DIRDIF MANUAL

DIRDIF-2008

A computer program for Crystal Structure Determination by
Patterson Methods and Direct Methods applied to Difference Structure Factors

How to solve crystal structures using chemical knowledge

Paul T. Beurskens,   Gezina Beurskens,   René de Gelder, Jan M.M. Smits (Nijmegen),
S. García-Granda (Oviedo) & R.O. Gould (Edinburgh)

Crystallography Laboratory, Radboud University Nijmegen,
Toernooiveld 1 , 6525 ED Nijmegen ,   The Netherlands
E-mail: PTBeurskens@hetnet.nl [subject only: DIRDIF !]

DIRDIF MANUAL

Contents

PRIMER

ORBASE - GALLERY

THEORETICAL BACKGROUND

DEMO structures for DIRDIF

many of those NOT YET AVAILABLE

Chapter 1. Handout

DIRDIF-2007   A computer program for Crystal Structure Determination by Patterson Methods and Direct Methods applied to Difference Structure Factors

Notation: in this MANUAL all names of programs, parameters, keywords, etcetera, are written in capitals, but all input data may also be given in lower case lettering,
All data file names are defined in *lower case* although in this MANUAL capitals may be used.

Documents: files stored in directory dirdif / DOCS
• the PRIMER files [1-handout.txt, 2-intro-, 3-options-, 4-files-, 5-monos- 6-final.txt]
• the file gallery.pdf : a visualization of 'orbase' = data base of molecular fragments for Vector Search
• this DIRDIF MANUAL   which includes the PRIMER files and the ORBASE GALLERY, and
 - contains theoretical background about major (sub) programs PATTY , ORIENT , TRACOR , etcetera,
 - and the write-up for some DIRDIF DEMO structures.

Input data: the usual crystal data and observed reflection data, and
1. ===> for heavy atoms structures: nothing else !
2. ===> for Patterson - Vector Search methods: one <u>or more</u> known molecular fragments.
    Some options (see below) require atomic coordinates.

When to use DIRDIF for routine structure analysis:
1. For structures with heavy atoms, including P and S   call: PATTY
2. For structures of molecules with (partly) known geometry   ORIENT

When to use DIRDIF for (special) problem structures:
•- When ab-initio direct methods gave a misplaced fragment   call: TRACOR
•- For expansion of a (very) small fragment to the complete structure   PHASEX
•- For an enantiomorph, and/or a super- or pseudo-symmetry problem   PHASEX

How to run DIRDIF (CCODE identifies the structure)   [Change to working directory CCODE ...!]
                    (For demo test runs on structure MONOS, see file 4-monos.txt)
In most options, the structural parameters (atomic coordinates, either user-supplied or found by Patterson methods) are expanded and recycled (calling repeatedly the appropriate programs) to determine the complete structure. Output: atoms file.

1. Patterson option PATTY for Heavy Atom Patterson interpretation.
To solve a heavy-atom structure (including S or P in a light atom structure), use PATTY.
No input atoms needed. Enter
=====> DIRDIF CCODE PATTY

2. Patterson option ORIENT for application of Vector Search methods.
Vector Search methods are used when a (relatively small) part of a structure has known geometry.Prior to running ORIENT, the search model (atmod file, fractional or Cartesian coordinates) must be prepared. To verify the format of your prefab atmod file, or to retrieve a model from orbase (see ORBASE-GALLERY), and/or to modify the model interactively, enter:
=====> DIRDIF CCODE atmod
Then, for application of ORIENT (valid atmod file needed), enter:
=====> DIRDIF CCODE ORIENT

The orientation of the model is found by program ORIENT, using Vector Search. Thereafter, programs TRACOR and TRAVEC are called to position the model, and then PHASEX and FOUR to expand the structure to completeness.

Additional options:
•- Positioning of a misplaced fragment (fractional atomic coordinates given in an atomsS file):
=====> DIRDIF CCODE TRACOR
•- Expansion and recycling of a partial structure (fractional atomic coordinates in atoms file):
=====> DIRDIF CCODE PHASEX
•- For Fourier refinement of an almost completed structure (fractional coordinates in atoms file):
=====> DIRDIF CCODE FOUR

Special options:
•- To determine the absolute structure by calculating the Bijvoet - coefficient, enter:
=====> DIRDIF CCODE BIJVOET   (input: CCODE.ins [priority] or atoms)
•- To generate multiple search fragments for a flexible search model, incase the flexibility can be
        described by rotations around atom bonds:
=====> DIRDIF CCODE ORFLEX   (input: atoms, output many models in atmod)
•- For some other utilities, enter:
=====> DIRDIF CCODE INVERT   to invert the structural atomic parameters (input: atoms)
=====> DIRDIF CCODE AT2X   to convert atoms file into CCODE.res
=====> DIRDIF CCODE X2AT   to convert CCODE.ins file into atoms

In case of problems:
=====> DIRDIF H   : invokes a limited help session.
=====> DIRDIF   : starts an interactive run.
=====> DIRDIF CCODE PHASEX NORECY : call for PHASEX , but NORECY means: NO recycling!
=====> DIRDIF CCODE NORECY : call for FOUR , but NO recycling!
=====> DIRDIF CCODE NOFREE : re-assign atom names, following cell contents given in file
                crysin , using one Fourier synthesis.

Aliasses and abbreviations:
The instruction DIRDIFX may be prepared to call a plot program after the final structure expansion has been completed. The abbreviations DD and DDX may be used for DIRDIF and DIRDIFX, respectively. After the first DIRDIF run for a given compound, the compound code name (CCODE) may be omitted in all following calls.

DIRDIF files

The **filename** dictates the contents and the format of the file. Filenames sometimes are represented (in all programs , documents, and output listings) by capitals. The actual file names are in lower case , but some concatenate the name to the compound code (up to six characters, denoted CCODE).

Reflection data files (options); DIRDIF finds out which one is input:
fref   formatted reflection data file (F values)
CCODE.hkl   formatted reflection data file (F or F-squared;
                an HKLF record defines whether F or F**2 is given (default: F**2).

Crystal data files (options); DIRDIF finds out which one is input:
crysin   primary crystal data
CCODE.ins or CCODE.res   SHELX control data files (may contain an HKLF record).
CCODE.cif   the IUCr-ActaCryst CIF file (for crystal data only)

Atomic parameters files:

atoms   atomic parameters (in fractional coordinates) of the structure or of a part of the structure, either as input or as output. Various programs overwrite the atoms file with intermediate or final atomic parameters, but some input data are appended to file atold.

atmod   atomic parameters (fractional or Cartesian coordinates) of a molecular model (fragment) for input to ORIENT. This file can be prepared manually, or by calling ORBASE and retrieving a model from orbase/oruser. Various procedures overwrite this file but input data are copied/appended to the atold file for back up. The atmod file may contain more than one fragment (with Cartesian coordinates only), e.g. as generated by program ORFLEX .

atold   file with a collection of produced and/or used atoms files (= back up file).

CCODE.res   file with output atomic parameters in SHELX format. This file is written by program AT2X at the end of the recycling procedure.

CCODE.ins   file with refined atomic parameters in SHELX format for use in program BIJVOET.

CCODE.spf   atomic parameters in PLUTON format (A.Spek)

Other DIRDIF files: [2008: files lis1 and lis2 give too much nonsense now.]

lis1   output file with summary of results and comments to be inspected and (optionally) printed.

lis2   auxiliary output file with more (intermediate) information, only to be inspected when lis1did not show the complete structure and you want to figure out what went wrong.

ddlog   file on which the system keeps some data on the subsequent DIRDIF runs.

ddstop   file with information after an error stop.

Note: many more files will be generated: don't change any of them.

Results of DIRDIF

When the structure has been solved correctly the output atomic parameters are written onto the atoms file (and CCODE.res, CCODE.spf); one proceeds with this file as input to a plot or refinement program. If your structure has not come out correctly, the output listing files lis1 and lis2 give indications to detect where the solution of the structure (probably) went wrong, and you can set up an appropriate strategy to run DIRDIF again: either with non-default parameters, or with a changed model. Sometimes the automatic recycling in the expansion part of the procedure (PHASEX / FOUR) does not work properly (*e.g.* when the cell contents are far off): do inspect the results of the first cycle of PHASEX / FOUR or rerun DIRDIF without the automatic recycling. (An additional calling parameter NORECY suppresses the automatic recycling.)

To change the space group or the contents of the unit cell: change the crysin file .Note: after a change of crysin (especially the cell contents) you may reassign the atomic designation by entering: DIRDIF CCODE NOFREE.

Major change in DIRDIF-2008:

1. The former collection of many individual programs now are united into one single program DIRDIF.

2. Former computer dependent routines are removed: DIRDIF is believed to be computer independent.

3. Heavy atoms: the program analyses all possible PATTY solutions! or

4. Vector Search: the program will orient all given fragments (from atmod), TRACOR will position all oriented fragments, then TRAVEC analyses all positions for each orientation of each input model, and decides how many atom sets should be subjected to the PHASEX/FOUR recycling procedure.

5. The program ORFLEX can be used to generate (many) models of a flexible molecular fragment

6. The program BIJVOET is called and if appropriate (wrong abs.structure), the structure will be inverted.

7 All file names are in lower case (important for unix users !)

8. Conventions for the use of SHELX files in DIRDIF are changed: beware, res=output, ins=input!

Chapter 2 Introduction

DIRDIF-2008 A computer program for Crystal Structure Determination by Patterson Methods and Direct Methods applied to Difference Structure Factors

Notation: in this MANUAL all names of programs, keywords, etcetera, are written in capitals. as is used in the FORTRAN coding. All input data may also be supplied in lower case lettering but in some installations (unix!) lower case must be used for filenames.

Documents: the DIRDIF documentation, distributed with the program DIRDIF, consists of
• the PRIMER files : six tekst files with practical user information:
  1-handout.txt,   2-intro.txt, 3-options.txt, 4-files.txt,. 5-monos.txt, 6-final.txt .
  (they are ascii copies of the first 8 chapters of this manual).
• this DIRDIF MANUAL [which includes the PRIMER files = practical write-ups, and with]
   - the ORBASE GALLERY: a visualization of molecular fragments for Vector Search,
   - theoretical background about major (sub) programs PATTY , ORIENT , TRACOR , etcetera,
   - plus write-up for DIRDIF DEMO structures.

DIRDIF = using your chemical knowledge !

DIRDIF is a program for solving crystal structures. Major features are
• powerful Patterson methods for use of your chemical knowledge,
  both for heavy atoms structures and for equal atoms structures, and
• direct methods applied to (phases of) difference structure factors,
  with special techniques for solving symmetry problems, all followed by
• automated R2-driven dual-space structure expanding procedures.

The program is designed to operate under a wide variety of circumstances using automated as well as interactive routines. Most options are fully fully automated and can be used as a black box, but the (experienced) user has on-line control for adapting his strategy to the problem at hand. An on-line help-facility is provided; it is limited in scope and not meant to replace the present PRIMER files.

All programs are written in standard FORTRAN-77 and are believed to be fully computer independent.

When to use DIRDIF. Main options & programs

When to use DIRDIF for routine structure analysis:
1. For structures with heavy atoms, including P and S   call   PATTY
2. For structures of molecules with (partly) known geometry   ORIENT

When to use DIRDIF for special problem structures:
3. When ab-initio direct methods gave a misplaced fragment   call   TRACOR
4. For expansion of a small fragment to the complete structure   PHASEX
5. For an enantiomorph, or a super- or pseudo-symmetry problem   PHASEX

And more ....
6. For Fourier refinement of a partly completed structure   call   FOUR
7. For finding the absolute structure of a noncentrosymm. structure   BIJVOET
8. For generating search models for a flexible molecular fragment   ORFLEX

Two main options in using Patterson methods

1.   To solve a heavy atom structure (including S or P in a light atom structure) use the program PATTY. Input: crystal data and reflection data. No input atoms or control data needed. After locating the heavy atoms the structure is automatically expanded until completeness.

2.   To solve a structure with a partly known model (molecular fragment, often a rigid part of the molecule) use Vector Search methods. The search model (file with atomic coordinates) must be prepared in advance. To retrieve a model from the ORBASE fragment database and/or to modify the model and/or verify the format of your prefab file (all inter-actively), use the option ORBASE .
           To solve the structure with this search model, use the program ORIENT. After orientation, the model is automatically positioned and further expanded to the complete structure.

The main individual routines (denoted *'programs'*) ; a short write-up:

DDSTART and DDMAIN are system programs which are automatically executed when needed.

PATTY, ORIENT, TRACOR, PHASEX and FOUR are the main structure-solving programs. When the programs are called, the system automatically performs all calculations necessary for the completion of the structure.

PATTY   is a program for the interpretation of a sharpened Patterson. It uses Buerger's implication theory (*i.e.* the so called symmetry map) and checks all cross vectors using the minimum-function value as a selection criterion.
           The program is used for heavy atom structures with unknown heavy atom positions including not-so-heavy-atoms like S or P in a light atom structure).

ORIENT   is a program to find the orientation of a molecular fragment (model) by means of Vector Search methods (Nordman). The input model is used for the calculation of interatomic vectors. The shape function of a single interatomic vector peak is approximated from the shape of the origin peak of the Patterson function, and it is used for the calculation of the overlap inbetween neighbouring vectors of the model. Vectors to be used the search are selected on weight (including overlap), length, and mutual separation. A fast cyclic search system in angular space (Eulerian angles denoted A, B, C), employing increasing resolution per cycle, leads to the best fitting orientation of the model. The fitting criterion is the Nordman 'minimum average' function value.
           The program is used for structures with known geometry for a (relatively small) part of the molecule. Note that often such a molecular fragment (model) is available from the users own collection of related structures.

TRACOR   is a program to find the position of a molecular fragment (with known correct orientation) by means of reciprocal space correlation functions. The input fragment (the atoms file) is used for the calculation of partial structure factors for all reflections (for the entire expanded data set). The partial structure factor (which is the sum of partial structure factors of symmetry related fragments) depends on the vector **t** used for shifting   the input fragment to another position. The correlation between calculated and observed structure factors (intensities) determines the best value for the shift vector **t**. The calculations are done by the   Fast-Fourier-Transform method employing all symmetry elements simultaneously.
           The program is used for positioning a structural fragment with correct orientation but unknown position. The program is automatically executed in the procedure initiated by calling ORIENT (see above).

A correctly oriented fragment sometimes is available as the result of a failure of ab-initio direct methods; when a recognizable fragment does not allow expansion or refinement, then the fragment may be misplaced, though the orientation is correct.

TRAVEC is a program which is automatically executed after the execution of TRACOR: it is based on vector search methods. It is used to test the (maybe many) results from ORIENT and and TRACOR, and it decides which partial structure is promising enough to warrant further structure expansion by PHASEX.

PHASEX is a program to EXpand and refine the PHASes of the difference structure factors by direct methods. The input fragment (a correct but incomplete set of atomic parameters, for instance known heavy atoms or an oriented and positioned molecular model) is used to calculate normalized difference structure factors (Wilson-Parthasarathy) giving 'E1' values. Weights are calculated (Woolfson or Sim) and the E1 values with the most reliable phases are input to a modified tangent formula to refine the input phases and to calculate phases for unphased reflections.

The application of DIRect methods to the DIFference structure factors is particularly powerful:
• when the known part of the structure is only marginally sufficient to solve the structure,
• when the model has higher translation symmetry than the space group (superstructure),
• when the centrosymmetric input model comprizes an enantiomorph problem,
• when the known atoms comprise another pseudosymmetry problem such as a 'chicken wire' fragment.
The program recognizes the symmetry problem and uses a special symbolic addition procedure to solve the enantiomorph and/or origin ambiguity problem.

The program PHASEX is used for expanding a partial structure. The program is automatically executed in the procedures initiated by calling PATTY, ORIENT or TRACOR.

FOUR is a program for the calculation of Fourier and Patterson maps. It implies calculation of distances and angles, assignment and shuffling of new peaks into connected atoms, plotting of the asymmetric part of the structure. The program FOUR also arranges for further expansion of the structure and recycling (reactivating programs DDMAIN, PHASEX and FOUR). Note that the program's decision of how to assign peaks and which atoms to use is based on peak heights, partly peak - integral values and geometrical considerations, but not on valid chemical arguments: the user should apply his chemical knowledge and make the appropriate modifications in the final output atomic parameters.
The program uses input files generated by other programs (via DDMAIN). It is automatically executed in the procedures initiated by calling PATTY, ORIENT, TRACOR or PHASEX. When the known part of the structure is relatively large, program FOUR is executed instead of PHASEX because the difference structure factors then are unreliable.

Finally we describe some additional useful tools, options, parameters and programs.

ATMOD is an option which can be used to prepare a suitable model for input to the Vector Search program ORIENT. A model can be selected interactively from the ORBASE or ORUSER database of molecular fragments. (See ORBASE-GALLERY.) Some facilities are available for modifying the model.
Note: this option generates one model only, although many models may be (manually) stored in the atmod file.

ORFLEX is an intractive program to generate multiple search fragments for a flexible molecular fragment, in case the flexibility can be described by rotations around atom bonds.
Input model stored in file atoms, the resulting many models are stored in the atmod file.

DIRP1 is an option that can be useful for the solution of a structure in case the user is uncertain about the space group, the composition of the compound, and/or the position of some heavy atoms. The

option DIRP1 causes the reflection data to be expanded to space group P1 (or centered equivalent e.g. C1) and calls the program PHASEX for elucidation of the structure in P1. The input model may be, for instance, one heavy atom in the origin!

NORECY is an additional calling parameter which is used to suppress the automatic recycling procedure. The keyword NORECY is added to the calling parameters. The recycling should be bypassed only when the automatic procedure failed to solve the complete structure.

BIJVOET is a program to find the absolute structure by calculating the Bijvoet-coefficient from differences in observed intensities of Friedel-related reflections. The ideal result would be either +1 for the correct structure or -1 for the inverted structure. The result is also given in Flack notation.
(Input file: CCODE.ins [priority] or atoms.)

NUTS	is a collection of sub-programs for various utility functions which contains for intance:
AT2X	conversion of file atoms to file = CCODE.res and CCODE.spf for SHELX, PLUTON, etc.
X2AT	conversion of file CCODE.ins (SHELX format) to file atoms (DIRDIF format)
SHAT	shift atoms
EULER	rotation of a rigid fragment (by A,B,C, in angular space)
INVERT	inversion of atomic parameters given in file atoms

Chapter 3.   How to run the various options of DIRDIF

Preliminary comments on input/output files (for details see Chapter 4). Most programs need a reflection data file and a crystal data file. For the application of vector search methods (ORIENT), the user has to prepare the atmod file (with the a-priori known molecular geometry) either before the automatic execution of ORIENT, or (for small models) in an interactive session on request.

For some options (in cases with problem structures) the user has to prepare an atoms file. No control data is needed at the outset of a job.

Atomic parameters of all possible solutions obtained by programs PATTY, or ORIENT, TRACOR and TRAVEC, and also atomic parameters of some intermediate results (program FOUR) are stored in the atold file or in some other files (to be listed is Chapter 4).

The ddlog file keeps record of some data of subsequent runs.

When the structure has been solved you find results and comments on the lis1 and lis2 files and the atomic parameters of the structure on the atoms file as well as on a SHELX's CCODE.res file and on some other files (to be listed is Chapter 4).

Instruction syntax: DIRDIF [CCODE] [PROGRAM [PARAMETER(S)]]
  CCODE: compound code, name of the structure, working directory
    Note: after the first DIRDIF run for a given compound, the compound code (CCODE)
    may be omitted in all following calls..
  PROGRAM: program name or option,
  ====> means : enter the instruction (being in working directory CCODE)

For the execution of any of the structure-solving programs PATTY, ORIENT, TRACOR, PHASEX, and FOUR the user has the choice between the automatic mode and the interactive mode:
=====> DIRDIF CCODE PROGRAM   for automatic execution
=====> DIRDIF   for interactive execution
                                    In the interactive mode every question is provided with a help facility.

The execution of some additional options (see below) and the execution of the program NUTS and any of the programs collected in NUTS (AT2X, BIJVOET, etc., see Chapter 2) are interactive, see below.

Patterson option 1: run PATTY for Heavy Atom Patterson interpretation.
When the structure contains Heavy Atoms (including S or P in a light atom structure):

=====> DIRDIF CCODE PATTY

No input atoms needed. The system automatically arranges for the following procedure:
- first calculate the Patterson function (program FOUR),
- then locates the heavy atom(s)   (program PATTY),
- expand the partial structure (program PHASEX, followed by FOUR), and
- recycle several times (programs DDMAIN, PHASEX and FOUR) for complete structure elucidation.
  Output: structural parameters in the atoms file, and tranformed into xyzn = CCODE.res.
  Note : all acceptable PATTY solutions will be tested, and sorted on a FOM (figure of merit) based on the Patterson fit and the R2 value; the most probable solutions are partly expanded, and the result with the lowest R2 value is fully expanded to find the complete structure.

Patterson option 2: **run ORIENT for application of Vector Search methods.**

Vector Search methods are used when a (relatively small) part of a structure has known geometry. The known part usually (but not necessarily) is a rigid molecular fragment. The search model (atmod file, fractional or Cartesian coordinates) must be prepared before executing ORIENT (see Chapter 4 for the write-up of the atmod file):

=====> DIRDIF CCODE ATMOD

Procedures:
>   When the user has prepared an atmod file in advance (from literature data, molecular   modelling
>   or his own archieves): checking the format of the file.
>   Else: interactive retrieval of a model from the database ORBASE and/or ORUSER
>   (see the fragments listed in the ORBASE-GALLERY).
>   In either case: interactive fragment modification (add, delete or rename atoms, etcetera).
>   Output: an (updated) atmod file with Cartesian coordinates.

Note: the user may manually store more then one atoms set in the atmod file
>   (in this case, the atoms must be given in Cartesian coordinates)

=====> DIRDIF CCODE ORFLEX
>   When a flexible molecular fragment is available (stored in file atoms), and in case the   flexibility
can be described by rotations around atom bonds, the user is asked to supply these   bonds   and   the
program ORFLEX generates multiple search fragments.
>   Output: many search models, stored in the atmod file.

=====> DIRDIF CCODE ORIENT

Input: atmod file with one or more Cartesion molecular fragments.

When the user calls for ORIENT, the system automatically arranges for the following procedure:
- first check and perhaps rewrite the atmod file with the atomic parameters of the model,
- then calculate the Patterson function (program FOUR),
- search for the orientation of the model (pgm ORIENT), and repeat this for all search fragments,
- use translation functions to position the model according to space group symmetry (TRACOR) and repeat this for all accepted ORIENT results,
- call program TRAVEC to collect all results of TRACOR for all orientations for all input models, and to calculate a FOM based on TRACOR correlation results, TRAVEC vector fit results, and calculated R2 for all resulting trial structures. Output: selected trial structures.
- expand all selected partial structures, and recycle several times (programs DDMAIN, PHASEX, FOUR) for partial structure elucidation.
- select the best result, based on R2, and recycle again to achieve complete structure elucidation.

Output: atoms file (and other files) with final atomic parameters.

Additional options (for various kinds of problems)

=====> DIRDIF CCODE TRACOR

>   Input fractional atomic coordinates given in an atoms file. The program is used for expanding structural fragments with correct orientation but unknown position. The program is automatically executed in the procedure initiated by calling ORIENT. The program is explicitly called by the user in a number of cases

A correctly oriented fragment sometimes is available as the result of a failure of ab-initio direct methods; when a recognizable fragment does not allow expansion or refinement, then the fragment may be misplaced, though the orientation is correct. The user may supply this fragment (input: atoms file) and call for TRACOR.

The program is also a powerful tool for the elucidation of troublesome heavy atom structures. For instance, the origin and the next largest non-Harker Patterson peak define a pair of heavy atoms which can be used as a well oriented model to be positioned by the program TRACOR.

When the user calls TRACOR, the following procedure is executed:

• first expand the reflection data to a half-sphere and use the fragment to calculate partial structure factors (program DDMAIN),

• then find the position of the fragment (program TRACOR) and

• continue as described above (with TRAVEC), and structure expansion (with programs DDMAIN, PHASEX and FOUR) to complete the structure elucidation.
  Output: atoms file with atomic parameters.

Note: the user cannot call program TRAVEC individually.

=====> DIRDIF CCODE PHASEX

Expansion and recycling of a partial structure, i.e. when some atoms are known (on correct positions). Input fractional atomic coordinates are given in the atoms file. The program is automatically executed after PATTY, ORIENT or TRACOR.

The program is explicitly called by the user in a number of cases.

• The user should call for PHASEX when he has his own suggestions for atomic positions: for   instance he may have modified the atoms in the atoms file available from a foregoing DIRDIF run (which, of course, is only useful if something went wrong ...).

• When the user calls for PHASEX, the system automatically arranges for structure factor calculation and normalization (program DDMAIN), then executes the program PHASEX to expand and refine the phases of the difference structure factors, calculates and interprets a Fourier synthesis (program FOUR), and finally organizes recycling several times (programs DDMAIN, PHASEX and FOUR) for expansion of the fragment and completion of the structure.
  Output: atomic parameters in atoms file.

=====> DIRDIF CCODE FOUR

The program FOUR is automatically executed after PATTY, ORIENT, TRACOR or PHASEX. The program is explicitly called by the user in a number of cases similar as for PHASEX (see above). Input: atoms file.

• When the user calls for FOUR, the system will automatically arrange for structure factor calculation (by program DDMAIN) and then calls program FOUR for a default Fourier synthesis. The program FOUR then arranges for recycling (programs DDMAIN and FOUR) until the structure evaluation is completed.   Output: atomic parameters in the atoms file.

=====> DIRDIF CCODE BIJVOET

This call for the NUTS option BIJVOET is especially useful after the structure has been refined (for instance by SHELXL) for finding the absolute structure in case of a non-centrosymmetric space group. Input atomic parameters: CCODE.ins [priority] or atoms . Output only when structure inversion is required: CCODE.res and atoms

=====> DIRDIF CCODE AT2X
=====> DIRDIF CCODE X2AT

AT2X is a subprogram for the conversion of the (final) atoms file into files for other propgrams (SHELXL, PLUTON). One output file is CCODE.res .

X2AT converses a SHELX CCODE.ins file into an atoms file.

=====> DIRDIF CCODE NUTS

This call invokes an interactive session for the execution of various utility calculations.such as: AT2X, X2AT, BIJVOET, SHAT, EULER, INVERT: call NUTS for more information.

The programs AT2X and BIJVOET, and maybe INVERT, are automatically executed in all structure solving procedures after the final execution of program FOUR.

=====> DIRDIF CCODE CRYSDA

To prepare a 'permanent' crysda file with extended crystal data. Usually the crysda file is generated automatically, and deleted at the end of a job. This call is useful when the user wishes to inspect the extended crystal data. At next call for DIRDIF, the crysda file is erased. When the user wants to modify the crystal data (e.g. space group or cell contents) he should make those changes in the crysin file.

In case of problems:

=====> DIRDIF H

Invokes a short help session. (No CCODE given, no data needed.) For the new DIRDIF user it is really useful to try out all possibilities in order to get used to the system.

=====> DIRDIF   Starts an interactive run.

• When DIRDIF is activated in interactive mode the user is asked to select an option or program (ORIENT, PATTY,...) and then whether or not special control data is wanted. Interactive help facilities are available. For a first run we strongly advise to use the default values.

=====> DIRDIF CCODE DIRP1

Starts a procedure which helps the user to solve the structure in space group P1 or centered equivalent.

• It is used in case the space group, the composition of the compound, and/or the position of some heavy atoms are very uncertain. The input partial structure (atoms file) may be, for instance, one atom in the origin!

• Procedure: the reflection data are expanded to space group P1 (or centered equivalent e.g. C1) and the program PHASEX is executed for elucidation of the structure in P1. (The one-atom case is made asymmetric by the enantiomorph-fixing procedure.) After inspection of the results the user decides how to continue. There is no automatic recycling.

• [2008 : at present, this option DIRP1 is not anymore of importance.]

=====> DIRDIF CCODE FOUR NORECY
=====> DIRDIF CCODE NORECY

Prepare one more Fourier systhesis with interpretation, which may be very useful if just a few atoms are renamed, or the cell contents are slightly modified in crysin.

Restarting DIRDIF - summary

When you want to rerun one of the options of DIRDIF you have to consider which atomic parameter set is to be used as input. You can start your own recycling procedure using the existing atoms file (output from last DIRDIF run), or you can select one of the parameter sets stored in the back-up file ATOLD and copy it to the atoms file. Use local editing facilities to modify the atoms file to suit your idea of the best set of atomic parameters. To decide which option of DIRDIF you want to call, consider their consecutive actions:

```
            heavy atoms          Vector Search                completion
            or   super-      fragment     fragment        fragment    make
                 structure   orientation  positioning     expansion   Fourier
call:
PATTY  :    PATTY   --------------------------------------------->  PHASEX --> FOUR
ORIENT :                     ORIENT  --> TRACOR  ------>  PHASEX --> FOUR
TRACOR:                                  TRACOR  ------>  PHASEX --> FOUR
PHASEX :                                                 PHASEX --> FOUR
FOUR   :                                                             FOUR
```

Chapter 4. DIRDIF file definitions

Filenames are dependent on the computer and on local use. The different files of the DIRDIF system are referred to by its functional type. (The filename dictates the contents and the format of the file.) File names always are written in lower case, although within the FORTRAN programs and in some documents, filenames are represented by capital letters. Some files maybe concatenated by compound code. Example: for the test compound MONOS the primary crystal data is given in the crysin file while output parameters are given in the atoms file and in the file monos.res .

Input data:
- DIRDIF needs an input reflection data file.
- The primary crystal data may be supplied manually,
  but it is preferred to prepare the crysin file in advance, using local conversion programs.
- For some options atomic parameters files are needed:
      for ORIENT: the model parameters => atmod,
      for TRACOR or PHASEX: parameters of the partial structure => atoms.

Standard file structure: Most files consist of free-format records of at most 72 characters each.
                                   The order of words (literals, numbers) in a record usually is fixed.
- The first record usually is a header record with at least FILENAME and CCODE.
- The first word of a record is a keyword for identification.
- REMARK records (keyword=REMARK) (with printable information) may be inserted anytime.   •
The last record is an END or a FINISH record.
Note: reflection files have fixed format; REMARK records are not permitted.

4a. Listing files lis1 and lis2

The system produces a file for printing (lis1 = printable output) which gives the most important information on the solution of the structure. In addition a longer listing file, lis2, is produced which gives information on the input data, the execution of the various programs, and their results. Inspect the file lis2 only if you are interested or when the structure did not come out as you hoped or expected. With the aid of the detailed information you might be able to detect where things went wrong, then change input data and start DIRDIF again. Certainly lis1 and lis2 should not be printed routinely. But if things really go wrong, do send the lis1 and lis2 files to Nijmegen: we will be glad to help you!
Note: before next run is executed, the files lis1 and lis2 are copied to lis1x and lis2x, repectively .

4b. Atomic parameter files atoms and atmod

The input and output atomic parameter files of the DIRDIF system are:
- atoms file: input to most programs, overwritten with/by output parameters,
- atmod file with the model parameters input to the program ORIENT,
- atold file: a collection of parameter sets, to be used as back-up file,
- CCODE.ins and CCODE.res : SHELX control data files
      Note: CCODE.res is a file ready for input to the SHELXL least-squares refinement program.
- CCODE.spf : input for PLUTON (A.Spek),
- schakal : input for plot program SCHAKAL.

The atoms file consists of the following records, each containing a keyword followed by data:
- atoms   CCODE   more-info   (CCODE = compound code)
- ATOM   atomname x y z   (x,y,z: fractional atomic coordinates)
                                      (one atom / record, as many as needed)
- REMARK   comments   (optional, as many as desirable)
- END   (last record)

The atomname begins with the chemical symbol and may be followed by one or more characters (*e.g.* C7, C+7, C7+, C7A are carbon atoms; CA is a calcium atom, CX is an error).
Alternatively the atomname may consist of the chemical symbol, one or more blanks, and one unsigned integer number (e.g. C 27).
Uninterpreted (residual) peaks of a Fourier map are given atom name = Q .

It is possible to supply
- a site occupancy factor sof (sof = 1 also for atoms on special positions; sof < 1 for disordered atoms),
- and an isotropic temperature factor (B) on the ATOM record, but do so only if you are sure about the data, because it will have a significant effect on the scaling procedure.

When the structure has been solved, the output ATOM records are provided with a site occupancy factor (sof = 1) and an isotropic temperature factor (B):   ATOM   atomname x y z sof B

At the end of a structure solving run, the program AT2X converts the output atoms file to a SHELX type file CCODE.res , and to CCODE.spf and schakal files (input to graphics programs PLUTON and SCHAKAL, respectively).

Note: the atoms file may contain more atoms sets, after the closing 'END' may follow another 'atoms' record, etc., but in most cases only the first atoms set is used.

The atmod file has the same structure as the atoms file.

Possible header records are:
ATMOD   MCODE   more-info   (MCODE = Model code)
ATMOD   MNUM MCODE   (MNUM = Model number)
ATMOD   MCODE   MCELL a b c alpha beta gamma
ATMOD   CART
ATMOD   MCODE CART MNUM
ATMOD   CCODE   (using cell of present structure = CCODE)

ATOM records with atomic parameters (one atom / record, as many records as needed) contain either
        fractional coordinates: ATOM   atomname x y z
or   Cartesian coordinates: ATOM   atomname X Y Z
For the atom name, see under atoms file.

REMARK records can be inserted (after the header) whenever needed.
END is the last record.

Notes:
The information CART (for Cartesian) is optional as DIRDIF finds out whether the parameters are fractional or Cartesian. The information 'MCELL a b c alpha beta gamma' is necessary only when the fractional atomic parameters of the model or fragment are represented in a unit cell that is different from

the present compound CCODE. (In stead of 'MCELL' also 'CELL' is accepted.) In an interactive session the MCELL data can also be provided at the terminal.

Atomic parameters of a known molecular model can be retrieved from the DIRDIF-ORBASE fragment file at an interactive terminal session. For larger structures these fragments may be too small. The Vector Search method can often be employed more powerful if you retrieve molecular models from your own solved structures, or from the literature, or by molecular modelling. It is convenient to prepare an atmod file in advance, and modify the model (delete, rename, and add atoms) interactively.

The atmod file, described so far, is input (e.g. by instruction: DIRDIF CCODE ORBASE), and after checking, editing, and possible re-orientation, a new atmod file is output with Cartesian coordinates (the original input file is saved in the atold file for back-up).

Note: the atmod file may contain more atoms sets, after the closing 'END' may follow another 'atmod' record, etc., and each atmod set is used by ORIENT. If more than one model is given, checking of the models is limited and all models must be given in Cartesian coordinates.
Such a multiple atmod file may be created by program ORFLEX.

4c. Crystal data files crysin and crysda

crysin   : primary crystal data: standard DIRDIF input file.
CCODE.ins or CCODE.res: SHELX control data files (contains the HKLF record, see 3d)
CCODE.cif :   IUCr-ActaCryst CIF data file for crystal data only
crysda: extended crystal data, generated by subroutine crysda, removed after each run.

The program CRYSDA (usually called automatically) reads crystal data from a crysin file (highest priority) and/or from other input possibilities (existing crysda, INS/RES, CIF, keyboard) and produces a crysda file which contains the input crystal data and extended data such as cell volume, calculated density, tables of scattering factors, etc. If no crysin file was available, or if the crystal data was modified interactively, a (new) crysin file will be output. The crysin files is to be kept. Normally, the crysda file is deleted at the end of the job.

The crysin file contains the following records:

```
CRYSIN  CCODE   more-info      (header)
TITLE         any user supplied information      (optional, to be printed)
CELL          a b c alpha beta gamma             (Angstrom, degree)
CELLSD        esd's                              (six numbers)
SPGR          e.g. P 1 or P 21 21 21 or R -3     (axial directions are ( separated by blank(s))
FORMUL        At1 Nr1 At2 Nr2 At3 Nr3 ......     (Ati=chem.symbol,  Nri=nr of atoms Ati )
              At7 Nr7 At8 Nr8 At9 Nr9 ......     (continuation record allowed, max. 10 atom kinds )
                    [Example: for  Na2CO3.7H2O:  FORMUL  NA 2  C 1  O 3   H 14  O 7]
Z             number of FORMUL units / cell      (cell contents = Z * FORMUL)
                                                 (Note: Z is not a symmetry factor !)
WAVE          Cu or Mo or Fe or Ag or Cr         (one atom type; no number)
ORIN          crystal orientation matrix         (OPTIONAL,  3 records)
HKLF          number                             (3 or 4 to indicate F or F**2 on HKL file)
END
```

Notes: When during the crystal structure analysis you wish to alter the cell contents or the space group, you should do this in the crysin file.

4d. Reflection data files

Input :   fref   or   CCODE.hkl

                     DIRDIF finds out which input data file is present
                     Formats of the input reflection data files:

fref : formatted reflection data file, 28 characters/record

                                   (standard DIRDIF file) with Fobs values

       first record: header with 'FREF' (or 'FREFA' or -B or -C) and CCODE

       following records: one reflection each, FORMAT (A1,3I3,I2,F9.2,F7.2)

                                   for: ' ', h, k, l, JC, Fobs, sigma

                                       JC=2 for 'unobserved' or 'unreliable',

                                         else JC=1 or blank

       last record: 'E'

CCODE.hkl: SHELX formatted reflection data file,   28 characters/record

                             with |Fobs| or |Fobs|**2 values (defined by a HKLF record: default F**2)

       First record: HKLF header (optional, not SHELXL convention)

                     First word: 'HKLF' on columns 1 - 4

                     Second word: the CCODE (optional, not checked)

                     Then: one number, either 3 or -3 : |Fobs| expected,

                                    or 4 or -4 : |Fobs|**2 expected !

       Following records: 1 reflection/record,   FORMAT (3I4, 2F8.2)

                                   for: h, k, l, |Fobs|,   sigma or: h, k, l, |Fobs|**2, sigma

                                   (Note: the SHELXL batch number on cc. 29-32 is ignored.)

       Last record: h = k = l = 0 (or: all blanks)

(Note about the SHELXL indices transformation matrix Rij given on the HKLF record: This feature is available, but should be used with care !! It is not used on crystal data!)

Mind that a CCODE.cif file or a CCODE.ins/res file (SHELX) can only be used for crystal data input, not for reflection data input.

4e. ddlog file ('readable data')

This file contains a summary of DIRDIF runs with pertinent data. This file is to be kept.

4f. orbase and oruser files

orbase : a data base with molecular fragments.
oruser : a private extension of orbase (with your own favourite models) A write-up of these files is given in the header lines of these files. The user is urged to add (manually) his own structural molecular fragments to the file oruser for future use when solving 'similar' compounds.

4g. Note about the use of CCODE.ins and CCODE.res files

Note: DIRDIF uses these SHELX control and parameter files as follows:
*   CCODE.ins and CCODE.res files may be used to create the crysin file
*   CCODE.res is output after structure expansion and if called by AT2X
*   CCODE.ins is input to X2AT to create an atoms file, and
       CCODE.ins has priority input when the user calls program BIJVOET , in which case
               - if the structure has to be inverted - CCODE.res is output.

Chapter 5.   Examples (structure MONOS)

You may wish to get acquainted with DIRDIF by running an example.
We have provided the data for the test structure MONOS.

Preliminaries

- Change working directory to dirdif / MONOS
- Run the DIRDIF system for help.
    The help facility can be used without the presence of the data of the test structure.
- Enter at the terminal: ====> DIRDIF H
  You will be given some information.
    Please, try out some possibilities, in order to learn about various conventions and options.
- Use your local editor to look at the MONOS data files .
    The crystal data for MONOS are given in the crysin file. The molecule contains a sulfur-
    bridged six-membered ring which is given in ORBASE under the model name MONOS.
- What to do if more (unknown) data files are present (e.g. from former test runs)? Nothing !

Proceed to run DIRDIF with the data of test structure MONOS, solving the structure of MONOS along
six different routes, depending on the apriory information we assume to know:

route 1: call program PATTY, using DIRDIF in automatic mode   (RUN 1),
route 2: call program PHASEX, assuming to know the S position   (RUN 2).
route 3: call option atmod, then program ORIENT   (RUN 3+4).
route 4: call program TRACOR, using one S in the origin for fun   (RUN 5).
route 5: call program ORIENT, using three models in file atmod (RUN 6).
route 6: call program ORFLEX to create multiple models in atmod   (RUN 7).

RUN 1.   Route 1: option PATTY in automatic mode

We know there is a sulfur atom, but we assume not to know its position. We start an automatic (default)
run of DIRDIF program PATTY. The following files are input:
crysin   crystal data
fref   reflection data file

Enter at the terminal:
=====> DIRDIF MONOS PATTY

The program PATTY finds the sulphur atom at a pseudo-special position, with x=0.0 !
This site generates a mirror plane in the model.
To handle this problem, the program PHASEX runs through an enantiomorph fixing procedure. The
course of the recycling procedure can be followed on the screen. When the program has finished the
structure has been solved. The lis1 file gives the most interesting features of the procedure and a line-plot
of the structure. The atoms file contains the parameters of the atoms of the structure. It appears that all
atoms are correctly nominated (S, O, N, C).
[2007: one interchange: O <--> N ! ??]

The following files have been created (look at these files using your local editor):

atold   atomic parameters of consecutive steps in the procedure
atoms   atomic parameters of the complete structure
monos.res converted atoms file to SHELX format
ddlog   information on this run and some important data
lis1   file with details of the DIRDIF run, suitable but too long (!) for printing
lis2   (ignore, use only in case of problems)

The information on the atold file and on the ddlog file will be extended in following runs of DIRDIF. The files atoms, xyzn, lis1 and lis2 files are overwritten in a next run. So, do not delete the files that have been created by this run before you run RUN 2.

RUN 2. Route 2: option PHASEX

Assume for test RUN 2 that we know the position of the sulfur atom. To put in the position of the sulphur atom you modify the file atoms which has been created in RUN 1 so that it contains the atomic parameters of the sulphur atom only. So make the atoms file to contain:

atoms   MONOS
ATOM   S -0.020 0.098 0.146
END

The following files now are available for input: atoms, crysin, fref

We start an automatic (default) run of DIRDIF program PHASEX.
Enter at the terminal:
=====> DIRDIF MONOS PHASEX

The sulphur position on x = -0.020 does not have the pseudo-symmetry which occurred in RUN 1, so PHASEX does not run through the enantiomorph fixation. (Note: x=+0.02 gives the enantiomer!) When the program has finished the structure has been solved, the lis1 file shows the structure, and the atoms file contains the parameters of the atoms of the structure. The final results are almost identical to the outcome of RUN 1. (Note: one can not predict whether PATTY finds a positive or a
negative x value for the sulphur position).

The following files have been re-created (look at these files using your local editor):

atoms   atomic parameters of the complete structure
monos.res converted atoms file to SHELX format
lis1   file for inspection
lis2   (ignore, use only in case of problems)

New results have been appended to the following files:

atold   atomic parameters of consecutive steps in the procedure
ddlog   information on this run and some important data

RUN 3+4.
Route 3: option atmod + program ORIENT in interactive mode

Assume that we know a rigid fragment of the structure, which is available in the ORBASE file. We start (RUN 3) with calling ORBASE for an interactive retrieval of the rigid fragment from ORBASE as a set of atomic parameters (7 atoms) which is to be stored in file atmod.
Then (RUN 4) we call an automatic run of ORIENT.

The following files are available for input: crysin and fref .

RUN 3, enter at the terminal:
=====> DIRDIF MONOS ATMOD

In the following dialogue you may also answer in lower case.

On the screen appears	*You answer at the terminal:*	
(Please give TITLE	anything)	
No atmod file. Can you supply the atomic		
parameters now at the terminal (T)		
or did you select or do you suggest an item		
from ORBASE (S)	S	
• Enter model code or number	MONOS	
Schematic picture of the model. Just try some things		
• Enter first letter of Edit option	X	10
• Enter first letter of Edit option	X	80
• Enter first letter of Edit option	G	S1
• Enter first letter of Edit option	Q	
• Is this result acceptable? (Y/....)	Y	

The atmod file with model coordinates (Cartesian) is output.

RUN 4, enter at the terminal:
=====> DIRDIF MONOS ORIENT

The program ORIENT reads the model and rotates it, the program TRACOR shifts it to the correct position (verified by TRAVEC) This is done for all acceptable orientations. Then the program PHASEX expands the models , and the resulting structure with the lowest R2 value is further expanded to the complete structure. When the recycling procedure is finished, the structure is solved. The lis1 file shows some intermediate results and a line-plot of the structure. The output atoms file contains the parameters of the atoms of the structure. It appears that two nitrogen atoms are placed at carbon positions, and v.v.

The following files have been updated (look at these files using your local editor):

atold	atomic parameters sets of various steps in this run (and in former runs)
atoms	atomic parameters of the complete structure
monos.res	converted atoms file to SHELXL format
ddlog	information and data on this and preceding runs
lis1	file for inspection
lis2	(ignore, use only in case of problems)

[2007] RUN 5,6,7 for Route 4,5,6 will be described later.

Chapter 6. How to interpret the results, How to restart

How to interpret the results

Use your own graphics and your chemical knowledge to edit the final parameters (maybe delete or rename some atoms). The table of bond distances and angles in lis1 will be of help. lf necessary, restart DIRDIF to find some more atoms (use: DIRDIF CCODE FOUR NORECY).
The final CCODE.res file is ready for use by the program SHELXL.

Trouble shooting

The best way to learn about DIRDIF is to use it as a routine tool for solving crystal structures. Although DIRDIF is designed for delivering automatically the complete set of atomic positions, it is useful to read some of the output listings (lis1) in order to learn about the way things are done for normal structures.

1.   The 'NORECY' option is used in case you hope to find a chemically reasonable fragment from a Fourier peak list in those troublesome cases where the automatic recycling failed. In this case, however, the R2 criterion is not used for rejecting atoms.

2.   A failure of the computer or a technical error in our programs may lead to a supervisor-interrupt, in which case the system may stop without properly deleting or closing various files. This should not cause problems at the next run!

3.   In case of a technical program failure, please give us the details: we wish to correct the programs. In case the DIRDIF system cannot solve your structure, please let us know: maybe we can help, probably we can learn from it.

4.   About ORIENT: a very small fragment, especially a simple 5- or 6-membered ring, fits almost everywhere in the Patterson. As a consequence, many orientations are acceptable, and for each orientation many position may be accepted by TRACOR, which all have to be analized! Try to find a bigger fragment with more characteristic geometry, even at the cost of accuracy of the model.

5.   Reading messages and looking at numbers in the output lis1 file ?

• If an uninterpretable error message occurs, write to us: we know and can tell you.
        And we wish to improve the message.
• Look for error messages or possible WARNINGS.
• Are the temperature factors normal ?
• Are high-order reflections adequately measured (not too many unobs)?
• Look at the Patterson peaks: is all O.K. ? No apparent space group error?
• See if (for ORIENT) the Patterson origin peak is about zero.
• See if (in PHASEX) the average E**2 converges to about 1.00
  ..... and the symbol-consistency   decreases to below 0.50
  .... and the number of participating reflections is 'normal' ......
• How is the distribution of peaks in the final Fourier map? Too many clusters?
By looking at those numbers and messages after the structure solution of normal structures, one knows what to expect, and one can often find clues in the output lis1 file for failures or problems encountered with 'difficult' structures. (The output listing file lis2 also might give information.)

[2007: at present lis1 and lis2 give too much ... to be shortened !]

Restarting DIRDIF

If your structure does not come out as you wish or expect, and you have detected where the solution of the structure (probably) went wrong, you can rerun part of DIRDIF either with non-default parameters, or with a changed model (ORIENT), or with mofified crysin data .

Sometimes DIRP1 is an interesting option (especially if many things are uncertain). It requires that the user selects his own set of atoms, and (when nesessary) updates the cell contents in the crysin file, and interactively modifies the scale factor and the temperature factors. As early as possible he must find possible positions of the symmetry elements, and select atoms in such a way that the superfluous artificial symmetry is reduced. Experimental TRACOR runs may help to locate the symmetry elements. The set of atoms may be shifted using the program NUTS/SHIFT.

Note: it is easy to solve a space group uncertainty by restarting DIRDIF using different space groups (just modify the space group in the crysin file !).

[2007: the distinction between atoms with almost the same scattering power (C <-> N, N <-> O, S <-> P, ...) is reduced in the new DIRDIF version: our former procedures too often led to erroneous results; we hope to have an inproved procedure in some months time.]

Chapter 7.   Notes for various computers

Some technical details are given in the various readme-, contents-, and EXEC- files, when relevant for the implementation of the distributed DIRDIF files.

When you start working on a new compound you must first create a directory, probably named after your compound code CCODE, in which all files relating to this compound are (will be) stored. When working on a particular compound you have to switch over (change directory: cd) to its directory.

Filenames: all file names are in lower case, although written in this manual in capitals!
The file names express the function of the files, and are identical for all compounds.
Hence the importance of using the correct directory.
Example for PC and unix: the file with name 'crysin' is written here as crysin
Example for VAX: the file 'crysin.DAT' is witten here as crysin .

Hardware and software requirements for DIRDIF PC-version: no special requirements.
Operating system: we have been working with Windows-2000 and WINDOWS-XP, using the MSDOS prompt (Windows > instruction processor).
Dr Louis Ferrugia (Glasgow) has prepared a WINDOWS version.

 [2008: exit ?]
How to use PLUTON (A. Spek) for DIRDIF results on a PC
At the end of a successful DIRDIF run the file CCODE.SPF is ready for use by PLUTON.
•.Enter at the terminal: =====> PLUTON CCODE   (asuming that the path to pluton has been defined)
•.For a first model you may enter respectively: labels on / stick color / plot / quit .
•.For a nicer plot, use: labels on / straw color / omit q / plot / quit .

Note: the file CCODE.spf (generated by AT2X) contains some of these instructions at the end, which means that routinely you will get a nice view immediately.

We hereby gratefully acknowledge and thank Dr. A.L. Spek from the University of Utrecht for allowing us to use and distribute his plot program PLUTON together with the DIRDIF program system. Normally the program is available free of charge from the author for use within the academic community under the condition that it is not redistributed. A licence fee is charged to profit organizations. Therefore you are not allowed to forward this program to others without asking explicit permission from the author: Dr. A.L. Spek, Kristal- en Structuurchemie, Universiteit Utrecht, Padualaan 8, 3584 CH Utrecht, The Netherlands.E-mail: SPEA@CHEM.RUU.NL
[end of PLUTON ---> the input to PLATON will be investigated ???]

Chapter 8. Acknowledgements and References

The following PhD-students (a), co-workers (b) and special guests (c) have greatly contributed to the development of DIRDIF and its sub-programs:

a. H.J. Bruins Slot, H.M. Doesburg, Th.W. Hummelink, O.R. Israel, E.J.W. Lam, P.H.J.Prick, Th.E.M. Van den Hark,
b. G. Admiraal, H.J. Behm, W.P. Bosman, J.H. Noordik, C. Smykalla,
c. R.C.Haltiwanger, V. Parthasarathi, S.B. Sani, G.F. Schaefer, M. Strumpel, W.K.L. Van Havere.

The following colleagues have contributed to the implementation on various computers:
    G. Baudoux, J.P. Declercq, R. Driessens, A. Meetsma, R. Olthof-Hazekamp, A.L. Spek, N.P.C. Walker

Special thanks for technical advice and for peparing the WINDOWS version of DIRDIF are due to
    L.J. Farrugia .

For part of this research financial aid was obtain from the Dutch National Science Foundations
    FOMRE, SON and STW.

Selected DIRDIF references   [2007 ... to be updated]

- Program PHASEX, general procedures: Van den Hark, Th.E.M., Prick, P.A.J. and Beurskens, P.T. (1976) Acta Crystallogr. A32, 816.
- Pseudo-symmetry: Prick, P.A.J., Beurskens, P.T. and Gould, R.O. (1983) Acta Crystallogr. A39, 570-576.
- Statistical procedures: Beurskens, P.T., Bosman, W.P., Doesburg, H.M., Van den Hark, Th.E.M., Prick, P.A.J., Noordik, J.H., Beurskens, G., Gould, R.O. and Parthasarathi, V. (1983) Conformation in Biology, R. Srinivasan and R.H. Sarma, eds. (Adenine Press, New York), p. 389.
- The DIRDIF program system, general: Beurskens, P.T. (1985) Crystallographic Computing, Vol. 3, G.M. Sheldrick, C. Krueger and Goddard, eds. (Clarendon Press, Oxford), p. 216.
- Program ORIENT: Beurskens, P.T., Beurskens, G., Strumpel, M. and Nordman, C.E. (1987) Patterson and Pattersons, J.P. Glusker, B.K. Patterson, and M. Rossi, eds. Clarendon Press, Oxford), p. 356.
- Program TRACOR: Beurskens, P.T., Gould, R.O., Bruins Slot, H.J. and Bosman, W.P. (1987) Z. Kristallogr. 179, 127.
- PHASEX phase expansion procedure: P.T.Beurskens and C.Smykalla (1991), Direct Methods of Solving Crystal Structures, ed. H.Schenk, Plenum Press, New York and London, pp. 281.
- Program PATTY: Beurskens, P.T., Admiraal, G., Behm, H., Beurskens, G., Smits, J.M.M. and Smykalla, C. (1991), Z. f. Kristallogr. Suppl.4, p.99.
- Program BIJVOET

Reference to DIRDIF-2008

Please refer to the present program as:

P.T. Beurskens, G. Beurskens, R. de Gelder, J.M.M. Smits , S. García-Granda, R.O. Gould ,
DIRDIF-2008 *, Crystallography Laboratory, Radboud University Nijmegen, The Netherlands.*

===

End of PRIMER files

    Following sections:

ORBASE - GALLERY
                    The following 2 Chapters of this manual are provided with the DIRDIF implementation as a separate file, 'gallery.pdf', stored in directory 'dirdif/docs'.

THEORETICAL BACKGROUND

DEMO structures for DIRDIF

Chapter 9 .   Inroduction to   DIRDIF.ORBASE GALLERY

A visualization of a database of molecular fragments (file DIRDIF.ORBASE) which can be input to the program ORIENT for the solution of crystal structures using vector search methods.
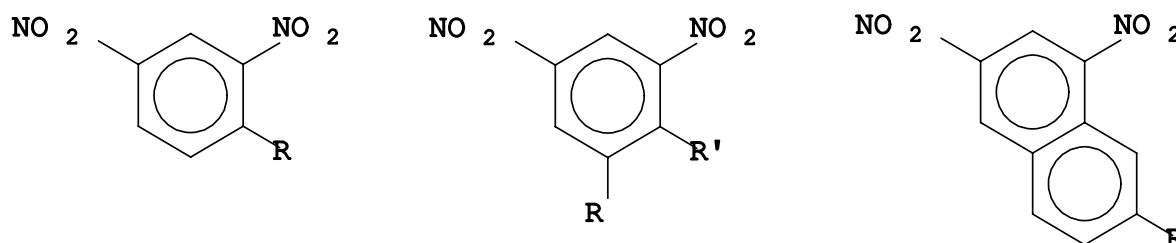
Why using vector search or molecular replacement for small molecules?

-       Because it makes sense to use available chemical information.
-       If you have a good fragment, vector search is stronger than direct methods only.
-       Because it is fun seeing your fragment growing.
-       Because by using vector search you get familiar with it and you can use it when you need it: in some problem cases it is the only way to solve your structure.

And why not?

-       It is easier than you might think it is.
-       You do not need to be a crystallographer.
-       The only thing to do before an automatic structure solution is selecting a proper fragment.

Your interactions with DIRDIF.ORBASE are:

Retrieving ready-to-use fragments: rigid fragments, or fragments with one or more preferred orientations (e.g. nitrobenzene, steroids) are available in this file. (Let your fingers do the walking!)

Storing your own fragments: when you work on a series of related compounds you can store your coordinates in a file DIRDIF.ORUSER (this is a 'personal' file, analogous to the system file DIRDIF.ORBASE).

Modifying a fragment to suit your specific needs (e.g. replace C by N, add a methylcarbon, fuse an aromatic ring).

How to use vector search?

When the vector search program ORIENT is called, a molecular fragment is used as input. The program determines its correct orientation, and the DIRDIF system passes automatically control to the program TRACOR to put the fragment on its correct position relative to the symmetry elements. Then control is passed on to the program PHASEX for phase refinement and phase expansion (using direct methods

applied to difference structure factors). Finally the program FOUR organizes an automatic recycling procedure for completion of the structure.

Running the program ORIENT: the program reads the parameters of the fragment or model, from a file named atmod.

When you want to solve the structure with ORIENT in the automatic mode (instruction: DIRDIF CCODE ORIENT), the atmod file has to be present in advance.

When you run ORIENT interactively (instruction: DIRDIF CCODE) you either select a fragment in advance or during execution of the program. An existing atmod file can be modified, or a new atmod file can be selected from ORUSER and/or ORBASE.

How to use DIRDIF.ORBASE?

Retrieving and modifying fragments: The instruction DIRDIF CCODE ORBASE allows you to retrieve and/or modify a fragment in advance, see page 3 for edit instructions. The resulting fragment is stored in a file atmod.

Storing your own fragments: For your own structural research you can build up your private database in a file named DIRDIF.ORUSER which is analogous to DIRDIF.ORBASE. The file DIRDIF.ORUSER is one of the distributed files and its header contains instructions for the addition of your fragments. It contains some examples as well. The atomic parameters of your own fragment may be obtained by previously solved structures, the Cambridge Structural Database, molecular modeling, and so on.

Notes
- Do not hesitate to use a model with slightly deviating geometry: vector search methods do not depend critically on precise geometry.
- N-compounds: for many nitrogen compounds the geometry of the corresponding C-compound is sufficiently accurate.
- O-compounds: for many oxygen compounds the geometry of the corresponding C-compound is sufficiently accurate, unless the aromaticity is broken.
- S-compounds: for many sulphur compounds an out-of-ring replacement is sufficiently accurate, but some compounds with S in a ring are listed additionally.
- Warning! Most light-atom structures with S are favourably considered as heavy-   atom structures and you had better solve it with PATTY (instruction: DIRDIF CCODE PATTY).
- Building up a large fragment in many steps is likely to introduce too large deviations in the geometry. Try instead to get the fragment from related structures or molecular modeling.

COFRED EDITOR

This editor is a simple COmmon FRagment EDitor, which is called when executing:

DIRDIF CCODE ORBASE   or
DIRDIF CCODE   (for an interactive run of ORIENT).

In the editor the following instructions are possible:

H   Help

D C4   Delete atom C4.

R C4 S5   Rename or replace atom C4 by S5; Adjust bond length if end-of-chain.
R C4 N4 1.2   Replace atom C4 by N4 at distance 1.20 from parent atom if end-of-chain.

A C4 C   Add one or more C-atoms to C4. (The number of atoms added and their positions depend on the hybridization of the parent atom C4. The new atoms will be numbered automatically.)
A C4 C –T   Add one or more C-atoms to C4, forcing a specified hybridization. (The number of atoms added and their positions will be determined automatically.)
Possible keys are:
-T   tetrahedral(sp3)
-P   planar (sp2)
-L   linear (sp)
A C4 =O   Add =O to atom C4 in planar hybridization.
A C4 –CN   Add -CN to atom C4.
A C4 -NO2   Add -NO2 to atom C4 (Beware of geometry!).

F C2 C3 5   Fuse a 5-membered aromatic ring to bond C2-C3 of an aromatic ring or planar system.
F C2 C3 6   Fuse a 6-membered aromatic ring to bond C2-C3 of an aromatic ring or planar system.

G C4   Supply geometrical information about atom C4.
S   Save (store present atoms in back-up array).
U   Undo (retrieve atoms stored at last save instruction).
X 35   Rotate model around horizontal axis by 35 degrees.
Y 35   Rotate model around vertical axis by 35 degrees.
Z 35   Rotate model in the plane of the screen.
Q   Quit: finish session. Any present H-atom will be removed. (The user will have the opportunity to accept or reject the model obtained so far.)

After finishing the session the parameters of the fragment are stored in a file atmod. When this file is used by the program ORIENT, these parameters are saved in a file called ATOLD.

CLASSIFICATION

The fragments are classified according to chemical classes as given in Molecular Structures and Dimensions Vol.A1, Ed. by Olga Kennard *et al.*, Crystallographic Data Centre Cambridge (1972). As the 'rigidity' of the fragment is of greater importance than the chemical functionality, we occasionally deviate from these classes. Note that several classes have no entries, as fragments are not available, or are easily generated.

1	Aliphatic Carboxylic Acid Derivatives	34	Hetero-Nitrogen (>6-Rings Monocyclic)
2	Aliphatic Carboxylic Acid Salts	35	Hetero-Nitrogen (2 Fused Rings)
3	Aliphatic Amines	36	Hetero-Nitrogen(>2 Fused Rings)
4	Aliphatic N and S Compounds	37	Hetero-Nitrogen (Bridged Ring Systems)
5	Aliphatic Miscellaneous	38	Hetero-Oxygen
6	Enolates (Aliphatic and Aromatic)	30	Hetero-Sulphur and Hetero-Selenium
7	Nitriles (Al. & Ar.)	40	Hetero-(Nitrogen and Oxygen)
8	Urea Compounds (Al. & Ar.)	41	Hetero-(Nitrogen and Sulphur)
9	Nitrogen-Nitrogen Comp. (Al. &Ar.)	42	Hetero-Mixed Miscellaneous
10	Nitrogen-Oxigen Comp. (Al. & Ar.)	43	Barbiturates
11	Sulphur and Selenium Compounds	44	Pyrimidines and Purines
12	Carbonium Ions, Carbanions, Radicals	45	Carbohydrates
13	Benzoic Acid Derivatives	46	Phosphates
14	Benzoic Acid Salts	47	Nucleosides and Nucleotides
15	Benzene Nitro Compounds	48	Amino-Acids and Peptides
16	Anilines	49	Porphyrins and Corrins
17	Phenols and Ethers	50	Antibiotics
18	Benzoquinones	51	Steroids
19	Benzene Miscellaneous	52-57	Terpetines
20	Monocyclic Hydrocarbons (3,4,5-Rings)	58	Alkaloids
21	Monocyclic Hydrocarbons (6-Rings)	59	Miscellaneous Natural Products
22	Monocyclic Hydrocarbons (7, 8-Rings)	60	Molecular Complexes
23	Monocyclic Hydrocarbons (>8-Rings)	61	Clathrates
24	Naphtalene Compounds	62-66	B, Si, P, As, Sb, Bi Compounds
25	Naphtoquinones	67-68	Group IA, IIA, III Compounds
26	Anthracene Compounds	69-70	Ge, Sn, Pb, Te Compounds
27	Hydrocarbons (2 Fused Rings)	71	Transition-metal compounds
28	Hydrocarbons (3 Fused Rings)	72-75	Metal pi-Complexes
29	Hydrocarbons (4 Fused Rings)	76-86	Metal Complexes
30	Hydrocarbons (>4 Fused Rings)	87-99	Miscellaneous
31	Bridged Ring Hydrocarbons		
32	Hetero-Nitrogen (3,4,5-Rings Monocyclic)		
33	Hetero-Nitrogen (6-Rings Monocyclic)		

Chapter 10.   The ORBASE GALLERY

'd.a.: 5' denotes a dihedral angle of 5 degrees

| 0.001 | MONOS | Test model (7-atom fragment of model 41.41) |

1.	For anhydrides, see class 38.	
1.11	D-Tartaric acid	$COOH(CHOH)_2COOH$
1.12	Ethylacetate	$CH_3COOC_2H_5$
1.13	Valeric acid	$CH_3(CH_2)_3COOH$
1.14	Valeric acid amide	$CH_3(CH_2)_3CONH_2$
1.21	Aminobutenoate	
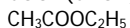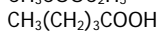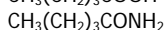
| 3.11 | Butylamine | $CH_3(CH_2)_3NH_2$ |
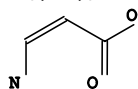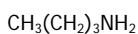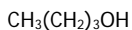
5.11	Butanole	$CH_3(CH_2)_3OH$
5.12	Butadiene (trans conformer)	
5.13	Butadiene (cis conformer)	
5.21	Hexatriene	
5.31	Undecane	
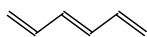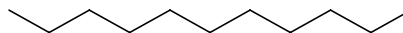
| 13.11 | Hydroxybenzoic acid | |

| 13.12 | Nitrobenzoic acid | |

| 13.21 | Dinitrobenzoic acid | |

| 15.11 | TNB | |

| 16.21 | Azine derivative | |

| 18.11 | Benzoquinone | |

| 18.21 | Tetracyano-p-quinodimethane (TCNQ) | |

| 19.11 | Benzene | |

| 19.21 | Dicyanamidebenzene dianion | |

| 19.31 | Tolane | |

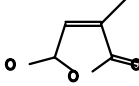|		
20.11	Cyclopropane	
20.12	Cyclopropene	
20.13	Cyclopropene-dispiro derivative	
20.21	Cyclobutane (flat)	
20.22	Cyclobutane (twist)	
20.23	LOXY	
20.24	Dimethylene-cyclobutane	
20.31	Cyclobutene	
20.41	Cyclopentane (twist-envelope)	
20.51	Cyclopentene	
20.61	Cyclopentadiene	
21.11	Cyclohexane (chair)	
21.12	Cyclohexane (twist-boat)	
21.21	Cyclohexene	
21.31	1,4-Cyclohexadiene (flat)	
21.32	1,3-Cyclohexadiene	
21.41	Phloroglucinol	
22.111 22.112	Cycloheptane, chair Same, twist-chair	
22.113 22.114 22.21	Same, bed Same, twist-bed Cycloheptatriene	
22.22	Same, other conf.	
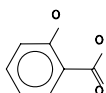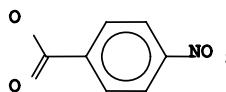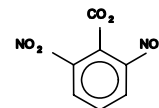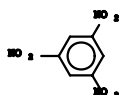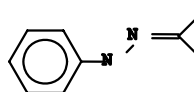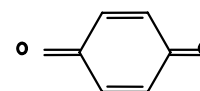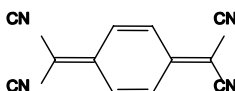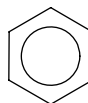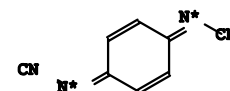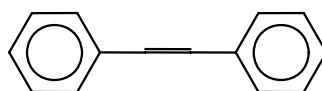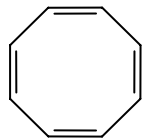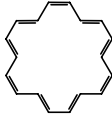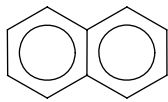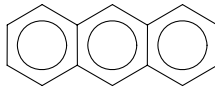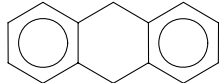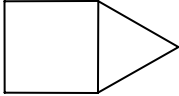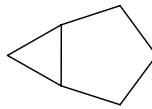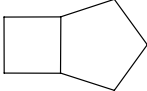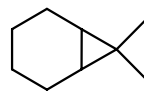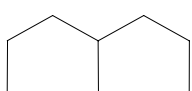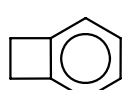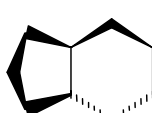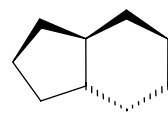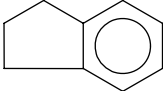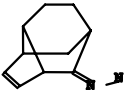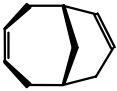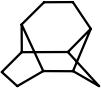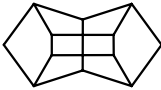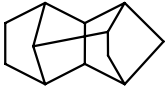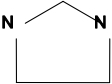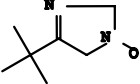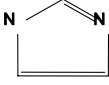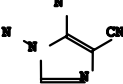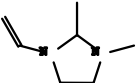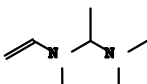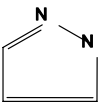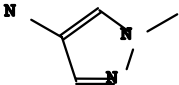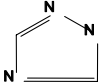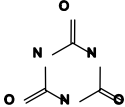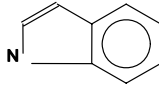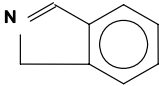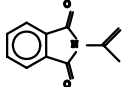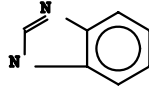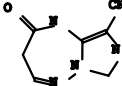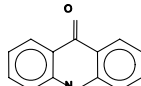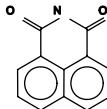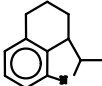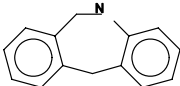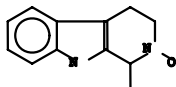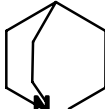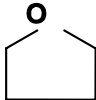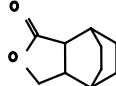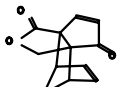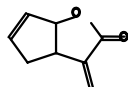
31
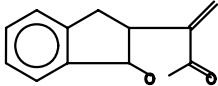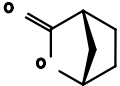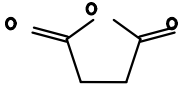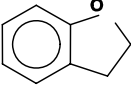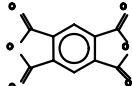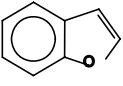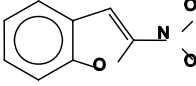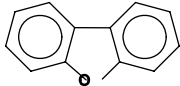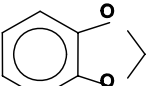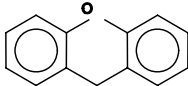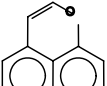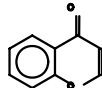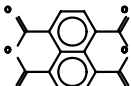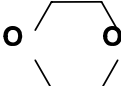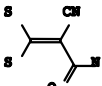
22.31	Cyclooctatetraene	
22.32	1,3,5-Cyclooctatriene	
22.33	Same, other conf.	
23.11	(18)-Annulene	
24.11	Naphthalene	
25.11	Naphthoquinone	
26.11	Anthracene	
26.211	Dihydroanthracene, d.a.: 4	
26.212	Same, d.a.: 42	
26.213	Same, d.a.: 65	
27.11	Bicyclo[2.1.0]pentane	
27.21	Bicyclo[2.2.0]hexane	
27.22	Bicyclo[3.1.0]hexane	
27.31	Bicyclo[3.2.0]heptane	
27.32	DiMe Bicyclo[4.1.0]heptane	
27.41	Bicyclo[3.3.0]octane	
27.45	Benzocyclobutane	
27.51	cis-Hydrindane	
27.52	trans-Hydrindane	
27.53	Indane = Benzocyclopentane	

32
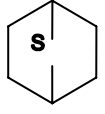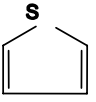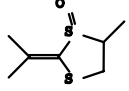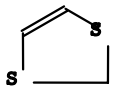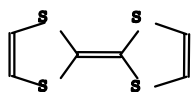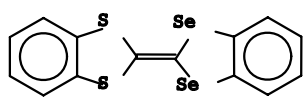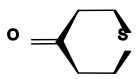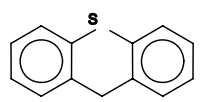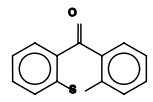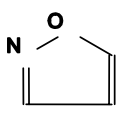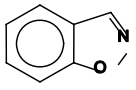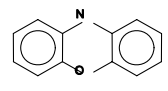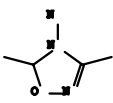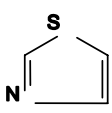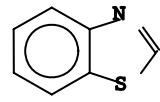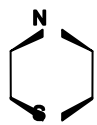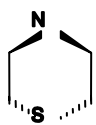
27.55	Indan-1-on
27.56	Acylindanone
27.57	Indene
27.61	Tetralin
28.21	cis-anti-cis-Perhydroanthracene
28.22	cis-syn-cis-Perhydroanthracene
28.23	cis-trans-Perhydroanthracene
28.24	trans-anti-trans-Perhydroanthracene
28.25	trans-syn-trans-Perhydroanthracene
28.31	Phenanthrene
28.32	Dihydrophenanthrene
28.41	Phenalene (aromatic)
29.31	Tetrahydrobenzo anthrachinone
29.41	Tetrahydrobenzo anthrone (d.a.: 32)
30.31	Coronene
31.01	Bicyclo[1.1.1]pentane
31.11	Bicyclo[2.1.1]hexane
31.21	Bicyclo[2.2.1]heptane = Norbornane
31.22	Norbornene

33

31.23	Norbornene-spiro derivative	
31.251	Fused Norbornane-pentane	
31.252	Same	
31.253	Same	
31.254	Same	
31.26	BULAC	
31.27	CARBET	
31.28	DIENON	
31.29	DOLS1	
31.31	Fused Norbornanes	
31.32	Fused Norbornanes	
31.331	Same, C5=C6	
31.332	Same, C3=C4	
31.333	Same, C1=C2, C5=C6	
31.341	Fused Norbornanes	
31.342	Same, C3=C4	
31.41	Bicyclo[3.1.1]heptane	
31.51	Bicyclo[2.2.2]octane	
31.521	Dibenzobicyclooctane, bridge: single bond (1.55)	
31.522	Same, aromatic bond (1.39)	
31.523	Same, double bond (1.30)	
31.55	Bicyclo[3.2.1]octane	
31.56	Tricyclooctane	
31.57	DELE8	
31.58	Cubane	
31.61	Bicyclo[3.2.2]nonane	
31.64	Bicyclo[3.3.1]nonane	
31.71	Bicyclo[3.3.2]decane	
31.74	Adamantane	

34

31.75	NN=decane		
31.79	Bicyclodecane		
31.81	Tetracycloundecane		
31.83	Dodecahedrane (pentagonal dodecahedron)		
31.84	Pentacyclododecane		
31.86	Hexacyclododecane		
31.88	Diamantane C14		
31.91	Buckminster-fullerene (C60 ball)		
32.21	Pyrrolidine		
32.31	Pyrrole		
32.41	Imidazolidine		
32.43	N-oxyimidazol derivative		
32.51	Imidazole		
32.52	Imidazolyl derivative		
32.53	Imidazolium-ion		
32.54	Same		
32.61	Pyrazole		
32.62	Pyrazole derivative		
32.81	1,2,4-Triazole		
33.	For pyridines, see benzene derivatives		
33.31	Dihydropyridine		
33.51	1,3,5-Triazine (flat)		
33.52	Trioxotriazine		

35

35.11	Indole
35.21	2H-Isoindole
35.22	STRIAN
35.31	Benzimidazole
35.51	Imidazotriazepine
36.21	Oxohydroacridine
36.31	Naphthalimide
36.32	Ergolin-fragment
36.411	Dibenzoazacycloheptadiene, fold-line: N-C
36.412	Same, fold-line: C-C
36.51	BEWIJS
37.21	Azabicyclooctane
38.11	Oxirane
38.21	Oxetane
38.31	Tetrahydrofuran
38.32	VADER
38.33	FENBUT
38.34	GAMDEL
38.35	FENLAK
38.36	MANG1

36

38.37	MANG4
38.38	Camphanic-lacton
38.41	Succinic anhydride
38.42	STRIA
38.43	Dihydrobenzofuran
38.44	PMDA
38.51	Furan
38.52	Benzofuran
38.53	Nitrobenzofuran
38.54	Dibenzofuran
38.61	Benzodioxole
38.62	1,3-Dioxole
38.72	Xanthene
38.73	Oxaphenalene
38.74	Dimethylpyrone (flat)
38.75	Benzopyrone
38.76	NTDA
38.81	1,4-Dioxane (chair)
38.82	Pyran
39.01	NOVOA

39.11	Thiirane	
39.21	Thietane	
39.31	Thiabicycloheptane	
39.32	Thiophen	
39.34	SULFOX	
39.35	1,3-Dithiole	
39.36	TTF-Cation	
39.37	SSe-fulvalene	
39.41	Thiacyclohexanon, bed	
39.42	Same, chair	
39.511	Thioxanthene, d.a.: 4	
39.512	Same, d.a.: 40	
39.52	Thioxanthone	
40.11	Isoxazole	
40.22	Benzisoxazole	
40.311	Phenoxazine, d.a: 5	
40.312	Same, d.a.: 33	
40.41	Oxazo	
41.11	Thiazole	
41.12	Benzothiazole	
41.22	Thiazocyclohexane bed, one C-C: aromatic (1.39)	
41.23	Same chair, C-C: single bonds (1.51)	

41.311	Phenothiazine, d.a.: 5	
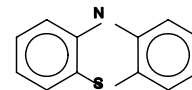
41.312	Same, d.a.: 29
41.313	Same, d.a.: 42
41.41	MONOSX
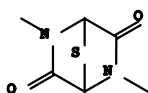
43.21   Alloxan
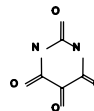
44.11   Uracil
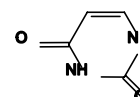
44.12   Cytosine protonated
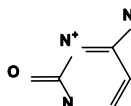
44.13   Thymine
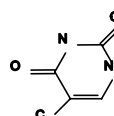
44.21   Adenine protonated
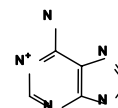
44.22   Guanine protonated
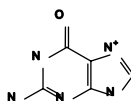
44.31   Benzopyrimidone
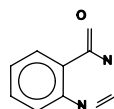
45.11	beta-D-Ribose and 45.111
45.12	alpha-D-Ribose and 45.121
45.21	beta-Fructopyranose
45.51	beta-D-Glucopyranose
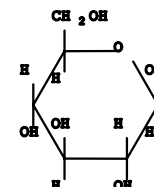
45.52	alpha-D-Glucopyranose
45.61	beta-D-Mannopyranose
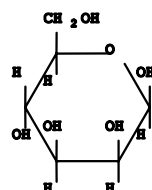
45.71   beta-D-Arabinopyranose

48.11	Glycine (pi = 5.97)	NH_2CH_2COOH
48.21	gamma-Aminobutyric acid zwitterion	$NH_3^+(CH_2)_3COO^-$

50.11   Cephalosporin-C

51.11	Estradiol
51.12	Testosterone
51.13	5a-Steroid
51.14	Cortisone
51.15	Progesterone skeleton
51.16	Cholesterol skeleton
51.17	Ster1
51.18	Ster2
51.19	Ster3
51.20	Ster4
51.21	Ster5
58.11	Morphine skeleton
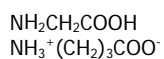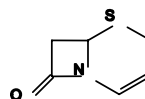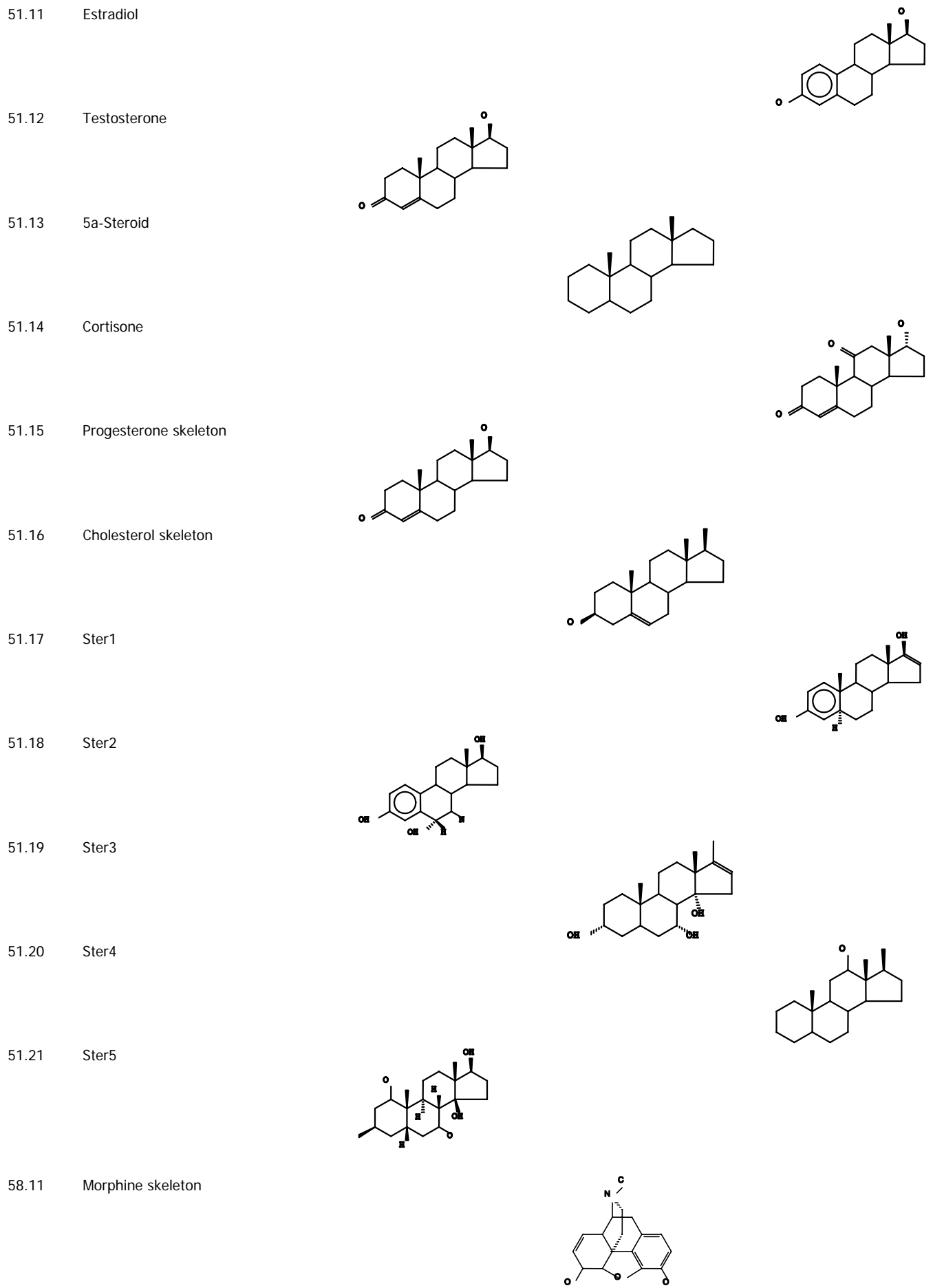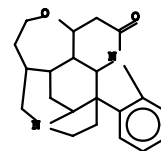
| 58.21 | Strychnine |

| 58.22 | Same: cation |
| 58.51 | Vincadifform. |

| 58.52 | Oxotabersonine |

| 59.11 | (+)Biotin skeleton |

| 59.21 | Gliotoxin skeleton |

| 63.11 | Silano derivative (staggered) |

| 64.31 | CLAM2 |

| 67.51 | Cu-bipyridine |

| 90. | Calixarenes |
| 90.01 | HUMMEL |
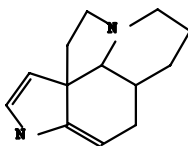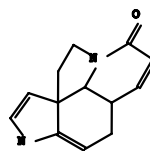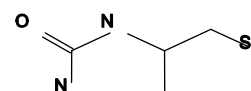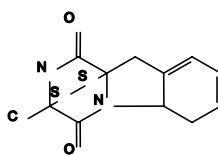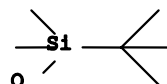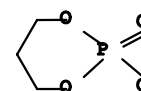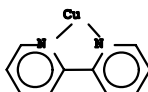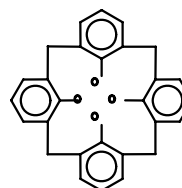
six more different geometries: 90.02, 90.03, 90.04, 90.05, 90.06, 90.07

DIRDIF.ORUSER GALLERY

Molecular fragments in DIRDIF.ORUSER analogous to those in DIRDIF.ORBASE (see header lines of the file DIRDIF.ORUSER for instructions and suggestions for use). The distributed file may contain additional fragments.

| 0.1 | IODINE | I_2 |
| 0.2 | PHENA | |
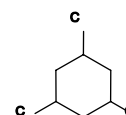
0.3	PO4	PO_4
0.4	GOLD	$AuPC_3$
0.5	PF6	PF_6
8	BASKET	A basket-shaped fragment used in supramolecular chemistry.

41

End of ORBASE - GALLERY

Contents of the following chapters:

THEORETICAL BACKGROUND

DEMO structures for DIRDIF

DIRDIF MANUAL
Chapter 11. Abstract & Introduction

DIRDIF-2007 A computer program for Crystal Structure Determination by Patterson Methods and Direct Methods applied to Difference Structure Factors

How to solve crystal structures using chemical knowledge

Contents

Origin: chapters 11-18 of this MANUAL are adapted from lecture notes on *'Patterson Methods and Completion of the Structure by Direct Methods'* at ECM-15 in Dresden, 1994 by Paul T. Beurskens

Abstract

The methods described in this lecture are available in the computer program system DIRDIF for the automatic solution of X-ray crystal structures using *a priori* structural and/or chemical knowledge. The system knows two major pathways: one to follow when one or more heavy atoms are present, another when the structure contains a geometrically known molecular fragment. Each pathway leads to a partial structure which is expanded to the complete structure.

The pathway for heavy atom structures begins with the program PATTY (automatic Patterson interpretation). The program uses the symmetry-minimum function to find all possible heavy atom positions, calculates all cross-vectors and inspects all combinations of heavy atom positions. Criterion: any vector of a set of atoms (including symmetry-related atoms) must correspond to an acceptably large Patterson value. The most promising set of heavy atom parameters is sent forward to the program PHASEX for further expansion of the structure.

The pathway which is followed when we know a molecular fragment begins with the program ORIENT. The known fragment is used to calculate a set of interatomic vectors, which is rotated in Patterson

space to search for the best fitting orientation, *i.e.*, the rotated vectors are compared with the observed Patterson values. The best fitting orientation of the fragment maximizes the so-called weighted minimum-average function. The corresponding atomic parameters for all possible orientations are sent forward to the program TRACOR for positioning of the rotated fragments.

The translation search program TRACOR searches for a translation vector which shifts the oriented fragment to the best fitting position relative to the symmetry elements of the space group: a correlation function between structure factors is evaluated to find the best fit between the calculated and observed data. A small fragment, with only marginally sufficient scattering power to solve the structure, often leads to a multiplicity of TRACOR solutions.

The program TRAVEC calculates a figures of merit (FOM), based upon vector search technques, for all results from ORIENT and TRACOR, and selects several promising result which are then sent forward to the program PHASEX.

The two pathways of the DIRDIF system come together in the program PHASEX where the phases of the structure factors are refined and expanded by the application of direct methods to the difference structure factors. In special cases symbolic-addition procedures are use to solve enantiomer and/or (pseudo/translational) symmetry problems. In all cases three or four cycles of tangent-formula refinement are executed.

Finally, R2-driven dual space recycling, with Fourier techniques, automatic peak interpretation, calculating R2-rejection criteria, and recycling via PHASEX, etcetera, are used to accept new atoms and reject false peaks in order to expand a partial structure to the complete structure.

In the DIRDIF-2007 version, all possible and acceptable results of PATTY or ORIENT/TRACOR are subjected to a preliminary PASEX/FOUR expansion procedure, and the best partial structure (i.e. the atoms set witch gives the lowest R2 value) is further expanded to completeness.

Technical note on the use of DIRDIF. DIRDIF is designed for fully automatic structure solution, and essentially makes use of chemical/structural information that is available before the system is called into execution. For instance, if the keyword 'PATTY' is used in starting up the system, then control moves automatically from one program to another, to do all that is to be done, from the initial scaling procedure, via the programs PATTY and PHASEX, up to the final Fourier recycling procedure. The final output will be a set of atomic coordinates ready to be input to a least-squares structure refinement program. It is possible, however, to use DIRDIF in interactive mode: on-line HELP is provided, and to some extent the experienced user is able to control the flow of the calculations to be performed.

Introduction

The majority of '*small molecule*' crystal structures is solved using direct methods and/or Patterson methods, see Fig. 1.

There are several good programs based on direct methods that solve a crystal structure *ab initio*, that is, with no knowledge about the structure. But in many cases one knows at the outset of the structure analysis that the structure contains one or more heavy atoms, or some molecular fragments with known geometry. Using this information gives a better start and one may solve the structure easier and faster, and it may even solve the structure where other methods fail. The use of *a priori* structural information in the DIRDIF system is depicted in Fig. 2.

Direct   Methods   Patterson   Methods

ab initio   applied to   heavy atom   fragment
                   difference structure interpretation   search

MULTAN
SHELX   PHASEX   PATTY   ORIENT
SIR   TRACOR
MITHRIL   TRAVEC
etc

Figure 1. Major structure solution techniques. Under' ab-initio', some classical programs names are given. Program names in the rectangular box are part of program DIRDIF

What is known   **path 1**   **path 2**
                                                      | |
Heavy atoms present | V
at unknown positions ------------------------- >   PATTY |
                                                      | |
                                                      V |
Geometrically known molecular | |
fragment in unknown position------------------- ---------->   ORIENT
                                                      | |
Fragment in known orientation but | |
in unknown position = misplaced fragment---- ---------->   TRACOR
                                                      | |
Rigid group at known position (optimize)----- ---------->   TRAVEC
                                                       \   /
                                                        \   /
Atoms at known positions ------------>--------   --->   PHASEX
                              / |
                             /   V
Full structure <----------recycle ? |
                              \ |
                               \-- <----------   --<------ FOUR
                                                 atoms and new peaks

Figure 2. The use of chemical information in DIRDIF.

Chapter 12. PATTY

Determination of heavy atom positions by Patterson methods
The Computer Program PATTY[1]

A Patterson synthesis, which is calculated directly from the measured intensities, can be considered as the vector-map of interatomic vectors in the crystal structure. Heavy atoms can be located by interpretation of this vector-map.

As a starting point for the locations of heavy atoms, the computer program PATTY uses all possible vectors between symmetry-related positions. A so-called symmetry-map is calculated. Heavy atom positions have high values in the symmetry-map and normally correspond to peak positions in this map.

If there is more than one symmetry-independent heavy atom all vectors between not-symmetry related atoms (cross-vectors) should also be present in the Patterson. For such a case PATTY selects combinations of peaks for which all cross-vectors are present. Often many combinations can be made which fulfil this condition, and all represent possible models for the heavy atom structure. Selecting the best model is done by a minimum function of the P/W values of all vectors of the model, in which P is the Patterson value and W is the weight of the vector (which includes a contribution from overlap with other vectors).

A slightly sharpened origin-removed Patterson synthesis is calculated and brought to a Z^2-scale, such that a single vector between atoms j and k with atomic numbers Z_j and Z_k has approximately peak height Z_j x Z_k if there is no overlap with other vectors.

Symmetry minimum function. A position $\mathbf{r}$ and its symmetry related positions $\mathbf{r}_s$ give rise to a number of so-called self-vectors or Harkervectors. The symmetry minimum function or symmetry-map (Simpson *et al.*, 1965), S($\mathbf{r}$), is defined as a minimum function of weighted Patterson values of the self-vectors where the weight for one self-vector is equal to the number of times the self-vector has exact overlap (by symmetry). The symmetry map is expected to have peaks at positions of heavy atoms with $S(\mathbf{r}) \geq Z_j^2$ for atom j. In the following, $S(\mathbf{r}_p)$ denotes a peak of the symmetry map at position $\mathbf{r}_p$, p = 1,...N_p. Due to the extra symmetry of the symmetry-map a peak $\mathbf{r}_p$ represents all possible atomic positions $\mathbf{r}_{p,q} = \pm\mathbf{r}_p + \mathbf{t}_q$ where $\mathbf{t}_q$ is an origin shift vector ranging over all permissible origin choices. For q = 1,2, ... m_q (either discrete shifts or small steps along polar directions) we have $2m_q$ positions $\mathbf{r}_{p,q}$.

Peaks $S(\mathbf{r}_p)$ are selected as possible positions for heavy atoms j if

$$S(\mathbf{r}_p) \geq P_{min,j} = \alpha Z_j^2 \quad (0.4 < \alpha < 0.8) \tag{1}$$

Because of the variation of form factors of the atoms a lower Patterson value (αZ_j^2) than the theoretical value (Z_j^2) is used as a safe limit.
Note: A special method is used to locate the heavy atoms for space group P1. This method only differs at a few points.

Cross-vectors. When two or more atomic positions have to be found, also cross-vectors between peaks have to be inspected. Let the peaks numbered p and p' at positions $\mathbf{r}_p$ and $\mathbf{r}_{p'}$ respectively, originate from atoms j and k, respectively, then the cross-vectors between peaks p and p' include symmetry related atoms, and the minimum value of all corresponding m Patterson values is defined as

$$P_{cross}(\mathbf{r}_p,\mathbf{r}_{p'}) = min_s\{P(\mathbf{r}_p - [R_s \mathbf{r}_{p'} + \mathbf{t}_s])\} \tag{2}$$

---

[1]See G. Admiraal, H. Behm, C. Smykalla, and P.T. Beurskens, *Z. Kristallogr*., suppl. **6**, 522 (1992).

Each of the cross-vectors is expected to have a Patterson value $\geq Z_j Z_k$, but because of the inaccuracy of peak positions for atoms, a lower Patterson value ($\beta Z_j Z_k$) is used as a safe limit, which leads to the following condition:

$$P_{cross}(\mathbf{r}_p, \mathbf{r}_{p'}) > P_{min,jk} = \beta\, Z_j\, Z_k \qquad (0.4 < \beta < \alpha) \tag{3}$$

Origin shifts and inversion. In order to include all possible atomic positions, and starting from atom j, we have to consider not only $\mathbf{r}_{p'}$ as position for atom k but all $2m_q$ positions mentioned above. The condition for each of these positions is

$$P_{cross}(\mathbf{r}_p, \mathbf{r}_{p',q}) > P_{min,jk} \tag{4}$$

We need to be careful with computing time and storage space and it proved to be very inefficient to work with origin and inversion defined positions at the initial stages of the analysis. A preliminary selection of 'possible' pairs of atoms is based on the maximum function Q_{cross} (Behm *et al.*, 1988) for the $2m_q$ cross-vectors:

$$Q_{cross}(\mathbf{r}_p, \mathbf{r}_{p'}) = \max_q \{P_{cross}(\mathbf{r}_p, \mathbf{r}_{p',q})\} \tag{5}$$

The preliminary triple condition for peaks p and p' to originate from atoms j and k is then

$$(S(\mathbf{r}_p) > P_{min,j}) \wedge (S(\mathbf{r}_{p'}) > P_{min,k}) \wedge (Q_{cross}(\mathbf{r}_p, \mathbf{r}_{p'}) > P_{min,jk}) \tag{6}$$

When the structure contains heavy atoms at two symmetrically independent positions all pairs of peaks fulfilling (6) are accepted. For more than two heavy atoms, say N_H atoms, a combination of N_H peaks should be found in which all pairs of peaks fulfil (6). Brute force inspection of all combinations of N_H peaks is impossible for larger values of N_H, due to the very large number of combinations (for example the number of combinations of 5 peaks out of a list of T=150 peaks is about equal to 6×10^8). Therefore a tree-search algorithm is used, which efficiently can find all combinations of N_H peaks fulfilling (6) for all pairs of peaks. In this search procedure the combinations are built up step by step. Efficiency is caused by the fact that in each step the number of peaks which can be used for building up a complete combination is strongly limited.

Extracting atomic positions from selected peak positions

Atomic positions corresponding to peak positions $\{\mathbf{r}_p, \mathbf{r}_{p'}, \dots\}$ of a combination of N_H symmetry-map vectors, as found by the foregoing procedure, are preselected by (6) and not yet associated with origin translation vectors. Condition (4) is not yet tested. The two atomic positions $\mathbf{r}_p$ and $\mathbf{r}_{p',q}$ can shift together along origin translations, while their position relative to each other remains unchanged. Therefore it is useful to relate these two positions by a vector.

All vectors between all pairs of positions are determined according to (4). After choosing the origin (and enantiomorph for acentric space groups) by locating one atom j at $\mathbf{r}_p$, possible positions for an atom k are defined by these vectors relative to $\mathbf{r}_p$. Among these possible positions for all N_H-1 atoms a tree-search procedure now searches for solutions for the atomic positions, in which *all* interatomic vectors agree to condition (1) for each atom and (4) for each pair.

Calculation of a figure of merit (FOM). The procedure described so far yields all solutions for heavy atoms which are possible under the given conditions. To select the best model of heavy atoms, for each solution a FOM is calculated which is based upon the Patterson values of all interatomic vectors of the model. The FOM is defined as the weighted minimum-average function (Nordman & Schilling, 1970), which is also used in the program ORIENT.

Chapter 13. ORIENT

Rotation Functions in Vector Space
The Computer Program ORIENT[2]

The program ORIENT is a fully automated program for the determination of the orientation of a known molecular fragment. It is based on the Vector Search programs of Nordman and Schilling (1970). A slightly sharpened Patterson function is stored. Intramolecular vectors of the known molecular fragment are calculated and selected; vector weights take into account the overlap with neighbouring vectors. The image seeking function is defined as the 'weighted minimum average' function. It is evaluated at each point as the vector set is rotated through the symmetry-independent part of the Eulerian angle space. A coarse scan using only short vectors is followed by a number of fine scans to determine the best possible orientations of the fragment.

Patterson methods for 'equal atom' structures provide powerful alternatives to direct methods for a large number of different cases (rigid groups, expected geometries, coordination polyhedra). Molecular geometries are readily available in ever-growing crystallographic databases, or can easily be calculated by Molecular Modelling programs.

Patterson search methods

Correlation functions in reciprocal space can be used to find the orientation of the molecule, but there are good reasons to prefer the use of rotation functions in direct space, *i.e.,* to use Patterson vector search methods. Each individual interatomic vector must be 'present' in the Patterson function! Therefore, rotation functions in Patterson space (= vector space) are very reliable. However, they are very sensitive to various executional parameters, and the success of these methods (and their strength!) has been enhanced by carefully designed default strategies. In addition, and in contrast to a general perception, the calculation of a Patterson rotation function can be very fast.

Vector search strategy. The approach is briefly described as follows:

1 A rigid molecular group of known geometry (calculated, or retrieved from data in the literature) is defined by a set of atomic positions r_i. The vector set $[v_i]$ is obtained from all interatomic vectors $r_i - r_j$, properly weighted, but modified by various weighting schemes and cut-off options.

2 The vector set is rotated stepwise through three Euler angles A, B and C, to scan all possible orientations. For each orientation of the vector set (point (A,B,C) in angular space) its 'presence' is looked up in the Patterson vector map. A measure of fit, denoted by R, is evaluated. The rotation function $R(A,B,C)$ thus describes the fit of the model vector set to the Patterson vector map for all possible orientations of the model.

3 The highest values (peaks) in the rotation function represent possibly correct orientations of the known fragment. The best fitting orientation (A,B,C) is selected for further structure elucidation. In all space groups but $P1$, the program TRACOR can be used to find the position of the fragment.

It is well known that the sum function, when all vectors in the structure are used, is equivalent to rotation functions based on $|F|^2$ correlations using all reflections. The additional strength of the vector search method is the use of 'weighted minimum average function' (Nordman, 1966):

$$R(A,B,C) \ = \ \min (N) \ = \ \Sigma^N P \ (\ v_i \ (A,B,C) \ ) \ / \ \Sigma^N \ W_i$$

where P is the Patterson function value for vector v_i rotated by angles A,B,C, W_i is the weight of vector v_i and the summation Σ^N is over the N vectors with the lowest values of P_i/W_i. For $N=1$, $R(A,B,C)$ is just the

minimum function. For N equals the total number of vectors, $R(A,B,C)$ is the weighted sum function. A good choice for N is about one fifth of the total number of vectors.

The program ORIENT

A slightly sharpened Patterson function is calculated and stored for fast Patterson function value retrieval. The peak shape of a single C-C Patterson peak is approximated from the shape of the origin peak. This peak shape is used for the calculation of vector weights to account for overlap of neighbouring vectors. Accidental overlap with symmetry related vector sets is not taken into account (to save computer time). The important systematic overlap due to the presence of symmetry in the Patterson function is easily accounted for by modifying the Patterson function values on and near to symmetry elements. (Instead of increasing the weights of vectors approaching a Patterson mirror plane, the stored Patterson values are proportionally decreased; this prevents planar molecules from lining up with the mirror plane.)
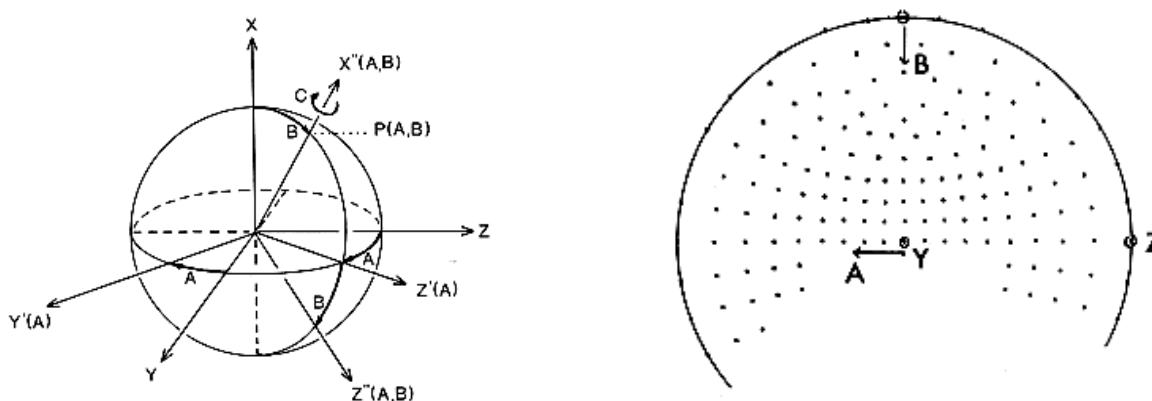
Figure 1. Definition of Eulerian angles   Figure 2. The coarse (A,B) grid plotted on a Wulff net, i.e., a stereographic projection of points P(A,B)

Definition of the Eulerian angles A,B,C. Several systems of angular variables can describe the orientation of the molecule. Eulerian angles are most commonly used. A second angular system is defined by a spin about an axis given by spherical polar coordinates. It is useful to consider both systems simultaneously. The definition of Eulerian angles used here are as follows (Fig. 1):

(1) A is a rotation of the axial system and the molecule about the Cartesian X axis
    B is a rotation of the axial system and the molecule about the rotated Y axis (Y')
    C is a rotation about the rotated X axis (X'')

Note that the A and C rotations are not independent. We also consider an interpretation of A, B and C in terms of polar spherical coordinates. Define:

(2) A is a rotation of the axial system about the X axis (the molecule remains fixed)
    B is a rotation of the molecule and the axial system about the rotated Y axis (Y')
    $(C+A)$ is the spin (rotation of the molecule) about the rotated X axis (X'').

The calculations are performed using definitions (1), but the following discussion uses the interpretation (2). Thus $X''(A,B)$ is the direction to which the original molecular X axis has been rotated. This direction is characterized by a point $P(A,B)$ on the surface of a sphere, see Fig. 1. The sphere represents all possible angular coordinates A,B.

The angular distance ΔAB between two points $P(A,B)$ and $P(A',B')$ is the smallest angle that rotates $X''(A,B)$ to $X''(A',B')$. The sphere (A,B) is covered by points $P(A,B)$ which are only approximately equidistant as measured by ΔAB. The choice of the step size ΔAB is determined by the length of the longest vector, and the resolution of the data. But as the spin is independent of the direction of $X''(A,B)$, the step size ΔC can be chosen independently. When the search model happens to be elongated along X, the longest vector

component perpendicular to the molecular *X* axis is shorter than the longest vector. We may then use larger steps for the *C* rotations.

The cyclic procedure of the program ORIENT is given in Table 1. The generated coarse (*A,B*) grid is given in Fig. 2. The *C* rotation is executed in the innermost loop, for fixed *A* and *B*. The order of (*A,B*) values (points on the (*A,B*) sphere) is irrelevant.

Cycle #	*1*	*2*	*3*	*4*
Sampling	full grid	regions	blocks	blocks
Appr. step size (^{o})	10	5	2	0.7
Max. vector length (Å)	4.5	6.0	7.5	≥ 7.5
Patterson interpolation	none	4-point	8-point	8-point
Number of points	large	large	medium	small
relative level	2 σ	50%	70%	70%
		of the largest peak		

Table 1: The default ORIENT sampling strategy

Concluding remark. The most crucial step in the recycling procedure is the definition of blocks for the fine scans. At present we use 4 x 4 x 4 points in a block. There is no problem, if well-defined peaks result from the medium scan.

But we do have a problem when the rotation function shows bananas as a result of orientational Patterson overlap (because of symmetry, or because of the presence of similar fragments, or because the model is not completely correct). We then have many neighbouring points with about equal values for $R(A,B,C)$. For such cases, blocks for fine scans are defined as follows. The point (*A,B,C*) with the highest *R* value is accepted and its neighbours ($\Delta'A,B,C < 8^{o}$) are removed. Thereafter, the next maximum is accepted, and so on. In such cases, the final results will be a collection of similar orientations of wich one or more may lead to the solution of the structure. As the 'best' of all possible orientations not necessarily leads to succes, it is adviced to feed all orientations into a translation function program, to shift the corresponding molecular fragments to the most probable positions.

===

Chapter 14. TRACOR

Translation Functions for the positioning of a molecular fragment
The Computer Program TRACOR[2]

Reciprocal space correlation functions, which make use of FFT-techniques, are fast and reliable. Best results are obtained when all symmetry elements are used simultaneously, and geometrically unacceptable positions are rejected. For small fragments one should use all observed reflections. The correlation function takes care of the reciprocal space equivalent of sharpening, origin removal, and subtraction of intramolecular vectors.

In this Chapter we discuss the positioning of a well-oriented fragment relative to the symmetry elements of the space group. If the initial application of direct methods yields a recognizable model which cannot be expanded, this model is usually correctly oriented but misplaced relative to the symmetry elements and the model has to be shifted to the (unknown) correct position. The orientation of a rigid part of the structure can also be found by rotation searches (program ORIENT): also in this case the correctly oriented fragment remains to be positioned before the model can be expanded successfully.

Symbols
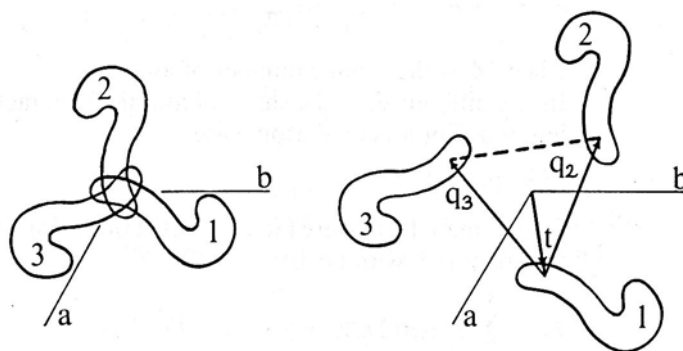
*Figure 1. Definition of the shift vector **t**. Left: original fragment (1) and symmetry transformed fragments (2,3) for a threefold rotation axis. Right: Partial structure for a tentative shift vector **t**.*

$s = 1, ..., m$ m = multiplicity of the space group, excluding Bravais lattice translations ($s = 1$ is the identity operation).

$R_s+\mathbf{d}_s$ symmetry operation on $\mathbf{r}$, where R_s and $\mathbf{d}_s$ are the rotation matrix and translation vector components, respectively.

$\mathbf{T}$ translation vector or shift vector, to be applied to the atoms of the search fragment.

$T(\mathbf{t})$ a translation function of which peaks correspond to possible translation vectors.

$\mathbf{H}$ reflection *hkl*. For typographical reasons we will often omit the subscript $\mathbf{h}$.

$\mathbf{h}_s$ a symmetry related reflection: $\mathbf{h}_s = \mathbf{h}R_s$, $\mathbf{h}_1 = \mathbf{h}$.

$|F_o|$ observed structure factor of reflection $\mathbf{h}$ (on absolute scale).

p^2 the scattering matter of the sum of the input fragment and the symmetry related fragments, relative to the total scattering matter in the primitive unit cell.

$p_1{}^2$ the scattering matter of the sum of the input fragment only, relative to the total scattering matter in the primitive unit cell.

F_s the calculated structure factor for a symmetry transformed fragment (with $\mathbf{t} = 0$).

$F_p(\mathbf{t})$ the calculated structure factor for the entire set of symmetry related fragments, after the application of a shift vector $\mathbf{t}$ to the input fragment.

$|E_o|$, E_s, $E_p(\mathbf{t})$ the normalized structure factors. Note: $pE_p(\mathbf{t}) = p_1 \sum_{s=1}^{m} E_s(\mathbf{t})$.

Theory

We first consider translation functions in Patterson space. A measure of fit between calculated (P_p) and observed (P_o) Patterson functions (Buerger, 1959) is:

$$T_1(\mathbf{t}) = \int P_o(\mathbf{u}) \, P_p(\mathbf{u},\mathbf{t}) \, d\mathbf{u}$$

where $\mathbf{u}$ is a point in vector space, and $\int d\mathbf{u}$ denotes the integration over the unit cell. The self-Pattersons (P_s) and the origin peak (P_{OR}) are subtracted, which gives:

$$T_2(\mathbf{t}) = \int \left[P_0(\mathbf{u}) - \sum_{s=1}^{m} P_s(\mathbf{u}) - (1-p^2)P_{OR}(\mathbf{u}) \right]\left[P_p(\mathbf{u},\mathbf{t}) - \sum_{s=1}^{m} P_s(\mathbf{u}) \right] d\mathbf{u}$$

The importance of these convolutions is that they provide a sound background for the derivation of correlation functions defined in reciprocal space. The convolution T_1 is readily converted (see for instance Crowther and Blow, 1967) into the well known reciprocal space correlation function

$$T_3(\mathbf{t}) = \left\langle \left| F_o \right|^2 \left| F_p(\mathbf{t}) \right|^2 \right\rangle \qquad \text{or} \qquad T_4(\mathbf{t}) = \left\langle \left| E_o \right|^2 \left| E_p(\mathbf{t}) \right|^2 \right\rangle$$

where $\langle \, \rangle$ denotes averaging over all reflections $\mathbf{h}$. T_3 and T_4 differ in that sharpening is used. Reciprocal space functions can be expressed in terms of F's or E's, but the analogue to the removal of the Patterson origin peak and of the self-Patterson vectors is most easily expressed in terms of E's. Thus the reciprocal space analogue to T_2 leads to:

$$T_5(\mathbf{t}) = \left\langle \left[\left| E_o \right|^2 - p_1^2 \sum_{s=1}^{m} \left| E_s \right|^2 - (1-p^2) \right] \left[\left| E_p(\mathbf{t}) \right|^2 - m^{-1} \sum_{s=1}^{m} \left| E_s \right|^2 \right] \right\rangle$$

where the averaging is over all reflections.

The reciprocal space translation functions are evaluated by Fourier transformations. T_3 gives:

$$T_6(\mathbf{t}) = \sum_{\mathbf{h}} \left| F_o \right|^2 \sum_{s=1}^{m} \sum_{s'=1}^{m} F_s F_{s'}^* \exp\left[2\pi i \mathbf{h}\left(R_s - R_{s'} \right)\mathbf{t} \right]$$

where the summation is over a complete sphere of reflections. Define the coefficient $G_{ss'}(\mathbf{h}) = \left| F_o \right|^2 F_s F_{s'}^*$ for a reflection $\mathbf{h}$ and one pair of symmetry operation s and s', and define the transformation of indices $\mathbf{k} = -\mathbf{h}(R_s - R_{s'})$ for one pair of symmetry operations. This transformation implies that for all symmetry elements which give rise to one- and two- dimensional translation searches, many different reflections $\mathbf{h}$ contribute to the same vector $\mathbf{k}$. Substitution in T_6 gives:

$$T_7(\mathbf{t}) = \sum_k \sum_{s=1}^{m} \sum_{s'=1}^{m} \sum_{\mathbf{h}|\mathbf{k}} G_{ss'}(\mathbf{h}) \exp\left[-2\pi i \mathbf{kt}\right]$$

where $\mathbf{h}|\mathbf{k}$ denotes that the summation is over all reflections $\mathbf{h}$ which transform to the same vector $\mathbf{k}$. The summation index $\mathbf{k}$ does not denote a reflection, but is a transformed vector. This expression represents a Fourier synthesis which can be calculated by the program FOUR.

Modifications of T_7 as defined for T_5 are easily incorporated. The result is:

$$T_8(\mathbf{t}) = \sum_k \sum_{s \neq s'}^{mm} \sum_{\mathbf{h}|\mathbf{k}} \left[|E_o|^2 - p_1^2 \sum_{s=1}^{m} |E_s|^2 - \left(1 - p^2\right) \right] E_s E_{s'}^* \exp(-2\pi i \mathbf{kt})$$

Note that in this formula all symmetry elements are used simultaneously.

The TRACOR procedure consists of the following steps:
1.   Expand the reflection data to a complete hemisphere, and calculate normalized structure factors $|E_1|$ for the expanded set of reflections.
2.   Examine the symmetry of the space group, and prepare Fourier coefficients.
3.   Carry out the one-, two-, or three-dimensional Fourier summations.
4.   In addition, the program also calculates the R2 disagreement factor.
5.   Reject the shift vectors with too short intermolecular contacts.
6.   Apply the most probable shift vectors to the input model, the result being a collection of trial structures ordered by their probabilities.

The size of search fragment. The scattering power of the search fragment and its symmetry related fragments (p^2) is appropriate for the positioning of the search fragment relative to all symmetry elements simultaneously. As $p^2 = mp_1^2$, the minimum size of a search fragment (p_1^2) depends on the number of symmetry elements. For acentric distributions, the peak-to-background ratio is expected to be $(1+p^2)/(1+p_1^2)$ (Beurskens, 1981). This ratio is of course affected by Patterson overlap.

Note for very small fragments: we need not find a single solution! It is not troublesome to test tentative shift vectors by expanding the implied partial structure. This is automatically done in DIRDIF.

Note:
There is a problem regarding TRACOR: for colliding fragments (contacts betwee symmetry related fragments) the FOM's as well as the R2 values become very unreliable. The corresponding shift vectors are rejected, but is is not at all impossible that the input fragment is part of a symmetrical molecule. Something for DIRDIF 2009.

==

Chapter 15. TRAVEC

Validation of a rigid fragment by Vector Search method
The Computer Program TRAVEC

A fragment smaller than, say, 20% of the scattering power of the symmetry independent part of the structure, could have been inaccurately positioned by the program TRACOR because of the uncertainty in its orientation. Consequently, the relative peak heights of the TRACOR results are less reliable.

In the past, we used the program TRAVEC to subject all TRACOR peaks (*i.e.* all different positions for one given orientation of the fragment) to a group-refinement procedure. Variable parameters are rotations around the cartesian X, Y and Z axes of the fragment and shifts in the fractional *x*, *y* and *z* directions (except for the polar directions). Thus the total number of variable parameters ranges from 3 (space group *P*1) to 6 (all non-polar space groups).The refinement was done by a steepest-descent procedure, which is not described in this manuscript.

We have abandoned the optimization, but we maintained the program TRAVEC to calculate the overlap of inter-atomic vectors with the Patterson map. To our experience, the optimization could easily go astray, but on the other hand, using the calculated measure of fit (MOF) for Patterson vectors was strong enough to discriminate between good and bad TRACOR solutions.

The measure of fit, MOF, is the 'weighted minimum average function', as described for ORIENT. Intramolecular vectors are limited to 10Å, and all corresponding intra- en intermolecular vectors are used; all vector-overlap is taken into account. Note: intramolecular vectors, which already have been used by ORIENT, are also included in the presnt FOM.

Output
The program TRAVEC (1) calculates figures of merit (FOM) for all results from ORIENT and TRACOR, and selects several promising results. The final sets of atomic coordinates are ordered according to the FOM's and output in file atoms, which is then sent forward to the program PHASEX for testing, selecting, and finally expanding the most promising atoms set to completness.

Note about the number of trial structures:
Re ORFLEX: the program ORFLEX (not descibed in this manual) may be used to generate some, or a great many of molucular configurations of a flexible molecular fragment, all to be used as input to ORIENT.
Re ORIENT: a good model may lead to one or two accepted orientations, a very poor model easily leads to 20 or 30 orientations which have almost eaqual significance, and as the FOM (level of significance) used in ORIENT does vary only slowly, it is best to have TRACOR trying out a large number of orientations. The ORIENT results for different input models have no absolute significance, so each input model must be investigated by TRACOR.
Re TRACOR: a well-oriented model leads to one accepted position, a poor model may leads to many positions with almost eaqual FOMs. The FOM, however, is a correlation coefficient which has an absolute significance and can be used to discriminate between good and bad orientations for correct as well as for incorrect molecular configurations. So, for many models input to ORIENT, and many many output orientations (maybe thousands), TRACOR is able to select and output a reasonable number of trial structures.
Re TRAVEC: with a combined FOM (including results of TRACOR and R2) , the number of possible solutions is greatly reduced again! [Program limitation: select 300 best TRACOR, and output best 20 TRAVEC resuts.]

(1) G. Admiraal, and P.T. Beurskens

Chapter 16. PHASEX

Direct Methods applied to difference structure factors
The Computer Program PHASEX[3]

When part of a structure is known, direct methods can be used to solve the unknown part of the structure. Often the known part of the structure consists of one or more heavy atoms, either on general, or on special or pseudo-special positions. The known part of the structure may also consist of a molecular fragment, found by *ab-initio* direct methods, or by Patterson rotation search techniques.

The difference structure factors, phased by the partial structure, are used as input to a weighted tangent-refinement procedure for phase extension and for the refinement of input phases and amplitudes. If the known atoms do not uniquely define the structure, symbolic addition techniques are introduced to solve the (pseudo-symmetry) ambiguities.

The method is referred to as 'PHASEX'. It is most useful if the known part is only marginally sufficient to solve the structure, when the known atoms lie in special or pseudo-special positions (origin ambiguity), or when, for noncentrosymmetric structures, the known atoms form a centrosymmetric arrangement (enantiomorph ambiguity).

The idea of applying direct methods to difference structure factors was the consequence of trying to solve structures with pronounced superstructure effects. Classical (*ab initio*) direct methods are in general not applicable to structures in which heavy atoms lie on special or pseudo-special positions. However, after subtracting the heavy atoms, the remainder of the structure is an ordinary structure which doesn't give any troubles in standard direct methods.

And more: a substantial number of phases are known at the outset of the direct methods procedure. As a consequence direct methods applied to difference structure factors appear to be much more powerful than standard *ab initio* direct methods.

Since 1970 the basic principles of these difference methods have been developed into various procedures and strategies some of which are presented in this chapter.

It is the purpose of this chapter to show how known chemical information can be incorporated in various 'direct methods' procedures such as the tangent formula, symbolic addition, multisolution, quartets, etc. The phase refinement procedure PHASEX, described in this chapter, allows the expansion of a small known fragment, and the solution of various types of phase ambiguities and symmetry problems.

Pictograms: structure factors are represented by vectors in the complex plane. The known F_p is given by a heavy line. The observed represented by a *circle* with radius $|F_o|$. Fig. 1 represents eqs. (4) and for a given vector F_r.
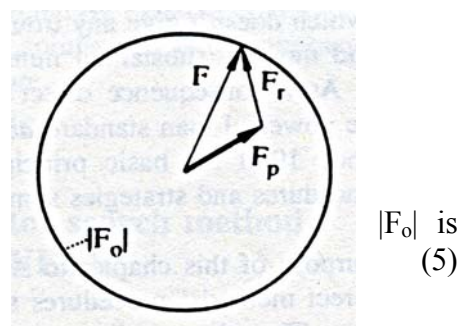
$|F_o|$ is

(5)

Figure 1. Vector diagram for F, F_p and F_r.

[3]Excerpts from: P.T. Beurskens, and C. Smykalla. In: *Direct methods of solving crystal structures*. Ed.: H. Schenk, Plenum Press, New York. Pp. 281-290 (1991).

Notation and definitions

H   reciprocal space vector (h,k,l)
In most formulae the indices **h** will be omitted to simplify the notation.

φ phase of reflection **h**. In principle this phase is not known; for all practical purposes φ represents the best available estimate for the phase.

$|F_o|$ observed structure factor for reflection **h** on absolute scale!

F   a complex number representing the true structure factor, or the 'best' phased value for the observed structure factor: $F = |F_o| \exp(i\varphi)$ (1)

F_p partial structure factor; sum over N_p known atoms

φ_p phase of F_p

F_r structure factor of the remainder; sum over N_r remaining unknown atoms.
In principle, F_r is not known; for all practical purposes F_r represents the best available estimate for the true value.

φ_r phase of F_r

Σ_p, Σ_r summation over all N_p known atoms, and over all N_r unknown atoms.

Z   atomic number of an atom

p^2 scattering fraction of the partial structure: $p^2 = \Sigma_p Z^2 / (\Sigma_p Z^2 + \Sigma_r Z^2)$ (2)

r^2 scattering fraction of the rest structure: $p^2 + r^2 = 1$ (3)

The relation from these definitions: $F = F_p + F_r$ (4)

gives the fundamental PHASEX equation: $|F| = |F_p + F_r|$ (5)

where F_r is to be determined by 'direct methods'.

E   normalized value for F

E_p, E_r normalized values for F_p and F_r: $E_p = (\varepsilon \Sigma_p f_p^2)^{-\frac{1}{2}} F_p \exp(B_p s^2)$ (6)
$E_r = (\varepsilon S_r f_r^2)^{-\frac{1}{2}} F_r \exp(B_r s^2)$ (7)

s $\sin \vartheta / \lambda$

ε conventional symmetry enhancement factor

B_p, B_r overall temperature factor parameter of the known atoms, and of the remaining unknown atoms.
From the definition of the normalisation it follows then: $E = p E_p + r E_r$ (8)
Normalisation and determination of B_p and B_r is achieved by the method of Gould, van den Hark and Beurskens (1975).

ΔF the conventional difference Fourier coefficient: $\Delta F = (|F_o| - |F_p|) \exp(i\varphi)$ (9)
Note that the phase of ΔF is either φ_p or $\varphi_p + \pi$

E_1 normalized value of ΔF: $E_1 = (\varepsilon \Sigma_r f_1^2)^{-\frac{1}{2}} \Delta F \exp(B_r s^2)$ (10)

φ_1 is the phase of E_1, *i.e.*, $\varphi_1 = \varphi_p$ or $\varphi_1 = \varphi_p + \pi$

W_1 weight associated with the reliability of φ_1

φ_t, W_t subscript t refers to tangent formula results.

Preliminary considerations. The phase problem

Ab initio direct methods can be employed to determine the direction of the vector F in the above diagram, Fig. 2a. Multisolution techniques, for instance, may be initiated by assigning a four-fold ambiguity to some selected reflections, Fig. 2b.
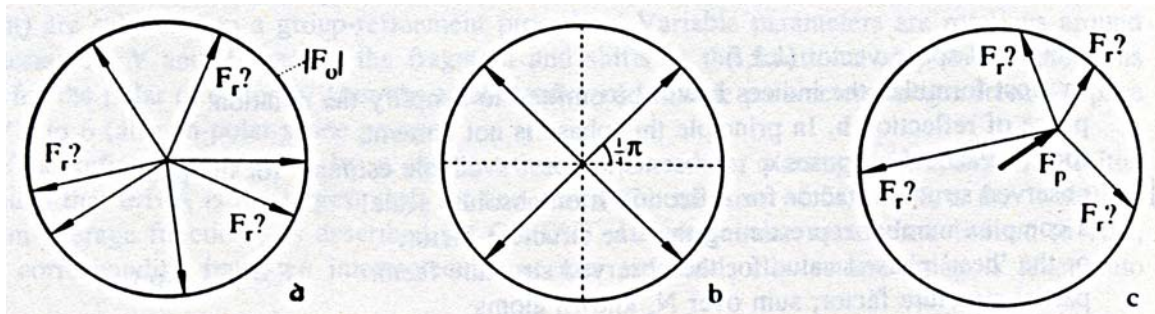
Figure 2.
a:Ab initio direct methods: find direction of F.
b:Fourfold ambiguity phases $\pm\frac{1}{4}\pi$, $\pm\frac{3}{4}\pi$.
c:Structure factor F_p, known fragment: find remainder of structure; i.e., find F_r.

The situation changes if a part of the structure is known. Then, the calculated structure factor of the partial structure may be drawn (Fig. 2c), but the phase problem is not yet solved. From the end-point of the known vector F_p an unknown vector F_r, representing the structure factor of the remaining part of the structure, is drawn with its end-point lying on the $|F_o|$-circle.

Comparing Figs. 2a and 2c we notice two important differences:
Fig. 2a: *a priori*: all phases are equally probable, all vectors F have a length equal to $|F_o|$.
Fig. 2c: known fragment: the length of the unknown vector F_r is not fixed; it depends on φ_r (*i.e.*, the phase of F_r) according to eq. (4) and, consequently, possible phases φ_r do not have equal probability.
In fact, when the known fragment is sufficiently large, one can calculate a conventional difference Fourier synthesis using the assumption that the most probable phase of F_r is equal to φ_p, *i.e.*, the calculated phase of F_p (eq. (9)): see Fig. 3a.
Distribution functions for φ_r have been given by Sim (1960); we will use these later, but for the time being it is sufficient to associate a weight W_1 with the ΔF value. The Fourier transform of $W_1\Delta F$ is the weighted difference electron density map. But this is certainly not the same as the electron density map of the difference structure!

When the known fragment is relatively large, the (difference) Fourier synthesis can be used to find the rest of the structure, however, if the known fragment is small (or exhibits higher symmetry than the space group symmetry!) most of the ΔF values are a poor estimate of the true F_r values. Nevertheless, the ΔF's contain valuable phase information, and we shall see that direct methods can be used to improve these phases (Fig. 3b). Note that a change of phase (φ_r) implies an adjustment of the magnitude ($|F_r|$, see Fig. 3c).
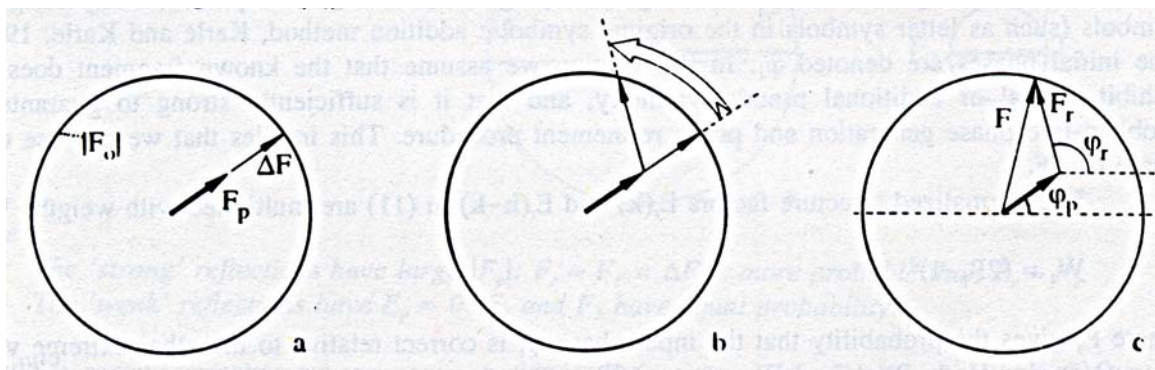

Figure 3.
a: Difference Fourier synthesis: use ΔF.
b: Phase refinement: input ΔF, output F_r.
c: The phase φ_r obtained by direct methods is used to construct F_r and F.

The tangent formula applied to difference structure factors

It has been intuitively assumed that direct methods are applicable to a hypothetical structure consisting of the complete structure minus the known part of the structure. The tangent formula for the difference structure is

$$E_r(\mathbf{h}) \approx c \, \Sigma_{\mathbf{k}} \, E_r(\mathbf{k}) \, E_r(\mathbf{h-k}) \tag{11}$$

where c is a positive scaling factor defined to satisfy (4) (for infinite resolution: $c = N_r^{-\frac{1}{2}}$). We usually wish to see the electron density function for the entire structure. Substitution of (11) in (8) gives:

$$E(\mathbf{h}) = pE_p(\mathbf{h}) + c \, r \, \Sigma_{\mathbf{k}} \, E_r(\mathbf{k}) \, E_r(\mathbf{h-k}) \tag{12}$$

The Fourier transformation of E(**h**) (or of its de-normalized equivalent F(**h**)) gives all the known atoms (Fourier transform of E_p) as well as all the unknown atoms (Fourier transform of E_r)
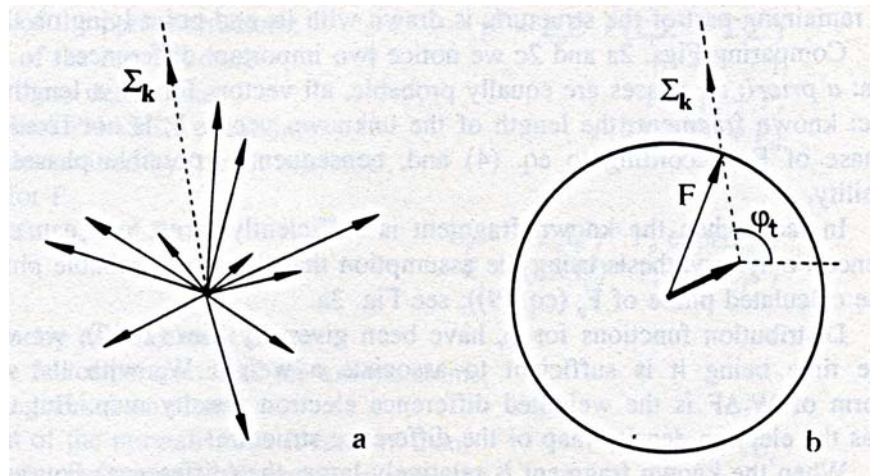

Figure 4.
  a: shows a graph of the tangent formula for E_r;
      arrows: all terms $E_r(\mathbf{k}) \, E_r(\mathbf{h-k})$; dotted line: sum of all arrows (eq. 11).
  b: shows a construction of F; dotted line: direction of F_r (see Fig. 4a).

The application of (11) can be initiated by using phases which, in principle, may be obtained from either the known fragment, or by arbitrarily specifying multi-solution trial phases, or by assigning symbols (such as letter symbols in the original symbolic addition method, Karle and Karle, 1963). The initial phases are denoted φ_1. In this Chapter we assume that the known fragment does not exhibit special or additional pseudo-symmetry, and that it is sufficiently strong to guarantee a problem-free phase generation and phase refinement procedure. This implies that we can use φ_1 = phase of ΔF.
   The normalized structure factors $E_r(\mathbf{k})$ and $E_r(\mathbf{h-k})$ in (11) are multiplied with weigths W_1

$$W_1 = (2P_1-1)^2 \tag{13}$$

where P_1 gives the probability that the input phase φ_1 is correct relative to the other extreme value $\varphi_1+\pi$ (Van den Hark, Prick and Beurskens, 1976). This is a measure to select E_1, and to use it as a first estimate for E_r. The weight assigned to the phases that result from the tangent formula (11) is W_t:

$$W_t = (2P_t-1)^2 \tag{14}$$

With $P_t = \frac{1}{2} + \frac{1}{2}\tanh\left[(\sigma_3/\sigma_2^{3/2}) \, |E_h| \, S_{\mathbf{k}}W_{\mathbf{k}}W_{\mathbf{h-k}}E_{\mathbf{k}}E_{\mathbf{h-k}}\right]$

And $\sigma_n = \Sigma_r Z^n$,   Z being the atomic number of the atoms of the rest structure.

In the first cycle of tangent refinement only E_1 values with phases φ_1 are used as first estimates for E_r for the reflections $\mathbf{k}$ and $(\mathbf{h}\text{-}\mathbf{k})$. Whether or not the output phases φ_t are accepted depends on the corresponding weights.

If $W_t > W_1$ then the new phase $\varphi_r = \varphi_t$ is accepted with weight W_t.

If $W_t < W_1$ then the calculated φ_t value is only partially accepted when $|\varphi_t\text{-}\varphi_1|$ is less than $\pi/2$: $\varphi_r(\text{new}) = \varphi_1 + (W_t/W_1)[\varphi_t\text{-}\varphi_1]$. The new φ_r value is used to calculate a new value for E_r, which may be used as input for the next tangent refinement cycle.

Symbolic addition and phase correlation

In the original symbolic addition method (Karle and Karle, 1963) as few 'letter symbols' as possible were used to represent unknown phases. Alternatively, in the sign-correlation method (Beurskens, 1963) many (up to 20) letters were used, which enabled the statistical analysis of a large number of more or less consistent phase relationships. For the present purpose (symbolic addition applied to difference structure factors) we found that about ten letter symbols are sufficient.

Let us rewrite (11) for one term in the form of a 'sigma-2' relation:

$$\varphi(\mathbf{h}+\mathbf{h}') \approx \varphi(\mathbf{h}) + \varphi(\mathbf{h}') \tag{15}$$

The primary set H is obtained by selecting the best phased reflections (*i.e.*, with the highest $W_1|E_1|$ values) and, in addition, assigning phase symbols X (= A, B, C, ...) to ten selected unphased reflections. Substitution of these phases in (15) gives the set H+H' where the phases $\varphi(\mathbf{h}+\mathbf{h}')$ are expressed by numerical values and symbols. The most reliable results are selected to constitute the secondary set K. Application of (15) now gives phases for sets H+K and K+K'. Symbolic relations are obtained from the many multiple phase indications. (For instance, if $\varphi(\mathbf{h})$ is found as A, and via another route as B+π, then A = B+π is a symbolic relation.) These relations are analysed as shown below.

Origin fixation (centrosymmetric)
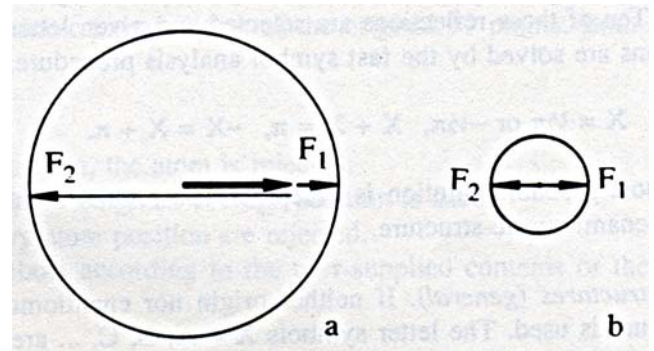
Figure 5.
   a:The 'strong' reflections have large $|F_p|$: $F_r \approx F_1 = \Delta F$ is more probable than $F_r \approx F_2$.
   b:The 'weak' reflections have $F_p = 0$: F_1 and F_2 have equal probability.

Consider a centrosymmetric superstructure problem. F_r is either F_1 or F_2. The 'weak' reflections (*i.e.*, the superstructure reflections) have no (or negligible) contribution from the partial structure (see Fig. 5b). The ten strongest reflections (*i.e.*, having the largest $|E_1|$ values) will be given letter symbols X = A, B, C, ... and many relations between the symbols are obtained.

The symbolic relations are easily solved by the fast symbol analysis procedure (Beurskens and Prick, 1981) using the symmetric mode:

X = 0 or π, X + X = 0, -X = +X.

The most probable solution (*i.e.*, the assignment of numerical values to the letters A, B, C, ... which gives the highest consistency with respect to (11) or (15)) is assumed to be correct. Substitution of the value in all symbolic phases fixes the origin and solves the phase problem. Several cycles of tangent refinement (11) are used before calling the Fourier and peak search routine.

Enantiomorph fixation

Consider a non-centrosymmetric structure with one or more known heavy atoms having a centrosymmetric arrangement. The origin may have to be shifted to put the center of symmetry at the origin, and the symmetry relations are changed accordingly. All calculated phases are then either 0 or π. This constitutes the enantiomorph problem. We need to find reflections of which the phases deviate substantially from the value 0 or π.
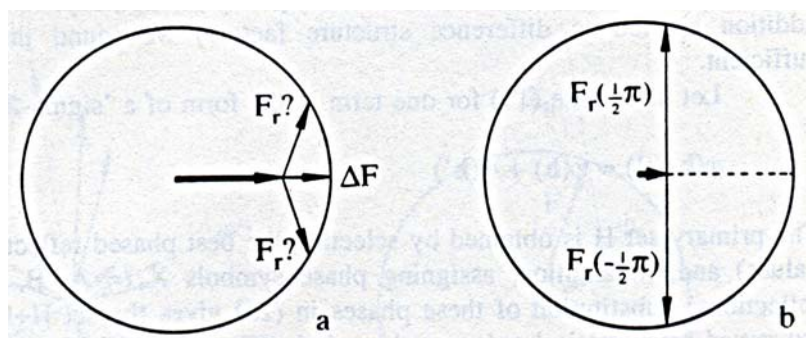

Figure 6.
    a:Large F_p: F_r is probably close to ΔF.
    b:Small F_p: φ_r may be $+\frac{1}{2}\pi$ or $-\frac{1}{2}\pi$? Enantiomorph sensitive?: input to selection procedure.

Enantiomorph sensitive reflections are selected by inspection of the tangent formula summation (11). Reflections which are found most inconsistently as 0 or π, are expected to be either $\frac{1}{2}\pi$ or $-\frac{1}{2}\pi$. Ten of these reflections are selected and given letter symbols X = A, B, C, ... The symbolic relations are solved by the fast symbol analysis procedure, using the anti-symmetric mode:

$$X = \tfrac{1}{2}\pi \text{ or } -\tfrac{1}{2}\pi, \ \ X + X = \pi, \ \ -X = X + \pi.$$

The most probable solution is accepted. The second solution, with the same probability (!), refers to the enantiomeric structure.

Superstructures (general). If neither origin nor enantiomorph is fixed, the same phase correlation procedure is used. The letter symbols X = A, B, C, ... are assumed to represent phases $\pm\frac{1}{4}\pi$, $\pm\frac{3}{4}\pi$: it is easily seen (if we represent phases X by vectors exp(iX)) that the symbolic relations have to be satisfied for the real part (projection of the vectors on the real axis) as well as for the imaginary part (projection on the imaginary axis).

Small fragments. The symbolic phasing procedure is also used for cases where the scattering fraction is less than, say, 0.15. The solution with the highest consistency from the fast symbol analysis (Beurskens and Prick, 1981) is not necessarily correct. To select the most probable phase set we use the Ψ_o test (Cochran and Douglas, 1955) and negative quartets (De Titta *et al.*, 1975; Schenk, 1974).

Conclusion

After the symbol analysis has been finished, the resulting phases are refined by the tangent formula as described above, and the phased structure factors will then be input to the Fourier program for further structure elucidation.

Chapter 17. FOUR

The Fourier program FOUR
and the DIRDIF recycling strategy using R2

The program FOUR uses FFT-techniques for the Fourier transformation, and continues with peak search, peak interpretation, and peak selection. Various options (such as Patterson, or difference Fourier map) will **not** be discussed in this Chapter: here we are interested only in the Fourier synthesis (electron density map) that is calculated in order to find more atoms in the process of a routine structure determination.

The input file with the Fourier-coefficients is prepared by the program DDMAIN and may be the result of a preceding user-supplied instruction (*e.g.* FOUR, PHASEX, ORIENT, ...) or of the recycling instruction generated by the program FOUR itself. The program DDMAIN is an important 'background' program that takes care of several common calculations, such as scaling, structure factor calculation, and preparation of input data files to various programs. It is also the program DDMAIN that takes care of the R_2-driven recycling strategy. Thus recycling implies the selection of peaks by FOUR to be used as atoms for the next cycle, and the rejection of atoms by DDMAIN based on the R_2 criterion. In the following we will discuss these topics without further reference to the programs FOUR or DDMAIN. Whether or not the phases of the calculated structure factors are refined by PHASEX is of no relevance for the present discussion.

The Fourier peak list

After the preparation of the electron density map, a peak search routine collects all peaks in the map which are found above a threshold value, and with a minimum separation. The peak height is obtained by a 19-point quadratic interpolation. The peak sum is obtained by integrating over neighbouring points using a moderated quadratic extrapolation of the original 19 points. Both peak height and peak sum are on absolute scale.

Peaks are compared with input atoms:
    -if no peak is found at the atomic position, the atom is rejected;
    -if peak and atomic position are close, a weighted average position is used ;
    -peaks which are too close to a heavy atom position are rejected.
Accepted peaks are assigned chemical symbols according to the user-supplied contents of the unit cell.

The R₂ criterion

A known partial structure, which may include one or more heavy atoms, and which may exhibit pseudo symmetry, is subjected to an R_2-criterion test (van Havere and Lenstra, 1983) to allow some (misplaced) atoms to be rejected prior to the start of the expansion procedure.

The Fourier synthesis, as described above, leads to a new set of atoms. This new partial structure also is subjected to the R_2-criterion test, which allows one or more atoms (original and/or new) to be rejected. The disagreement factor R_2 using 'sharpened' structure factors, is given by:

$$R_2 = \Sigma_{\mathbf{h}}\, w^2\, (\, |F_o|^2 - |F_p|^2)^2 \, / \, \Sigma_{\mathbf{h}}\, w^2\, |F_o|^4$$

where w is the sharpening function: $w = 1 \, / \, \Sigma_i\, f_i^2$, f_i is the scattering factor of atom i, which includes the temperature factor.

In order to test individual atomic contributions to the R_2-function for atoms j,j', ..., we need to store matrices with cross-terms j,k for all atoms k:

$$f_j f_k \{\cos(2\pi \mathbf{h.r}_j) \cos(2\pi \mathbf{h.r}_k) + \sin(2\pi \mathbf{h.r}_j) \sin(2\pi \mathbf{h.r}_k)\}$$

The influence of one atom or a group of atoms is then easily calculated without recalculating all structure factors. Atoms may be sorted on their contribution to the R_2-value. Atoms which raise the value of R_2 may be removed from the model. Atoms which hardly lower the R_2-value may also be considered as 'suspect' atoms.

During the expansion process the number of atoms converges to the number of atoms of the complete model. R_2^{est}, which is the estimated value of R_2, can be calculated *a priori* (van Havere and Lenstra (1983)): the comparison of the observed R_2 with R_2^{est} is used as a criterion to stop the procedure.

Chapter 18. BIJVOET

The Bijvoet - coefficient for finding the absolute structure

[.. this section is under construction
     here follows a preliminary write-up..]

The program Bijvoet determines the absolute configuration or "absolute structure" of a crystal, expressed by the weighted Bijvoet-coefficient B and its estimated standard deviation σ (B):

$$B = \Sigma \, (w * \varDelta Fc * \varDelta Fo) \, / \, \Sigma \, (w * | \, \varDelta Fc * \varDelta Fo \, |)$$
$$\sigma \, (B) = (1.- | \, B \, |) * \text{sqrt} \, (\Sigma \, (\, 2w * \varDelta Fc^{**}2)) \, / \, \Sigma \, (\, w * | \, \varDelta Fc * \varDelta Fo \, |)$$

in which ΔFc is the difference between Fcalc of the two reflections of a Bijvoet pair, and $\varDelta$Fo is the difference between the Fobs values, and $w = 1/\sigma^{**}2 =$ weight of one measured Bijvoet pair, in which $\sigma^{**}2 =$ variance of $\varDelta$Fo obtained from the accuracy of the measurements.
The sum is taken over selected Bijvoet pairs. The Bijvoet pairs are carefully selected on criteria related to $\varDelta$Fc and $\varDelta$Fo and sorted on $|w * \varDelta Fc * \varDelta Fo|$. The program calculates individual Bijvoet coefficients, and Bijvoet coefficients for individual batches of Bijvoet pairs, and also cumulative values. The final value for the Bijvoet coefficient is based upon about 100 pairs. The consistency of the results for individual batches may help to judge the reliability of the final result.

Note about the probability P(B).:
Inversion of the structure will change the calculated B into -B, which is a change of 2|B| . For a normal error distrtibution, assuming that the structural parameters are error free, and that σ (B) is small relative to |B| , the probability that the observed value of |B| deviates less then 2B/σ (B) standard deviations is:
$$P(B) = \text{erf} \, (\, (2.* |B|) \, / \, (\text{sqrt}(2) * \sigma \, (B))$$
which may be the probability that the sign of B is correct. The assumptions, however, are not met for weak anomalous scatters. A more realistic question is: what is the probability that B does not change sign with acceptable changes in the measurements? Thus, the probability that the sign of B is correct is given by: the probability that the true (+/-)B value lies between zero and (+/-)infinite, which is given by:

$$P(B) = 0.5 + 0.5 * \text{erf} \, (\, |B| \,) \, / \, (\text{sqrt}(2) * \sigma \, (B))$$

For very weak anomalous effects, however, σ (B) is in the order of |B|, and the result for P(B) is over-estimated because the expression for P(B) assumes that the calculated anomalous effects are error free, thus that the atomic positional and vibrational parameters are perfect. For a structure with oxygen as heaviest atom, measured with Mo-radiation, very careful L.S. refinement is escential.

Chapter 19.

References and Acknowledgements

References

Behm, H., Smykalla C., and Beurskens P.T. (1988), *Z. Kristallogr*. **182**, 17-18.
Beurskens, P.T. (1963): Technical report on *Sign Correlation by the Sayre Equation*.
    The Crystallography Laboratory, Univ. of Pittsburgh, Pennsylvania.
Beurskens, P.T. (1981), *Acta Cryst.* **A37**, 426-420.
Beurskens, P.T., and Prick, P.A.J. (1981), *Acta Cryst.* **A37**, 180.
Buerger, M.J. (1959): *Vector space*. New York: John Wiley.
Cochran, W., and Douglas, A.S. (1955), *Proc. Roy. Soc.* **A227**, 486.
Crowther, R.A., and Blow, D.M. (1967), *Acta Cryst*. **23**, 544-548.
De Titta, G.T., Edmonds, J.W., Langs, D.A., and Hauptman, H. (1975), *Acta Cryst*. **A31**, 472.
Gould, R.O., van den Hark, Th.E.M., and Beurskens, P.T. (1975), *Acta Cryst.* **A31**, 813.
Van den Hark, Th.E.M., Prick, P., and Beurskens, P.T. (1976), *Acta Cryst.* **A32**, 816.
Van Havere, W.K.L., and Lenstra, A.T.H. (1983), *Acta Cryst*. **A39**, 920-924.
Karle, I.L., and Karle, J. (1963), *Acta Cryst.* **16**, 969.
Nordman, C.E. (1966), *Trans. Am. Cryst. Assoc.* **2**, 29-38.
Nordman, C.E., and Schilling, J.W. (1970), In: *Crystallographic Computing*. Pp. 110-114.
    Edited by Ahmed, F. R.. Copenhagen: Munksgaard.
Schenk, H. (1974), *Acta Cryst.* **A30**, 477.
Sim, G.A. (1960), *Acta Cryst.* **13**, 511.
Simpson, P. G., Dobrott, R. D., and Lipscomb, W. N. (1965), *Acta Cryst*. **18**, 169-179.

Acknowledgement

Please refer to the present program as:

P.T. Beurskens,   G. Beurskens,   R. de Gelder,   J.M.M. Smits , S. García-Granda,   R.O. Gould , DIRDIF-2007 ,   Crystallography Laboratory, Radboud University Nijmegen, The Netherlands.

Chapter 20.

DEMO structures for DIRDIF

List of DEMO structures for
PATTY

	page
ANTZN	65
EB2207	65
FE2S3	
LAP2	
KAP3	
MARS	
MONOS	19, 69?
NGUY	
OXATON	
PBAG	
PBCU7	
S6MO4	
TETRAS	

List of DEMO structures for
ORIENT

	page
ACNORT,	
CAROL	
GERA	
HUMMEL	
JOOST1	
MONOS	21, 71?
MUEN	
NUNE	
OXATON	
STRIAN	

_____

ANTZN [....]

The following pages are under construction.

[back to start]