Superflip — A computer program for solution of crystal structures from x-ray diffraction data in arbitrary dimension.

User Manual

written by Lukáš Palatinus
palat@fzu.cz

Laboratoire de Cristallographie
Le Cubotron
Ecole Polytechnique Féderale de Lausanne
output1015 Lausanne

version: 13/03/2006

Contents

Chapter 1

Introduction

The solution of the phase problem in crystallography is the central issue in the process of recovering structure information from x-ray diffraction data. Thanks to the development of direct methods, this problem can be solved routinely now for a large part of small- to medium-sized structures. Nevertheless alternative methods still can make a decisive contribution to solution of structures that are difficult or impossible to solve by direct methods and related techniques.

Charge flipping is an iterative algorithm for reconstructions of approximate electron densities from structure factor amplitudes. While it probably does not supersede direct methods in terms of the maximum size of solvable structures, it is superior in terms of the amount of prior information required. While most other structure solution methods are based on the assumption of the atomicity of the structure and the symmetry is also usually exploited, charge flipping needs neither symmetry nor the atomic character. This makes it particularly suitable for structure solution of modulated structures and quasicrystals, where the atoms form continuous domains in a (3+d)-dimensional space.

Superflip (the name stands for charge *flip*ping in *super*space) is a computer program that has been written to provide an effective tool for applications of the charge flipping algorithm. It allows for an (almost) automated structure solution of simple structures as well as a detailed control and advanced options for investigation of difficult structures.

Chapter 2

Theoretical background

The algorithm dubbed charge flipping has been described in two papers by Oszlány and Sütő (Oszlányi and Sütő, 2004; Oszlányi and Sütő, 2005). The reader interested in the details of the algorithm and the deeper theoretical background is strongly advised to study these two articles. The extension of charge flipping towards solution of incommensurately modulated structures has been described in Palatinus (2004).

In this manual only the basic concept of the algorithm will be outlined together with the basic terms that occur further in the manual.

The electron density ρ is sampled on a grid with $N_{pix} = N_1 \times N_2 \times N_3$ pixels. The density values ρ_i are evaluated in each pixel $i = 1, \ldots, N_{pix}$ of the grid. $|F^{obs}(\mathbf{H})|$ are the experimental amplitudes of the structure factors. The algorithm is initiated in the zeroth cycle by assigning random starting phases $\varphi_{rand}(\mathbf{H})$ to all experimental amplitudes and making all unobserved amplitudes equal to zero:

$$F^{(0)}(\mathbf{H}) = \begin{cases} |F^{obs}(\mathbf{H})| \exp\left(i\varphi_{rand}(\mathbf{H})\right) & \text{if } |F^{obs}(\mathbf{H})| \text{ is known} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

The iteration cycle then proceeds as follows:

1. The density $\rho^{(n)}$ is calculated by inverse Fourier transform of $F^{(n)}$.

2. The modified density $g^{(n)}$ is obtained by flipping the density of all pixels with density values below a certain positive threshold δ and keeping the rest of the pixels unchanged:

$$g_i^{(n)} = \begin{cases} \rho_i^{(n)} & \text{if } \rho_i^{(n)} > \delta \\ -\rho_i^{(n)} & \text{if } \rho_i^{(n)} \leq \delta \end{cases} \tag{2.2}$$

3. Temporary structure factors $G^{(n)}(\mathbf{H}) = |G^{(n)}(\mathbf{H})| \exp\left(i\varphi_G(\mathbf{H})\right)$ are calculated by Fourier transform of $g^{(n)}$.

4. New structure factors $F^{(n+1)}$ are obtained by combining the experimental amplitudes with the phases φ_G and setting all non-measured structure factors to zero:

$$F^{(n+1)}(\mathbf{H}) = \begin{cases} |F^{obs}(\mathbf{H})| \exp\left(i\varphi_G(\mathbf{H})\right) & \text{if } |F^{obs}(\mathbf{H})| \text{ is known and strong} \\ |G^{obs}(\mathbf{H})| \exp\left(i(\varphi_G(\mathbf{H}) + \pi/2)\right) & \text{if } |F^{obs}(\mathbf{H})| \text{ is known and weak} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

In the standard variant of the algorithm no reflections were treated as weak. In the improved variant (called the "π-half" variant sometimes) the reflections are sorted by their amplitudes and certain proportion of the smallest amplitudes is considered to be weak.

These modified structure factors then enter the next cycle of iteration.

The $F(\mathbf{0})$ structure factor is set to zero in the zeroth cycle of the iteration and allowed to change freely in the subsequent cycles.

δ is the only adjustable parameter of the algorithm. Its value should be selected small relatively to the maximum density, but larger than the typical amplitude of the Fourier artifacts induced by the series termination error. In practice, the value of δ is determined by trial and error.

An important aspect of the algorithm is that all operations are performed in the whole unit cell with symmetry $P1$. The origin of the structure is thus not fixed and the structure can emerge anywhere in the unit cell.

The progress of the iteration can be monitored for example by observing the R-value of amplitudes $|G^{(n)}(\mathbf{H})|$ with respect to $|F^{obs}(\mathbf{H})|$. It is large in the initial cycles of the iteration, and the onset of the convergence is signalled by a sharp decrease of the R-value. The iteration is converged, if the R-value stops decreasing and oscillates around a constant value. The final R-values are larger than the values typical for successful structure refinement, typically 20-30%. However, the R-value is not used as a measure of the quality of the reconstruction, but merely as an indicator of convergence.

The generalization of the algorithm for reconstructions of incommensurately modulated and composite crystal structures is straightforward. Following the method of embedding of aperiodic crystal structures in superspace, the $3D$ density is replaced by a $(3+d)D$ superspace density sampled using a $(3+d)D$ grid with $N_{pix} = N_1 \times N_2 \times \ldots \times N_{3+d}$ pixels, where d is the number of independent modulation vectors. The structure factors are indexed by $(3+d)$ integer indices. They represent the coefficients of the Fourier transform of the superspace density. With these modifications, the algorithm described at the beginning of this section can be applied directly to incommensurate structures.

Chapter 3

Installation and execution

3.1 Compilation of SUPERFLIP

The program is written in standard Fortran 90 and as such should be compilable with any f90 compiler. Most providers of unix workstations provide their own compiler suite. In addition to that, free fortran 90 compilers are now available (`http://gcc.gnu.org/wiki/GFortran`,`www.g95.org`).

The distribution package should be unpacked in a separate directory. The header of the `Makefile` must be modified manually if the commands calling the fortran and c-compilers are different from the default (currently gfortran and cc, respectively), or if the compiler options are not suitable for the given platform. The program is compiled by running `make` from the command line:

```
$ make
```

Prior to running make, the FFTW3 library has to be installed on the system and available to the linker (see below). After the compilation passes successfully, the resulting executable named `superflip` should be copied to a desired location, preferable one that is contained in the system variable PATH.

3.2 FFTW library

In order to speed up the performance of the program, `superflip` uses the library FFTW for computing of the discrete Fourier transform. Corresponding library (version FFTW 3.X or higher) must be therefore installed on your system prior to compiling the program itself.

The source codes for FFTW are available at www.fftw.org. Download the source codes and unpack to a separate directory. The compilation consists of three steps. First run the configure script:

```
$ ./configure --enable-float
```

The option `--enable-float` is necessary, because the default precision of the FFTW is double precision, but SUPERFLIP works in single precision (which, however, does not make the results less precise...).

Attention! After running `configure`, always check the file `config.log` for a string `disable-fortran` by typing:

```
$ grep "disable-fortran" config.log
```

On some systems the configure script has difficulties in figuring out the way to link fortran and C programs. This results in disabling the fortran interface to the C-routines. A message about disabling this interface is written in the file `config.log`. Thus, if `config.log` contains the string `disable-fortran`, measures have to be undertaken to allow configure to find the way to link fortran and C. Currently, this problem is known to occur on Mac OS X. The file `config.log` contains a bunch of error messages starting with `"/usr/bin/ld: multiple definitions of symbol "`. The solution is to replace following line in the file `configure`:

```
-lang* | -lcrt0.o | -lc | -lgcc | -libmil | -LANG:=*)
```

by line:

```
-lang* | -lcrt[012].o | -lc | -lgcc | -libmil | -LANG:=*)
```

After a successfull run of configure the library is compiled by running make:

```
$ make
```

and the compiled libraries are exported to their destination (usually /usr/lib) by typing

```
$ make install
```

You will need the administrator's access rights to do the last step.

If the libraries are placed in a standard location, the linker will usually find them without problems. However, if the location is non-standard (for example: you do not have the administrator's rights and want to install FFTW only on your account), you have to tell the linker where to find the library. This can be done by modifying the following line in the Makefile:

```
linklib = -lfftw3f -lm
```

to

```
linklib = -Lpath_to_FFTW -lfftw3f -lm
```

where ”`path_to_FFTW`” is a full path to the location of the library libfftw3f.a.

3.3 Execution

The program is executed from the command line by typing:

```
$ superflip filename [maxcycles]
```

Filename is the name of the ACII input file containig the instructions for the program. The optional argument maxcycles defines the maximum cycles of the iteration. If the calculation does not converge within maxcycles, the iteration is interrupted. If maxcycles is omitted, it is set to a default value 10000.

Chapter 4

Format of the ASCII input file

The input file is a free-format ASCII file based on keywords. Each keyword represents a specific command or parameter for the program and must be given a value.

Multiple spaces anywhere in the file are handled as a single space. If the character `'#'` or `'!'` occurs anywhere in the line, the rest of the line after this sign is treated as comment and not interpreted. Blank lines anywhere in the input file are ignored. The length of the interpreted part of the line is 132 characters, any text exceeding this length is ignored.

4.1 Specification of keywords

There are two basic types of keywords. The first type is followed by one or more values on the same line:

```
keyword value1 [value2 value3...]
```

The second type has the form:

```
initial_keyword
line 1
line 2
...
final_keyword
```

Each line may contain one or more values.

The name of the keyword of the first type is a single word without spaces. The name of the keyword of the second type is a pair of an initial and a final word (separated by a hyphen in the following text).

Each value can be a constant of type real, integer or character. The type of the parameters and their allowed values are specified. Alternative values are separated by slashes.

The keywords are either compulsory or optional. The compulsory keywords must be specified for the analysis to proceed. The optional keywords can be omitted. If an optional keyword is omitted, the default value is used. Compulsory keywords are indicated by "compulsory keyword – no default value" in the item "default".

The item "description" describes the function of the keyword, its influence on the output and relations to other keywords.

4.1.1 name: bestdensities

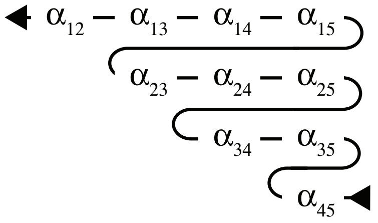
- value: positive integer
- default: 1
- desctription: Supeflip has the option to repeat the calculation. See keyword `repeatmode` to learn more about this option. If the calculation is repeated, the program saves the densities with the best R-values. If bestdensities = 1, only one density with the name given by the keyword `outputfile` is saved. This density is rewritten each time a new density has a better R-value than the saved density. If bestdensities > 1, the corresponding number of best densities are saved with the name `bestXX_`*outputfile*, where `XX` stands for a serial number of the density and runs from 1 to bestdensities. Outputfile is the filename given as a value of the keyword `outputfile`. The program writes out the list of the best R-values after each calculation to the logfile and, with the setting `terminal yes`, also to the standard output.

4.1.2 name: biso

- value: positive real number
- default: 0.0
- description: Defines the overall isotropic Debye-Waller factor B_{iso}. This factor is used to approximately cancel the decrease of the intensities by the thermal motion and effectively sharpens the resulting map. This sharpening can be vital for the success of the algorithm, if $B_{iso} > 4\,\text{Å}^2$ or so. For most inorganic materials B_{iso} is small and need not be taken into account.

4.1.3 name: cell

- value: a b c α β γ
- default: compulsory keyword – no default value
- description: Gives the dimensions of the unit cell. For two dimensional real space the order of parameters is a b γ, for one-dimensional case only one entry a is expected. For dimensions of the real space higher that 3 (see keyword `realdimension` for explanation) the lengths of the unit cell dimensions must be listed first, and then the angles between the individual pairs of basic vectors according to following convention (demonstrated on a five-dimensional example):



Here α_{ij} is an angle between the basic vector $\mathbf{a_i}$ and $\mathbf{a_j}$. This convention is a generalization of the convention used for the common 3D case.

4.1.4 name: centers

- value: each line contains one centering vector
- default: zero vector
- description: If the cell is non-primitive, a complete list of centering vectors must be given here. The components of the centering vectors can be either decimal numbers, or fractions with nominator and denominator separated by slash. Examples:

  A-centering:

  ```
  centers
  0 0 0
  0 0.5 0.5
  end centers
  ```

  R-centering:

  ```
  centers
  2/3 1/3 1/3
  1/3 2/3 2/3
  end centers
  ```

  As can be seen from the examples, the zero (trivial) centering vector may, but need not be included in the list.

4.1.5 name: coverage

- value: yes/no
- default: yes
- description: If `yes`, the coverage of the data as a function of resolution is performed. It is very recommended to check this table since a poor coverage might indicate some data inconsistency or simply a too low coverage for charge flipping to work. The calculation is very fast, only for very large problems the calculation of coverage can take considerable time and in that case it can be desirable to switch it off.

4.1.6 name: dataformat

- value: intensity/amplitude/a/b/phase/group/dummy and combinations of these items
- default: a b
- description: This keyword defines the format of the reflection entries in the keyword `fbegin - endf`. The keyword `dataformat` must be followed by one or more items from this list:
  - intensity: gives the intensity ($|F|^2$) of the reflection
  - amplitude: gives the amplitude ($|F|$) of the reflection
  - a, b: define the real and imaginary components of the structure factor
  - phase: Phase of the structure factor. Phase is expressed as a multiple of 2π, i.e. 0.5 corresponds to π, 0.25 corresponds to $\pi/2$. Questionmark (?) can be used in place of an unknown phase.
  - group: Reflection group. Applicable for powder diffraction data, where some reflections are overlapped.

  - dummy: Stands for any entry that should be ignored by the program. Can be used for example to make the program skip a column with standard deviations or some other irrelevant entry in the reflection list.

Each entry in the reflection list (see keyword `fbegin - endf`) is then asssumed to consist of the reflection indices and items corresponding to the list of items after `dataformat`.

The information in the list must not be redundant, i.e. it is not possible to give amplitude and intensity at the same time, and it is not possible to combine `a`, `b` with either of `intensity`, `amplitude` or `phase`. On the other hand, the items `group` and any number of items `dummy` can be combined with any of the other entries.

If a particular entry from the reflection list has less numbers than the number of items after `dataformat`, the behavior of the program depends on the type of the missing item. If `intensity`, `amplitude` or "`a`" is missing, program terminates with an error message. If "`b`" or `group` is expected, value 0 is assumed. If `phase` is expected, unknown phase is assumed (like if there were a questionmark).

If a reflection entry has more numbers than the items after `dataformat`, these additional numbers are silently ignored.

The default "`a b`" das been chosen for a backward compatibility with the files produced for BayMEM, for example by Jana2000.

Although all this seems very complicated, the practice is much simpler. Unless you want to use a special format of the data or some special features of superflip, you are likely to use either `dataformat ampitude` or `dataformat intensity`. Each reflection entry will then consist of the reflection indices and one number, this number being either $|F|^2$ (for intensity) or $|F|$ (for amplitude).

4.1.7 name: delta

- value: positive real number/AUTO [static/dynamic]
- default: AUTO
- description: Determines the handling of δ, the central parameter of the charge flipping algorithm. There are two basic ways of setting δ: automatically by the program, or directly be the user. With the setting `delta AUTO` the program tries to find the proper δ automatically. This is the recommended option in most cases, because for most of the structures the automatic procedure works well. The other option is to set delta to a fixed value. This is achieved by setting `delta` *value*.

  The original algorithm was developed with δ that is defined at the beginning of the iteration and then kept fixed during the iteration. Wu, Spence, O'Keeffe and Groy (2004) have developed another way of handling δ. In their modification, δ is defined as a fraction of all pixels that are to be flipped. At each iteration, the pixels are sorted according to their electron density, and the pixels with order lower than δN_{pixels} are flipped. Clearly, delta must be now smaller that one, and it is typically around 0.75. This dynamic determination of the flipping threshold has the advantage of being independent of the scale of the structure-factor amplitudes, but otherwise does not seem to outperform the original variant, and the sorting of pixels to find the dynamic threshold represents an additional computational cost.

  The original way of handling δ is defined by setting `delta` *value* `static` (synonym `absolute`), the dynamic δ of Wu and Spence is defined by `delta` *value* `dynamic` (synonym `fraction`). Default is static. Automatic search of delta uses the original, static variant.

4.1.8 name: dimension

- value: positive integer
- default: 3
- description: Defines the dimension of the problem. For standard structures dimension is 3, for modulated structures dimension is $3+d$ where d is the number of independent modulation vectors.

4.1.9 name: expandedlog

- value: yes/no
- default: no
- description: During the whole run of the program there is a lot of information produced that can be of potential interest in some situations, but is uninteresting in most of the cases. `expandedlog` allows to switch between the concise form of the file `jobname.sflog`, which however contains the general information about the iteration, and the expanded form, which provides a detailed record of all steps of the data processing, iteration, and output. For the description of the differences between the two forms of the log-file see Section 5.

4.1.10 name: fbegin – endf

- value: each line contains one structure factor in the form:

  ```
  h k l m... reflection information
  ```

  where the reflection information can be one or more of the following: intensity, amplitude, A and B component of the structure factor, phase, group number.
- default: Compulsory keyword – no default
- description: Information about the input data. Each reflection is defined by its reflection indices (their number must correspond to the value of keyword `dimension`). The reflection information that follows the indices must define the amplitude of the structure factor, and optionally may define its phase. The phase is not used in the actual charge-flipping iteration, but it can be useful under special circumstances (see keywords `perform` and `usephases`). For more information on how the reflection information is handled see keyword `dataformat`. Note that SUPERFLIP makes no use whatsoever of $\sigma(F)$.

4.1.11 name: filebase

- value: valid filename
- default: the name of the input file without the extension
- description: SUPERFLIP writes the log of the calculation in a file *filebase*`.sflog`. This keyword defines the name of the output file without the extension `.sflog`. Most often this keyword is omitted and the default behavior is used. In the default case the filebase is derived from the input file in the following manner: If the name of the input file does not contain a dot (.), the filebase is equal to the name of the input file. Otherwise the filebase is equal to the name of the input file without the part starting at the last dot. In other words, the filebase is simply the input filename without the extension.

4.1.12 name: fixsymops

- value: d integer numbers representing symmetry operations from the list; d corresponds to the value of keyword `dimension`
- default: automatic determination of applicable symmetry operations
- description: NOTE: This keyword applies only to the `old` method of the symmetry search (see keyword `searchsymmetry`). Depending on the value of the keyword `searchsymmetry` the program will or will not try to locate the position of the symmetry operations in the reconstructed density. In order to do that, suitable symmetry operations have to be selected from the current space group. Each dimension in the electron density can generally be fixed by zero, one or more than one symmetry operation. For example, a two-fold axis 2_x can be used to locate the origin in the y and z directions, but obviously not of the x direction. The program determines the optimal choice of the symmetry operations for individual directions. The method is to select an operation with the largest invariant subspace, which implies low dimension of the search space. Only invariant subspaces parallel to the crystallographic axes are considered. Such operations can be found for all standard space groups and all directions apart from (intrinsically unfixable) polar axes. However, in soma exotic settings and in higher-dimensional groups (an icosahedral group $Pm\bar{3}\bar{5}$ being a prominent example), the invariant subspace of the "lowest-invariant-subspace" symmetry operation does not have its invariant subspace aligned with the symmetry axes. At present the program is unable to handle correctly such a case and the user has to use the keyword `fixsymops` to select manually a suitable operation for each direction, that can be unambiguously used to fix that direction. The number at the place of each dimension corresponds to the order of the symmetry operation in the list `symmetry - endsymmetry`. If certain dimension cannot be fixed (as is the case in most non-centrosymmetric spacegroups), the entry at the corresponding place is 0.

  Sometimes it can be difficult to find a suitable symmetry operation. If the structure is centrosymmetric, the inversion center can always serve this purpose, but this comes at the cost of high computational time, because the invariant subspace of the inversion center is zero-dimensional.

4.1.13 name: fullreflections

- value: valid filename
- default: no list
- description: Ocasionaly it can be useful to have a complete list of all reflections (input reflections and their symmetry-equivalents) with phases from charge flipping. If the keyword `fullreflections` is given, SUPERFLIP will write out such a complete list into a separate file with name given after the keyword `fullreflections`. The file contains one line per reflection with format `h k l...  A B`.

4.1.14 name: histogram - endhistogram

- value: each line contains one entry of the histogram (one real number)
- default: no histogram
- description: SUPERFLIP allows for an improvement of the density by a technique called histogram matching (Zhang and Main, 1990). In short, the density values are modified so that they match a predefined histogram, possibly derived from a similar structure or calculated. The histogram is defined by the values that delimit density classes with

equal number of pixels in each class. The histogram can be obtained so that the pixels of the reference density (from which the histogram should be calculated) are sorted from smallest to largest denity value. Then the number of pixels is divided by the number of desired number of classes (50 is usuallly sufficient) to get a step size and each "stepth" pixel's density value will represent one histogram entry. The first and last entries are the smallest and largest density values, respectively. See also keyword `hmparameters`.

4.1.15 name: hmparameters

- value: two integer numbers
- default: 200 100; applicable only if `histogram - endhistogram` is present
- description: The histogram-matching procedure (see keyword `histogram - endhistogram`) is usually not performed after every cycle. It is first performed at cycle number *hmstart* and then every cycle for which cycle mod *hmstep* = 0. *hmstart* and *hmstep* are the first and second number following the keyword `hmparameters`, respectively.

4.1.16 name: maxcycles

- value: non-negative integer
- default: 10000
- description: Defines the maximum number of iteration cycles. If this number of cycles is reached, the iteration is interrupted and no convergence is detected. Alternatively, this number can be passed to the program as a second command-line argument (see Section 3.3. The command-line argument has priority over the value in the input file.

4.1.17 name: outputfile

- value: valid filename [another valid filename...]
- default: compulsory keyword – no default
- description: Defines the name of the file holding the resulting electron density. More than one filename can be given, maximum is 10. Superflip will save the density in all the files in formats given either by the keyword `outputformat`, or guessed from the file extension. See keyword `rewriteoutput` for more information about handling the output files.

4.1.18 name: outputformat

- value: jana/xplor/m80 [formats of other output files according to the keyword `outputfile`
- default: guess the density format from the extension of the of the output filename
- description: Defines the format of the output electron density. The setting `outputformat jana` results in saving the density in the format of the crystallographic system Jana2000 (Petříček, Dušek and Palatinus, 2000). The standard extension is `m81`. It is a binary format suitable for saving electron density in arbitrary dimensions up to six. It can be viewed in Jana2000 in a form of contour plots in arbitrary sections or projections. Setting `outputformat xplor` results in an ASCII output format of Xplor (extension `xplor`) – a program system for computational structural biology. It can be viewed e.g. by the 3D plotting program Chimera. Being an ASCII format, it is easy to modify it to any other format. The native xplor format is suitable only for 3D data. Superflip

uses a generalized xplor format for saving electron densities with more that three dimensions. See section 5.1.1 for closer description of the format. Setting `outputformat m80` will produce a list of reflections with phases that can be used in Jana2000 to produce arbitrary section through the density. Standard extension is `m80`. See section 5.1.3 for closer description of the format. If `searchsymmetry` is set to `average`, only reflections present in the input file are listed. In all other cases the list contains complete set of reflections.

Other formats of the output file can be added upon request, provided a transparent description of the format (or a reference thereto) is supplied together with the request.

4.1.19 name: perform

- value: CF/fourier
- default: CF
- description: Apart from performing the density reconstruction by charge flipping (setting `perform CF`), SUPERFLIP can also do a simple Fourier transform of the input data (setting `perform fourier`), provided the phases of the structure factors are supplied (see keyword `fbegin - endf`). This can be occasionally useful for checking the internal consistency of the data, if the structure is already (partially) known and SUPERFLIP is used to test the behavior of charge flipping on that data set.

4.1.20 name: qvectors – endqvectors

- value: each line contains coordinates of one q-vector as decimal numbers
- default: compulsory keyword if dimension > realdimension, otherwise inapplicable
- description: The list holds the definition of the q-vectors in a modulated structure. The number of q-vectors must be equal to the difference between dimension and realdimension (see the corresponding keywords). The number of components of each q-vector is equal to realdimension.

4.1.21 name: randomseed

- value: AUTO/non-negative integer
- default: AUTO
- description: Defines the seed for initialization of the random number generator. Setting `randomseed AUTO` causes the program to generate the seed from the system time. Positive integer is taken as a random seed without any modification. In this way it is possible to exactly reproduce one calculation several times.

4.1.22 name: realdimension

- value: positive integer
- default: 3
- description: Defines the dimension of the real space. For normal crystal structures this is always 3. Other values might be used in case of data from the two-dimensional surface scattering, calculation of quasicrystal densities, or theoretical experiments with higher-dimensional crystallography.

4.1.23 name: referencefile

- value: a valid filename of an existing file
- default: no reference file
- description: Occasionally the result of the charge flipping needs to be compared with an electron density obtained otherwise, or with another result of charge flipping on the same data. The location of the origin of the space group in the electron density is not sufficient for bringing the density itself always to the same position. This has two reasons. First, some symmetry operations generate equivalent symmetry operations elsewhere in the cell, if combined with the lattice translation (a simple example: a mirror plane in the origin of an ortorhombic cell generates another mirror plane at 1/2). The second reason is that most non-centrosymmetric space groups have one or more directions that cannot be fixed by the symmetry at all and the resulting density, although averaged over symmetry, will be randomly shifted along these directions.

  For these reasons the user has the possibility to supply a reference electron density. Superflip will align the resulting electron density with the reference density, allowing for a direct comparison of them. Another application of this technique is the possibility of summing up several results of charge flipping, which should result in cleaner density map, since the signal sums up, but the random noise cancels upon the summation.

4.1.24 name: referenceformat

- value: jana/xplor
- default: compulsory, if the keyword `referencefile` is present, irrelevant otherwise
- description: The format of the reference density (see the keyword `referencedensity`). For more information about the formats see the keyword `outputformat`.

4.1.25 name: reflectionlist

- value: unique/complete
- default: unique
- description: Superflip accepts two forms of the list of reflections. The standard way is to list only symmetry-inequivalent reflections (setting `reflectionlist unique`). Superflip will expand them according to the list of symmetry operations given in the keyword `symmetry - endsymmetry`. Another possibility is to list all reflections in the full sphere, including the symmetry-equivalent reflections (setting `reflectionlist complete`). This can be occasionally useful, if the symmetry is doubtful or if the user wants to test, how the unaveraged reflection list complies to given symmetry. Note however, that even in the complete list doubled reflections (i.e. two entries with identical indices) are not allowed and thus the averaging in $P1$ (or better $P\bar{1}$) must be performed before. Note also that charge flipping needs a reasonably complete coverage of the reciprocal space. It is user's responsibility to supply a list that fulfills this condition.

  The two types of the input reflection list differ also in handling of the reflections that are forbiden by the symmetry (extinct or systematically absent reflections). With `reflectionlist unique` the extinct reflections are excluded from the reflection list and a warning is written in the log file. With `reflectionlist complete` the extinct reflections are allowed and included in the calculation. With the setting `expandedlog yes` the list of the extinct reflections is written in the log file.

The equality of intensities of Friedel-pairs (with anomalous dispersion neglected) is of a special interest. Strictly speaking the Friedel pairs are not independent and they should be listed in the list of doubled reflections. However, since most of the data-processing software does not average over Friedel pairs for obvious reasons, SUPERFLIP does not consider Friedel pairs in a non-centrosymmetric structure as doubled reflections, but averages over the two reflections and uses the average in the calculation. Information about the Friedel pairs is written in the log-file.

4.1.26 name: repeatmode

- value: never/nosuccess/always
- default: never
- description: SUPERFLIP can repeat the whole calculation. With setting `repeatmode never` (which is the default) the program will stop after one calculation, i.e. when the calculation converges or if the maximum number of cycles is reached (see keyword `maxcycles`). With the setting `repeatmode nosuccess` the calculation is repeated until the convergence is detected. `repeatmode always` will cause the program to repeat the calculation indefinitely. The saving of the best density (or several best densities) is controlled by the keyword `bestdensities`.

4.1.27 name: rewriteoutput

- value: yes/no
- default: yes
- description: With `rewriteoutput yes` both the log file and the density file are overwritten by the new log file and density file. With `rewriteoutput no` the log file is appended and the density file is not overwritten, if it exists. Instead of that the density is written to a file `sfrhoXX.`*ext*, where `XX` is a number between 00 and 99, and *ext* is a format-specific extension, namely `.m81` for the jana format and `xplor` for the xplor format. First unused number is selected. If all files `sfrho00.`*ext* through `sfrho99.`*ext* exist, an error message is written to the log file and the program terminates without writing the density. The information about the name of the density file is always written at the end of the log file.

4.1.28 name: searchsymmetry

- value: no/shift/average [old/new]
- default: average new
- description: The electron density is always reconstructed in P1, i.e. without the use of the symmetry. But of course, the resulting density still (approximately) obeys the underlying symmetry of the structure, only the origin of the space group is randomly shifted in the cell. Thus, the symmetry elements can be localized in the resulting electron density. This keyword defines, how the program should handle the symmetry. `searchsymmetry no` prevents any search for the position of the symmetry elements. `searchsymmetry shift` leads to the location of the origin of the space group in the density and subsequent shifting of the density. However, no averaging is performed, the density is only shifted. `searchsymmetry average` leads to the location of the origin, shifting of the density and averaging of the density over the symmetrically equivalent pixels, so that the resulting density has exactly the symmetry of the space group. The last setting is recommended, unless problems occur with the location of the origin.

The search for the symmetry is a more tricky business than it might seem at the first glance. Several methods were attempted. The default method (`new`) should works very well in all tested cases. However, if it should fail (i.e. the symmetry is not located properly although the calculation converges and the resulting density contains the symmetry), the user has the possibility to use another method (`old`), which is slower, and less general, but might provide a solution where the other method fails. If you encounter a problem with the default method, please DO report it to the author, so that the method can be improved.

4.1.29 name: symmetry – endsymmetry

- value: each line contains one symmetry operation in the one-line form
- default: compulsory keyword – no default
- description: Defines the symmetry of the structure. Its primary use is to expand the reflections to full sphere. It need not be known for the charge-flipping iteration, but it can be used to recover the symmetry of the resulting density (see keyword `searchsymmetry`. If the list of reflections is complete and the space group is uncertain, space group $P1$ can be always used.

  The symmetry operations are given in a on-line form known from the International Tables for Crystallography. However, since SUPERFLIP works in arbitrary dimensions, the individual coordinates are not denoted x, y, z, but rather $x1$, $x2$, $x3$, $x4$... The translational part can be given both as a fraction and as a decimal number. A complete space group must always be listed, including the identity operation. The centering vectors are, however, listed separately (see keyword `centers - endcenters`). Two examples will illustrate the form of the input:

  $P4_2/n$:

```
symmetry
    x1     x2     x3
1/2-x1 1/2-x2     x3
   -x2 1/2+x1 1/2+x3
1/2+x2    -x1 1/2+x3
   -x1    -x2    -x3
1/2+x1 1/2+x2    -x3
    x2 1/2-x1 1/2-x3
1/2-x2     x1 1/2-x3
endsymmetry
```

  (3+1)-dimensional superspace group $Cmcm(0\beta 0)s0s$:

```
symmetry
 x1  x2     x3     x4
-x1  x2     x3 1/2+x4
 x1 -x2 1/2+x3     x4
 x1  x2    -x3 1/2+x4
-x1 -x2    -x3    -x4
 x1 -x2    -x3 1/2-x4
-x1  x2 1/2-x3    -x4
-x1 -x2     x3 1/2-x4
endsymmetry

centers
```

```
1/2 1/2 0 0
endcenters
```

4.1.30 name: terminal

- value: yes/no [keep]
- default: yes
- description: If yes, a short information about the progress of the calculation is written on the standard output. This output can be suppressed by setting `terminal no`. Such setting can be useful if you run SUPERFLIP in a batch mode on a remote computer, in which case you do not want to bind the execution of the program to an existence of specific terminal window. If a word ”keep” is present as the second word after the keyword, the program will ask for pressing Enter before quitting. That is useful if the terminal window closes automatically after the end of the execution, and the user wishes to see the output before closing the window.

4.1.31 name: testsymmetry – endtestsymmetry

- value: each line contains one symmetry operation in a one-line form
- default: no test symmetry operations
- description: Occasionally the symmetry is not known with certainty and several space groups are possible. SUPERFLIP can take the list of symmetry operations, and test, how well is the corresponding symmetry present in the density reconstructed by charge flipping. In combination with `searchsymmetry no` the whole density is searched for the optimal position of each of the listed symmetry operations, with other settings only the positions allowed by the ambiguity of the position of the origin are tested (compare keyword `referencedensity`). The syntax of the symmetry operations is the same as in the keyword `symmetry - endsymmetry`.

4.1.32 name: title

- value: string of characters up to 132 characters long
- default: no title
- description: Title of the calculation. Is written in the log file and can serve for the identification of the job. The program makes no other use of this string.

4.1.33 name: usephases

- value: firstcycle/always/integer number
- default: no
- description: With `usephases firstcycle` the starting phases are not chosen randomly, but the phases present in the input file are used. If only some phases are known and some not (see keyword `dataformat` for details), the unknown phases are taken at random. With `usephases always` the known phases are reset to the input values in each iteration cycle. If a number n is given as a value to `usephases`, the phases are set to their input values only in the first n cycles of the iteration and after that they are allowed to change freely.

4.1.34 name: voxel

- value: list of positive integers; their number must be equal to dimension
- default: compulsory keyword – no default
- description: Defines the grid on which the density is computed. The numbers are subject to following restrictions:
  - The grid division in each dimension must be larger that two times the largest reflection index in that dimension.
  - The grid must be compatible with the symmetry, i.e. each grid point must be mirrored by all symmetry operations onto itself or onto another grid point. For example, a spacegroup containing a 6_1 axis along z must have the division along the third axis that is a multiple of 6.
  - The last restriction is not a must, but it is recommended that the grid divisions can be factorized into small primes, preferably 2 and 3. This speeds up the fast Fourier transforms, that take most of the time in the calculation.

4.1.35 name: weakratio

- value: real number between 0.0 and 1.0
- default: 0.0
- description: In the second article of the authors of charge flipping (Oszlányi and Sütő, 2005) it has been shown that the convergence can be substantially improved by perturbing not only the electron density in the direct space, but also the phases of the structure factors in reciprocal space. The perturbation is achieved by shifting the phases of certain portion of the weakest reflections by $\pi/2$ in each iteration cycle and by not replacing the calculated amplitudes with the observed ones. The keyword `weakratio` serves for defining the fraction of the reflections that are considered weak and subject to the phase shift. Typically the value of `weakratio` is between 0.2 and 0.3. However, note that the use of `weakratio` yields somewhat more noisy and less accurate maps due to the loss of information from the weak reflections. But even these maps usually represent a pretty good approximation of the real density.

4.2 Examples of an input file

4.2.1 A minimalistic example

This example contains a minimal set of instructions. Despite of its simplicity it is likely to work for most simple structures.

```
title A minimalistic input file
cell 5.2 5.8 4.4 90. 90. 98.
outputfile example1.m81

voxel 60 60 64

#Space group P2/m
symmetry
 x1     x2      x3
-x1    -x2      x3
-x1    -x2     -x3
 x1     x2     -x3
endsymmetry

fbegin
   7   0 -3     5.7445626
   7   1 -3     5.3385391
   8   1 -3     7.0710678
```

```
    .
    .
    .
endf
```

4.2.2 A realistic example for a modulated structure

This is a real example of a file that illustrates all the options available in SUPERFLIP. This file was used to produce the log-file that is described in Section 5.2. This file is included as a sample file in the distribution package of SUPERFLIP.

```
title Cr2P2O7 - incommensurate phase, room temperature
perform CF

# Keywords influencing the form of the files
outputfile Cr2P2O7_sf.m81
outputformat jana
expandedlog no
coverage yes
referencefile Cr2P2O7_ref.m81
referenceformat jana

# Basic crystallographic information
dimension  4
voxel   36  48  24  16
cell     7.0192    8.4063    4.6264   90.00  108.61   90.00
qvectors
  -0.361   0.000   0.471
endqvectors
centers
  0.0  0.0  0.0  0.0
  0.5  0.5  0.0  0.0
endcenters
symmetry
     x1     x2     x3     x4
    -x1     x2    -x3 1/2-x4
    -x1    -x2    -x3    -x4
     x1    -x2     x3 1/2+x4
endsymmetry
testsymmetry
     x1    -x2     x3     x4
endtestsymmetry

# Keywords influencing the algorithm
delta AUTO
weakratio 0.000
biso   0.000
randomseed AUTO
searchsymmetry average

# List of reflections
reflectionlist unique
dataformat amplitude
fbegin
   0   0   0   2    29.6141853
   0   0   0   4     5.9497900
  -2   0   0  -4     6.4498062
  -2   0   0  -2     7.9498429
   .
   .
   .
endf
```

Chapter 5

Description of the output

There are two main output files. The principal output of the calculation is the file with the electron density. The second output file is the log-file containing the information about the data processing, iteration itself and the results of the calculation.

5.1 The electron density file

The name of the output density file is defined by the keyword `outputfile`, the format is defined by the keyword `outputformat`. Currently SUPERFLIP supports three density formats: the xplor format (defined by `outputformat xplor`), the format m81 of the crystallographic package Jana2000 (`outputformat jana`), and the format m80 of Jana2000, which contains a list of structure factors (`outputformat m80`). These structure factors can be used directly by Jana2000 to calculate arbitrary sections through the density. Other formats can be added upon request, provided a transparent description of the format (or a reference thereto) is supplied together with the request.

5.1.1 format xplor

This is an ASCII format of the software package Xplor for structural biology. It can be read and displayed e.g. by the 3D plotting program Chimera. The format assumes the following form:

```
empty line
number of lines of the title
line 1 of the title
line 2 of the title etc. up to the number given on line 2
pixel division: number of pixels along x, first pixel, last pixel; ditto for y and z
cell parameters
ZYX string defining the order of axes from the most slowly to the most quickly varying
     0 - the number of the current layer along z
density value
density value
density value
nx.ny values, nx and ny is the number of pixels along x and y. x varying first, y second.
    1 - number of the next layer along z
density value
    .
    .
    .
-9999 always present at end of the density values
two real numbers for the average and sigma(average). Sigma is not computed by {\sc superflip}.
```

For the exact format (exact length and position of the numbers) please check the documentation of Xplor or consult a file in xplor format produced by SUPERFLIP.

5.1.2 Format jana

This is the format of the crystallographic software package Jana2000 (Petříček et al., 2000). Its standard extension is .m81. It stores the electron density in single-precision direct-access binary format. It is suitable for storing electron densities up to 6 dimensions and can be viewed in Jana2000. It is beyond the scope of this manual to fully describe this format. The reader interested in the details of the format should consult the source code of SUPERFLIP or Jana2000, or contact the authors of either of the programs.

5.1.3 Format m80

This is the format of the input file for the Fourier module of Jana2000 (Petříček et al., 2000). Its standard extension is .m80. Each line of the file has format (di4,i4,13e12.5), where d is the number of reflection indices. The information in each line is:

reflection indices, number of structure(always 1 in superflip),F_{obs}, F_{obs}, F_{calc}, A, B

The rest of the line is compulsory in the format but it is irrelevant for the output from superflip and is padded with zeroes.

5.2 The log-file

The log-file contains all information about the run of SUPERFLIP. Its name is *filebase*`.sflog`. The filebase is the name of the input file without the extension, unless it is explicitly redefined (see keyword `filebase`).

Depending on the value of the keyword `expandedlog`, the log-file can have two forms: the short, concise record of the main elements of the integration, and the long form, that should be necessary only for diagnostic purposes and in case of the difficulties with some part of the structure solution.

5.2.1 The concise form of the log file

This form is obtained by default or by setting `expandedlog no`. The form of the log-file will be illustrated on the example of a modulated structure of chromium(II)-diphosphate in order to include also the parts specific to the modulated structures. The input file is described in section 4.2.2

In the following subsections the individual parts of the file will be described.

5.2.2 Information about the input

The first part contains an information about all the control keywords read from the input file. This part is written to be self-explanatory and can be used in any later time to reconstruct the conditions under which the calculation has been performed.

```
-----------------------------------------------
Start of the calculation: 16.JAN 2006, 12:51:24
-----------------------------------------------

###################################################################
# Following data were read from the input file or set as default: #
###################################################################

Job title: Cr2P2O7 - incommensurate phase, room temperature

-----------------------
Information about files:
-----------------------
Name of the input file: Cr2P2O7.inflip
Density will be writen in jana format to file   Cr2P2O7_sf.m81
Warning: If the outputfile exists, it will be overwritten by the new density.
Logfile will contain only basic information about the calculation.
Logfile will contain information about the data coverage.
```

```
-----------------------------
Crystallographic information:
-----------------------------
Superspace dimension:             4
Dimension of the physical space:  3
Direct cell parameters:          7.0192   8.4063   4.6264  90.0000 108.6100  90.0000  Volume: 258.7095
Reciprocal cell parameters:      0.1503   0.1190   0.2281  90.0000  71.3900  90.0000  Volume:   0.0039
Q-vectors:
  -0.3610   0.0000   0.4710

4 symmetry operations found, their list follows:

   1:  1  0  0  0   0.0000   2: -1  0  0  0   0.0000   3: -1  0  0  0   0.0000
       0  1  0  0   0.0000       0  1  0  0   0.0000       0 -1  0  0   0.0000
       0  0  1  0   0.0000       0  0 -1  0   0.0000       0  0 -1  0   0.0000
       0  0  0  1   0.0000       0  0  0 -1   0.5000       0  0  0 -1   0.0000

   4:  1  0  0  0   0.0000
       0 -1  0  0   0.0000
       0  0  1  0   0.0000
       0  0  0  1   0.5000

The structure is centrosymmetric.
The symmetry operations are to be combined with the following centering vectors:
   0.0000   0.0000   0.0000   0.0000
   0.5000   0.5000   0.0000   0.0000


-------------------------
Settings of the algorithm:
-------------------------
Number of voxels:                36    48    24    16   Total:    663552
Delta will be determined automatically.
The random number generator will be initialized automatically.
Isotropic Debye-Waller factor:   0.000
Proportion of reflections to be treated as weak:   0.000
The resulting density will be shifted and averaged according to the symmetry operations given above.
The symmetry operations used for location of the origin of symmetry will be found automatically.
```

5.2.3 Information about the processing of the reflections

The concise form contains only the summary of the reflection-import, i.e. the number of reflections, the maximum indices in the expanded set (can be used to fine-tune the grid size), and (by default or with `coverage yes`) the coverage as a function of the resolution. Always check the coverage at the first run of a new data set to make sure it is sufficient. Low coverage will prevent convergence!

Note: The calculation of the coverage does not work properly for composite structures. This is because of the difficulties with the definition of what is a satellite in such structures (in other words, the W matrix of the composite is not available to the program). However, this problem concerns only the calculation of coverage and has no impact on the calculation itself. The coverage does not work well also for quasicrystals. This might change in the future versions of SUPERFLIP.

In addition to the example below, the listing may contain information about the doubled reflections, extinct reflections and independently listed Friedel pairs in a non-centrosymmetric cases. For the rules governing the handling of the reflections see description of the keyword `reflectionlist`.

```
##################################
# Information about reflections: #
##################################

Number of reflections in the input file:   2410
   The list of reflections will be assumed to contain only symmetry-inequivalent reflections.

Maximum indices in expanded reflection set:  10  10   7   4

Coverage statistics of the expanded reflections by shells:
Resolution (sin(th)/l):    0.050     0.100     0.150     0.200     0.250     0.300     0.350     0.400
Resolution (d_min):       10.000     5.000     3.333     2.500     2.000     1.667     1.429     1.250
Obs. refl. in shell:           0        11        53       108       151       236       337       431
Total refl. in shell:          1        15        53       108       151       236       337       431
```

```
Coverage in shell:          0.0%    73.3%   100.0%   100.0%   100.0%   100.0%   100.0%   100.0%
Commulative coverage:       0.0%    68.8%    92.8%    97.2%    98.5%    99.1%    99.4%    99.6%

Resolution (sin(th)/l):    0.450    0.500    0.550    0.600    0.625
Resolution (d_min):        1.111    1.000    0.909    0.833    0.800
Obs. refl. in shell:         557      710      828      959      631
Total refl. in shell:        557      710      828      959      631
Coverage in shell:        100.0%   100.0%   100.0%   100.0%   100.0%
Commulative coverage:      99.7%    99.8%    99.9%    99.9%    99.9%
```

5.2.4 Information about the iteration

This part deserves special attention and therefore the listing will be commented by comments interleaved in the sample text:

```
#############
# Iteration #
#############
```

The random seed used by the program is indicated. This can serve for reproducing the calculation at a later stage.

```
Random seed:  140625
```

The header for the iteration record gives the short explanation of the items that occur in each line. The total charge is simply $\sum \rho_i$ summed over all pixels (i.e. $F(\mathbf{0})$). Flipped charge is $\sum |\rho_i|$ summed over all pixels for which $\rho_i < \delta$.

```
Cycle  R-factor   Total Charge (flipped charge)
```

The next part is a record of the search for delta. This is present only for setting `delta AUTO`. The criterion for δ is $0.6 <$ total charge/flipped charge < 1.0.

```
Searching for a proper delta:
Current delta =   20.83529
Total/flipped ratio =    2.224. Increasing delta.

Current delta =   22.29377
Total/flipped ratio =    1.578. Increasing delta.

Current delta =   23.85433
Total/flipped ratio =    1.179. Increasing delta.

Current delta =   25.52413
Total/flipped ratio =    0.975.
Criterion for delta fulfilled, continuing iteration.
```

In the concise form the status is written out only at some cycles of the iteration, not all. Every tenth cycle is listed up to 100 cycles, every hundredth cycle from 100 to 1000 cycles, and every thousandth cycle from 1000 cycles upwards.

```
   10  R: 43.805  Charge:  2429.90( 2492.58)
   20  R: 45.256  Charge:  2080.52( 2620.42)
   30  R: 40.643  Charge:  1490.29( 2645.50)
   40  R: 35.989  Charge:  1412.62( 2672.35)
   50  R: 31.716  Charge:  1346.45( 2627.30)
   60  R: 29.206  Charge:  1349.93( 2612.52)
   70  R: 27.996  Charge:  1352.00( 2625.80)
   80  R: 27.133  Charge:  1339.73( 2633.87)
   90  R: 26.592  Charge:  1329.37( 2635.58)
  100  R: 25.301  Charge:  1328.46( 2646.54)

Calculation successfully converged after   107 cycles.
Last iteration record:
  107  R: 25.123  Charge:  1355.69( 2651.31)
```

If the value of the keyword `searchsymmetry` is not `no`, the search for the symmetry operations is performed.

The agreement factor determines the degree of coincidence between the original and symmetry-transformed density. The expected value of the agreement factor for a random shift is 100%. The smaller the agreement factor, the better the match. First the agreement factors for individual generators used for the symmetry search are listed, and then the overall agreement factor that includes all symmetry operations of the space group.

```
############################################
# Search for the origin of the space group #
############################################

Agreement factors of individual generators:
Number  agreement
   4      17.09
   2      15.46

Overall agreement factor: 16.09
```

If some symmetry operations for testing are present in the input file (keyword `testsymmetry`), the result of the testing is listed here. The coordinates give a point that lies in the symmetry element and has the smallest distance to the origin. This example shows that the tested symmetry operation is not present in the density, because the agreement factor is much larger than the values for the real symmetry elements.

```
###########################################################
# Testing for the presence of the test symetry operations #
###########################################################

Number   agreement         coordinates
   6     95.8104      0.0000  0.3854  0.0000  0.0000
```

If a reference file is given (see keyword `referencefile`), info is written about the alignment of the current density and the reference density. The agreement factor of about 1% proves that the two densities are indeed very similar.

```
The density was aligned with the reference file, agreement  1.325%.
```

Final notes on the possibility to get more info with expanded log and record of the name of the infut file and end of the calculation:

```
You can obtain more information about the reconstructed reflection phases by using 'expandedlog yes'.

Electron density written to file Cr2P2O7_sf.m81.


---------------------------------------------
End of the calculation: 16.JAN 2006, 16:47:15
---------------------------------------------
```

Chapter 6

Some know-how or what to do if things go wrong

This chapter is intended to help you start using SUPERFLIP, getting your structures as quickly as possible and understanding potential problems. I will try to summarize my experience with charge flipping here. So, please, be aware that the following observations are necessarily incomplete, maybe even inaccurate or partially wrong, because they apply to a limited, albeit large number of structures solved by myself. Having said that, I still believe these notes will be helpful to everybody who is just getting started with charge flipping.

6.1 The value of δ

δ is the crucial parameter of the whole calculation. Unfortunately, no way has been found up to now to determine δ for a given data set a priori. δ depends on the scale of the amplitudes, on the resolution, on the value of the Debye-Waller factor and, of course, on the contents of the unit cell. Despite of that there are ways to determine δ empirically. The most reliable way that I have discovered is to compare the amount of total charge and flipped charge (see subsection 5.2.4 for definition). In a typical iteration the initial three transition cycles are followed by a plateau of several to several thousand cycles before the convergence sets on. On this plateau, many characteristics of the iteration remain approximately constant, including the R-value, the total charge and the flipped charge. I have made the observation that the value total/flipped charge should be in an ideal case somewhat less that one. If the value is too large (too little flipped charge), δ must be increased and *vice versa.* This procedure can be easily automated and the automatic determination of δ works just in this way.

This criterion is closely related to the fact that the realistic structures have approximately the same amount of "signal/background" ratio, i.e. the proportion of pixels bearing the significant density tends to be about the same in realistic structures. Of course, this is not true exactly, the organic structures will have more significant pixels that a structure with heavy atoms and the same unit-cell size. This can be taken into account and δ can be correspondingly modified, if the automatic search does not lead to convergence. The larger is the proportion of the significant pixels, the larger must be the ratio of the total/flipped charge.

Occasionally you will encounter a rapid convergence, which however does not yield any sensible map. This can be easily recognized on a very small R-value (usually around 1% or even less). In addition to that either the amount of the total charge or the amount of the flipped charge is close to zero. In the first case the δ is way too large, in the second case it is way too small. This can be easily understood. If δ is too large, virtually all pixels in the starting density will be flipped, which yields almost $\rho_{new} = -\rho_{old}$, so almost no perturbation is performed, the amplitudes remain almost the same and the R-value is very low despite of the density being completely random. Similarly, if δ is too small, only very little perturbation is performed, the amplitudes also do not change significantly and the R-value again stays low. The program does not detect the false convergence, so if you use a fixed value of δ, be sure to check if the resulting R-value and the total and flipped charge have reasonable values.

6.2 The symmetry

As already mentioned, charge flipping reconstructs the density without any other information about the symmetry than what is contained in the amplitudes of the structure factors. Thus, the symmetry is contained in the resulting density, but only approximately and the origin is shifted. Superflip therefore searches for the origin of the space group. This usually works pretty well. Problems occur only if the disturbance of the symmetry is too small to exhibit itself in the approximate density. Therefore, do not completely rely on the symmetry search in cases of superstructures with small deviation of the superstructure from the basic structure. In case of doubts, whether the symmetry operations are correctly located, rerun superflip and do not use `searchsymmetry average`, but `searchsymmetry shift`. This option will try to located the origin, but it will not average the density, but instead of that it will only shift it. So the original density remains preserved and you can analyse it for small symmetry disturbances.

Another problem can occur, if you work with a structure that is almost centrosymmetric, but a substructure breaks the centrosymmetry. In that case it is likely that charge flipping will find a centrosymmetric structure that will contain both orientations of the non-centrosymmetric substructure superimposed. On the other hand, a robust acentricity of the structure is properly reconstructed by charge flipping.

6.3 The convergence

It is usually very easy to assess the convergence by visually inspecting the curve of the R-value vs. number of iteration steps. The convergence is marked by a quick drop of the R-value from the plateau down to values typically between 20 and 30%. It is somewhat more difficult to automate the recognition procedure. Superflip can recognize the convergence in most of the cases, but it fails occasionally. The two cases when the automatic convergence recognition fails are:

- The convergence is too fast and no plateau has developed that could be used as a reference.

- The resulting R-value is too high and the difference between the R-value of the plateau and the converged R-value is too small to be recognized.

If you encounter this situation, just run superflip again with a limited number of cycles (see section 3.3) and select the number of cycles so that the convergence is reached within this number.

6.4 Charge flipping converges, but I cannot refine the structure!

Before trying to pinpoint the problem, one has to keep in mind one thing: For most of the easy to moderate structures the direct methods are the method of first choice. Therefore, charge flipping is tried most frequently on structures that direct methods fail on. Such a selection of structures is likely to have some intrinsic problem, often with the data quality or with twinning. If charge flipping converges but the resulting structure seems to make no sense or cannot be refined well, it is an indication of one of the following issues:

- The data correspond to a twinned structure. This is actually the most probable explanation.
- The data are only a subset of all reflections, some superstructure reflections have been omitted and thus the result is a superposition structure.
- The structure has a non-centrosymmetric substructure and charge flipping reconstructed a superposition structure with both non-centrosymmetric substructures superimposed.

With the exception of the last item, which is quite rare, the effect of ”convergence without refinement” is indicative of a problem in the data set rather that with the structure being too difficult to solve.

Bibliography

Oszlányi, G. and Sütő, A. (2004), '*Ab intitio* structure solution by charge flipping', *Acta Cryst. A* **60**, 134–141.

Oszlányi, G. and Sütő, A. (2005), '*Ab intitio* structure solution by charge flipping. ii. use of weak reflections', *Acta Cryst. A* **61**, 147–152.

Palatinus, L. (2004), '*Ab intitio* determination of incommensurately modulated structures by charge flipping in superspace', *Acta Cryst. A* **60**, 604–610.

Petříček, V., Dušek, M. and Palatinus, L. (2000), *The crystallographic computing system JANA2000*, Institute of Physics, Praha, Czech Republic.

Wu, J. S., Spence, J. C. H., O'Keeffe, M. and Groy, T. L. (2004), *Acta Cryst. A* **60**, 326–330.

Zhang, K. Y. J. and Main, P. (1990), 'Histogram matching as a new density modification techique for phase refinement and extension of protein molecules', *Acta Cryst. A* **46**, 41–46.