

Chapter 3

XDINI – Importing Data into XD

XDINI provides an interface between XD and certain crystallographic computer packages used to solve and refine the structure. It creates the master file (**xd.mas**) with default options and settings, the corresponding input-parameter (**xd.inp**) and data files (**xd.hkl**) for **XDLSM**. The program either requires keywords given in the command line or input from the file **xd_ini.inp**. The current version supports data transfer from SHELX (SHELXTL) and CIF files. It also accepts free-format, as well as fixed-format data files, making it possible to communicate with other computer packages.

XDINI in XD2024 has been substantially revised and now consists of three modules. The first module **XDINI** is essentially identical to the earlier version. The resultant output files **xd.inp** and **xd.mas** are then post-processed by two additional modules:

Module XDLSDB

This module provides an automatic reassignment of the local coordinate system for each atom, based on local connectivity and geometrical considerations. Chemically equivalent atoms and the local symmetry are identified based on these considerations and a new local coordinate system is determined. The methodology is primarily geared to organic molecules, or molecules with substantial organic content, e.g. organometallics or coordination complexes. It is less appropriate for purely inorganic systems, though the same principles are applied. If the module detects that the molecule is purely inorganic (no C and no H atoms) then the more extensive searches for rings etc. are bypassed. The user should be aware that the module **XDSLDB** may occasionally fail if encountering widely differing chemical situations from organic molecules. In the vast majority of cases however, it provides a much more sensible local coordinate system than the very basic one assigned in **XDINI**. The program also determines the restrictions on the multipole parameters based on the determined local pseudo-symmetry. This is of course an idealized symmetry, but it is probably the most appropriate one to use, especially in the earlier stages of refinement. The default action is NOT to apply these pseudo-symmetry restrictions in the final **xd.mas** file, but this can be overridden with the *-pseudo* command line option (see below). The corresponding key table for the pseudo-symmetry restrictions determined by **XDLSDB** is always written to file **key.table**, so that users may manually edit the **xd.mas** file and choose which pseudo-symmetries to apply. The symmetry codes which are used by **XDLSDB** are *always* written to the parameter file, regardless of whether or not the pseudo-symmetry is applied to the multipoles in the master file.

The module also extends commonly observed X-H distances to standard distances and includes a full set of associated RESET_BOND directives in the modified master file. In addition, a file (**riding.con**) containing the appropriate riding constraints for each H atom is created, and referred to in a (commented out) “include” statement in the new master file. This allows the user to choose either RESET_BOND directives or riding constraints for the X-H bonds during refinement.

IMPORTANT NOTICE: It is important to be aware that the local pseudo-symmetries chosen by **XDLSDB** should NOT be treated as definitive. They are but one possibility and it is up to the user to examine them all carefully and see which are the most appropriate for the desired refinement model. This is particularly so for metal atoms and other “non-organic” elements, where the assigned pseudo-symmetry is very likely to be inappropriate. For instance, in transition metal complexes the user might wish to assign the central atom high pseudo-symmetries such as O_h , at least in the initial stages of refinement, but this will never be offered by module **XDLSDB** (at least in the current version).

Module XDSPPOS

The symmetry constraints chosen by **XDLSB** for atoms lying in special positions (*i.e.* on symmetry elements of the space group) are in almost all cases inappropriate. For such atoms, there may be mandatory restrictions on the positional, thermal and multipole parameters which may be refined. Failure to comply with these mandatory restrictions will lead to an unstable refinement. In the past, the user was required to investigate Tables 4-5 to 4-9 in the **XDLSM** manual to determine the appropriate constraints on allowable parameters and the necessary relationships between them and also to decide on a suitable local coordinate system. This required a detailed manual editing of the **xd.mas** file to include these constraints and the modified local coordinate system. This process is now fully automated in the module **XDSPPOS**.

The module reads the **xd.mas** and **xd.inp** files written by **XDINI** (or those modified by **XDLSDB**) and determines the space group and associated special positions. If no atoms in special positions are found, the module immediately halts and the current master & parameters files are unmodified. If atoms in special positions are identified, the site symmetry restrictions on *xyz*, *Uij*, *Cijk*, *Dijkl* and the multipole parameters are determined automatically and appropriately modified versions of **xd.mas** and **xd.inp** are then produced, with the correct key tables. The necessary constraints (if any) for the anharmonic thermal parameters are written to the file **anharmonic.con** which is included in the master file as an unactivated "include" file, as this constraint list can be quite lengthy, especially if several atoms lie on special positions. This file also provides the user with a reference for the key table of the atom(s) in special positions, in the event that the key table in **xd.mas** is overwritten. The symmetry codes which are used by **XDSPPOS** (and **XDLSDB**) are written in the parameter file to the previously unused parameter *isym* (see Table 2-1). These are given in Table 3-1 in the Appendix. Note that for the atoms in special positions, the symmetry constraints are *always* applied to the multipoles in the master file. The pseudo-symmetry codes assigned by **XDLSDB** are also always written to the master file, but by default the symmetry constraints are not applied to the refinement keys of the multipoles (see command line option *-pseudo* below).

It is also possible to run **XDSPPOS** independently from the command line. In this case the original **xd.mas** and **xd.inp** files are left untouched and the modified versions written to **xd_spos.mas** and **xd_spos.inp**.

These two new modules are run automatically when invoking **XDINI** on the command line. The old version of **XDINI** is run first, to create a basic set of structure files, followed by **XDLSDB** and then **XDSPPOS**. If the user wishes to avoid running the new modules and modifying the original **xd.mas** & **xd.inp** files, then the new command line option *-nomod* achieves this – see below for details of the new command line options.

In addition to applying the correct restrictions pertaining to atoms in special positions in the starting input files for XD refinement, the new version of **XDINI** has these additional features:

- (i) If a non-standard X-ray wavelength (*i.e.* not MoK α or CuK α) is detected, then the values of f' and f'' for each element are now correctly computed for that wavelength. In earlier versions of **XDINI**, the f' and f'' values for MoK α were used and written to the **xd.mas** file when non-standard wavelengths were encountered.
- (ii) If it is detected that the reflection data are from neutron diffraction, then a comment as to this is issued. All the data banks now include an entry for deuterium (D), identical to that for hydrogen, except for the atomic weight & neutron scattering length.
- (iii) If the reflection data set is found to be unmerged, then a merged data file is automatically created using the XD utility program **XDHKL**. The resolution of the data and the number of low angle data and suitability for charge density refinement are also given in a comment. Note that it is preferable to use a more sophisticated data processing program such as SORTAV to produce the input reflection file.

(iv) The space group is determined from the SYMM entries given in the input files by the modules **XDINI** & **XDSPPOS** and is printed in the output listing files.

(v) Certain non-centrosymmetric space groups are polar, i.e. the origin is not defined by the symmetry elements along one or more crystallographic axes. In such cases it is necessary to define the origin along the polar axis/axes *by fixing (i.e. not refining)* the appropriate positional parameter(s) of one atom, preferably of the heaviest atomic type. Such restrictions are now automatically applied to the first atom in the list in the **xd.mas** file. The first atom in the list will always be of the heaviest atomic type (unless the *-nosort* command line option was given). This important issue regarding polar space groups was not presented in previous versions of the XD manual and it was left to the user to discover and apply the necessary restrictions, which otherwise would lead to unstable refinement with large correlations.

(vi) The atomic sites are checked for their occupancies and a warning is issued if disordered atoms (or partially occupied atoms) are found. XD is not well designed for refinement of disordered structures – specifically it is not possible to refine the atomic occupation factors in **XDLSM**.

(vii) The input of data via the CIF format has a number of changes (a) the **xd.cif** & **xd.fcf** files are now checked for correct syntax, as the library routines which XD uses give enigmatic error messages when reading cif files with syntax errors. A clear message about the nature and location of the syntax error is given on screen. (b) the first of the command line tokens is interpreted as a request for a specific data block (see below). (c) it is now possible to read the reflection data from a cif created by recent versions of SHELXL (see below).

(viii) When **XDINI** finishes, a summary of the important issues which have been encountered is issued as WARNINGS and COMMENTS. COMMENTS are merely for information purposes, but WARNINGS indicate more serious problems and should always be carefully considered.

The file **xd.mas** is the control file for running all XD programs and needs constant manual editing during the refinement process. The files **xd.inp** & **xd.hkl** should never normally need editing. When running **XDLSM**, the instructions in **xd.mas** take precedence over those in **xd.inp**, so that if, say, the number of kappa parameters need changing, it is only necessary to make the required editorial changes in **xd.mas**, and **xd.inp** can remain unmodified.

3.1 Instructions for XDINI

3.1.1 Command line mode

In the command-line mode, no **xd_ini.inp** file is required. The following simple syntax can be used:

xdini *cid prgname bnkname (-nomod -nosort -norename -pseudo)*

(e.g. `xdini test shelx SCM`)

cid is a maximum 48 character string and the *cid* compound identifier described before is taken as the first 8 characters of this token. When using cif format files as input to XDINI, the first token is taken as the desired data_ block name. The *prgname* can be **shelx** or **cif** with the following input-file requirements:

<i>prgname</i>	Crystal data & coordinate file	Reflection file
shelx	shelx.ins/shelx.res	shelx.hkl
cif	xd.cif/shelx.cif	xd.fcf/shelx.fcf

The legacy input formats XTAL & MOLLY are now deprecated and unsupported, though they are still available and should work as in earlier versions of XD. The parameters on the command line may be entered in any order, with the sole restriction that the first token on the command line is always taken to designate *cid*. The files are read in the order given above. If any of them is not found or its interpretation failed, XDINI turns to its standard input file **xd_ini.inp** for further instructions. *bnkname* can be CR, BBB, SCM or VM - see Ch. 2.5 for their meaning. If *bnkname* is not specified, **xd.mas** will be generated with the default Clementi/Roetti BANK CR.

If using the CIF input, the following points should be noted:

(i) The cif files containing the parameter and reflection data should be called **xd.cif** and **xd.fcf** respectively. Alternative names are **shelx.cif** & **shelx.fcf**.

(ii) If the **xd.cif/shelx.cif** file is written by a recent version of SHELXL which inserts the HKL reflection data in the cif, then both the atomic/crystallographic data and reflection data can be read from this same file. Note that the program will *only* attempt to read the reflection data from **xd.cif/shelx.cif** if the file **xd.fcf** is absent. If **xd.fcf** is present, the reflection data will preferentially be read from this file. This is the preferred option for XD, as the ideal reflection file for XD should contain only unique reflections and no systematic absences. If the FCF file was created by the program SHELX (LIST3/LIST4), then the data will be merged and have the systematically absent reflections removed

(iii) It is **essential** that the names of the data_ block names to be accessed are identical in both the parameter and reflection data files. In a cif file with multiple data blocks, it is possible to choose the desired data block through the first command line token *cid*. If the cif format is chosen, the first token (*cid*) is interpreted as the desired data block name. The available data blocks are displayed in the monitor output. If one of these data block names corresponds to the *cid* token, then that data block is read and processed in both files. If no data block name corresponding to token *cid* is found, then the first available data block is read (or the second one if the first one is called "global" as this data block normally contains no crystallographic information and is always ignored by **XDINI**). The same data block name in the **xd.fcf** file is then sought and if this is not found, then no reflection data are read and a warning message issued. It is of course mandatory that the data block name contains no embedded blanks.

(iv) it is assumed that all the reflection data in the FCF file are on the same scale and a single overall scale factor applies, unless the data_ item **_refln_scale_group_code** is present. The given code should always be an integer "1", "2" etc (the CIF dictionary categorises this data_ item as character). All scale factors are given an initial value of 1.0.

There are several new command line options which allow the user to select specific options

-nomod allows the user to run **XDINI** without running the subsequent modules **XDLSDB** & **XDSPOS**, i.e. in the exact same fashion as in previous versions of XD.

-nosort allows the user to run **XDINI** without sorting the atom list, i.e. retaining the original atom order in the input files.

-norename by default, the modified files written by the modules **XDLSDB** & **XDSPOS** are copied to **xd.mas** & **xd.inp**. This command line option overrides this default action. The resultant files are then stored in **xd_lsdb.*** & **xd_spos.***.

-*pseudo* by default, the local pseudo-symmetry restrictions determined by **XDLSDB** for the multipole parameters are NOT used (though the pseudo-symmetry is still used to calculate the local axial definitions). This command line option activates the pseudo-symmetry for the multipole key table, so that the key table in the final resultant **xd.mas** file is commensurate with the determined pseudo-symmetries. Regardless of whether this command line option is set, the pseudo-symmetric multipole restrictions determined by **XDLSDB** are always written to the file **key.table**, so this may be used at a later stage to manually adjust the pseudo-symmetry settings in the model.

3.1.2 File directed mode

The legacy mode for entering the crystallographic, structural and reflection data remains available. It will be activated if there no tokens on the command line or if the *prgname* type (shelx or cif) is not given. The program then expects further input from the file **xd_ini.inp**. If **XDINI** is invoked without any command line options and **xd_ini.inp** is absent, the program immediately stops and prints a short message informing the user of the command line possibilities.

The input file **xd_ini.inp** consists of three segments: general crystallographic, parameter and observation input. The first one corresponds to that in the master file containing the instructions **TITLE**, **CELL**, **CELLSD**, **SYMM**, **LATT**, **BANKFILE** and **WAVE**, among which the first two always have to be given. The latter two segments have common instructions described below.

3.1.3 FILE

FILE *filename*

The data are expected to be read from the input file **xd_ini.inp** unless otherwise specified. The **FILE** instruction redirects the default input to a file named *filename* – normally the case for the reflection data.

3.1.4 FORMAT

FORMAT (*format specification*)

The reflection data must be given in default order (see below) and in free format. If this is not the case a proper format instruction (standard FORTRAN) is to be given. The format specification in parentheses must be divided by a blank from the **FORMAT** command. Note that it is the **LOADREF** directive which specifies whether the data in the reflection file is *F* or *F*².

3.1.5 Default atomic parameter list

The following entries have to be given for each atom:

atomname *x* *y* *z* *mult* *uiso* or
u11 *u22* *u33* *u12* *u13* *u23*

The *atomname* (up to eight characters) MUST start with a proper element symbol followed by any character string. It is transformed to the atom identifier standard to XD (NA11 to Na(11) or h2a to H(2a)). *x*, *y* and *z* are fractional coordinates corresponding to the cell dimensions given by the **CELL** card. The last two entries, the atomic site occupation factor (*mult*) and the isotropic thermal parameter (*uiso*) can be omitted if the atom is in general position and anisotropic displacement parameters are supplied in the next line. Note that the standard order of the Uij parameters differs from that in SHELX.

3.1.6 LOADPAR

LOADPAR *nat* (*npar*) (*UIJSX*)

Followed this command line *nat* atom segments are read. The parameters have to be either in free format in the order specified above or according to a format statement given previously. In the latter case, the number of entries (including *atomname*) for each atom has to be given by *npar*. If the directive UIJSX is given, the anisotropic *Uij* parameters are read in the SHELX order, i.e. *u11 u22 u33 u23 u13 u12* instead of the standard order.

3.1.7 DTYP

DTYP **u**|**b**|**beta**

This instruction specifies the type of the displacement parameters in the atom line. A general expression for the anisotropic atomic thermal parameter is

$$t = \exp \left(- \sum_{ij} d_{ij} A_{ij} \right) \quad \text{with } i \geq j = 1, 2, 3$$

For the three options above the constants (*d_{ij}*) and the displacement amplitudes (*A_{ij}*) take the following forms:

<i>Dtyp</i>	<i>d_{ij}</i>	<i>A_{ij}</i>
U	$2\pi^2 a_i a_j h_i h_j (2 - \delta_{ij})$	U_{ij}
B	$a_i a_j h_i h_j (2 - \delta_{ij})$	$B_{ij} = 8\pi^2 U_{ij}$
Beta	$h_i h_j (2 - \delta_{ij})$	$b_{ij} = 2\pi^2 a_i a_j U_{ij}$

where

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$$

a_i are reciprocal axis lengths and *h_i* are the corresponding components of the scattering vector. If the input displacement amplitudes are in the form of **b** or **beta** they are transformed to **u** as required by XDLSM.

3.1.8 SCALE

SCALE (*sc(i),i=1,nsc*) [*1.0*]

nsc number of scale factors are read in. If omitted the data are supposed to be on absolute scale forming one scale group. More than one **SCALE** directive may be given if the number of scale factors is large.

3.1.9 NEUTRON

NEUTRON (TOF)

This instruction indicates that the reflection data are from neutron diffraction. The extra token TOF indicates that the data are time-of-flight neutron data and the wavelength for each reflection must then be given as an extra parameter in the reflection data (see below). In this case, the wavelength in the **WAVE** instruction is ignored and any value may be given.

3.1.10 Default observation input

The following entries can be given for each observation:

h k l obs sigobs scgrp tbar u1 u2 u3 v1 v2 v3

For TOF neutron data, the following formats are accepted:

h k l obs sigobs scgrp wavelength (ndat = 7) or

h k l obs sigobs scgrp tbar wavelength (ndat = 8) or

h k l obs sigobs scgrp tbar u1 u2 u3 v1 v2 v3 wavelength (ndat = 14)

These symbols are described in the previous chapter in connection with the reflection file **xd.hkl**. The first 5 entries always have to be given, all others are optional. If the reflection file is in FCF cif format (i.e. the filename has the extension ".fcf"), the expected data order is *h k l calc obs sigobs*, and the *calc* value is ignored.

3.1.11 LOADREF

LOADREF **F** or **F^2** *nref ndat*

nref observation lines, each containing *ndat* entries with *F* or *F²* data, are read either in the order specified above or according to a format statement given previously. If **F** or **F^2** is omitted then **F** is assumed.

3.1.12 SORT

SORT (*index1* [**h**] *index2* [**l**] | **sinthl**)

The reflections are sorted either with respect to indices (*index1* varies first and *index2* last) or to the absolute value of the scattering vector (**sinthl**). The current maximum number of reflections which can be processed is 500,000. Proper sorting with respect to indices speeds up the Fourier calculations. It is mentioned here that **XDINI** does not average symmetry equivalent reflections and neither does **XDLSM**. It is advised to enter into XD with unique (symmetry-averaged) data unless anisotropic extinction refinement is to be carried out.

3.1.13 END

END

The **END** card closes the **xd_ini.inp** file and terminates the program.

3.3 Files written by XDINI

The following files are produced by XDINI

1. **xd.mas** – the starting XD system master file
2. **xd.inp** – the starting XD system parameter file
3. **xd.hkl** – the starting XD system reflection data file
4. **xd.hkl.merged** – this is only written if the input reflection data file is an unmerged one. It may be renamed as **xd.hkl**.
5. **anharmonic.con** – this is only written if there are atoms in special positions. It contains the mandatory restrictions (if any) on the anharmonic thermal parameters for these atoms.
6. **riding.con** – this is only written if there are H-atoms attached to C, O, N atoms in well recognised chemical situations. It contains the riding instructions for these H-atoms, which may be used as an alternative to the RESET BOND instructions embedded in the **xd.mas** file in the **XDLSM** refinement.

7. **key.table** – this contains a back-up copy of the pseudo-symmetry key table entries, which is useful if the user wishes to apply these restrictions in future refinements.
8. **xd_ini.out** – the full listing file for module XDINI
9. **xd_lsdb.out** – the full listing file for module XDLSDB
10. **xd_spos.out** – the full listing file for module XDSPOS
11. **xd_hkl.out** – the full listing file for utility XDHKL
12. **xdini.out** – a copy of the summary listing file shown on the computer terminal

3.4 Examples of XD_INI.INP

```
TITLE oxal      (free format atom list)
CELL  6.093 3.469 11.9257 90. 105.69 90.
SYMM 1/2-X, 1/2+Y, 1/2-Z
LATT C P
WAVE 0.71073
BANKFILE SCM
SCALE 85.87513 89.84698 369.09409
LOADPAR 7
O1      0.085335 -0.055242 0.150354 1.0
      0.006503 0.009821 0.003786 0.002344 0.001042 0.000470
O2      -0.221518 0.244985 0.036284 1.0
      0.005563 0.009277 0.005550 0.002939 0.001704 0.000386
O3      -0.451596 0.634692 0.178431 1.0
      0.006991 0.009768 0.005222 0.001231 0.002255 0.000790
...
H3      -0.373817 0.487426 0.152675 1.0 0.03
FILE    ox.hkl
FORMAT  (3i4,2f8.0,i4)
LOADREF 1500 6
SORT    sinthl
END
```

In this example, the 1500 reflection data are read from a file `ox.hkl` in the SHELX fixed format and sorted in order of $\sin(\theta)/\lambda$.

```
TITLE NIEN3
CELL  8.8482      8.8482  33.1241  90.000  90.000  120.000
CELLSD 0.0001      0.0001  0.0004  0.000  0.000  0.000
SYMM  - X, - Y, 1/2 + Z
SYMM  - Y,  X - Y, 1/3 + Z
SYMM  - X + Y, - X, 2/3 + Z
SYMM  X - Y,  X, 1/6 + Z
SYMM  Y, - X + Y, 5/6 + Z
SYMM  X - Y, - Y, - Z
SYMM  - X, - X + Y, 2/3 - Z
SYMM  Y,  X, 1/3 - Z
SYMM  - Y, - X, 5/6 - Z
SYMM  X,  X - Y, 1/6 - Z
SYMM  - X + Y,  Y, 1/2 - Z
LATT  A P
WAVE  1.000
NEUTRON TOF
BANKFILE SCM
SCALE 1.00547 1.00547  0.98308  0.98416  0.99063  1.00605
```



```

SCALE 0.97794 0.94344 0.91942 0.93523 0.94915
LOADPAR 23 UIJSX
NI1 0.638747 0.277494 0.750000 0.50000
0.00555 0.00509 0.00390 0.00000 -0.00031 0.00255
N1 0.415606 0.084033 0.717796 1.00000
0.01100 0.00981 0.00606 -0.00173 -0.00146 0.00365
N2 0.441419 0.261185 0.790227 1.00000
0.00972 0.01156 0.00609 -0.00129 0.00051 0.00574
N3 0.849926 0.488564 0.781931 1.00000
0.00876 0.00957 0.00688 0.00002 -0.00158 0.00335
C1 0.259692 0.085501 0.733709 1.00000
0.00870 0.01016 0.01047 0.00033 -0.00316 0.00331
...
H3C 0.744677 0.653136 0.790701 1.00000
0.03495 0.03791 0.02008 -0.00475 0.00301 0.02567
H3D 0.972592 0.767719 0.779693 1.00000
0.02527 0.01652 0.03160 -0.00411 -0.00837 0.00137
FILE shelx.hkl
FORMAT (3i4,2f8.0,i4,f8.0)
LOADREF F^2 21600 7
END

```

In this example, the reflection data are from time-of-flight neutron diffraction. The number of scale factors is eleven, hence more than one SCALE directive is given. The 23 parameter data entries have *Uij*'s given in the SHELX order, hence the UIJSX entry in the **LOADPAR** instruction. The 21600 reflection data are read from a formatted file (actually SHELX HKLF 2 format), with seven data items in each record *h k l obs sigobs scgrp wave*. The TOF entry on the **NEUTRON** directive tells the program that the seventh item is actually the wavelength and a dummy *tbar* entry (0.05) is provided in the resultant NDAT 8 format **xd.hkl** file.

The input reflection data (**shelx.hkl**)

```

6 -8 0 3353.52 58.36 1 1.8500
5 -5 112318.51 100.34 1 2.9300
5 -5 -112714.86 107.39 1 2.9500
6 -6 2 7129.51 77.94 1 2.4300
6 -6 -2 7369.12 82.71 1 2.4600
5 -5 -5 7384.24 83.20 1 2.8400
11 -11 0 14779.01 170.73 1 1.3400
5 -5 -4 6945.49 80.51 1 2.8800
6 -6 -1 6562.23 79.32 1 2.4600
...

```

The output reflection data (**xd.hkl**)

```

NIEN3      F^2      NDAT      8
6 -8 0 1      3353.520      58.360 0.050000 1.8500
5 -5 1 1      12318.510      100.340 0.050000 2.9300
5 -5 -1 1      12714.860      107.390 0.050000 2.9500
6 -6 2 1      7129.510      77.940 0.050000 2.4300
6 -6 -2 1      7369.120      82.710 0.050000 2.4600
5 -5 -5 1      7384.240      83.200 0.050000 2.8400
11 -11 0 1      14779.010      170.730 0.050000 1.3400
5 -5 -4 1      6945.490      80.510 0.050000 2.8800
6 -6 -1 1      6562.230      79.320 0.050000 2.4600
...

```

3.4 Refining Neutron Data in XD2024

Although the main purpose of XD2024 is analysis of electron density from high resolution X-ray structure factors (either experimental or theoretical) it is also possible to refine neutron data with the program. This may serve a useful purpose as the more accurate positional & thermal parameters obtainable from neutron diffraction could be used as an effective starting model for the electron density analysis.

For monochromatic neutron diffraction data, the procedure in **XDINI** is essentially identical to that for X-ray diffraction data. The presence of neutron data is detected automatically when using SHELX or CIF input. For input via the XD_INI.INP route, there is an extra instruction NEUTRON to indicate this situation (see above). The neutron scattering model is automatically set (*model -4 2 0 0).

For time-of-flight neutron data, refinement is also possible, but in order to apply an extinction correction, it is necessary to read the actual wavelength associated with each reflection from the reflection file **xd.hkl**. This may be achieved by utilising the NDAT 7, NDAT 8 or NDAT 14 options in the header of that file. See above for the expected order of data items in this file. The current version of **XDINI** can utilise all options for input of TOF neutron data sets. For SHELX input, the TOF mode is indicated by the HKLF 2 instruction in **shelx.ins** and is expected to be accompanied by such a formatted file in **shelx.hkl**. See above for the requirements when using **xd_ini.inp** input. If cif input is used, the reflection file **xd.fcf** must contain the data item **_refln_wavelength** for each individual reflection, and also the data items **_refln_scale_group_code** (if more than one scale factor is present) and **_refln_mean_path_length_tbar** if this is known.