

Article

Stochastic and Deterministic Crystal Structure Solution Methods in GSAS-II: Monte Carlo/Simulated Annealing Versus Charge Flipping

Robert Von Dreele

Advanced Photon Source, Argonne National Laboratory, Lemont, IL 60439, USA; vondreele@anl.gov;
Tel.: +1-630-252-8178

Academic Editor: Silvina Pagola

Received: 28 June 2017; Accepted: 22 August 2017; Published: 29 August 2017

Abstract: Crystallographic studies frequently involve the determination of a previously unknown crystal structure; General Structure Analysis System (GSAS)-II provides two methods for this purpose. The Monte Carlo/simulated annealing method is fundamentally stochastic in nature; random trials are tested for suitability by comparing calculated structure factors with a suite of observed ones. In contrast, the charge flipping method may begin with a suite of random structure factor phases, but the subsequent mathematical steps are entirely deterministic even though they appear to display chaotic behavior. This paper will briefly describe these methods as implemented in GSAS-II, illustrating their use with examples.

Keywords: structure solution; charge flipping; Monte Carlo/simulated annealing; chaos mathematics; GSAS-II

1. Introduction

One of the goals in developing GSAS-II [1,2] was to expand from the capabilities of the original General Structure Analysis System (GSAS) which largely encompassed just structure refinement and post refinement analysis. GSAS-II has been written almost entirely in Python loaded with graphics, GUI, and mathematical packages (matplotlib, pyOpenGL, wxpython, numpy, and scipy). Thus, GSAS-II has a fully developed modern GUI as well as extensive graphical display of data and results. However, the structure and operation of Python has required new approaches to many of the algorithms used in crystal structure analysis. The extensions beyond GSAS include image calibration/integration as well as peak fitting and unit cell indexing for powder data which are precursors for structure solution. Structure solution within GSAS-II begins with either Pawley or LeBail extracted structure factors from powder data or those measured in a single crystal experiment. Both charge flipping (CF) and Monte Carlo-simulated annealing (MCSA) techniques are available; the former can be applied to (3 + 1) incommensurate structures as well as conventional 3D structures. This paper will briefly describe these methods as implemented in GSAS-II, illustrating their use with examples.

2. Methods

2.1. Monte Carlo/Simulated Annealing

The MCSA method in GSAS-II follows procedures similar to those implemented in a number of other MCSA programs (e.g., PSSP [3], FOX [4], DASH [5]) and is a local modification of the (now deprecated) scipy.optimize.anneal [6] routine. A set of structure factors is obtained from the powder pattern via either Pawley [7] or LeBail [8] refinement within GSAS-II; these are typically limited in d-spacing to be greater than 2 Å. If the Pawley method is used, then a covariance matrix (C) is

available that can be used in the MCSA calculations (see below, Equation (4)). Otherwise, a covariance matrix is created from the reflection widths and the reflection separations in the powder pattern, widely separated reflections would have a zero covariance while ones that nearly overlap would have a large covariance. For single crystal data, this is a unit matrix since there is no reflection overlap. A molecular model constructed with Cartesian coordinates via a program such as Avogadro [9] is also required; variable chain torsion angles can be included along with molecular position and orientation in the parameter suite to be explored by MCSA. For example, in the molecule D-lactose (Figure 1) there are four internal torsion angles that can be included in the MCSA model in GSAS-II.

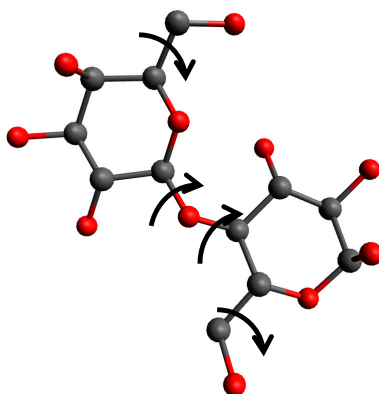


Figure 1. The molecular model for D-lactose; hydrogen atoms are not shown. The four variable torsion angles are indicated. C atoms are shown in gray and O atoms in red.

In most MCSA programs (e.g., PSSP), the molecular orientation is described by Eulerian angles (Ω , X , Φ) which, if randomly chosen, will concentrate orientations with $X \sim 0^\circ$ or $\sim 180^\circ$ because for these values $\Omega \sim \Phi$. To avoid this, GSAS-II describes the molecular orientation with a quaternion (Equation (1)) which gives an even distribution of trial orientations.

$$Q_{ijk} = r + ai + bj + ck \quad (1)$$

After normalization (Equation (2)) of the quaternion, the a , b , c components indicate a Cartesian space vector and r gives the cosine of $\frac{1}{2}$ the angle of rotation about this vector from the reference orientation.

$$r^2 + a^2 + b^2 + c^2 = 1 \quad (2)$$

The model coordinates (x) obtained after applying any internal torsion rotations to the original molecular model are transformed by the quaternion operation via the Hamilton product $x' = QxQ^{-1}$ which in matrix form is

$$x' = 2Mx + x, \text{ where } M = \begin{bmatrix} -b^2 - c^2 & ab - rc & rb + ac \\ rc + ab & -a^2 - c^2 & bc - ra \\ ac - rb & ra + bc & -a^2 - b^2 \end{bmatrix} \quad (3)$$

followed by transformation from Cartesian space to crystallographic space. Finally, the molecular position is applied to obtain the set of test atom coordinates. These are used in a simplified structure factor calculation (no thermal motion contribution) for the chosen reflection suite; a residual (R) is computed from the difference (ΔI) and its transpose $(\Delta I)^T$ from the observed intensities ($I_o = m \cdot F_o$, m = reflection multiplicity) via

$$R = \sqrt{\frac{\sum (\Delta I) C (\Delta I)^T}{\sum I_o^2}} \quad (4)$$

which is similar to that used in crystallographic refinements and covers the range 0.1–0.6 for most MCSA runs. The MCSA procedure in GSAS-II proceeds by first generating a set of random values in the parameter space (in the case of D-lactose this includes just x and z coordinates of the molecule in the unit cell since y is arbitrary for its space group, $P2_1$, the four components of the quaternion and four torsions) all restricted by their respective ranges, applying it to the molecular model and then calculating the structure factors and the residual.

Instead of using a flat random distribution of values across the parameter ranges, GSAS-II uses a temperature dependent distribution (Figure 2) that is peaked at half the range which is more pronounced as the temperature decreases. This provides more chances for major model changes than a flat distribution. The residual (Equation (4)) is subject to a Metropolis test [10] based on the temperature to determine if the move is retained. To repeat the process, a new set of parameter values is randomly chosen using the distribution in Figure 2. This is repeated a sufficient number of times (perhaps 10^3 or more) to ensure sufficient coverage of the parameter space to locate a point in the vicinity of the global solution. The temperature that controls the stiffness of the Metropolis test follows a cooling schedule to improve the probability of locating the global solution. Two cooling schedules are available: “log” (Equation (5)), where the temperature changes are dependent on a slope (s) and a temperature schedule increment (k)

$$T_k = T_0 s^k \quad (5)$$

and “fast” (Equation (6)), where two parameters (c, q) allow rapid cooling at the start and then slower cooling near the finish of the sequence.

$$T_k = T_0 \exp(-c k^q) \quad (6)$$

Typical cooling schedules are shown in Figure 3; a useful temperature range matches that of the residual (initial and final MCSA temperatures, $T_0 = 0.6$ to $T_f = 0.1$, respectively). A cooling sequence is terminated when $T_k < T_f$.

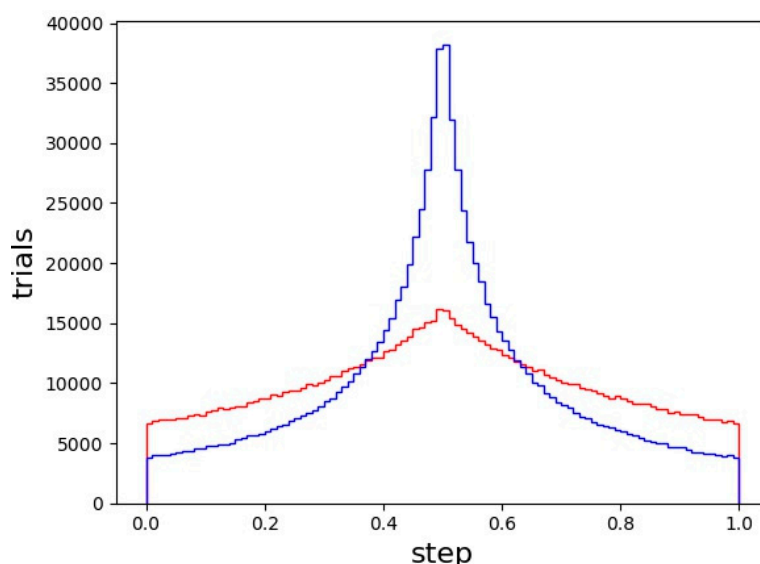


Figure 2. Temperature dependent random step distribution used for MCSA calculations in GSAS-II for $T = 0.7$ (red) and $T = 0.1$ (blue) for 1,000,000 trials. The steps are then scaled to the parameter ranges to give the values used in each MCSA trial.

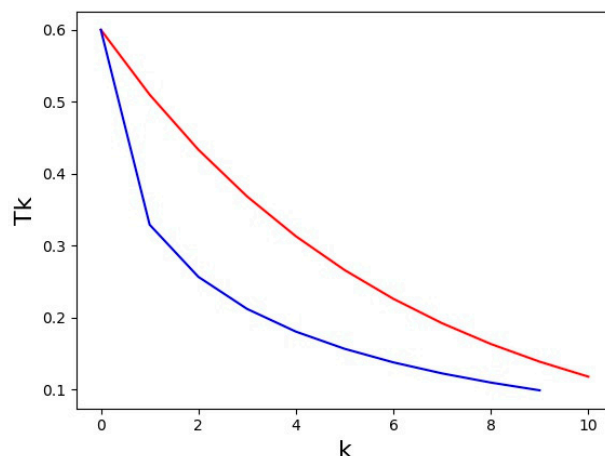


Figure 3. Cooling schedules for “log” (red, Equation (5), $s = 0.8$) and “fast” (blue, Equation (6), $c = 0.6$, $q = 0.5$) starting at $T_0 = 0.6$.

The model obtained at the end of a cooling sequence is then subjected to a bounded Broyden-Fletcher-Goldfarb-Shanno (BFGS) [11] local minimization to further improve the result. The cooling sequence is repeated several times and the best result, subject to user evaluation, is chosen as the solution to the crystal structure. In GSAS-II, the next steps are a least-squares refinement of this model against all the observed diffraction data.

2.2. Charge Flipping

The CF method in GSAS-II is essentially that originally proposed by Oszlányi and Sütő [12,13]. The set of structure factors extracted from a powder pattern by either the Pawley [7] or LeBail [8] methods is expanded to a complete sphere of reflections. In contrast to the MCSA method, for CF it is generally required that the structure factors extend to essentially “atomic resolution” with the minimum d-spacing of ~ 1 Å. To achieve a real space resolution of ~ 0.5 Å over the unit cell, the reflection set is further expanded to encompass a parallelepiped with the number of reflections matching the number of points in the real space unit cell. These additional reflections, as well as any missing ones (including space group extinctions), are zeroed. In this process, adjustments are made in the real space grid so that points fall on possible symmetry elements. Finally, a random set of phases are assigned to the expanded reflection set (CEhkl).

In Python the charge flip algorithm then consists of very few lines (Table 1) that are repeated until the user terminates the process usually when convergence has been noted in the displayed residuals (or 10,000 trials without user intervention). All calculations use the default (64 bit) Python precision.

Table 1. Charge flipping as implemented in numpy.fft/python in GSAS-II.

```

CErho = np.real(fft.fftn(fft.fftshift(CEhkl)))*(1. + 0j)
CEsig = np.std(CErho)
CFrho = np.where(np.real(CErho) >= flipData['k-factor']*CEsig, CERho, -CERho)
CFrho = np.where(np.real(CErho) <= flipData['k-Max']*CEsig, CFRho, -CFrho)
CFhkl = fft.ifftshift(fft.ifftn(CFRho))
CFhkl = np.where(CFhkl, CFhkl, 1.0)
phase = CFhkl/np.absolute(CFhkl)
CEhkl = np.absolute(CEhkl)*phase
Ncyc += 1
sumCF = np.sum(ma.array(np.absolute(CFhkl), mask = Emask))
DEhkl = np.absolute(np.absolute(Ehkl)/sumE - np.absolute(CFhkl)/sumCF)
Rcf = min(100., np.sum(ma.array(DEhkl, mask = Emask)*100.))

```

In the first step, the real space density (CERho) is formed via fast Fourier transform of the fully expanded set of structure factors (CEhkl); the fftshift operation accomodates the origin shift in reciprocal space from the center to one corner of the Python/numpy array (python does not permit negative array indices). The second step determines the standard deviation (CEsig) of the map which is used in the third step to determine the level in the map below which all densities are flipped in sign. The fourth step imposes a flip of densities that exceed an upper limit, thus reducing the chance of obtaining “U-atom” solutions where all the density is concentrated in a single position. Step five computes the reverse fast Fourier transform (CFhkl) and step six sets any accidental zeros to unity, thus avoiding zero divides in recovering the phase (step 7). Finally, the new phases are applied to the CEhkl; followed by a few lines needed to form the displayed residual. The algorithm then goes back to step 1 unless the user has intervened.

At the completion of the CF sequence, a new density map is prepared. Since the original CF map is arbitrarily located, GSAS-II evaluates space group equivalent reflection phases to determine the origin offset. Then the map is searched for peak positions; the user then identifies them by chemical knowledge to give a structural model suitable for a least-squares refinement against the observed diffraction data.

3. Discussion

3.1. Monte Carlo/Simulated Annealing

As a demonstration of the MCSA method as implemented in GSAS-II, we consider the structure determination of cimetidine ($C_{10}H_{16}N_6S$, $a = 10.3936 \text{ \AA}$, $b = 18.8176 \text{ \AA}$, $c = 6.8249 \text{ \AA}$, $\beta = 106.44$, $P2_1/a$, $Z = 4$) from synchrotron powder diffraction data obtained from a structure determination from powder diffraction data (SDPD) test suite [14] which has been used in a number of SDPD computer program tutorials (see [14] for details). A Pawley [7] refinement for the entire powder diffraction pattern yielded intensities for 924 reflections ($d_{\min} = 1.132 \text{ \AA}$, $R_{wp} = 6.31\%$). A model for cimetidine was developed using program Avogadro [9], which provides a simple energy minimization and molecular geometry optimization for the molecular structure. The hydrogens were stripped from the molecule (to save computer time in the MCSA calculations); 7 torsions (see Figure 4), in addition to (3) position and (4) orientation parameters gave 14 adjustable parameters to be determined by MCSA.

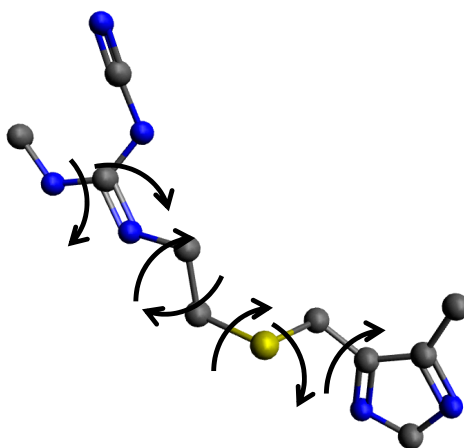


Figure 4. The molecular model for cimetidine; hydrogen atoms are not shown. The seven variable torsion angles are indicated. C atoms are shown in gray, N in blue, and S in yellow.

The variation ranges for the 14 adjustable parameters were developed to limit the MCSA search to the minimum required to (ideally) yield a single solution. Thus, the molecular position is limited to account for both the unique part of the unit cell and the effects of alternate origin choices (four inversion centers in $P2_1/c$); this gives $0-1/4$, $0-1/4$, $0-1/2$ for molecular position. The quaternion

(Equation (1)) and its inverse describe the same orientation so the range on r is $180\text{--}360^\circ$. The torsion angles have no limits so their ranges are $0\text{--}360^\circ$.

A set of 256 temperature schedules (“fast” as shown in Figure 3 comprising 10 steps) using 62 reflections ($d > 2.8 \text{ \AA}$) were successful in one instance ($R = 0.13$) in giving the correct structure (Figure 5) for cimetidine; the other 255 solutions gave poorer residuals and chemically unreasonable solutions (commonly with intra- and intermolecular clashes). At each temperature step, 1000 MCSA trials were used; on average approximately 65% passed the Metropolis test. The successful run gave a MCSA residual $R = 44.8\%$ which was improved to 13.0% by the BFGS refinement. A similar but failed run gave an MCSA residual $R = 45.5\%$ but the BFGS refinement only gave $R = 34.4\%$ and a poor structure. Clearly, using a BFGS refinement at the end of each temperature run enhances the performance of the MCSA in GSAS-II. Selection of the d_{\min} (2.8 \AA in this case) should be done with some care. If it is too high there is not enough information to discern between molecular orientations or side chain torsions that yield similar atom arrangements on the d_{\min} resolution scale. If it is set too small, then the global minimum is too sharply defined to be easily found by random MCSA trials or the BFGS refinement. Moreover, the larger reflection suite with a smaller d_{\min} requires more computer time for the MCSA structure factor calculations and increased reflection overlap at smaller d_{\min} increases the uncertainty in the extracted structure factors.

These computations were performed on an Intel Core I7 machine with four duplexed processors (a total eight); GSAS-II uses the Python multiprocessor module to do the MCSA runs in parallel. The total run time (seven processors) was 596 s for 285 million structure factor calculations; the total processor time for these calculations was 3795 s implying a speed enhancement of ~ 6.4 over a single processor calculation.

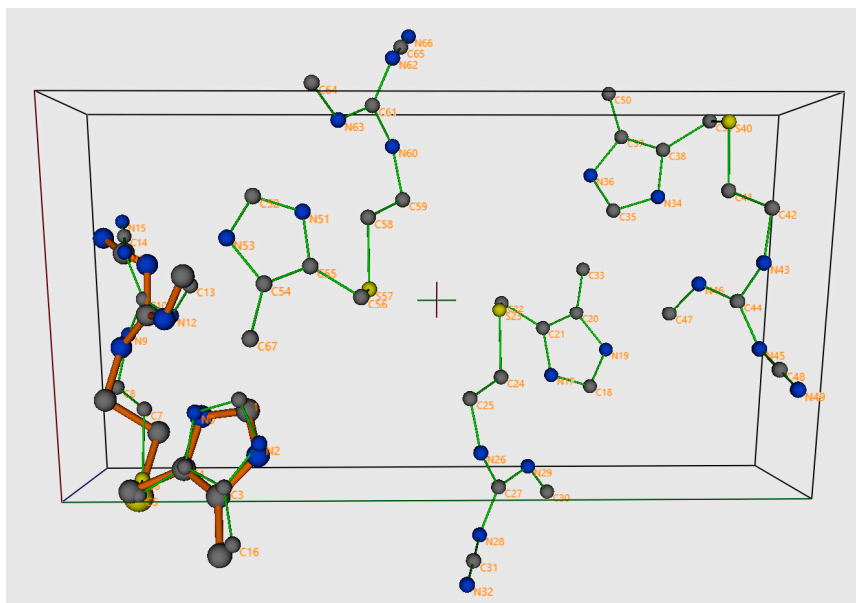


Figure 5. Successful MCSA result for cimetidine (atoms represented by small spheres and bonds in green) shown as the four crystallographically equivalent molecules in the unit cell viewed along the b-axis (a-axis vertical). The final refined structure is also shown (larger spheres and bonds in orange) for comparison. C atoms are shown in gray, N in blue, and S in yellow. Hydrogen atoms are not shown for clarity.

3.2. Charge Flipping

The CF method in GSAS-II works well for structure factors observed in a single crystal experiment or extracted in a Pawley refinement of powder data. By way of demonstration of the latter, we have used a high resolution X-ray powder diffraction pattern of sucrose (Figure 6).

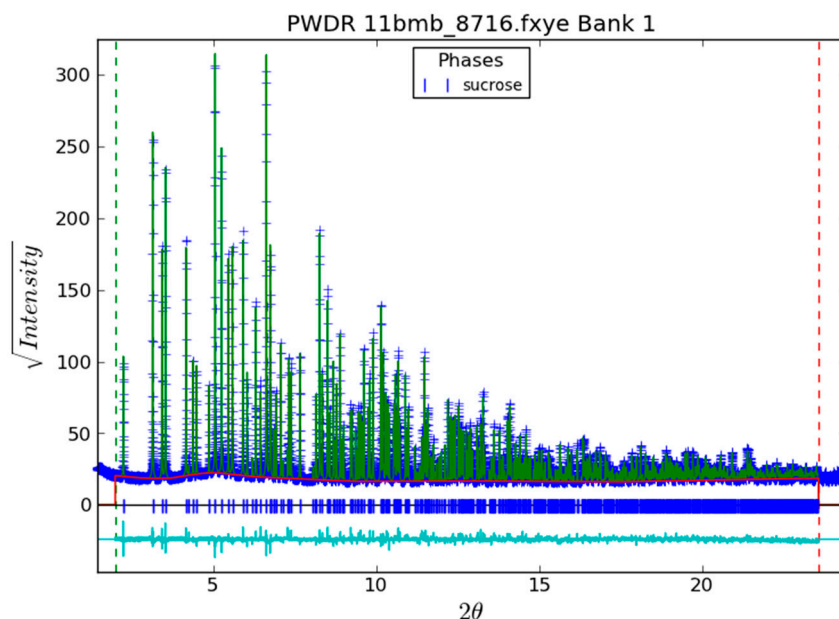


Figure 6. High resolution powder diffraction pattern of sucrose ($C_{12}H_{22}O_{11}$, space group $P2_1$, $a = 7.7157 \text{ \AA}$, $b = 8.6643 \text{ \AA}$, $c = 10.8101 \text{ \AA}$, $\beta = 102.983^\circ$, $Z = 2$) in a 1 h scan at $\lambda = 0.41326 \text{ \AA}$ taken at the Advanced Photon Source, beam line 11 BM. The portion of the data analyzed covers $2.0^\circ \leq 2\Theta \leq 23.55^\circ$ and comprises 21,554 profile points with a spacing of $0.001^\circ 2\Theta$. Observed data shown as blue (+), calculated pattern in green, background in red, difference shown below on same scale, and reflection positions indicated.

The result of the Pawley refinement that extracted 781 ($d_{\min} = 1.0138 \text{ \AA}$) unique structure factors ($R_{wp} = 5.122\%$) is shown in Figure 6. These were used in CF calculations; they were first expanded to 7680 reflections to cover the full sphere and then zero filled to a parallelepiped of 61,440 reflections to give a real space step size of 0.5 \AA . To begin, random values were assigned to all phases; CF solutions appeared approximately 50% of the time, giving a residual of $\sim 22\%$; otherwise the residual remained $>40\%$. In most successful cases, a solution appeared within 1000 CF cycles. Figure 7 shows the phases of five relatively strong reflections over 10,000 cycles and Figure 8 shows the detail at the point (\sim cycle 240) when convergence was achieved.

The behavior of these phases appears to be chaotic and span the entire 360° range of possible phase angles; this was noted in the original description of the CF method [12]. Before convergence, some phases show little variation while other show radical oscillations with little stability in their values. After convergence, there are sometimes strong oscillations (cf. the 020 and 17-1 reflections in Figure 8), but they are very regular with the phase oscillating between a pair of values which may differ by as much as 180° . There also does appear to be some “annealing” of the phases after the initial convergence (Figure 8 cycles 240–270). As seen in Figure 7, there may be a long term drift in the phases after convergence, however any one set comprises a converged CF solution. Figure 9 shows the much more chaotic behavior for the phases in a CF run that failed to achieve convergence.

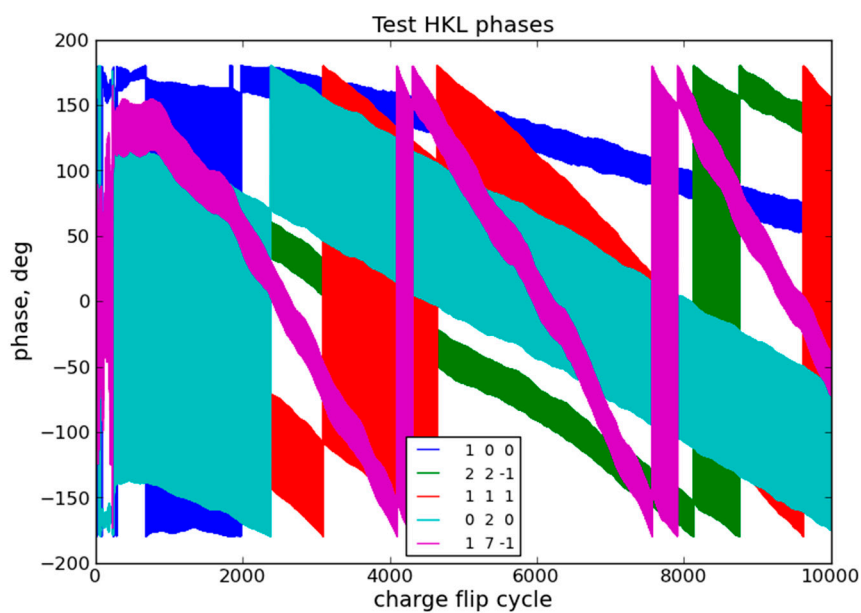


Figure 7. Phases of five reflections from a successful charge flipping run for sucrose. The residual dropped from ~40% to ~20% at cycle ~240. The reflection phases oscillate as well as drift over the succession of charge flip cycles (see detail in Figure 8).

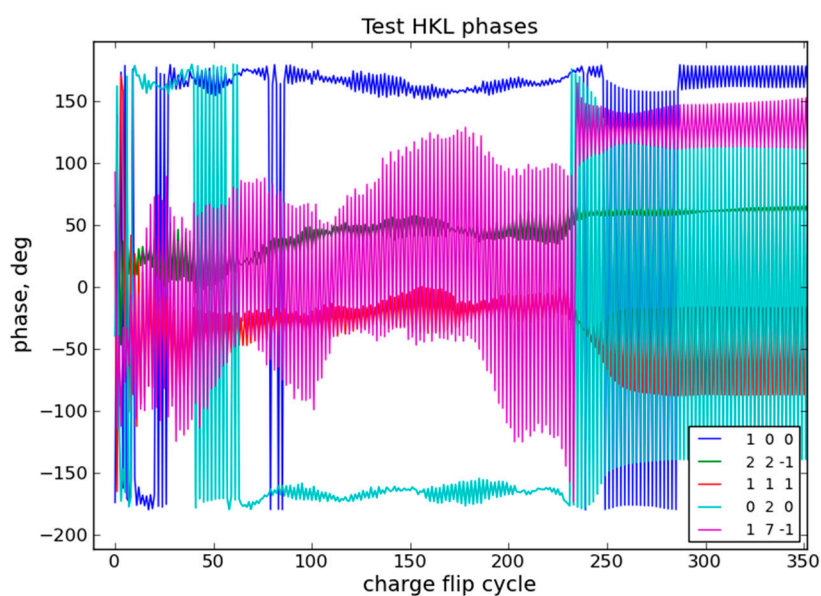


Figure 8. Detail of the initial CF cycles for sucrose showing phases of five reflections before and after the convergence (cycle ~240).

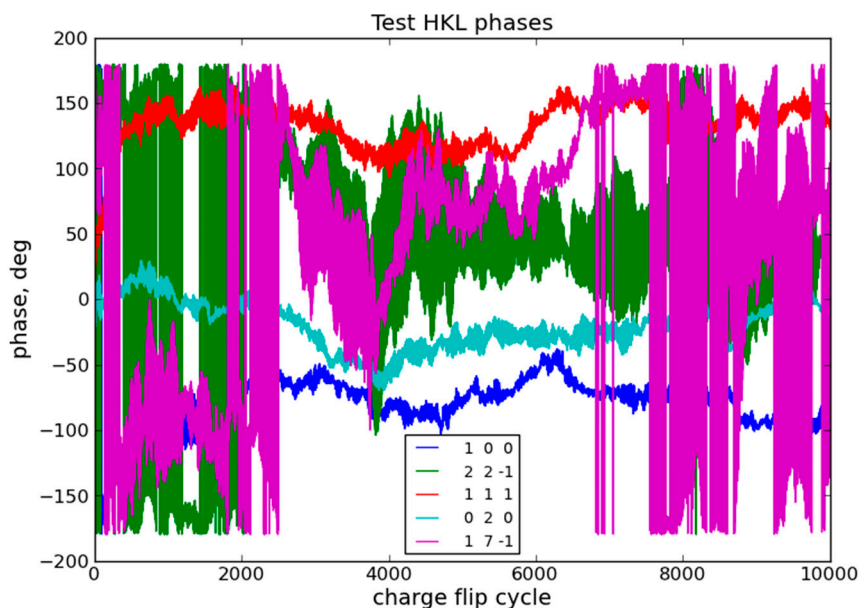


Figure 9. Phases of five reflections from an unsuccessful charge flipping run for sucrose. The residual stayed above ~40% for 10,000 CF cycles.

Thus, the calculations in CF are deterministic yet appear to be chaotic. As noted by Oszlányi and Sütő in their original description [12], there is strong sensitivity to the initial starting conditions; a sequence of CF runs will give different representations of the crystal structure when successful and complete failure when starting with a different random set of phases. For a chiral molecule such as sucrose, both versions will be found in a succession of CF runs. To explore this sensitivity, we modified the algorithm to invert five phases every 500 CF cycles; Figure 10 shows the result.

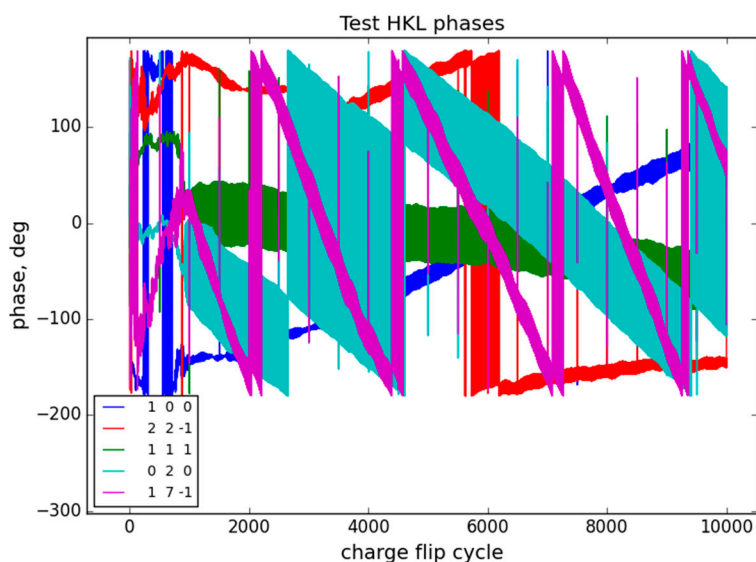


Figure 10. Phases of five reflections during 10,000 CF cycles where they were inverted at every 500th CF cycle (vertical lines).

Interestingly, there was no perturbation of the CF oscillations of these phases by the interruptions. Similarly, a failed (in 10,000 cycles—not shown) CF run with these perturbations showed no apparent effect on the phases over the course of the run. In chaos theory the level of sensitivity—e.g., how strong the “butterfly effect” is—to initial conditions is characterized by the “Lyapunov exponent” which

characterizes how fast a system will diverge after a perturbation or a change in initial conditions. If CF was characterized by a large Lyapunov exponent, the perturbations done here would push the CF run out of convergence or at least onto a new path. Evidently, in CF the Lyapunov exponent and the “butterfly effect” is very small so the sensitivity is to large changes in the phase set. It is also clear, from how the individual phases can explore the entire 360° range before convergence, that attempts to enforce limits (e.g., due to symmetry) would be counterproductive in CF as it would prohibit potentially successful paths to a solution.

4. Conclusions

Both CF and MCSA have been implemented in GSAS-II for both single crystal data and reflection intensities extracted from powder diffraction data. They have been demonstrated to work for moderate sized molecular crystal structures with powder data. Some of the chaotic features of CF were explored, showing that CF does not have a strong butterfly effect from changes to the phase set; further exploration is possible via modifications to the open source GSAS-II Python code but that is outside the scope of this publication. The MCSA algorithm has been shown to be effective; the inclusion of a BFGS fit at the end of each annealing run is particularly useful. This feature is a major component of an alternative MCSA scheme, “basinhopping” [15], which, as implemented in `scipy.optimize` [16], does a local minimization for every random step with a Metropolis test against a fixed temperature. Basinhopping has been implemented in GSAS-II as an alternative MCSA scheme, but preliminary tests have not shown it to be superior. It should be noted that even though we have used cimetidine to demonstrate the MCSA method, it can very readily be solved by CF from the same Pawley extracted suite of reflections within a few seconds. Clearly, one should try CF first and only use MCSA if that fails.

Acknowledgments: Supported by US DOE/OS/BES under Contract No. DE-AC02-06CH11357. The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>.

Conflicts of Interest: The author declares no conflict of interest.

References and Notes

1. Toby, B.H.; Von Dreele, R.B. GSAS-II: the genesis of a modern open-source all-purpose crystallography software package. *J. Appl. Crystallogr.* **2013**, *46*, 544–549. GSAS-II can be obtained from the site: <https://subversion.xray.aps.anl.gov/trac/pyGSAS>. [CrossRef]
2. Toby, B.H.; Von Dreele, R.B. What’s new in GSAS-II. *Powder Diffr.* **2014**, *29*, 1–5. [CrossRef]
3. Pagola, S.; Stephens, P. PSSP, a computer program for the crystal structure solution of molecular materials from X-ray powder diffraction data. *J. Appl. Crystallogr.* **2010**, *43*, 370–376. [CrossRef]
4. Favre-Nicolin, V.; Cerny, R. FOX, ‘free objects for crystallography’: A modular approach to ab initio structure determination from powder diffraction. *J. Appl. Crystallogr.* **2002**, *35*, 734–743. [CrossRef]
5. David, W.I.F.; Shankland, K.; van deStreek, J.; Pidcock, E.; Motherwell, W.D.S.; Cole, J.C. DASH: A program for crystal structure determination from powder diffraction data. *J. Appl. Cryst.* **2006**, *39*, 910–915. [CrossRef]
6. Scipy.optimize.anneal. Available online: <https://docs.scipy.org/doc/scipy-0.13.0/reference/generated/scipy.optimize.anneal.html> (accessed on 25 June 2017).
7. Pawley, G.S. Unit-cell refinement from powder diffraction scans. *J. Appl. Crystallogr.* **1981**, *14*, 357–361. [CrossRef]
8. LeBail, A.; Duroy, H.; Fourquet, J.L. Ab-initio structure determination of LiSbWO₆ by X-ray powder diffraction. *Mat. Res. Bull.* **1988**, *23*, 447–452. [CrossRef]

9. Hanwell, M.D.; Curtis, D.E.; Lonie, D.C.; Vandermeersch, T.; Hutchison, G.R. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminform.* **2012**, *4*, 17. [[CrossRef](#)] [[PubMed](#)]
10. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [[CrossRef](#)]
11. Scipy.optimize.minimize (method='L-BFGS-B'). Available online: <https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.optimize.anneal.html> (accessed on 25 June 2017).
12. Oszlányi, G.; Sütő, A. Ab initio structure solution by charge flipping. *Acta Crystallogr.* **2004**, *A60*, 134–141. [[CrossRef](#)] [[PubMed](#)]
13. Oszlányi, G.; Sütő, A. The charge flipping algorithm. *Acta Crystallogr.* **2008**, *A64*, 123–134. [[CrossRef](#)] [[PubMed](#)]
14. Solving an Organic Structure (Cimetidine—C 10 H 16 N6 S) from Powder Diffraction Data. Available online: http://www.ccp14.ac.uk/solution/powder_structure_solution_pathways/cimetidine.html (accessed on 27 June 2017).
15. Wales, D.J.; Doye, J.P.K. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *J. Phys. Chem. A* **1997**, *101*, 5111–5116. [[CrossRef](#)]
16. Scipy.optimize.basinhopping. Available online: <https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.optimize.basinhopping.html> (accessed on 25 June 2017).



© 2017 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).