdot C_{88}N_4O_{16}$
3.	`ibeyap` [14]	P$\bar{1}$	220	$2 \cdot C_{96}N_1O_{13}$
4.	`cotgib` [15]	P$\bar{1}$	244	$4 \cdot C_{53}Cu_1O_5P_2$
5.	`sisyey` [16]	P$\bar{1}$	326	$2 \cdot C_{98}Cl_2Mn_{12}N_1O_{50}$
6.	`valino` [17]	P1	156	$2 \cdot C_{54}N_6O_{18}$
7.	`pawveo` [18]	P1	164	$2 \cdot C_{72}N_4O_6$
8.	`gofmod` [19]	P1	188	$2 \cdot C_{77.5}N_4O_{12.5}$
9.	`qarpuu` [20]	P1	220	$2 \cdot C_{105}N_4Pd_1$
10.	`qibbuy` [21]	P1	240	$1 \cdot C_{181}Cl_{24}N_6O_{26}P_3$

In all cases we generated data up to 0.8 Å resolution using the coordinates and scattering factors of non-hydrogen atoms and without adding noise. Furthermore, we assumed that the absolute scale and the isotropic thermal parameter B are known. In practice these come from Wilson's plot. The knowledge of the absolute scale is not a serious issue, it is simply related to the proper choice of the δ parameter. With trial and error we can quickly find its realistic range and fine tuning is needed only for faster convergence. In our examples 15% accuracy of δ is sufficient. Using B=0 needs more consideration. Thermal vibration smears out the atomic charge density and weakens atomicity on which the algorithm is based. Therefore, it is strongly preferred to use low temperature data. How well the effect of B can be removed from the real data is beyond the scope of this paper but our numerical tests show that an error of $\sim 3\,\text{Å}^2$ can be tolerated.

All ten example structures were successfully solved using the charge flipping algorithm. The solution of each structure was attempted 100 times starting with different random phase sets and running the algorithm for a maximum of 5000 iteration cycles. The number of iterations leading to convergence greatly varies, only their distribution characterizes the difficulty of the problem. Solution statistics is compiled in Table 2.

Table 2: Solution statistics of the example structures. Columns: δ parameter in units of $6 \cdot V_{pixel}$, success rate, mean/minimum/maximum number of iterations

	δ	success	mean	min.	max.
1.	0.30	0.99	338	55	2005
2.	0.28	1.00	301	55	1650
3.	0.30	1.00	90	30	205
4.	0.37	1.00	101	15	230
5.	0.42	1.00	143	70	300
6.	0.28	0.95	1040	115	4220
7.	0.30	1.00	106	35	345
8.	0.30	1.00	268	40	1645
9.	0.42	1.00	198	75	690
10.	0.47	1.00	441	85	4115

It is informative to follow some basic quantities during the iteration. Figure 2. shows a typical run of example 1. The three subplots are: the total charge, the traditional R-factor and the phase-change. In all three quantities a sudden decrease starts at 210 iterations and ends after another 10 iterations. This sharp drop is an unmistakable sign of convergence and its width is independent whether it occurs after 10 or 10000 iterations. All curves show three different parts: an initial transient, a long stagnation period before the convergence and an equilibrium after. What really goes on in these periods is discussed in the next section.

Once a solution is found its quality must be evaluated. For this we locate the atoms by $3 \times 3 \times 3$ pixel peak picking and compare their number, centroid position and integrated weight to the original structure. The solutions are remarkably complete, all atoms of the original structure can be found. As we work in the spacegroup P1 the structure is always shifted relative to the original and for non-centrosymmetric structures the solution is often the enantiomer. When we check a large number of solutions the shift vector is uniformly distributed in the unit cell. After applying the shift and enantiomer correction the coordinates of non-hydrogen atoms are typically within 0.1 Å from the original structure and the integrated weight of a carbon atom scatters between 4 to 6. This is considered very good quality reconstruction, especially without the use of a separate refinement program.

5 Mathematical notes on the algorithm

Without giving a formal proof of convergence, we reconsider here some of the mathematical aspects of the iteration method described previously.

Clearly, a prerequisite of any ab initio structure solution is that apart from translations and point group transformations, the Fourier moduli determine a unique density. Thus, we suppose uniqueness and mention only one obvious condition of it. The density should not be strictly positive, otherwise any sufficiently small change in the phases which respects Friedel's law would lead to a different, nonnegative density.

Our method assumes that the density has extended regions of zeros. If the density

$$\rho_{\text{ideal}}(\mathbf{r}) = \frac{1}{V}\sum_{\mathbf{k}} F(\mathbf{k})e^{-i\mathbf{kr}}$$

has a sea of zeros then the value taken by the finite sum

$$\rho(\mathbf{r}) = \frac{1}{V}\sum_{k\leq K} F(\mathbf{k})e^{-i\mathbf{kr}}$$

is small positive or negative near this sea of ideal zeros. The discrete inverse Fourier transform provides a sampling of ρ at the centres of pixels and not an average over the volume of pixels. The oscillations around zero can be seen in the sampling, and the threshold $\delta > 0$ under which the sign flip is made has to be chosen in such a way that these small oscillations fall in the interval $[-\delta, +\delta]$. Therefore, the optimal choice of δ depends on the function to be determined. We demonstrated earlier that δ can be chosen without an *a priori* knowledge of ρ so that the algorithm converges, and for this reason δ should not be too small.

Given $\delta > 0$, we divide ρ in two parts, $\rho = \rho_1 + \rho_2$ with

$$\rho_1(\mathbf{r}) = \begin{cases} \rho(\mathbf{r}) & \text{if} \quad \rho(\mathbf{r}) \geq \delta \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and

$$\rho_2(\mathbf{r}) = \begin{cases} \rho(\mathbf{r}) & \text{if} \quad \rho(\mathbf{r}) < \delta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Since the input data are $F_{\text{obs}}(\mathbf{k}) \equiv |F(\mathbf{k})|$ for $0 < k \leq K$, the target function cannot be ρ_{ideal} but only ρ, including the total charge

$$F(0) = \int \rho_{\text{ideal}}\,\mathrm{d}\mathbf{r} = \int \rho\,\mathrm{d}\mathbf{r}$$

The iteration will generate $F(0)$, although with a limited precision because it sensitively depends on the choice of δ. If we are given $\rho_{\rm ideal}(\mathbf{r}) - \frac{1}{V}F(0)$, the charge $F(0)$ can be found by knowing that the flat regions of $\rho_{\rm ideal}$ have to be at zero level. No similar information about $F(0)$ can be used if we know only $\rho(\mathbf{r}) - \frac{1}{V}F(0)$. The best we can hope for is to reproduce ρ_1 whose mere definition depends on the positivity of δ. Another fundamental reason to work with $\delta > 0$ is that for any set of phases $\{\varphi(\mathbf{k})\}$ and a constant c large enough

$$\sum_{0<k\leq K} F_{\rm obs}(\mathbf{k})e^{i(\varphi(\mathbf{k})-\mathbf{kr})} + c \geq 0$$

This means that any set of phases $\{\varphi(\mathbf{k})\}$ is a fixed point of the iteration if $\delta = 0$ and $F(0)$ is not prescribed.

The scheme of iteration has been given in equation (1). More precisely, we do the following:

0th half-cycle.

We choose $\varphi^{(0)}(\mathbf{k})$ for (the half of) $\mathbf{k}$ with $0 < k \leq K$ independently, according to the uniform distribution in $[0, 2\pi]$. Then

$$F^{(0)}(\mathbf{k}) = \begin{cases} F_{\rm obs}(\mathbf{k})e^{i\varphi^{(0)}(\mathbf{k})} & \text{for } \ 0 < k \leq K \\ 0 & \text{for } \ \mathbf{k} = 0 \ \text{ and } \ k > K. \end{cases} \tag{4}$$

and by inverse FFT we compute $\rho^{(0)}$ determined in pixels $\mathbf{r} = \mathbf{r}_j$.

nth cycle ($n \geq 1$).

Given $\rho^{(n-1)}$, we divide it in two parts,

$$\rho^{(n-1)} = \rho_1^{(n-1)} + \rho_2^{(n-1)} \tag{5}$$

as in (2) and (3), and execute the sign flip on $\rho^{(n-1)}(\mathbf{r}_j) < \delta$ to obtain

$$g^{(n)} = \rho_1^{(n-1)} - \rho_2^{(n-1)} \ . \tag{6}$$

The Fourier transform of $g^{(n)}$ provides $G^{(n)}(\mathbf{k})$ for as many $\mathbf{k}$ as the number of pixels in the unit cell. Then

$$F^{(n)}(\mathbf{k}) = \begin{cases} F_{\rm obs}(\mathbf{k})G^{(n)}(\mathbf{k})/|G^{(n)}(\mathbf{k})| = F_{\rm obs}(\mathbf{k})e^{i\varphi^{(n)}(\mathbf{k})} & \text{for } \ 0 < k \leq K \\ G^{(n)}(0) & \text{for } \ \mathbf{k} = 0 \\ 0 & \text{for } \ k > K. \end{cases} \tag{7}$$

and through inverse FFT we find the next approximation $\rho^{(n)}(\mathbf{r})$ of the density in pixels $\mathbf{r} = \mathbf{r}_j$.

The real space transformation $\rho^{(n-1)} \rightarrow g^{(n)}$ is

(i) non-invertible, i.e. one cannot reproduce $\rho^{(n-1)}$ from $g^{(n)}$

(ii) norm-preserving

$$\sum_j g^{(n)}(\mathbf{r}_j)^2 = \sum_j \rho^{(n-1)}(\mathbf{r}_j)^2 \tag{8}$$

(iii) local in the sense that $g^{(n)}(\mathbf{r}_j)$ depends only on $\rho^{(n-1)}(\mathbf{r}_j)$.
Together with (8) this implies $|g^{(n)}(\mathbf{r}_j)| = |\rho^{(n-1)}(\mathbf{r}_j)|$. Its most important characteristics is, however

(iv) sign change in a broad region of the unit cell, $\rho^{(n-1)}(\mathbf{r}_j) < \delta$ occurs for the majority of pixels.

Locality in real space also means no charge displacement, implying that the position of the density evolves freely. This has consequences on the transformation we do in reciprocal space. Since the real space modification makes no use of symmetries, symmetry constraints on the phases of the Fourier components are not helpful or counter-productive. As we see from equation (7), the transformation $G^{(n)} \rightarrow F^{(n)}$ in reciprocal space is also local and non-invertible but does not preserve the norm

$$\sum_{\mathbf{k}} |F^{(n)}(\mathbf{k})|^2 \neq \sum_{\mathbf{k}} |G^{(n)}(\mathbf{k})|^2 \tag{9}$$

until convergence has not reached. After convergence, instead of (9) equality will hold, but not term by term. The step from $g^{(n)}$ to $G^{(n)}$ always creates nonzero Fourier components for $k > K$, while $F^{(n)}(\mathbf{k}) = 0$ for $k > K$. This means that

$$\sum_{0<k\leq K} |G^{(n)}(\mathbf{k})|^2 < \sum_{0<k\leq K} |F^{(n)}(\mathbf{k})|^2 = \sum_{0<k\leq K} F_{\text{obs}}(\mathbf{k})^2 \tag{10}$$

and therefore

$$|G^{(n)}(\mathbf{k})| < |F^{(n)}(\mathbf{k})| = F_{\text{obs}}(\mathbf{k}) \tag{11}$$

for the largest structure factors, dominating the sum of the squares.

We emphasize that in the present algorithm convergence means reaching a limit cycle and not a fixed point, in the sense that $|G^{(n)}(\mathbf{k})|$ becomes independent of n, but $\varphi^{(n)}(\mathbf{k})$ alternates between two values according to the parity of n. In Figure 3. we plotted the evolution of $G^{(n)}(\mathbf{k})$ in the complex plane for a few strong reflections. This is more spectacular than the evolution of $F^{(n)}(\mathbf{k})$ which stays on the circle of radius $F_{\text{obs}}(\mathbf{k})$. Equation (5) implies

$$F^{(n-1)} = F_1^{(n-1)} + F_2^{(n-1)} \tag{12}$$

where $F_i^{(n-1)}$ is the Fourier transform of $\rho_i^{(n-1)}$. Thus,

$$G^{(n)} = F_1^{(n-1)} - F_2^{(n-1)} . \tag{13}$$

According to (7), $F^{(n)}$ is a functional of $F_1^{(n-1)} - F_2^{(n-1)}$, but not of $F^{(n-1)}$ which is not uniquely determined by $F_1^{(n-1)} - F_2^{(n-1)}$. When convergence sets in $F_1^{(n)}(\mathbf{k})$ becomes nearly independent of n, while $F_2^{(n)}(\mathbf{k})$ alternates between two nearly collinear vectors in the complex plane which are nearly orthogonal to their respective $F_1^{(n)}(\mathbf{k})$. That collinearity and orthogonality are imperfect is due to (11) and (13). As a result, we can see an even-odd alternation of $G^{(n)}(\mathbf{k})$ *inside* the circle of radius $F_{\mathrm{obs}}(\mathbf{k})$, as shown in Figure 4.

If the number of independent phases is N, the algorithm has to find one of the good phase sets in the N-dimensional real space $\mathbb{R}^N$, where *good* means reproducing a translate of ρ_1. Any good set is represented by a point in $\mathbb{R}^N$, and these points form a three- dimensional manifold S having one or several connected components. If $\rho(\mathbf{r}) \neq \rho(-\mathbf{r})$ then their respective translates generate different connected sets. $\Phi = \{\varphi(\mathbf{k})\}$ and $\Psi = \{\psi(\mathbf{k})\}$ are in the same component if they are connected by a space translation,

$$\psi(\mathbf{k}) = \varphi(\mathbf{k}) + \mathbf{k} \cdot \mathbf{a}$$

for some $\mathbf{a} \in \mathbb{R}^3$. Phase retrieval is done in the cube $C = [0, 2\pi]^N$, and hence parts of S outside this cube have to be shifted back into it by subtracting integer multiples of 2π. Then even the connected components of S fall into three dimensional 'filaments' starting and ending on the surface of the cube, and our algorithm has to converge to a point of one of the filaments. Although S is infinite, when shifted back into C, the filaments do not fill densely the cube, otherwise any choice of the phases would do. Other than a circumstantial evidence of this fact can be obtained by noting that for any four vectors $\mathbf{k}_1, \ldots, \mathbf{k}_4$ and any $\mathbf{a} \in \mathbb{R}^3$ the numbers $\mathbf{k}_i \cdot \mathbf{a}$ are rationally dependent, i.e. $\sum_{i=1}^4 m_i \mathbf{k}_i \cdot \mathbf{a} = 0$ for suitably chosen integers m_i. Also, there is a large number of shift-invariant quantities formed by the coordinates of points of S, namely, if $\mathbf{k}' = m\mathbf{k}$ for some integer m then

$$\varphi(\mathbf{k}') - m\varphi(\mathbf{k})$$

is shift-invariant.

Having no *a priori* information about the position of the filaments, a random initial set of phases seems to be a good choice to start with. This is even more so, because starting with a good phase set the iteration leaves the neighbourhood of this point and returns to another one after convergence. Indeed, starting with the good phases implies $\rho^{(0)} = \rho - \frac{1}{V}\int \rho \, \mathrm{d}\mathbf{r}$. Then in real space, instead of shifting this function upwards, we start to flip the values below δ and the iteration leads farther away from ρ before it approaches again a translate of it. That it does, is due to property (iv) of the real space transformation which invokes a wide exploration of C. We note that there is no attraction along the filaments, because the algorithm makes no preference in the position of the sample. It takes a while to reach the basin of attraction of one of the filaments, showing that their complement in C has to be a large set of a complicated structure, similar to a 'strange repellent' of a chaotic system. However, in the neighbourhood of a solution convergence sets in like an avalanche, it is complete after a few steps of iteration. The iteration process is chaotic, in the usual sense of sensitive dependence on initial conditions, which are the starting phases in our case. An obvious sign is that the number of iteration cycles leading to convergence varies wildly with the slightest change of a given random phase set. Chaotic behaviour comes from two sources. The topological reason is the filamentation of the set of points to be attained in the very high dimensional cube of phases. The algorithmic reason is that successive points to be visited in C are decided by charge flipping in real space. Although this is done in a deterministic way, during the long stagnation period it appears very much like random jumps in the space of phases.

It is instructive to follow the evolution of the total charge $F^{(n)}(0)$ plotted in Figure 2.a. It starts with zero, in the first step of the iteration jumps to a high positive value, which is followed by a rapid decay to an intermediate value, a long stagnation, and a second abrupt drop to the final number. Its evolution follows rather closely that of the R-factor,

$$R^{(n)} = \frac{\sum_{0<k\le K} ||G^{(n)}(\mathbf{k})| - F_{\text{obs}}(\mathbf{k})|}{\sum_{0<k\le K} F_{\text{obs}}(\mathbf{k})} \tag{14}$$

shown in Figure 2.b. The distance

$$d(\Phi^{(n+2)}, \Phi^{(n)}) = \frac{\sum_{0<k\le K} F_{\text{obs}}(\mathbf{k})|(\varphi^{(n+2)}(\mathbf{k}) - \varphi^{(n)}(\mathbf{k}))(\text{mod } 2\pi)|}{\sum_{0<k\le K} F_{\text{obs}}(\mathbf{k})} \tag{15}$$

exhibits a similar behaviour (Figure 2.c). We recall that the R-factor is used only for monitoring the convergence, the fact that it does not tend to zero is due to (11), and has no bearing on the success of the iteration.

The initial overshooting of $F^{(1)}(0)$ for most of the starting phase sets can be understood as follows. Since

$$G^{(0)}(0) = F^{(0)}(0) = 0$$

and

$$F_1^{(0)}(0) = \int \rho_1^{(0)} \mathrm{d}\mathbf{r} = -\int \rho_2^{(0)} \mathrm{d}\mathbf{r} > 0$$

is the charge carried by the pixels in which $\rho^{(0)}(\mathbf{r}_j) > \delta$, we have

$$G^{(1)}(0) = F^{(1)}(0) = 2F_1^{(0)}(0) \ .$$

No similar charge doubling occurs in the subsequent steps, because typically

$$\int \rho_1^{(n)} \mathrm{d}\mathbf{r} \gg -\int \rho_2^{(n)} \mathrm{d}\mathbf{r}$$

for $n > 0$. For the example in Figure 2. the actual limit reached by $G^{(n)}(0)$ is about 30% lower than the true value of $F(0)$. In principle, a smaller δ could yield the correct value, but in practice we would not obtain convergence with a smaller δ. After convergence sets in, δ can be decreased and the iteration continued without destroying the result. A new limit cycle will be attained with suppressed oscillations and a higher total charge. This can go on down to $\delta = 0$, where the sea of zeros will be lost and the total charge will become too large.

6 Conclusion

In this paper we presented an ab initio structure solution method termed *charge flipping.* It uses high resolution data and alternates between real and reciprocal space in the manner of Fourier recycling. The real space modification simply changes the sign of the charge density below a threshold δ, which is the only parameter of the algorithm. In reciprocal space, observed moduli are constrained using the unweighted $F_{\rm obs}$ map while $F(0)$ corresponding to the total charge is allowed to change freely. All structures were handled in spacegroup P1 neglecting any symmetry constraints.

We tested the algorithm using 0.8 Å resolution synthetic data for a wide range of centrosymmetric and non-centrosymmetric structures taken from the Cambridge Structural Database. The size of our examples is somewhere at the upper limit of small molecule structures in spacegroups $\mathrm{P}\bar{1}$ and P1. While ideal data are not required for the working of the method, at this stage of development we needed a large pool of various structures and we had to be sure that any difficulty of solution is not the consequence of data quality, resolution or completeness. For the examples presented we followed single runs, analysed solution statistics of multiple runs and checked the quality of reconstruction. Here we only emphasize that all structures were solved with a high success rate, and all atoms were found without the use of a separate refinement program.

Finally, mathematical aspects of the iteration process were considered in detail. We clarified the most important properties of real and reciprocal space transformations, discussed the relevance of finite resolution and the choice of δ. By following the evolution of structure factors we also showed that the iteration process is chaotic and the solution is not a fixed point but a limit cycle.

The most important characteristics of the algorithm is its amazing simplicity. This is a big advantage for exact mathematical treatment, in the future we shall attempt to give a formal proof of convergence. When it comes to efficiency, such a simple method is likely to lag behind today's best programs. As long as symmetry is not helpful, charge flipping offers only an interesting alternative in the low symmetry spacegroups. Nevertheless, we encourage everyone to try it, it is a few lines of code plugged in an existing program, and only a few hundred lines as a stand alone application. We shall also continue our work to check the power of the algorithm on real data, and to make a fair comparison of success ratios to other methods. We have well defined plans to improve the algorithm but anticipate that the original simplicity will be lost in exchange for higher efficiency.

We thank Gyula Faigel, Miklós Tegze and Gábor Bortel for useful discussions. This research was supported by OTKA grants T043494 and T042914 and the work of G.O. was also funded by a Bolyai János Scholarship.

References

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

[2] J.R. Fienup, *Applied Optics* **21**, 2758–2769 (1982)

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

[4] R. Miller, G.T. DeTitta, R. Jones, D.A. Langs, C.M. Weeks, H.A. Hauptman, *Science* **259**, 1430–1433 (1993)

[5] G.M. Sheldrick, *Direct Methods for Solving Macromolecular Structures*, edited by S. Fortier, Dordrecht: Kluwer, pp. 401-411 (1998)

[6] L.D. Marks, W. Sinkler E. Landree, *Acta Cryst.* A**55**, 601–612 (1999)

[7] V. Elser, *Acta Cryst.* A**59**, 201–209 (2003)

[8] G.M. Sheldrick, R.O. Gould, *Acta Cryst.* B**51**, 423–431 (1995)

[9] M.C. Burla, B. Carrozzini, G.L. Cascarano, C. Giacovazzo, G. Polidori, *J. Appl. Cryst.* **33**, 307–311 (2000)

[10] J.P. Abrahams, A.W.G. Leslie, *Acta Cryst.* D**52**, 30–42 (1996)

[11] J.P. Abrahams, *Acta Cryst.* D**53**, 371–376 (1997)

[12] H. Irngartinger, A. Weber, T. Oeser, *Angew. Chem. Int. Ed.* **38**, 1279–1281 (1999)

[13] E. Nomura, M. Takagaki, C. Nakaoka, H. Taniguchi, *J. Org. Chem.* **65**, 5932–5936 (2000)

[14] R. Wang, J. Yang, Z. Zheng, M.D. Carducci, J. Jiao, S. Seraphin, *Angew. Chem. Int. Ed.* **40**, 549–551 (2001)

[15] É. Balogh–Hergovich, J. Kaizer, G. Speier, V. Fülöp, L. Párkányi, *Inorg. Chem.* **38**, 3787–3795 (1999)

[16] Z. Sun, D. Ruiz, E. Rumberger, C.D. Incarvito, K. Folting, A.L. Rheingold, G. Christou, D.N. Hendrickson, *Inorg. Chem.* **37**, 4758–4759 (1998)

[17] I.L. Karle, *J. Am. Chem. Soc.* **97**, 4379–4386 (1975)

[18] S. Pakhomova, J. Ondracek, J. Hodacova, I. Stibor, *Z. Kristallogr. – New Cryst. Struct.* **212**, 461–463 (1997)

[19] K. Biradha, D. Dennis, V.A. MacKinnon, C.V.K. Sharma, M.J. Zaworotko, *J. Am. Chem. Soc.* **120**, 11894–11903 (1998)

[20] T. Ishii, N. Aizawa, M. Yamashita, H. Matsuzaka, T. Kodama, K. Kikuchi, I. Ikemotoa, Y. Iwasa, *J. Chem. Soc., Dalton Trans.*, 4407-4412 (2000)

[21] J. Lacour, A. Londez, C. Goujon–Ginglinger, V. Buss, G. Bernardinelli, *Organic Letters* **2**, 4185–4188 (2000)

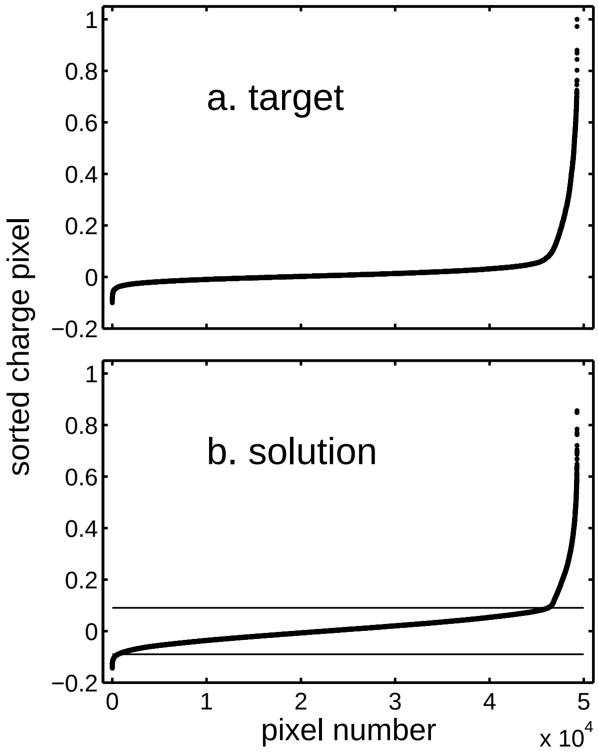

Figure 1: Charge density of a typical organic structure at 0.8 Å resolution. a: target charge density, b: solution after convergence. Pixel values are sorted in ascending order and are normalized to the maximum of the target. Note that the $[-\delta, +\delta]$ range within horizontal lines is approximately linear.

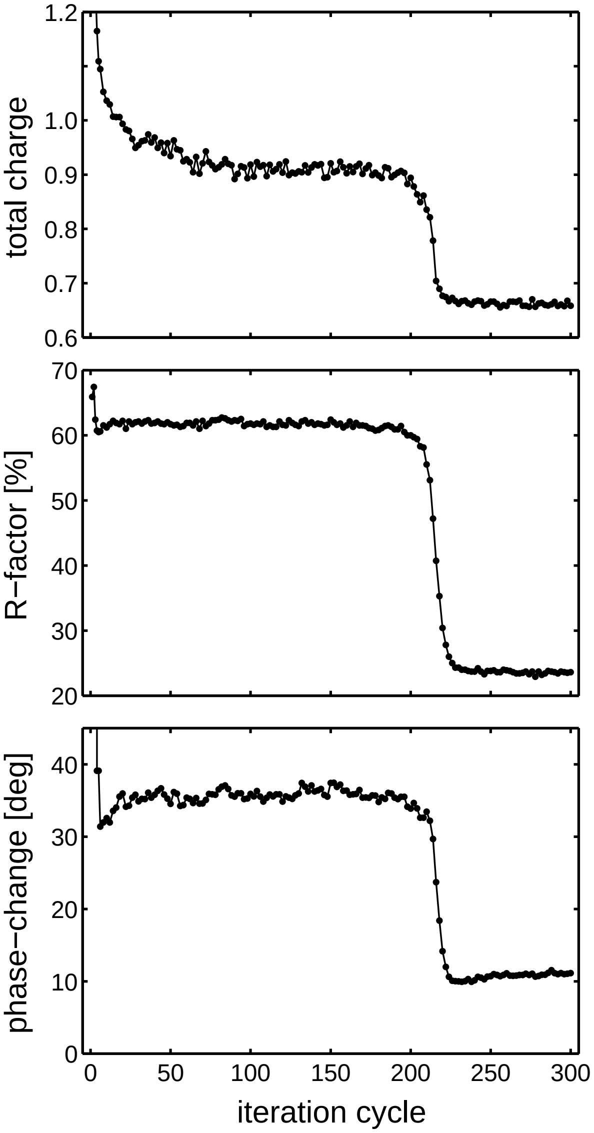

Figure 2: A typical run of the charge flipping algorithm leading to convergence. From top to bottom: total charge, R-factor and phase-change as a function of the iteration cycle. The total charge is normalized by its ideal finite resolution value. The R-factor and phase-change are defined by equations (14) and (15).

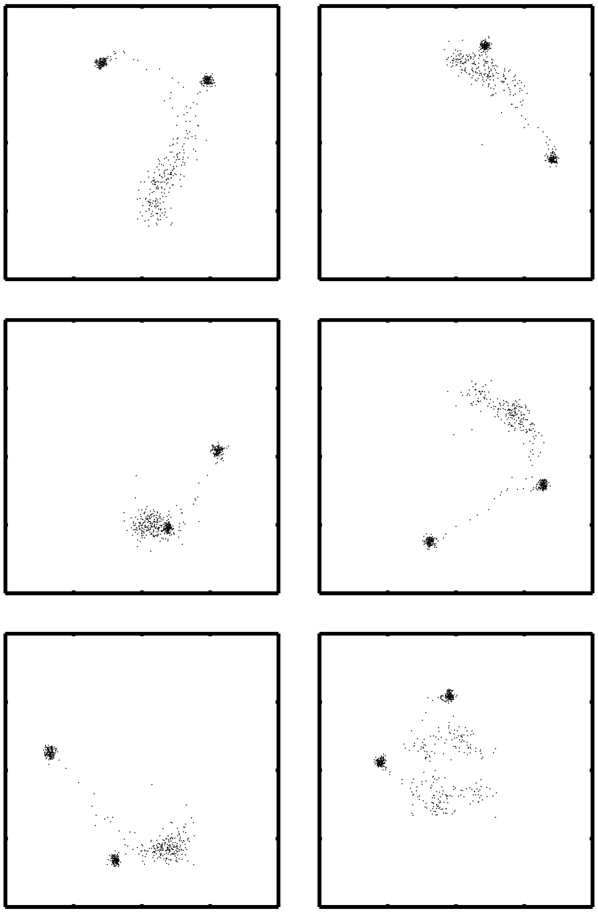

Figure 3: The evolution of the structure factor amplitude $G^{(n)}(\mathbf{k})$ in the complex plane for a few strong reflections. In each case it alternates between two values after convergence, showing that the solution is a limit cycle.

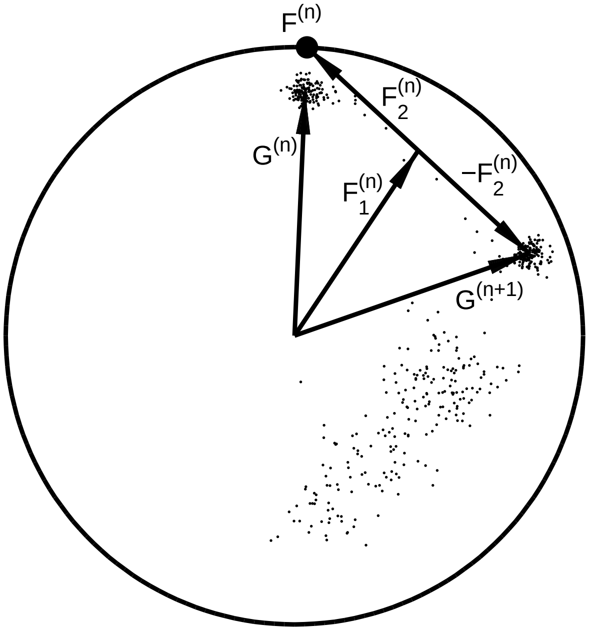

Figure 4: The evolution of a selected structure factor amplitude $G^{(n)}(\mathbf{k})$ in the complex plane. Arrows explain one cycle of iteration from $G^{(n)}$ to $G^{(n+1)}$ through $F^{(n)}{=}F_1^{(n)} + F_2^{(n)}$. The radius of the circle is $F_{\rm obs}(\mathbf{k})$.