A Computer Program Package for Multipole Refinement and Analysis of Electron Densities from Diffraction Data

Manual Version February 2004
Program Version 4.10 July 2003

T. Koritsanszky
Middle Tennessee State University
tkoritsa@mtsu.edu

S. Howard
University of Wales Cardiff
howardst@cardiff.ac.uk

P. Macchi
Università di Milano
piero.macchi@istm.cnr.it

C. Gatti
CNR-ISTM Milano
c.gatti@istm.mi.cnr.it

L. J. Farrugia
University of Glasgow
louis@chem.gla.ac.uk

P. R. Mallinson
University of Glasgow
paul@chem.gla.ac.uk

A. Volkov
State University of New York at Buffalo
volkov@chem.buffalo.edu

Z. Su
State University of New York at Buffalo

T. Richter
Freie Universität Berlin

N. K. Hansen
Université Henri Poincaré, Nancy I
hansen@lmcpi.u-nancy.fr
How to obtain XD

XD is available by subscription, which is currently US$ 1000 for an academic license.

New users wishing to obtain a license should contact Tibor Koritsanszky by email (tkoritsa@mtsu.edu) regarding the arrangements.

For registered users, the source codes for Unix/Linux and executables for MS Windows are available via the Internet.

Official XD web site : http://xd.chem.buffalo.edu

Please report any bugs by email to : xd-bugs@xd.chem.buffalo.edu

Acknowledgements

This work is made possible through the financial support of the following institutions:

- University of Glasgow under its New Initiatives Scheme.
- The Commission on Charge, Spin and Momentum Densities of the International Union of Crystallography.
- The State University of New York at Buffalo.

We would like to thank the following institutions for research facilities:

- Institut für Kristallographie, Freie Universität Berlin
- Department of Chemistry, University of Wales Cardiff
- Department of Chemistry, University of Glasgow
- Department of Chemistry, State University of New York at Buffalo
- Laboratoire de Minéralogie-Cristallographie Université de Nancy I
- Dipartimento di Chimica Strutturale e Stereochimica Inorganica, Università di Milano
- Istituto di Scienze e Tecnologie Molecolari CNR Milano
Notations

Throughout this manual we have tried to follow the following conventions in notation:

**BOLD face** letters are used for vectors and matrices in formulas, with vectors in lower case and matrices in upper case. The transpose of a matrix $\mathbf{M}$ is denoted as $\mathbf{M}'$ and its inverse as $\mathbf{M}^{-1}$.

For the description of commands, bold face denotes keywords which should be typed as shown. Uppercase and lowercase should not matter, they are only used to distinguish between command keywords and options in the manual. When in doubt, use uppercase.

*Italic* is used for parameters which should be replaced with actual values. *Emphasized* text is also typed in italic.

*Courier font* is used for examples and for file names.

Parentheses "( )" or question marks "?" enclose optional parts in command descriptions. In addition, lists which show a number of choices are enclosed in parentheses with a vertical bar "|" as the delimiter. (Yes, this is somewhat ambiguous.)

Brackets "[ ]" denote default values.
Corrections & Improvements

A number of major corrections, bug-fixes and improvements have been made since the last official release of XD (Rev 14, 1999)

- The exponents of 3d and 4s orbitals were interchanged by XDPROP when computing a default single-zeta exponent for the multipoles (option CSZD for DEFV in the SCAT Table). This error could cause severe problems in the topology, but it was quite easily detectable by a simple *check option in the output of XDPROP
- The same occurred for ns and (n-1)s/p for some closed-shell cations.
- The default radial exponent for fourth-row atoms was $n_l = 4$ instead of $n_l = 6$ (in both XDLSM and XDPROP)
- The configuration of the SCAT table was not read by XDPROP and the default configuration (stored in the databank) was always applied.
- When computing the radial function for multipoles with an orbitalic product (option CHFW for DEFV), the normalisation of the spherical harmonics was applied twice in XDPROP. This error gave rise to underestimated deformations around the atom. Because it was mainly adopted for describing 3d orbitals of transition metals, its effect was to reduce the polarisation of the inner valence shell, without affecting too seriously the density in the region of the bonding and the overall topology. Again, only the default valence orbitals were used, despite any different request by the user.
- The option GROUP atoms was not applied when computing the electrostatic potential (esp).
- Molecular quadrupole moments had an incorrect unit transformation.
- Molecular dipole and quadrupole calculations did not support the CHFW option for radial functions.
- TOPXD is now directly interfaced to XD.
- The anisotropic extinction refinement has been corrected (and a bug concerning interpretation of the wavelength has been removed - see Section 4.6.8 for more details)
- Symmetry operations 6, and 6, were not properly supported.
- $U$ factors for H atoms were erroneously transformed by XDINI from SHELX input if restraints were present.
- In the presence of atoms in special positions, XDPROP had some problems in correctly reproducing all the atoms requested by users through the APPLY symm option.

Updates

Further updates will be issued from time to time by the XD programming team. These will be placed on the official XD web-site given above. See this web-site for news of the latest bug-fixes and modifications to XD.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Overview</td>
<td>5.2 Files used and created by XDGEOM</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>5.3 Input instructions for XDGEOM</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>5.4 Example Master File segment for XDGEOM</td>
<td>53</td>
</tr>
<tr>
<td>6</td>
<td>XDPROP – ONE-ELECTRON PROPERTIES BASED ON THE MULTIPOLAR REPRESENTATION OF THE EDS</td>
<td>6.2 Input Instructions for XDPROP</td>
<td>55</td>
</tr>
<tr>
<td>7</td>
<td>XDFOUR – A GENERAL 2-D AND 3-D FOURIER SYNTHESIS PROGRAM</td>
<td>7.2 Files used and created by XDFOUR</td>
<td>71</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>7.3 Input instructions for XDFOUR</td>
<td>71</td>
</tr>
<tr>
<td>8</td>
<td>XDFFT - A FAST FOURIER TRANSFORM PROGRAM</td>
<td>8.2 Files used and created by XDFFT</td>
<td>75</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>8.3 Input instructions for XDFFT</td>
<td>75</td>
</tr>
<tr>
<td>9</td>
<td>XDGRAPH - VISUALISING THE RESULTS</td>
<td>9.2 The Command Line Interface</td>
<td>77</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.3 The Graphical User Interface</td>
<td>77</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.4 Running XDGRAPH</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.5 Commands</td>
<td>79</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.6 Common Options</td>
<td>88</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.7 Common Subcommands</td>
<td>89</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.8 Toolbox</td>
<td>90</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.9 Display Driver</td>
<td>91</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.10 Examples</td>
<td>91</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9.11 A few words about Tcl</td>
<td>92</td>
</tr>
<tr>
<td>10</td>
<td>TOPXD – FULL TOPOLOGICAL ANALYSIS</td>
<td>10.2 TOPXD Instructions</td>
<td>95</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.3 Topological Analysis of Electron Density</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.4 Topological Analysis of the Laplacian of Electron Density</td>
<td>104</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.5 Evaluation of atomic and/or NNA basin properties</td>
<td>106</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.6 2-Dimensional (2D) Plots</td>
<td>110</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.7 3-Dimensional (3D) plots</td>
<td>117</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.8 3-Dimensional (3D) visualization of atomic basins</td>
<td>119</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.9 Description of the atomic properties evaluated by TOPXD</td>
<td>121</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10.10 Frequently Asked Questions</td>
<td>122</td>
</tr>
<tr>
<td>11</td>
<td>XD UTILITY PROGRAMS</td>
<td>11.3 XDCIF – A Program to Create an Archive CIF</td>
<td>130</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>11.4 XDWTAN – A Program to Analyse the Weighting Scheme</td>
<td>131</td>
</tr>
</tbody>
</table>
11.5 AIM2TAB .......................................................... 132
11.6 LSDB ............................................................. 132
11.7 ADDGRID, SCALEGRID ........................................ 132

CHAPTER 12 .................................................................. 134

EXAMPLE INPUT FILES ......................................................... 134
11.1 Parameter file (XD.INP/XD.RES) .................................. 134
12.2 Master file .............................................................. 135

CHAPTER 13 ................................................................... 140

XD INSTALLATION .............................................................. 140
13.1 What you need .......................................................... 140
13.2 Installing XD on VMS systems (no longer supported) .... 140
13.3 Installing XD on Unix systems ..................................... 141
13.4 Installing XD for Windows ........................................... 142
Chapter 1

Introduction

1.1 Synopsis for non-crystallographers

Chemical crystallography and quantum chemistry encompass our knowledge about the detailed structure of molecules, their properties and reactions, and the distribution of electronic charge in their atoms and chemical bonds. On this insight are based all modern theories of chemical reactivity, and the design principles for new materials and drugs. Great advances in the last two decades have led to the present theoretical and experimental methods for determining molecular structure at the electronic level; we can in principle (and increasingly in practice) obtain not just the positions of atoms in molecules but all other topological properties of the associated electron distribution (ED).

A beam of X-rays is diffracted by the electrons in a crystalline material, just as visible light is diffracted by larger objects. Recombination of diffracted light by means of lenses can give a magnified image of the object; X-rays, having a wavelength about four orders of magnitude shorter than that of visible light, produce an image of the electron or charge density distribution characteristic of the diffracting crystal. There exist no lenses as such for X-rays, but recombination of diffracted rays into an image can be brought about by suitable detection followed by computational Fourier transformation. The experiment is effectively an X-ray microscope for the disposition of electronic charge.

In practice we can bypass the Fourier transformation, because quantum mechanics enables us to construct a mathematical model of the charge density in a crystal. The parameters of such a model can be adjusted to reproduce the experimentally-measured pattern of diffracted X-rays, given prior knowledge of the arrangement of atomic nuclei in the crystal lattice. For chemical (as distinct from biological) molecules this can usually be found routinely using the methods of conventional crystal structure analysis programmed in widely available computer packages. This leads to a "ball and stick" model of the atoms and bonds representing the topology of the charge density at the level of its most salient features, found at the positions of the atomic nuclei. It is obtained by Fourier transformation of the diffracted X-ray pattern at relatively low resolution. Next we can proceed with a far more elaborate, so-called "multipole" model of the crystalline density, fitting it to a diffraction experiment carried out at high resolution, such that two points as close together as 0.4x10^{-10} m can be distinguished. As mentioned earlier, we need no Fourier transformation at this stage because the charge density in fine detail can be computed directly from the fitted multipole model. One major component of the XD package is the program for least squares (lsq) fitting of a multipole model to the experimental data.

Once a charge distribution has been obtained experimentally, various chemical and physical properties that depend on the distribution can be derived. The chemical structure of molecules can be extracted from an analysis of the topology of the charge distribution, the features of which are summarized by the curvatures of the charge density at its critical points. Each feature, maximum, minimum or saddle has associated with it a point in space called a critical point, where the density is flat. One type of critical point has all three curvatures in 3-D space negative; it is found at the sites of atomic nuclei. Other types, with both positive and negative curvatures, are associated with bonding interactions between atoms. Because the strength and nature of the interactions are characterized by topology, the chemistry of the molecule can be recovered as a property of its charge distribution. A program for deriving molecular properties from the multipole model of the charge
distribution is thus another major component of XD. Many of these properties can be displayed pictorially, using the 2-D and 3-D graphics programs which plot contour, relief and iso-surface maps of selected properties such as the deformation density, the Laplacian of the total density, the electrostatic potential etc.

1.2 Experimental electron densities

X-ray diffraction was first applied with the purpose only of determining the positions of atoms in crystals and hence the geometrical structure of crystals and molecules. With the development of single-crystal diffractometers and computing facilities from the middle 1960s onwards came studies aimed at obtaining an experimental description of the chemical bonding to compare with the picture given by quantum chemistry theoretical calculations [1-4]. Accurate experimental measurement of the charge density in a crystal has been feasible since that time, following the development of sufficiently compact parameterized descriptions of molecular densities [5,6]. One of the most exciting applications of such an analysis is the evaluation of one-electron properties in molecular crystals. In a pioneering paper [7] Coppens et al. demonstrated the feasibility of this technique for a number of centrosymmetric crystals. However, applications to non-centrosymmetric materials, such as organic materials with non-linear optical applications, have been relatively few. In part, this is certainly due to the increased difficulty of obtaining accurate model structure factors when the phase is a continuous variable. Nevertheless, recent applications have demonstrated the usefulness and potential accuracy of the technique in the non-centrosymmetric case [8,9].

ED determinations [10] are based on intensity measurements of X-ray photons elastically scattered by crystals. In the next section a brief summary is given on some theoretical aspects of the procedure to extract the ED from X-ray diffraction data. For more detailed descriptions the reader is referred to references [11,12].

1.3 Theoretical aspects of electron density determination

According to the kinematical theory of scattering [13] the total diffraction intensity is

\[ I_{\text{tot}} = \langle I \rangle_T \propto \langle |F(h,q)|^2 \rangle_T \]  

(Eq. 1-1)

where \(F(h,q)\) is the Fourier transform of \(\rho(r,q)\), the static ED at a given nuclear configuration \(q\). \(h\) is the Bragg vector with integral components \(h_1, h_2, h_3\) relative to the reciprocal axes \(a^*, b^*, c^*\). \(V\) is the unit-cell volume and \(\langle \rangle\), means thermal averaging over all vibrational states. By disregarding the diffuse scattering altogether

\[ I_{\text{tot}} = I_{\text{Bragg}} = \langle |F(h,q)|^2 \rangle_T \]  

(Eq. 1-3)

it is assumed that the averaged scattering from a dynamic system can be well approximated by its main component, the scattering from the average structure [14,15]. This expression relates the intensity to the ED and its derivation implicitly includes assumptions not directly deducible from the experiment; assumptions on the coupling between nuclear and electronic motion and on the partitioning of the molecular ED into atomic components (convolution approximation). Based on this equation the ED in the crystal can be given by a Fourier summation

\[ \rho(r) = V^{-1} \sum_h F_h \exp(-2\pi i h r) \]  

(Eq. 1-4)
This direct evaluation of \( \rho \) to a desired level of resolution, is subject to severe limitations: (i) the observed structure factors are affected by experimental errors, (ii) the phases are not measured, (iii) only a finite number of reflections can be collected. Due to these limitations the interpretation of the X-ray data necessarily involves modelling the ED and optimizing its parameters by adjusting the calculated structure factors to those measured.

1.4 Electron density - Structure factor models

Within the convolution approximations the dynamic ED is

\[
\langle \rho(\mathbf{r}) \rangle_T = \sum_k \rho_k(\mathbf{r} - \mathbf{q}_{ko} - \mathbf{u}_k) P(\mathbf{u}_k) d\mathbf{u}_k
\]  

(Eq. 1-5)

where the summation runs over the density units \( \rho_k \) centered at \( \mathbf{q}_{ko} \) and \( P(\mathbf{u}_k) \) is the probability distribution function (pdf) describing the displacement \( \mathbf{u}_k \) of the \( k \)-th center with respect to its equilibrium position. The structure factor is then the Fourier transform of \( \langle \rho(\mathbf{r}) \rangle_T \)

\[
F(h) = \sum_k f_k(\mathbf{h}) t_k(h) \exp(2\piih\mathbf{q}_{ko})
\]  

(Eq. 1-6)

where \( f_k \) is the static scattering power of the \( k \)-th density unit and \( t_k \) is the associated temperature factor. The commonly used scattering models differ in the description of \( f_k \) and \( t_k \), both of which are, in general, complex functions of static and dynamic parameters, respectively.

1.5 Conventional formalism

This generalized form (1.6) is reduced to the conventional model if \( \rho_k \) is taken as the spherical atomic density and the nuclear motion is described within the harmonic approximation. This formalism disregards static deformations due to the chemical bonding and the least squares estimates of the corresponding parameters are likely to be biased. Such errors ("asphericity" shifts) usually manifest themselves in significantly shorter bond distances and smaller bond angles (at atoms with lone-pair electrons) relative to the values obtained by neutron diffraction. The accuracy of the thermal parameters is even more doubtful as the anisotropic displacements can absorb charge deformation. To overcome the inadequacy of the isolated atom model several methods can be applied.

1.6 High order refinement

In the atomic regions where the electron density is less affected by the bonding the isolated atom model is expected to be a fair approximation. The sharp core density has appreciable contribution to reflections at high Bragg angle where the scattering by the more diffuse valence or bond density is negligible. For this reason a refinement emphasizing the high-order data is expected to yield atomic parameters less biased by the inadequacy of the spherical-atom model [16].

1.7 The aspherical-atom formalism

The accuracy of the parameters can be significantly increased by implementing aspherical density models into the fit of all measured data. To account for the density deformations due to chemical bonding, several methods have been developed and applied [17,18]. One of the most successful refinement techniques is based on the nucleus-centered finite multipole
expansion of the ED [6]. This formalism, refined by Hansen & Coppens [19] is implemented in XD. The aspherical atomic ED is divided into three components:

\[ \rho(r) = \rho_c(r) + P_v \rho_v(kr) + \rho_d(k'r) \]  
(Eq. 1-7)

where \( \rho_c \) and \( \rho_v \) are the core and spherical valence densities (sphv), respectively and

\[ \rho_d(k'r) = \sum_l R_l(k'r) \sum_{m=-l}^l P_{lm} y_{lm} \left( \frac{k'}{r} \right) \]  
(Eq. 1-8)

is the term accounting for valence deformations. The \( y_{lm} \) are density normalized, real spherical harmonics, such that:

\[ \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} | y_{lm} | d\Omega = 2, l > 0 \]  
(Eq. 1-9)

\[ = 1, l = 0 \]  
(Eq. 1-10)

while \( R_l \) are properly chosen radial functions, and an element of solid angle \( d\Omega = \sin \theta d\theta d\phi \).

The isolated-atom valence density and the radial functions \( R_l \) are modified by the screening constants (\( \kappa \) and \( \kappa' \), respectively) to account for the radial expansion or contraction of the valence shell. The corresponding scattering factor is

\[ f(h) = f_c(H) + P_v \left( \frac{h}{\kappa} \right) + \sum_l \left\{ J_l \left( \frac{h}{\kappa'} \right) \right\} \sum_{m=-l}^l P_{lm} y_{lm} \left( \frac{h}{\kappa'} \right) \]  
(Eq. 1-11)

where \( \langle J_l \rangle \) is the \( l \)-th order Fourier-Bessel transform of \( R_l \):

\[ \langle J_l \rangle = 4\pi^2 \int j_l(2\pi hr) R_l(r) r^2 dr \]  
(Eq. 1-12)

with \( j_l \) being the \( l \)-th order spherical Bessel function. Closed-form expressions for evaluating \( \langle J_l \rangle \) using different types of radial functions have been given in reference [18].

### 1.8 Orbital vs. Multipole formalism

For a single-Slater determinant atomic wavefunction composed of orthogonal spin-orbitals the electron density is given by

\[ \rho = \sum_i n_i | \phi_i |^2 \]  
(Eq. 1-13)

where \( n_i \) is the orbital occupation number (1 or 2) of the \( i \)-th atomic orbital,

\[ \phi_i = \phi_{nlm} = R_{nl} y_{lm} \]  
(Eq. 1-14)

If the radial part \( R_{nl} \) is expanded in terms of basis functions

\[ R_{nl} = \sum_j C_{nlj} O_{lj} \]  
(Eq. 1-15)

the density unit \( \rho_{nlm} \) corresponding to \( \phi_{nlm} \) is given by the following linear combination:

\[ \rho_{nlm} = \sum_{jk} D_{njk} O_{lj} O_{lk} y_{lm} y_{lm} = R_{nl}^2 y_{lm} y_{lm} \]  
(Eq. 1-16)
The spherical harmonics form a complete basis set, thus their product can be expanded over spherical harmonics:

\[ y_{lm} y_{l'm'} = \sum_{LM} C_{Ll'Mm'} y_{LM} \]  

(Eq. 1-17)

The Clebsch-Gordon coefficients \( C_{Ll'Mm'} \) are given for both complex and real spherical harmonics (up to \( l, l' = 2 \)) in the literature [12]. It follows that the orbital product representation of the atomic density is completely equivalent to the multipolar description. This equivalence does not hold for molecules because of the two-center orbital products occurring in expression (1.13).

### 1.9 Radial functions and scattering factors

The core and spherical valence density are calculated from Hartree-Fock atomic wavefunctions [20] expanded in terms of Slater-type basis functions:

\[ O_l = [(2n(l)!)^{1/2} (2\zeta_l)^{n(l)+1/2} r^{n(l)} \exp(-\zeta r)] \]  

(Eq. 1-18)

where \( z_i \) are energy optimized orbital exponents.

The radial functions of the deformation density are also taken as simple Slater functions:

\[ R_l(r) = \frac{a_i^{n(l)+3}}{(n(l) + 2)!} r^{n(l)} \exp(-a_i r) \]  

(Eq. 1-19)

with \( n(l) \geq l \) to obey Poisson’s equation [21] and with values for \( a_i \) as deduced from the single-\( \zeta \) wavefunctions. As shown above, the evaluation of the scattering factor of an orbital product requires the calculation of \( L \)-th-order Fourier Bessel transforms of \( O_l \) \( \langle j_L \rangle \). The simple scheme below shows how \( L \) is related to \( l \) and \( l' \) \( (l=0,1,2 \) for s,p and d, respectively):

<table>
<thead>
<tr>
<th>( l ) ( l' )</th>
<th>s</th>
<th>p</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>p</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Taking the carbon atom as an example, the following scattering factors can be generated from the wavefunction:

- core: \( \langle j_0 \rangle (1s1s) \)
- sphv: \( \langle j_0 \rangle (2s2s) + \langle j_0 \rangle (2p2p) \)

Dipolar \( (l = 1) \) and quadrupolar \( (l = 2) \) radial scattering functions included in the deformation term in (1.8) could be composed as the Fourier-Bessel transforms of sp and pp type orbital products:

- defv: \( \langle j_1 \rangle (2s2p), \langle j_2 \rangle (2p2p) \)

### 1.10 The temperature factor

In harmonic approximation the vibrational pdf of the nuclear displacement vector \( \mathbf{u} \), taken with respect to the equilibrium position \( \mathbf{q} = \mathbf{q}_0 \), is a normal distribution:
Chapter 1 - Introduction

\[ P_o (u) = (2\pi)^{-3/2} (\det U)^{-1/2} \exp(-1/2u'U^{-1}u) \]  
(Eq. 1-20)

where \( U \) is the mean-square displacement amplitude (MSDA) matrix.

The corresponding atomic anisotropic temperature factor is the Fourier transform of \( P_o(u) \):

\[ t_o (h) = \exp(-2\pi^2h'Uh) \]  
(Eq. 1-21)

Anharmonic models in practical use are based on statistical approaches. If the anharmonicity is small the corresponding pdf can be expanded about the normal distribution. In the Gram-Charlier expansion [22] implemented in XDLSM, the anharmonic pdf is approximated in terms of zero and higher derivatives of the normal distribution:

\[ P(u) = (1 + \frac{1}{3!}C_{jkl}H_{jkl} + \frac{1}{4!}C_{jklmn}H_{jklmn} + ...)P_o \]  
(Eq. 1-22)

where \( H_{jklmn} \) are three dimensional Hermite polynomials being functions of \( U \) and \( u \), while the coefficients \( C_{jkl} \) are the quasi-moments being related to the moments of the pdf. The advantage of this form is that its Fourier transform is reduced to a simple power series expansion about the harmonic temperature factor:

\[ T(H) = (1 - \frac{4}{3}\pi^3C_{jkl}h_jh_kh_l + \frac{2}{3}\pi^4C_{jklmn}h_jh_kh lh_m + ...)T_o (H) \]  
(Eq. 1-23)

1.11 Deformation electron density

The conventional model is based on the pro-molecular density which is the superposition of the spherical atomic densities \( \rho_k(r) \) centered at the actual nuclear positions in the molecule. The promolecule can serve as a reference state relative to which charge migrations due to bond formations are expected to become visible [23].

\[ \delta\rho(r) = \rho_{mol}(r) - \sum_k \rho_k (r - r_k) \]  
(Eq. 1-24)

To interpret the \( \delta\rho(r) \) one always has to critically examine not only the method yielding the molecular electron density but the effect of the preconceptions applied in composing the promolecule. For atoms with a degenerate ground state, \( \rho_k \) is obtained by sharing the valence electrons among orbitals of different angular dependence regardless of their ability to form a bond in the actual arrangement of the atoms. As a result the obtained deformation electron density may not show the expected features of the covalent bond or lone-pair density [24].

In order to obtain a chemically meaningful deformation electron density, an alternative promolecule has been proposed for which the configuration and the orientation of the ground state of each constituent atom is correctly specified by a fitting procedure [25]. To elucidate important aspects of delocalization, effects of substitution or intermolecular interactions, one can consider fragments or molecules to choose as the basis for comparison [26,27]. Atoms prepared for bond formation can also serve as references [28].

If the deformation electron density is evaluated by a Fourier summation

\[ \delta\rho(r) = \sum_h [F_o (h) - F_r (h)]\exp(-2\pi h r) \]  
(Eq. 1-25)

the series termination error is considerably decreased. The phases and the \( F_r \) are usually calculated from the promolecule with atomic and positional parameters obtained from (i) neutron diffraction data (X-N) [29], (ii) conventional refinement on high-order X-ray data (X-X\textsubscript{ho}), (iii) full-data aspherical-atom refinement (X-X\textsubscript{mul}).
1.12 Experimental requirements

The applicability of the above formalism depends on the compound to be studied and its crystalline form, the radiation used and the method of the data collection. The kinematic theory is valid only in a certain frequency range: \( \mu_r > \mu > \mu_K \), where \( \mu_K \) corresponds to the K absorption edge of any atom in the molecule and \( \mu_r \) is the frequency limit, where relativistic effects occur. Accordingly, atoms with high atomic number (\( Z > 18 \)) are not well suited for charge density studies when a standard X-ray source is used. Bonding effects are likely to be invisible for atoms with small valence to core electron ratio [30].

The most important requirement for an accurate measurement is to maintain kinematical conditions or to make the systematic errors, due to dynamic scattering, correctable. To reveal these effects equivalent reflections should be measured. To minimize the diffuse scattering the data should be collected at low temperature. Details of the data reduction can be found in references [31-34].

1.13 Determination of atomic and structural properties from charge distributions

1.13.1 Critical points of the charge density

Once a charge distribution has been obtained experimentally, various chemical and physical properties that depend on the distribution can be derived. Bader [35] shows how the chemical structure of molecules can be extracted from an analysis of the topology of \( \rho(r) \), the features of which are summarized by the curvatures of \( \rho(r) \) at its critical points. Each feature, maximum, minimum or saddle has associated with it a point in space called a critical point, where the first derivatives of \( \rho(r) \) vanish. At such a point, denoted by position vector \( r_c \),

\[
\nabla \rho(r_c) = \hat{i} \frac{\partial \rho}{\partial x} + \hat{j} \frac{\partial \rho}{\partial y} + \hat{k} \frac{\partial \rho}{\partial z} = 0
\]

where \( \hat{i}, \hat{j}, \hat{k} \) are unit vectors. Whether a function is a maximum or minimum is determined by the sign of its second derivative, or curvature, at the stationary point. In general, for an arbitrary choice of coordinate axes, there will be nine second derivatives of the form \( \frac{\partial^2 \rho}{\partial x \partial y} \) in the determination of the curvatures of \( \rho \) at a point in space. Their ordered 3\( \times \)3 array, the Hessian matrix of the charge density, can be diagonalized to yield the principal axes of curvature, with respect to which the magnitudes of the three second derivatives of \( \rho \) are extremized. The principal axes and their corresponding curvatures at a critical point in \( \rho \) are obtained as the eigenvectors and corresponding eigenvalues (\( \lambda \)) of the Hessian matrix of \( \rho(r) \). The rank \( \omega \) of a critical point is the number of non-zero eigenvalues or curvatures of \( \rho \) at the critical point, while its signature \( \sigma \) is the algebraic sum of the signs of the curvatures at that point. The critical point is labelled by giving the pair of values \( (\omega, \sigma) \). With few exceptions the critical points of charge distributions for stable molecules are of rank three, and there are four possible signature values and labels:

(3,-3) all curvatures are negative and \( \rho \) is a local maximum at \( r_c \).

(3,-1) two curvatures are negative and \( \rho \) is a maximum at \( r_c \) in the plane defined by their corresponding axes. \( \rho \) is a minimum at \( r_c \) along the third axis, perpendicular to this plane.

(3,+1) two curvatures are positive and \( \rho \) is a minimum at \( r_c \) in the plane defined by their corresponding axes. \( \rho \) is a maximum at \( r_c \) along the third axis, perpendicular to this plane.

(3,+3) all curvatures are positive and \( \rho \) is a local minimum at \( r_c \).
The traditional association of nuclear positions with local maxima in $\rho(\mathbf{r})$ can now be formalized as the statement that nuclear positions behave topologically as (3,-3) critical points in the charge distribution.

### 1.13.2 Interatomic surfaces and chemical bonds

A useful function is obtained in the form of the gradient vector field of the charge density, represented through a display of the trajectories traced out by the vector $\nabla \rho$. The gradient vector points in the direction of the greatest increase in $\rho$, so these trajectories are perpendicular to the contour lines of $\rho$. They have the property of originating or terminating at critical points in $\rho$. The charge distribution is partitioned into disjoint regions by surfaces for which

$$\nabla \rho(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0$$

where $\mathbf{n}$ is the vector normal to the surface. These so-called zero flux surfaces are the interatomic surfaces or quantum mechanical boundaries of the atoms, and contain (3,-1) critical points when the atoms are chemically bonded. The pairs of gradient paths which originate at each (3,-1) critical point and terminate at the nuclei define a line through the charge distribution linking the neighbouring nuclei, along which $\rho(\mathbf{r})$ is a maximum with respect to any neighbouring line. This line is called a bond path and the (3,-1) critical point is referred to as a bond critical point. This is the topological definition of a chemical bond, formalizing the theoretically predicted and experimentally observed accumulation of charge between bonded nuclei. Chemical structure can thus be recovered as a property of the charge distribution. The strength and nature of the chemical bond can be characterized by the value of various properties evaluated at the bond critical points, e.g. bond order, bond ellipticity, $\rho(\mathbf{r}_c), \nabla^2 \rho(\mathbf{r}_c)$ [22].

The value of $\rho_c$ in a bond measures its strength [36]; the trace of the Hessian at $\mathbf{r}_c$ measures the extent of depletion or concentration of charge; and the ratio of eigenvalues of this matrix (the bond ‘ellipticity’ $\epsilon$) measures the degree of planarity or conjugation. More precisely, $\epsilon = (\lambda_2/\lambda_1) - 1$, where the $\lambda$’s are the two eigenvalues of the Hessian corresponding to directions perpendicular to the bond.

Stationary points in $\rho$ have been applied in characterizing benzenoid aromaticity [37], homoaromaticity and hyperconjugativity [38-40], and electrophilic substitution [41,42]. A number of applications of the topological properties of experimental charge distributions obtained from neutron and X-ray diffraction data for organic molecular crystals have been reported [43-46].

### 1.13.3 Lewis electron pairs - the Laplacian

The trace of the Hessian matrix, the quantity

$$\nabla^2 \rho(\mathbf{r}) = \frac{\partial^2 \rho}{\partial x^2} + \frac{\partial^2 \rho}{\partial y^2} + \frac{\partial^2 \rho}{\partial z^2}$$

is termed the Laplacian of $\rho$ and has physical meaning as representing local concentrations, where $\nabla^2 \rho < 0$, and depletions, where $\nabla^2 \rho > 0$, of the charge density. Electronic charge is compressed above its average distribution in regions where the Laplacian is negative, and expanded relative to its average distribution where the Laplacian is positive. Maxima and minima in the function $\nabla^2 \rho$ are to be distinguished from local maxima and minima in the charge density itself. Although the topology of $\rho$ yields a faithful mapping of the chemical concepts of atoms, bonds and structure, there is no indication of maxima in $\rho$ corresponding to the localized electron pairs of the Lewis model of electronic structure, of great importance to our interpretation of chemical reactivity and molecular geometry. The physical basis of this model is one level of abstraction above the visible topology of the charge density and
Chapter 1 - Introduction

appears instead in the topology of the Laplacian of \( \rho \), the scalar derivative of the gradient vector field of the charge density.

The Laplacian distribution recovers the electronic shell model of an atom by exhibiting a corresponding number of pairs of shells of charge concentration and charge depletion. For a spherical free atom, the outer or valence shell of charge concentration (VSCC) contains a sphere of uniform concentration of electronic charge. Upon entering into chemical combination, this shell is distorted and maxima, minima and saddles appear. The maxima correspond in number, location and size to the localized pairs of electrons assumed in the Lewis and VSEPR models of electron pairs. A local charge concentration is a Lewis base or nucleophile, while a local charge depletion is a Lewis acid or electrophile, and a chemical reaction corresponds to the combination of complementary features of the VSCC of the base and acid. The Laplacian distribution can thus be used to locate possible sites of nucleophilic attack, and to predict characteristics (such as hydrogen bonding) of the chemical reactivity in general.

Stationary points in \( \nabla^2 \rho(r) \), points of maximum charge concentration or depletion, are being extensively applied in studies of basicity and acidity [47-52]; to more general reactivity [53-56]; in accounts of molecular geometries [57]; and to directionality of hydrogen bonding [58,59]. Such points may generally be associated with either bonded or non-bonded electron pairs. Experimental determinations of \( \nabla^2 \rho \) distributions are included in [60-62].

Bibliography

Chapter 2

The XD System Files and the Master Control Program

2.1 File Name Conventions

The xd files are named according to the convention

\[ xd\(_{\text{model-ID}}\)(\_\text{property} \mid \text{program})\text{.type} \]

The prefix \( xd \) serves to distinguish the file as an XD system file from others that the user may want to keep in the same directory. The fields in parentheses are optional.

2.1.1 The Model-ID Field

The \textit{model-ID} field can be used to distinguished different refinement models. It is applied to the parameter file and related files only when the model-ID is specified on the command line (\textit{cf.} Section 2.11).

2.1.2 The Property Field

The \textit{property} field refers to the property used for the calculations. It is mainly used for grid files and output from the property program.

\begin{itemize}
  \item \texttt{rho} \quad \text{electron density}
  \item \texttt{defden} \quad \text{deformation density}
  \item \texttt{gradrho} \quad \text{gradient of the electron density}
  \item \texttt{d2rho} \quad \text{Laplacian}
  \item \texttt{esp} \quad \text{electrostatic potential}
  \item \texttt{fou} \quad \text{Fourier map}
  \item \texttt{fft} \quad \text{Fourier map}
  \item \texttt{core} \quad \text{core density}
  \item \texttt{valence} \quad \text{valence density}
  \item \texttt{nucpot} \quad \text{nuclear potential}
  \item \texttt{sigrho} \quad \text{error of the electron density}
  \item \texttt{siglap} \quad \text{error of the Laplacian}
\end{itemize}

2.1.3 The Program Field

The \textit{program} field specifies the program which created the file. \textit{program} can be ini, lsm, pro, geom, fft or fou.

2.1.4 The File Type

The file \textit{type} can be one of the following: (a * marks files for which the \textit{model-ID} can be used; filenames for which the \textit{property} field is applied are marked with a †, while those which make use of the \textit{program} field are marked ‡)
mas    the master file

mas    the master file

cyc    cycle information

cyc    cycle information

hkl    reflection data

hkl    reflection data

out*‡  list-able output

out*‡  list-able output

inp*    see res

inp*    see res

res*    all atomic and all refinable parameters

res*    all atomic and all refinable parameters

fou*    Fourier file (binary)

fou*    Fourier file (binary)
cov*    variance-covariance matrix (binary)
cov*    variance-covariance matrix (binary)
der*    matrix of structure factors derivatives (binary)
der*    matrix of structure factors derivatives (binary)
mat*    normal equation matrix (binary)
mat*    normal equation matrix (binary)
grd*†   property on a grid

grd*†   property on a grid

pth*    bond path

pth*    bond path
cps*†   critical points

cps*†   critical points

2.1.5 Examples of File Names

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xd.mas</td>
<td>(master file)</td>
</tr>
<tr>
<td>xd.lsm.out</td>
<td>(least squares results listing)</td>
</tr>
<tr>
<td>xd.res</td>
<td>(least squares parameter file on output)</td>
</tr>
<tr>
<td>xd.fou</td>
<td>(Fourier reflection file)</td>
</tr>
<tr>
<td>xd.cov</td>
<td>(variance-covariance matrix)</td>
</tr>
<tr>
<td>xd_defden.grd</td>
<td>(deformation density on a grid)</td>
</tr>
</tbody>
</table>

2.1.6 The Cycle Field and the Cycle File

The cycle field is applied only in case the file xd.cyc is present. In this case it is used to add cycle (version) numbers to the .res file. XDLSM usually reads from xd.inp and writes to xd.res. The cycle file contains instructions about the cycle to read and the first cycle to write. It consists of the single line

**READ rdcyc WRITE wrcyc**

If the read cycle equals zero, xd.inp is read. If the write cycle equals zero, the behaviour is undefined. After each refinement cycle, XDLSM sets rdcyc to wrcyc and increments wrcyc by one. After the last cycle, the new values are written to xd.cyc. When XDLSM is run again, it reads the last parameter file written and continues to write new parameter file following the existing ones. In case XDLSM dies before finishing the last cycle, no new cycle file is written and possibly created parameter files of this run are ignored.

In case the cycle naming scheme is to be used, one starts with **READ 0 WRITE 1** after XDINI has run. There is no need to touch the cycle file after that if things go right. To restart the refinement from a specific cycle, edit the **READ** field. The **WRITE** field only needs to be edited if parameter files should be overwritten after a successful run of XDLSM or, conversely if parameter files should not be overwritten after an unsuccessful run.

Note, that the cycle mechanism is only available on platforms which support multiple periods in file names (Unix/Linux and Windows all accept this mechanism).

2.2 The Master File xd.mas

Execution of the component programs of the XD package is started by a master control program directed by a master file. xd.mas is a free-format ASCII file. A line beginning with
Chapter 2  The XD System Files and Master Control Program

the exclamation mark (!) will be treated as a comment. This allows the user to keep all instructions in the file even if many of them are not in actual use. A single line can have up to 256 characters. If a line ends with a backslash (/), the next line will be read as a continuation line. The total length of a concatenated lines can be up to 256 characters. The input is not case sensitive. Two special tokens, \texttt{+inf} and \texttt{–inf}, can be used in places where numbers are expected. They represent plus and minus infinity.

The master file contains all instructions and options needed by all the programs of the XD package. It is created by XDINI, which provides an interface between XD and other commonly used crystallographic packages. The master file is divided into segments. Each program has its own input segment. The only segment which is shared by all the programs contains only general crystallographic information. Each line in a segment begins with a mnemonic string, usually followed by further keywords and/or numeric strings offering different sub-options or assigning default values to variables. In case of a multiple choice a sub-option can be selected by the asterisk (*) right before the corresponding keyword. Multiple flags, if not otherwise specified, are generally not allowed. Their presence should not normally terminate the program, but only the last selected option is actually activated.

A segment should begin with the module name as follows:

\textbf{MODULE} \texttt{(*)xdprogram}

where \texttt{xdprogram} is one of the program names (for example XDLSM, XDFOUR or XDPROP). A new line is read until the \textbf{END \ xdprogram} card which is the normal way to terminate the program.

A special type of input section is a structured sub-segment, which can be called a Table. It consists of a set of lines with ordered alphanumeric fields. The first row is a heading composed of keywords showing the content of the columns. The first keyword in this row serves as Table Identifier (TI) the others are not interpreted (one can’t use them to change the order of the fields, for example). No field can be skipped but the last one can always be omitted if a default value is available. A Table ends with the \textbf{END TI} instruction.

Example:

\begin{verbatim}
ATOM atom0 ax1 atom1 atom2 ax2 r/l tp tbl kap lmx sitesym chemcon
O(1) O(2) X O(1) C(1) Y R 2 1 1 4 NO
O(2) O(1) X O(2) C(2) Y R 2 1 1 4 NO O(1)
...
END ATOM
\end{verbatim}

\textbf{2.2.1 General instructions}

The master file begins with a section containing basic crystallographic data, common to all programs, as follows.

\textbf{2.2.1.1 \texttt{TITLE}}

\texttt{TITLE compound-id title-string}

The first eight characters serve as a compound identifier (CID) which is used to check if certain files belong together. The CID found in the first record of \texttt{xd.hkl} and \texttt{xd.inp} files has to match with that in the master file, otherwise XDLSM terminates with an error message.

\textbf{2.2.1.2 \texttt{CELL}}

\texttt{CELL a \{1\} b \{1\} c \{1\} alpha \{90\} beta \{90\} gamma \{90\}}
Unit cell parameters are given in Å and degrees. Default angles are 90 degrees, default axis lengths 1 Å.

The additional entry **CELLSD** may also be given. It is used by XDGEOM (Chapter 5) to compute errors on derived parameters which take the cell errors into account.

**CELLSD** $\sigma(a) \sigma(b) \sigma(c) \sigma(\alpha) \sigma(\beta) \sigma(\gamma)$ - no defaults

### 2.2.1.3 WAVE

**WAVE** *wavelength*

Radiation wavelength in Å.

### 2.2.1.4 LATT [C P]

**LATT** *centrosymmetry-flag lattice-type*

The centrosymmetry flag must be given as either **A** (non-centrosymmetric) or **C** (centrosymmetric). Lattice type may be **P**, **I**, **R**, **F**, **A**, **B**, or **C**. Rhombohedral lattices indexed on hexagonal axes (lattice type **R**) must be given as the obverse cell $(h+k+l=3n)$. Note that rhombohedral lattices indexed on rhombohedral axes have lattice type **P**.

### 2.2.1.5 SYMM

**SYMM** *general-position-coordinates*

**SYMM** $tx \ r11 \ r12 \ r13 \ ty \ r21 \ r22 \ r23 \ tz \ r31 \ r32 \ r33$

The positions may be given exactly as in the International Tables, the three coordinates being separated by commas (spaces are insignificant here). Alternatively, the operator elements may be separated by spaces (with no embedded spaces). Positions generated by a center of symmetry or corresponding to lattice centering should be omitted, and the origin must be at the center of symmetry in centrosymmetric structures. The entry 'SYMM $x,y,z$' is always assumed and will be ignored if given. More than one position may be given within one **SYMM** entry, if desired, by placing a semicolon between each of them, *e.g.* for space group $I\bar{4}$, (no. 80):

**LATT** A I

**SYMM** -X, -Y, Z; -Y, 1/2+X, 1/4+Z; Y, 1/2-X, 1/4+Z

or a **SYMM** entry may be given for each position.

The symmetry operation can also be written in a purely numerical way by giving a translation vector and a $3\times3$ rotation matrix, for example:

**SYMM** 0.0.0 -1.0 0.5 1.0 0. -25 0.0 1.

Note, that a mixture of the two ways of giving a symmetry operator (SHELX-type input) is not allowed.

### 2.2.1.6 BANK [CR | SCM | BBB]

**BANK** *databank type*

The databank type can be CR, BBB or SCM (see Section 2.5 for a detailed description). In the absence of the **BANK** instruction, the databank CR is used as default.
2.3 The parameter file \texttt{xd.inp} and \texttt{xd.res}

Type: ASCII, free-format, sequential

These are the input and output parameter files of XDLSM, and contain the information needed to calculate the electron density and related properties by XDPROP. \texttt{xd.res} is overwritten after each least squares cycle. See Table 2-1 (end of this Chapter) for a detailed description.

*Important! These files should not normally need to be edited.*

Many entries are also present in the master file. Specifications given in the master file have the priority. It means that the \texttt{xd.inp} and \texttt{xd.res} may differ according to any changes made in \texttt{xd.mas}.

2.4 The reflection file \texttt{xd.hkl}

Type: ASCII, free-format, sequential

An input file containing the observations. It consists of as many records as observations are available. See Table 2-2 (end of this Chapter) for a detailed description.

2.5 The databank files \texttt{xd.bnk}*

Type: ASCII, free-format, sequential

These files contain ground-state STO-HF atomic wave functions for elements from H to Xe including chemically relevant ions. The basis functions are Slater type orbitals of the form:

\[ b_{nl} = N_{nl} r^{(n-1)} \exp(-\zeta r)Y_{nl} \]

An atomic function is

\[ a_{nl} = Y_{nl} \sum_{k} b_{knl} C_k \]

where \( Y_{nl} \) are complex spherical harmonics. The orbital coefficients \( C \) and exponents \( \zeta \) are stored and used to calculate core and valence scattering factors according to a given electronic configuration. Additional data are also stored. A segment for an atom contains the entries given in Table 2-3. The files can be extended by introducing new segments identified by new atomic symbols. The element names are those conventionally used; first character upper case, second (if any) lower case. For ions the element name is followed by the order of ionization and the sign. Correct element names are: H Na Cu2+ F-

*Important! One should not modify the values of the available entries. This can be done, if necessary, in the \texttt{xd.mas} file with the SCAT table entries.*

The databank file \texttt{xd.bnk}, distributed in the previous versions of XD (up to Rev 1.14, 1999), is no longer valid because the introduction of new wave functions and analytical scattering factors required some changes in the format.

Three new databanks are now available:

\texttt{xd.bnk\_RHF\_CR}: (flag CR)

This contains the original XD databank in the new format. Clementi and Roetti [1] wave functions are tabulated for all neutral atoms and principal ions up to Kr. Single-\( \zeta \) functions
are taken from Clementi and Raimondi [2]. Analytical spherical scattering factor are from International Tables [3]. There are two changes with respect to the original file:

- For the metals Cr and Cu (both 4s\(^{3}d^{n-1}\)), the 4s orbital is now included in the "core" in agreement with the default convention adopted for all other transition metals
- The analytical spherical scattering factor (SPH) is now written with 13 entries: \((a(i),b(i),l=1,6),c\). Because the standard expansion of International Tables is up to \(i = 4\), entries 9-12 are 0.0; the 13\(^{th}\) entry is the constant term (which used to be the 9\(^{th}\) entry, when only nine fields were present in the old file).

**xd.bnk_RHF_BBB** (flag BBB)

Wave functions are taken from non-relativistic calculations by Bunge et. al [4] and include all neutral atoms up to Xe. Single-\(\xi\) functions are taken from Clementi and Raimondi [2] or Clementi and Roetti [1] (for atoms of the 5\(^{th}\) row). All the other parameters are identical to CR databank.

**xd.bnk_RDF_SCM** (flag SCM)

Wave functions fitted to a relativistic Dirac-Fock solution are taken from Su and Coppens [5] for neutral atoms up to Kr and from Macchi and Coppens [6] for neutral atoms Rb-Xe and all chemically relevant ions up to I. The analytical spherical scattering factor is taken from the same publications, where a six-term fitting was used (without constant term). Single-\(\xi\) functions as for databank BBB.

**Sample databank entry**

```
:C     6 12.0110 0.0033 0.0016 0.0181 0.0091 6.646 77 77 185
  SPH 2.3100 20.8439 1.02 10.2075 1.5886 0.5687 0.865 51.6512 0.000 0.000 0.000 0.000 0.216
  5.6727   1.6083   0.0000   0.0000   1.5679   0.0000   0.0000   0.0000
  0.0000  0.0000
  STO 1s  2   6 2s -2   6 2p -2   4 0.932620 5.435990 1 0.069310 9.482560 1 0.080990 1.057490 2 0.231950 6.510030
  2.684350 2 -0.001760 1.524270 2
  0.005590 2.684350 2 -0.005590 2.684350 2 -0.010710 4.200960
  0.335490 5.435990 1 -0.147650 5.435990 1 -0.010710 9.482560 1 -0.010710 9.482560 1
  0.282410 0.980730 2 -0.208140 5.435990 1 0.546970 1.443610 2 0.231950 2.600510 2 0.010250 6.510030
```

**2.6 The Fourier file xd.fou**

Type: binary, sequential
A binary Fourier file is created by XDLSM, if requested, after the last least-squares cycle. It has as many records as observations were included in the structure factor calculation. Each record contains the following entries:

- \(h\), \(k\), \(l\) reciprocal lattice components of the scattering vector
- \(fobs\) observed structure factor with anomalous dispersion removed
- \(sig\) error of \(fobs\)
- \(phase\) phase calculated with the final parameters according to the model the refinement was based on
- \(amod1\) real part of the calculated structure factor \((fmod1)\) based on an input dependent model (MODEL1)
- \(bmod1\) imaginary part of \(fmod1\)
- \(amod2\) real part of \(fmod2\)
- \(bmod2\) imaginary part of \(fmod2\)
2.7 The design matrix xd.der

Type: binary, sequential

\[ nref, \text{ibuf}(4), \text{nv} \]
followed by \( i=1,nref \) records of
\[ h(i), k(i), l(i), (D(i,,j), j=1,nv), wt(i) \]

\( D \) (real*8) is the matrix of derivatives of the structure factors with respect to the parameters refined (the design matrix), \( nv \) and \( nref \) are the number of variables and observations respectively, and \( h(i), k(i), l(i), \) (integer) and \( wt(i) \) (real*8) are the Miller indices and weight of the \( i \)’th observation.

2.8 The normal equation matrix xd.mat

Type: binary, sequential

\[ nv \]
\[ (B(i,j), j=1,i), \delta y(i), i=1,nv) \]

where \( B \) and \( \delta y \) (real*8) are the coefficient matrix and vector of the system of least squares equations and \( nv \) is the number of variables refined:

\[
B(i, j) = \sum_{h,k,l} w(h,k,l) * \frac{d|F(h,k,l)|}{dp_j} * \frac{d|F(h,k,l)|}{dp_j}
\]

\[
\delta y(i) = \sum_{h,k,l} w(h,k,l) * \frac{d|F(h,k,l)|}{dp_j} * (F_o - F_c)
\]

2.9 The variance-covariance matrix xd.cov

Type: binary, sequential
See Table 2-4 (end of this Chapter) for a detailed description.

2.10 Grid and path file format

Type: ASCII, formatted, sequential
See Table 2-5 (end of this Chapter) for a detailed description.

2.11 XD – The Master Control Program

SYNOPSIS

\[ \text{xd (options) (cid (mid))} \]

OPTIONS

\(-v\)
be more verbose (in XD itself, does not influence the output of individual modules)

\(-d\)
dump mode, show which commands would be executed, do not actually start them.
Chapter 2 The XD System Files and Master Control Program

- **e** `<exclude-list>`
  modules *not* to start, although they might be flagged in the master file

- **i** `<include-list>`
  modules to start, whether flagged in the master file or not

- **o** `<only-list>`
  exclude all modules not mentioned in this list

**DESCRIPTION**

The modules flagged 'active' in the master file (`xd.mas`) are started. (Subject to change with the `-i`, `-e`, `-o` switches). The parameters `cid` and `mid` are passed to each module. If no master file can be found, XDINI will be started.

It is also possible to start each program by its own command:

`xdini cid prgname bnkname`

See Chapter 3 for details about starting XDINI.

`xdlsm cid (mid)`
`xdprop cid (mid)`
`xdfour cid (mid)`
`xdgeom cid`
`xdfit cid`
`xdgraph options`
`topxd <output file>`

See Chapters 9 & 10 for details about starting programs XDGRAPH and TOPXD respectively.
Table 2-1: The content\(^5\) of the parameter file. The order of the parameters \((U_{ij}, U_{ijk}, U_{ijkl}, P_m, extcn)\) corresponds to the list given in Table 4-2

<table>
<thead>
<tr>
<th>Record</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>xdparfile version [1/2]</code></td>
<td>version of the parameter file (1 is the older, 2 the most recent one)</td>
</tr>
<tr>
<td>2</td>
<td><code>cid</code></td>
<td>Compound identification</td>
</tr>
<tr>
<td>3</td>
<td><code>nat, ntx, lmx, nzz, nto, nsc, ntb, nov</code></td>
<td>software limits for some parameters in the current program version</td>
</tr>
<tr>
<td>4</td>
<td><code>(k_u(i), i = 1, 14)</code></td>
<td>Dimensions of certain arrays in XDLSM, see table 4.1 in Chapter 4 for their meanings.</td>
</tr>
<tr>
<td></td>
<td>1: number of atoms ((n_a)), 2: number of displacement tensor components ((n_tmx)), 3: maximum level of multipole expansion ((npolmax)), 4: number of kappa sets ((n_z)), 5: not used ((n_t0)), 6: scale factors ((n_q)), 7: extinction model ((n.extc)), 8: number of constraints ((n_con)), 9: number of scattering factor tables ((n_tb)), 10: number of symmetry cards ((n_s)), 11: number of variables ((n_v)), 12: ((n_qq)), 13: number of cycles ((n_c)), 14: number of dummy atoms ((n.ad))</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>r10, r20, r1, r2, r1w, r2w, gof, sig</code></td>
<td>Statistics of the fit</td>
</tr>
<tr>
<td>6</td>
<td><code>a, b, c, d, e, f</code></td>
<td>Parameters of the lsq weight</td>
</tr>
<tr>
<td>7+nad</td>
<td><code>dx(i), dy(i), dz(i)</code></td>
<td>Dummy atom coordinates</td>
</tr>
<tr>
<td>do n=1,na</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td><code>atom, icor1, icor2, nax, nay1, nay2, jtf, itbl, isfz, lmax, isym, ichcon</code></td>
<td>Atom name (\text{character*8}) Code integers for defining the site coordinate system.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coordinates, multiplicity, Isotropic or anisotropic U</td>
</tr>
<tr>
<td>+2</td>
<td><code>x, y, z, amult</code></td>
<td>3. order anharmonic tensor components</td>
</tr>
<tr>
<td>+3,+4</td>
<td>(U_{ij}) for (j_{tf}(n)=1) or (U_{i}) for (j_{tf}(n)\geq 3)</td>
<td>4. order anharmonic tensor components</td>
</tr>
<tr>
<td>+5...+7</td>
<td>(U_{ij}) for (j_{tf}(n)=4)</td>
<td>Multipole populations</td>
</tr>
<tr>
<td>+8...+10</td>
<td>(P_{lm}(j, j=1, n_{px})) (n_{px}=l_{max}^2\times l_{max}+2\times l_{max}+2)</td>
<td>Scattering factor table to which the (i^\text{th}) kappa set refers, 6 kappa values</td>
</tr>
<tr>
<td>end do</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+1...+nzz</td>
<td>(f2(i), (z(j)=0, l_{max}+2))</td>
<td>Extinction parameters</td>
</tr>
<tr>
<td></td>
<td><code>extcn(i, i = 1, 7)</code></td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td><code>out</code></td>
<td>Overall thermal parameter</td>
</tr>
<tr>
<td>+1</td>
<td><code>sc(i, i = 1, n_{sc})</code></td>
<td>Scale factor</td>
</tr>
</tbody>
</table>

\(^5\)The previous format is still accepted and interpreted by the code
Table 2-2: The content of the reflection file.

<table>
<thead>
<tr>
<th>Record</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cid, fcod, NDAT ndat</td>
<td>Compound ID ( F ) or ( F^2 ) Number of entries for each observation (min.=6, max.=13)</td>
</tr>
<tr>
<td>do n=1,nref</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>h, k, l</td>
<td>Reflection indices</td>
</tr>
<tr>
<td></td>
<td>iscgrp, obs, sigobs, tbar, u1,u2,u3, v1, v2, v3</td>
<td>Scale group number ( F ) or ( F^2 ) as given by \textit{fcod} Sigma(( F )) or sigma(( F^2 )) Path length Direction cosines of a vector defined with respect to the real crystal axes and normal to the plane of diffraction Direction cosines of a vector defined with respect to the real crystal axes lying in the plane of diffraction and perpendicular to the incident beam</td>
</tr>
<tr>
<td>end do</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2-3: The content of an entry in the databank file.

<table>
<thead>
<tr>
<th>Record</th>
<th>Rec.-Id</th>
<th>Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>:ELEM</td>
<td>( z ) ( w ) ( dfpmo ) ( dfppmo ) ( dfpcu ) ( dfppcu ) ( sclt ) ( ira ) ( irc ) ( irn )</td>
</tr>
<tr>
<td>2</td>
<td>SPH</td>
<td>( a1 b1 a2 b2 a3 b3 a4 b4 a5 b5 a6 b6 c )</td>
</tr>
<tr>
<td>3</td>
<td>SZ</td>
<td>( { \text{zet1}(i),i=1,12 } ) ( \text{zet1} )</td>
</tr>
<tr>
<td>4</td>
<td>STO</td>
<td>( { \text{orb}(i),ioc(i),nbf(i),i=1,12 } ) ( \text{orb} ) ( ioc ) ( nbf )</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>( { \text{bc}(i,j),bx(i,j),nr(i,j),i=1,noc } ) ( \text{bc} ) ( bx ) ( nr ) ( noc )</td>
</tr>
</tbody>
</table>
Table 2-4: The content of the variance-covariance file.

<table>
<thead>
<tr>
<th>nv, errwt (real*4)</th>
<th>Number of variables, square of GOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>((A(i,j),j=1,nv),i=1,nv) (real*4)</td>
<td>( A = \text{inv}(B) \times \text{errwt} ), where ( B ) is the least squares matrix</td>
</tr>
<tr>
<td>(iatom(i),i=1,nv)</td>
<td>&gt; 0 the sequence number of the atom to which parameter ( i ) relates</td>
</tr>
<tr>
<td>(itype(i),i=1,nv)</td>
<td>( = 0 ) ( i ) is not an atomic parameter</td>
</tr>
<tr>
<td>(isfz(iatom(i)),i=1,nv) if iatom(i)&gt;0</td>
<td>(&lt; 0 ) point to Kappa set</td>
</tr>
</tbody>
</table>

Table 2-5 The content of grid and path files.

<table>
<thead>
<tr>
<th>Record</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>filetype version</td>
<td>filetype can be any of 2DGRDFILE, 3DGRDFILE or PATHFILE. version gives the version number of the file format (currently 0).</td>
</tr>
<tr>
<td>2</td>
<td>cid property</td>
<td>Compound id and name of the mapped property</td>
</tr>
<tr>
<td>3</td>
<td>title</td>
<td>Title string</td>
</tr>
<tr>
<td>4</td>
<td>nx ny (nz)</td>
<td>Number of grid points</td>
</tr>
<tr>
<td>5</td>
<td>ox oy oz</td>
<td>Origin of the grid in ‘world-coordinates’</td>
</tr>
<tr>
<td>6</td>
<td>xdim ydim zdim</td>
<td>Physical size of the grid (in Å)</td>
</tr>
<tr>
<td>7+1…+no</td>
<td>nc object1 object2</td>
<td>Number of objects</td>
</tr>
<tr>
<td>+1+1+…+n*</td>
<td>nc</td>
<td>Number of connections</td>
</tr>
<tr>
<td>+1+…+n*</td>
<td>object1 object2</td>
<td>List of bonds or other connecting lines to be drawn</td>
</tr>
</tbody>
</table>

The body for grid files

| values | List of grid values, \( x \) varying fastest. |

The body for path files

| +1 | ncurve times |
| ncurve | Number of bond path curves |
| +1+…+npoints | object type x y z | object gives a starting point (usually a CP object). type is BOND. |

Bibliography

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 with default options and settings, the corresponding input-parameter (xd.inp) and data files (xd.hkl) for XDLSM. An output file (xd_ini.out) is also written. 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 (SHELXL), CIF, XTAL and MOLLY files. It also accepts free-format, as well as fixed-format data files, making it possible to communicate with other computer packages. The files created by XDINI need to be checked. The default setting corresponds to a spherical-atom refinement. It is necessary to edit and modify the ATOM table in the master file before switching to multipole refinement. The default definition of the atomic site coordinate systems are based on the connectivity (the two closest neighbours together with the atom considered define the Z,X plane). This is, in most of the cases, not appropriate for site symmetry implementations. The level of the multipole expansion (the default is monopole) as well as the number of kappa sets should be extended. The xd.inp and xd.hkl files usually do not need to be modified.

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
(e.g. xdini test shelx SCM)

cid
is a maximum 8 character long compound identification described before and prgname can be shelx, cif, xtal and molly with the following input-file requirements:

<table>
<thead>
<tr>
<th>prgname</th>
<th>file1</th>
<th>file2</th>
<th>file3</th>
</tr>
</thead>
<tbody>
<tr>
<td>shelx</td>
<td>shelx.ins</td>
<td>-</td>
<td>shelx.hkl</td>
</tr>
<tr>
<td>cif</td>
<td>xd.cif</td>
<td>-</td>
<td>xd.fcf</td>
</tr>
<tr>
<td>xtal</td>
<td>xtal.inp</td>
<td>xtal.stm</td>
<td>xtal.hkl</td>
</tr>
<tr>
<td>molly</td>
<td>molly.inp</td>
<td>molly.par</td>
<td>molly.hkl</td>
</tr>
</tbody>
</table>

The files file1-3 are read, each after the other, 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 or SCM (see section 2.5 for their meaning). If bnkname is not specified, xd.mas will be generated with the default BANK CR.

3.1.2 File directed mode

xdini cid

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, SYM, LATT 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 read from the input file xd_ini.inp unless otherwise is specified. The **FILE** instruction redirects the default input to a file named *filename*.

3.1.4 FORMAT

**FORMAT** *(format specification)*

The data are supposed to be given in default order 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.

3.1.5 Default atomic parameter list

The following entries have to be given for each atom:

\[ \text{atomname \ x \ y \ z \ mult \ uiso or} \]
\[ u11 \ u22 \ u33 \ u12 \ u13 \ u23 \]

The *atomname* (up to eight characters) should 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.

3.1.6 LOADPAR

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

Followed this command line *nat* atom segments are read. The parameters have to be either 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*.

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

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

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

### 3.1.9 Default observation input

The following entries can be given for each observation:

\[ h \ k \ l \ \text{obs} \ \text{sigobs} \ \text{scgrp} \ \text{tbar} \ u_1 \ u_2 \ u_3 \ v_1 \ v_2 \ v_3 \]

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.

### 3.1.10 LOADREF

**LOADREF** \( F \) or \( F^2 \) \( nref \ ndat \)

\( nref \) observation line, containing \( ndat \) entries with \( F \) or \( F^2 \) data, are read either in the order specified above or according to a format statement given previously.

### 3.1.11 SORT

**SORT** \((index1 [h] \ index2 [l] | \text{sinthl})\)

The reflections are sorted either with respect to indices \((index1 \) varies first and \( index2 \) last) or to the absolute value of the scattering vector \((\text{sinthl})\). An in-memory sorting algorithm is implemented which can handle 15000 reflections. If more data are to be sorted, the parameter NO has to be changed in the source. 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.

### 3.1.12 END

**END**

The **END** card closes the \( xd\_ini.inp \) file and terminates the program.
3.2 Examples

Example 1.

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
SCALE 3 85.87513 89.84698 369.09409
LOADPAR 7
O1  0.085335 -0.055242 0.150354
   0.006503 0.009821 0.003786 0.002344 0.001042 0.000470
O2  -0.221518 0.244985 0.036284
   0.005563 0.009277 0.005550 0.002939 0.001704 0.000386
O3  -0.451596 0.634692 0.178431
   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
LOADREF 1500 6
SORT sinthl
END
Chapter 4

XDLSM - Least Squares Program for Multipole Refinement

4.1 Overview

XDLSM is a full-matrix least squares program based on the generalized scattering model detailed in the Introduction. Its present version includes multipole expansion up to $l=4$ and anharmonic treatment of the thermal motion up to 4th order of the Gram-Charlier expansion. XDLSM, being based on the Hansen-Coppens formalism [1], necessarily has many common elements with MOLLY, the algorithm of which has been rebuilt and extended to allow for further developments. XDLSM supports sophisticated density modelling, and features of previous refinement programs have been incorporated (LSEX [2]). Further important aspects of XDLSM provide methods to locate inadequacies in the model, to control the refinement and to monitor the results.

4.1.1 The method of least squares

In this chapter some aspects of the method of least squares are discussed, whose knowledge are necessary for the user to handle the input and output of XDLSM. This introduction is based on reference [3], to which the reader is referred for more details.

Consider a given set of $m$ observations $y_o\{y_{o1}, y_{o2}, y_{o3}, \ldots y_{om}\}$ represented by the corresponding set of model functions $y_c\{y_{c1}, y_{c2}, y_{c3}, \ldots y_{cm}\} = y_c(x)$, where $x$ is the $n$-component vector of the parameters $x\{x_1, x_2, x_3, \ldots x_n\}$. The best unbiased estimates of $x$ can be obtained by minimizing the square of the residual:

$$R^2 = (y_o - y_c)^TW(y_o - y_c) = (y_o - y_c)^TQ^TQ(y_o - y_c)$$  \hspace{1cm} (Eq. 4.1)

where $W$, the weight matrix, is to be chosen as the inverse of the variance-covariance matrix of the observations (in practice, it is taken to be diagonal), and $Q$ is an upper triangular matrix, i.e. $Q^TQ$ is the Cholesky decomposition of $W$. If $y_c$ can be expanded about $x_o$ in Taylor’s series retaining only the first order terms, then

$$y_c = y_c(x_o) + D(x - x_o)$$  \hspace{1cm} (Eq. 4.2)

with $D_i = dy_i/d_y$ being the design matrix. Eq. (4.1) becomes:

$$R^2 = [\Delta y - Z\Delta x]^T[\Delta y - Z\Delta x]$$  \hspace{1cm} (Eq. 4.3)

where $\Delta y = Q(y_c - y_c(x_o))$, $\Delta x = x - x_o$ and $Z = QD$.

The $n$ conditions

$$\left(\frac{dR^2}{dx_i}\right)_{x = x_o} = 0 \text{ for } i = 1,2,3\ldots n$$  \hspace{1cm} (Eq. 4.4)

lead to the system of normal equations

$$Z^TZ\Delta x = Z^T\Delta y$$  \hspace{1cm} (Eq. 4.5)

whose solution vector is
\[ x = x_o + B^{-1}Z'\Delta y \quad (\text{Eq. 4.6}) \]

with \( B = Z'Z \).

An alternative solution of the least-squares problem is provided through the singular value decomposition of the standardized design matrix \( Z \). Let

\[ Z = UGV' \quad (\text{Eq. 4.7}) \]

where \( U \) is an \( m \times n \) column orthogonal matrix, \( G \) is a diagonal matrix of the singular values and \( V \) is an \( n \times n \) orthogonal matrix.

A solution of the over-determined system of equations

\[ Z\Delta x = \Delta y \quad (\text{Eq. 4.8}) \]

can be given as

\[ \Delta x = Z^{-1}\Delta y \quad (\text{Eq. 4.9}) \]

where

\[ Z^{-1} = VG^{-1}U' \quad (\text{Eq. 4.10}) \]

This solution can be proved to be the best possible solution in the least-squares sense as \( \Delta x \) is the vector which minimizes the residual:

\[ R = |Z\Delta x - \Delta y| \quad (\text{Eq. 4.11}) \]

The solution of the least-squares problem through the system of normal equations (4.6) has the disadvantage that it fails if \( B \) is singular or ill-conditioned. A clear distinction should be made between ill-conditioning of an analytical and that of a numerical nature. The former case is likely to occur for an over-parametrized model, when some combination of basis functions are irrelevant to the fit. The normal equation matrix has zero or nearly zero eigenvalues and the inversion gives no or only a formal solution. This problem manifests itself in undesirable correlations among the variables. The method used for establishing hidden indeterminacies in the model is the singular value decomposition of the matrix of observation-equations (4.8). This procedure gives a diagnosis of the degeneracies and provides a solution minimizing the residual. The matrix can be considered ill-conditioned if its inverse condition number, the ratio of the smallest to the largest eigenvalue, is comparable with the machine precision. The components of the eigenvector (a row or column vector of \( V \)) corresponding to the smallest eigenvalue define a linear dependence among the variables (orthonormal basis for the null-space) which leads to the singularity. Zeroing an eigenvalue in the calculation of the inverse matrix (4.10) means introducing the constraint given by the corresponding eigenvector. The term numerical ill-conditioning refers here to an unbalanced least-squares matrix which is due to the fact that the model function is simply not equally sensitive to the changes of the different parameters, i.e. the components of the design matrix can differ by many orders of magnitude. A condition number of the order of 10 is typical for the multipole-model based structure factor least-squares matrix. This number indicates that a small change (error) in an element of the design matrix \( Z \) can cause large changes in the elements of \( B^{-1} \). That is why the solution via the inversion of the normal equation matrix is susceptible, to a considerable extent, to roundoff errors and requires double precision arithmetic. This problem can be overcome if the elements of the \( Z \) (or \( B \)) matrix are brought to a common scale. In XDLSM the normal equation matrix is analyzed and its conditioning is accomplished by a similarity transformation:

\[ B_c = \text{diag}(B)^{-1/2} B \text{diag}(B)^{-1/2} \quad (\text{Eq. 4.12}) \]
Inversion, based on the Gauss-Jordan elimination method [4], is the default option to solve eq. (4.5). If the matrix inversion fails or if diagonalization has been selected as the method of solution, the eigenvalues are calculated by the Jacobi method [4], and the singularities are reported and eliminated. The eigenvalue filtering is based on the inverse condition number. The lowest eigenvalues are rejected (zeroed) from the inverse calculation until the inverse condition number reaches a user specified limit. While this procedure gives a mathematically correct solution, its indiscriminate application does not necessarily reveal the physical meanings of the indeterminacies that made the least-squares equations singular or nearly singular in the first place.

4.1.2 Model ambiguities

The XD package will be available for a wide scientific community. This Section tries to help those who have not yet been involved in charge density research. In particular, it tries to help those users who have not yet had the uncomfortable feeling of getting stuck at a certain stage of the refinement. This happens when decisions need to be made as to which parametrization is preferable among several alternative ones which perform equally well in fitting the data.

The scattering model described in the Introduction formally allows 66 parameters per atom (in the present implementation of XD) to be included in the refinement. However, any interpretation of the data set using an “all-parameter” fit is hardly feasible, nor is it appropriate. Even if one could afford it (i.e. even if enough data points were available) and even if convergence was reached with a satisfactory fit, the physical significance of the results would certainly be doubtful. While the total dynamic ED obtained could account for the data very well, any property which is a function of a subset of the variables could well be meaningless. As mentioned above, the reason for this is that many basis functions of the structure factor expansion have a similar dependence on the components of the scattering vector. Consequently the data cannot differentiate between them. A typical example of this type of bias is that introduced into the static density deformations by the inadequate decomposition of the thermal smearing. This is caused by the formal similarity between density basis functions and pdf’s of the nuclear displacements. Strong correlations, as high as 80-90%, are likely to occur between quadrupole populations and second order displacement parameters. The Gram-Charlier model has been shown to be as adequate as the multipole expansion in accounting for static density asphericities [5]. Such indeterminacies can appear especially pronounced for non-centrosymmetric structures.

The flexibility of the model and the limited number of observations forces one to limit the optimization to a subset of parameters or to their combinations. The variables are usually selected on the basis of simple chemical arguments or preconceptions. The outcomes must be tested in order to judge their physical significances. A careful study should not neglect an independent analysis of static and dynamic parameters.

4.1.2.1 Testing the results

The most important test to judge the success of the model and the quality of the fit is to evaluate the residual ED through a Fourier summation \( F_{\text{obs}} - F_{\text{model}} \). This provides a direct-space representation of the extent to which the model accounts for the observations. A featureless residual map is a necessary condition for the adequacy of a model, but is far from being a sufficient one for judging its physical significance. Another usual procedure is to compare the static deformation density obtained from X-ray data with that calculated theoretically. Deformation peak-shapes and peak-heights are subject to specific conditions that are characteristic for the different methods to be compared. The ab-initio ED depends on the level of the theory applied and on the quality of the basis sets. Both factors place severe limitations on any direct comparisons, especially for larger systems. However, without such comparisons, the interpretation of the results in terms of the deformation ED remains
only of a qualitative nature. This is because of the arbitrariness in selecting the reference state and the sensitivity of the ED to the structural parameters.

We suggest that the experimental ED is tested through its local and global topological characteristics and by evaluating its integrated properties. XDPROP makes it feasible to trace the refinement process almost "continuously" by inspecting the different stationary points of the total ED and related scalar properties. In this respect the Laplacian of the ED, as a sensitive measure of charge concentrations, should play an important role. A static ED which fails to reproduce the characteristic topological features of a typical covalent bond, e.g. (3,-1) CP’s, bond charge concentrations shown by the Laplacian, is likely to be suspect.

One-electron properties are directly obtainable from the ED and their comparison with the outcomes of independent measurements and/or theoretical results are of great importance. The molecular dipole moment and the electrostatic potential are the quantities most frequently evaluated from the experimental ED. Such applications are being explored with a promising success.

One way to gain information on the physical significance of the thermal parameters is to test them against the rigid-body motion model [6] which is based on the observation that in molecular crystals the external (lattice) vibrations make the major contribution to the atomic motion. Satisfactory agreement between observed and calculated anisotropic displacement parameters may suggest that the molecule is rigid to a good approximation or the thermal parameters are uniformly affected by systematic errors. Significant residuals after the rigid-body fit may indicate either the importance of soft internal modes or simply a bias in the atomic displacements. A directly applicable test for the correctness of the atomic displacement parameters is the rigid-bond test [7].

If \( z_{A,B}^2 \) denotes the mean square displacement amplitude of atom A in the direction of atom B, then for every covalently bonded pair of atoms A and B

\[
\Delta_{A,B} = z_{A,B}^2 - z_{B,A}^2 = 0
\]

Conversely, if in parts of the molecule this rigid bond postulate is not fulfilled, one may deduce that the structural model is insufficient. Hirshfeld estimated that for atoms at least as heavy as carbon \( \Delta_{A,B} \) should normally be smaller than 0.001 Å\(^2\). Verification of the model and the anisotropic displacement parameters by this test strengthens confidence in the experimentally determined ED.

A very useful visualization of the atomic displacement parameters is provided by the computer-graphics program PEANUT [8], developed recently to analyze observed (fitted to diffraction data), calculated (as given by a model) or residual (observed- calculated) thermal parameters in terms of closed surfaces defined by the root-mean-squares displacements \( \langle u(n) \rangle^{1/2} = \langle n' U n \rangle \), where \( n \) is a unit vector in any direction). Applications are given in reference [9].

A plausible approach to reduce ambiguities in the model is to introduce constraints into the refinement. It is desirable to replace external checks on one of the possible, mathematically equivalent solutions by internal constraints applicable to support the physically most relevant solution. An advanced feature of XDLSM is to allow for general linear restrictions on any set of variables. Efforts are being made to further develop this option in order to incorporate more 'physics' into the refinement model.

### 4.1.2.2 Constraints in XDLSM

The treatment of constraints in XDLSM is based on the technique of direct elimination. Consider a system of \( nc \) linear equations, each of which defines a constraint among \( nv \) variables:
By decomposing the matrix $C$,

$$
C = \text{PSR}' = P(nc,nv)\begin{pmatrix}
S(nr,nr) & 0 & R'_1(nr,nv) \\
0 & 0 & R'_2(nr-nr,nv)
\end{pmatrix}
$$

(Eq. 4.14)

with $S$ being a diagonal matrix of $nr$ non-zero singular values ($nr \leq nc$), two sets of new variables can be introduced:

$$
\Delta \vec{x}_1(nr) = R'_1 \Delta \vec{x} \quad \Delta \vec{x}_2(nr-nr) = R'_2 \Delta \vec{x}
$$

(Eq. 4.15)

where the first set can be eliminated by means of eq. (4.13) and (4.14):

$$
\Delta \vec{x}_1 = S^{-1}P' a
$$

(Eq. 4.16)

This leads to a decomposition of the unconstrained variables

$$
\Delta \vec{x} = \text{R} \Delta \vec{x} = R'_1 \Delta \vec{x}_1 + R'_2 \Delta \vec{x}_2 = R'_1 S^{-1} P' a + R'_2 \Delta \vec{x}_2
$$

(Eq. 4.17)

The equations of observations 4.2 becomes

$$
\Delta y - DR'_1 S^{-1} P' a = DR'_2 \Delta \vec{x}_2
$$

(Eq. 4.18)

and the system of normal equations is reduced to

$$
B'_2 \Delta \vec{x}_2 = R'_2 Z' \Delta y_2
$$

(Eq. 4.19)

where

$$
B'_2 = R'_1 Z' Z R'_2 \text{ and } \Delta y_2 = \Delta y - DR'_1 S^{-1} P' a
$$

(Eq. 4.20)

The elimination through the singular-value decomposition of the constraints matrix has two advantages:

1. the dimension of problem is reduced by the number of independent constraints
2. the restrictions can be formulated in an automatic way as all accidental redundancies are easily filtered out.

Some of the constraints mentioned below have already been implemented in a user-friendly way, others will be available in subsequent releases of XDLSM.

### 4.1.2.3 Restrictions on the multipole populations

**Electro-neutrality constraint.** The sum of the monopole populations, by definition, gives the number of valence electrons in the molecule (unit cell). This statement is part of the multipole expansion formalism which involves "atomic" partitioning and thus provides a particular assignment of the atomic charge to the corresponding monopole population. The electro-neutrality constraint keeps the unit cell neutral. In XDLSM it is possible to define any subset of atoms (i.e. any functional group) for which the total number of valence electrons is kept constant. This option then precludes any charge transfer between the group(s) selected and the rest of the atoms in the unit cell.

**Local pseudo symmetry, "chemical" symmetry.** Preconceptions based on chemical intuition can also be applied to reduce the number of multipole populations to be refined. One can assume a simple hybridization scheme which corresponds to the actual geometrical arrangement of the atoms. This is usually achieved by imposing site symmetry in a properly chosen local Cartesian frame and using symmetry adapted angular functions. The symmetry restrictions for real spherical harmonics are given in Table 4-4. Another feasible restraint is to keep the valence density of chemically equivalent or similar atoms to be the same during the refinement. This is a widely accepted practical approach in studies on larger molecules. The real question is how to judge the actual applicability and success of
our chemical expectations implemented in such a way. Static equivalences might be hampered in an unconstrained refinement by dynamic non-equivalences of the atoms considered. Another important aspect is that in crystals, the 'chemical symmetries' characteristic of the isolated molecules may not be preserved. Any subsequent enforcement of static equivalencies may result in the effects of the crystal field becoming unobservable.

4.1.2.4 Restrictions on the radial functions

The shape of $\mathbf{R}_l$'s are controlled by $n(l)$ and $a_l$ (see eq. 1.19), the latter being estimated from the Hartree-Fock-optimized single-$\xi$ values. In case of quadrupolar atoms (which have only ss, sp, and pp type orbital products) the selection of $a_l$ for $l > 2$ is not straightforward. The corresponding "virtual" density basis functions are shown to account for bond densities [10]. The usual practice is to keep $a_l = a$ for all $l$ and optimize $\kappa$' scaling of $a$. Even under this severe restriction $\kappa$ becomes highly correlated with the populations and convergence can be troublesome. In this respect, $\kappa$ is by far the most critical parameter of the formalism. This may indicate that the constraint implemented is not adequate. Model studies on di-atomic molecules showed that a satisfactory fit of the HF ED with one-center multipole densities requires, in certain cases, highly structured radial functions while in other cases, depending on the level of expansion, simple Slater functions are sufficient [11]. The extent to which this statement applies to many-atom molecules remains to be examined. A trivial choice for improving the situation is the use of radial functions corresponding to extended basis HF atomic orbitals. In studies on transition metal complexes, the HF radial scattering factors were shown to be superior to those of single Slater functions [12].

4.1.2.5 Restrictions on the vibrational parameters

Rigid-body or segmented-rigid-body models could be incorporated into the structure factor refinement. Both approaches require a linear transformation of the design matrix leading to a reduction in the number of dynamic variables. Severe indeterminacies, depending on the formalism, can be introduced.

A more elegant alternative procedure applied in XDLSM is to define rigid molecules or segments by invoking rigid-bond and rigid-link constraints. This is a very efficient way to define the degree of flexibility, but a full control requires a detailed knowledge of the intramolecular motion. Normal coordinate analysis, if a suitable force-field is available, provides the MSDA matrix associated with any normal mode. For molecules of first row elements, standard force fields are readily available and procedures are in general use to refine them against spectroscopic data. Frequencies at the HF level are typically 10% larger than those of measured, and even semi-empirical methods can provide fair estimations. This suggests that incorporation of calculated ADP's due to intramolecular motion into the refinement is feasible. See Chapter 11.1 concerning the program XDVIB1.

An easy to handle approach is to apply constraints of the rigid-bond (rigid-link) type to the shift of the ADP's calculated from an intramolecular force field. Such shifts give only rigid-body type contributions to the ADP's and the procedure preserves atomic displacements due to intramolecular vibrations. The success of such applications depends on the extent to which the mean-field approximation is valid. Another difficulty is that the optimized molecular geometry needed to calculate the harmonic force field can considerably differ from that found in the crystal. Another approach is to start from a set of ADP's predicted by the TLS model. These ADP's satisfy the Hirshfeld condition for all internuclear separations. By invoking the rigid-body constraint to all covalent bonds between atoms of comparable mass the bias in the ADP's can be reduced significantly.
4.2 Refinement strategy

A general rule, it is strongly advised that the complexity of the model should be increased in a stepwise manner. Each stage of the refinement could provide a hypothesis for the next step. In this respect it is difficult to suggest a specific scheme, in advance, according to which one should proceed. The spherical-atom refinement could serve as a reference for comparison during the whole fitting procedure. This could be followed by a restricted multipole refinement in which all possible chemical constraints and atomic pseudo-symmetries are applied. As argued above, the extent to which these restrictions should be applied depends on many factors. In most cases the number of observed intensity data limits the number of free variables. The ratio of the number of reflections to the number of variables should not fall considerably below 10. Atoms with the same valence and first coordination sphere should always be considered chemically equivalent at this stage of the fit. The spherical HF radial screening parameters ($\kappa$) can already be included. These variables, in contrast to those scaling the Slater exponents ($\kappa'$), are much more stable and their changes should stay below 5-10%. If the resolution and accuracy of the observations allows, the different restrictions can be released in subsequent refinement cycles, in the hope of testing the adequacy of the assumed chemical equivalences. In this way, 'second order effects' (crystal field, conformation differences, second neighbours, etc.) on the valence density might become visible. To decide if a new variable contributes significantly to the fit, the ratio of its value to its standard deviation and the change in the goodness of fit are to be checked. More sophisticated statistical tests will be available in follow-up versions of XDLSM.

Because of their low scattering power and intense thermal motion, hydrogen atoms should be treated with a special care. A poor model for their static density manifests itself in unreliable dynamic parameters and conversely, no reasonable estimate of the charge transfer can be obtained without meaningful displacement parameters. In organic molecules a considerable amount of the charge transfer occurs at the expense of charge on the hydrogen atoms. Due to the electro-neutrality constraint these uncertainties can seriously affect the result. To overcome this difficulty, the following strategies can be applied. The position and thermal parameters of the hydrogen atoms should be fixed at the values obtained by neutron diffraction, when such data are available. An overall scaling of the neutron displacement tensor components should be applied to account for the temperature difference (or rather the difference in the diffuse scattering) between the two data collections [26]. In the absence of neutron data, the parameters of the hydrogen atoms could be obtained from spherical-atom refinement using the contracted scattering factors of Stewart et al. [13]. The isotropic displacement parameters can then be fixed during the multipole refinement. The correctness of this estimation can be judged by the distance of the bonds to the corresponding hydrogen atoms and by their net charges obtained in such a way. The ADP's of the hydrogen atoms can also be estimated by fitting the rigid-body or segmented rigid-body model to the motion of the non-hydrogen atoms. A simple riding model could also be feasible ($U(H) = 1.5 \times U_{eq}(\text{non-H})$). Such a constraint can easily be incorporated. The density asphericities of the hydrogen atoms can be represented by a bond-directed dipole. For those involved in a strong hydrogen bond an additional quadrupole can also be introduced. The RESET BOND command (Section 4.6.5) is very useful here to constrain X-H distances to neutron determined standard values.

4.3 Dimensioning

The parameters in Table 4-1 are used in certain DIMENSION and COMMON statements. They can be modified according to the user’s needs.

4.4 Variable names and order numbers

See Table 4-2 for a list of symbols and code numbers to be used as variable identifications.
4.5 Files used and created by XDLSM

Input: xd.mas, xd.inp, xd.hkl, xd.bnk_*
Output: xd.lsm.out, xd.res
Optional output: xd.fou, xd.der, xd.mat, xd.cov

4.6 Input instructions for XDLSM

The next section describes those commands which are interpreted by the program. All of these instructions must be placed between the MODULE *XDLSM and the END XDLSM lines in the xd.mas file.

4.6.1 Control instructions

4.6.1.1 SELECT

SELECT (*model m1 m2 m3 m4 based_on (f|f^2) (*)test
SELECT cycle cycles dampk dampk cmin cmas cmax eigcut r

(*)model m1 m2 m3 m4 This option provides a global control over certain parameters which characterize the structure factor formalism applied in the refinement. These parameters are shown in Table 4-3

The values given after the model option are applied for all atoms only as an upper limit. The option has only limited applications but can provide an easy way to reduce the complexity of the scattering formalism without having to modify all necessary parameters one by one. Note that certain combinations of the control parameters are meaningless, which might not be recognized by the program.

based_on (f|f^2) The refinement is based on structure factors or on their squares. The data in the reflection file xd.hkl are transformed accordingly.

(*)test If flagged an input test is performed. This includes calculation and printing of
1. the scattering factor tables,
2. the local coordinate systems,
3. the variable-parameter list,
4. the matrix of constraints, together with the result of its singular value decomposition
5. a file xd_scat_atom.out is printed for each atom type read in scat table in order to check two different calculations of the scattering factors (from the analytical expansion and from the wave function databank selected)

cycle cycles [0]
>0 The number of least squares cycles requested.
=0 Structure factor calculation.
<0 Scale factor refinement.

dampk dampk [1.0]. This is a damping parameter applied to refinement of kappa’ s.

cmin cmin [0.6] cmax cmax [1.] Lower and upper limit used as a criteria for printing the correlation matrix elements.

eigcut r [1.e-10] If the solution of the system of normal equations are obtained through diagonalization, r is used as a cutoff limit for the singularity test. Eigenvalues are
considered to be zero and omitted from the calculation of the inverse matrix until the inverse
condition number is smaller than $r$.

$$\min(eigenvalue) / \max(eigenvalue) < r$$

This test is applied to the eigenvalues of the reduced matrix (derived from the constraints)
and the conditioned matrix (see Introduction). The same parameter is used as a criteria for
eliminating linear dependencies among the constraints. (Singular value decomposition of the
matrix of the constraints.)

4.6.1.2 SAVE

SAVE (*)deriv (*)lsqmat (*)cormat

deriv  If starred, the structure factor derivatives for each reflection (design matrix) in the
last cycle are saved in the binary file $xd.der$ (see Section 2.7).

lsqmat  If starred, the least squares matrix and vector in the last cycle are printed to the
binary file $xd.mat$ (see Section 2.8).

cormat  If starred, the variance-covariance matrix is written to the binary file $xd.cov$.
This file is needed for estimating the standard deviations of different properties. The
structure and the content of the file is given in Table 2-4.

4.6.1.3 SOLVE

SOLVE [*]inv [*]diag [*]cond

The solution of the least squares normal equation can be obtained through inversion or
diagonalization.

inv  If starred, the Gauss-Jordan for inversion method is implemented. The program will
automatically switch to diagonalization if the matrix is found ill-conditioned (or singular)
during the inversion in the first cycle. The matrix inversion is the default option.

diag  If starred, the Jacobi algorithm for the calculation of the eigenvalues and eigenvectors
is used. This is considerably more time-consuming than other 'modern' diagonalization
methods but it is steady and works reliably even with an unconditioned least square
matrix. If an eigenvalue fails the test based on the condition number (see eigcut), the corresponding
eigenvector is printed.

cond  If starred, the normal equation matrix is conditioned via the transformation 4.12,
irrespective of the method of solution selected.

4.6.1.4 SKIP

SKIP (*)obs obsmin obsmax [*]sigobs sigmin sigmax [*]sinthl snlmin snlmax

The SKIP instruction defines criteria for rejecting observations from the refinement (not from
the structure factor calculation). To make a criterion active the corresponding option should
be starred, but note that the default rejection criteria will always be applied if these items
are not selected. If more than one are starred, the "AND" logic is applied. The available
options are:

obs  [0.0, 1.0e10] observations for which $obsmax > obs > obsmin$ will be used in the
refinement

sigobs  [3.0, 1.0e10] observations for which $sigmax*obs > obs > sigmin*obs$ will be included

sinthl  [0.0, 2.0] lower and upper limit in $\sin\theta/\lambda$.  

34
4.6.1.5 PRINT

PRINT (*)sinthl snlmin snlmax (*)obs obsmin obsmax (*)delta dmin dmax [*]del% min% max% (*)extcn extmin extmax (*)abssc

The PRINT instruction defines criterion for printing observations. After the last cycle the following quantities can be printed:

\[ \text{no h k l sinthl scgrp obs calc delta (del% flag) extcn code} \]

where

- \text{no} \quad \text{the order number of a reflection}
- \text{h k l} \quad \text{reciprocal-lattice components of the scattering vector}
- \text{sinthl} \quad \text{sin(θ)/λ}
- \text{scgrp} \quad \text{scale group number}
- \text{obs} \quad \text{F}_o \text{ or } \text{F}^2_o
- \text{calc} \quad \text{F}_c \text{ or } \text{F}^2_c
- \text{delta} \quad \text{F}_o - \text{F}_c \text{ or } \text{F}^2_o - \text{F}^2_c
- \text{flag} \quad \text{a flag based on } f = |100 \times (\text{obs} - \text{calc}) / \text{obs}| \text{ It is a four character long string as follows:}
  - ‘.’ for \(0 < f < 5\)
  - ‘*’ for \(5 < f < 10\)
  - ‘**’ for \(10 < f < 15\)
  - ‘***’ for \(15 < f < 20\)
  - ‘****’ for \(20 < f < 25\)
  - ‘????’ for \(25 < f < 30\)
- \text{del%} \quad \text{f is printed instead of a flag}
- \text{extcn} \quad \text{the extinction correction in percentage}
- \text{code} \quad 0 \quad \text{included in the refinement}
  -1 \quad \text{rejected based on criterion obs}
  -2 \quad \text{rejected based on criterion sigobs}
  -4 \quad \text{rejected based on criterion sinthl}
  -3 \quad \text{rejected based on criteria obs and sigobs}
  -5 \quad \text{rejected based on criteria obs and sinthl}
  -6 \quad \text{rejected based on criteria sigobs and sinthl}
  -7 \quad \text{rejected based on criteria obs and sigobs and sinthl}

The options, if flagged, serve as a lower and an upper limit applied for printing. Again, the ‘AND’ logic applies.

\[ (*)\text{sinthl snlmin snlmax [0 2]} \]
\[ (*)\text{obs obsmin obsmax [0 10]} \]
\[ (*)\text{delta dmin dmax [-50 50]} \]
\[ (*)\text{del% min% max% [80 100]} \]
\[ (*)\text{extcn extmin extmax [80 100]} \]
\[ (*)\text{abssc} \quad \text{if flagged the observations are printed on an absolute scale} \]

4.6.2 The SCAT table

The SCAT table provides a compact format for defining different scattering factors or modifying the entries in the databank file xd.bnk_*. In contrast with previous versions of the program, the SCAT table now includes all atomic orbitals. If an old xd.mas file is used, this table must be modified otherwise it will not be read correctly. The heading of the SCAT table is:
Chapter 4 XDLSM Least Squares Program for Multipole Refinement

SCAT core sphv defv 1s 2s 3s 4s 2p 3p 4p 3d 4d 4f 5s 5p 6s 6p 5d 7s 6d 5f Δf ′ Δf ′ ′ nsctl

| core   | core scattering factor |
| sphv   | spherical valence scattering factor |
| defv   | scattering factors due to valence deformation functions |
| 1s 2s 3s ... | occupations of HF atomic orbitals |
| Δf ′  | real part of anomalous dispersion correction |
| Δf ′ ′ | imaginary part of anomalous dispersion correction |
| nsctl  | neutron scattering length |

This SCAT line has to be followed by as many input lines or subsegments as atom types are present in the unit cell. Each row should begin with the element name that must be identical to one of the atom types stored in xd.bnk_* (see element naming convention in section 2.5). If the element name is the only string in the input line, the data on the corresponding segment of the databank file will be used to create the scattering factors. The databank file can be extended by introducing new segments assigned to dummy atom names. In this way considerable freedom is provided for designing scattering factors from atomic wavefunctions expanded over Slater-type basis functions. All the entries indicated above have default assignments. To change the default assignment of a particular entry all preceding entries in the list have to be given. For example, to change the default values for the anomalous dispersion corrections (Δf ′ and Δf ′ ′) all three types of scattering factors as well as the occupations have to be input.

For the scattering factors the following options are available:

| core   | chfw | rdtb   |
| sphv   | chfw | rdtb   | rhft   |
| defv   | chfw | rdtb   | cszd   | rdsc   |

4.6.2.1 chfw - Clementi’s Hartree-Fock Wavefunction

This is the default option for the core and sphv scattering factors and it means that the Slater-type atomic orbitals stored on the xd.bnk_* file are used. The user has the freedom to decide what to consider core and what valence density. This is done by specifying the orbital occupations, which have to be positive or negative integers for core or valence orbitals respectively. If they are omitted, the default configuration in xd.bnk_* is taken. The order of the orbitals is given in the heading of the SCAT table. For example, the default configuration of the ground state carbon atom is (1s^2), (2s^2,2p^2) and the corresponding line in the input table is:

C chfw chfw cszd 2 -2 0 0 -2

In this case 2\langle j_0 | (1s1s) + (2j_0 | (2s2s) + 2(j_0 | (2p2p))/4 is calculated, respectively, for the core and the spherical valence scattering factors. Note, that the sphv scattering factor is normalized, but not the core.

A ‘frozen’ spherical atom (only core or spherical atom scattering) could be defined as

C chfw chfw cszd 2 2 0 0 2

while that of with radial screening (only valence or spherical atom scattering)

C chfw chfw cszd -2 -2 0 0 -2

Another application of the orbital occupations is to form spherical valence scattering factors corresponding to an assumed hybridization. For example, one can ‘generate’ an sp3type carbon atom with the following input

36
which assigns \((\langle j_0 \rangle (2s^2s) + 3\langle j_0 \rangle (2p^2p))/4\) to the spherical valence scattering factors. Since more than one sets of scattering factors can be generated from the same wavefunction, the multiple use of an element name is allowed.

### 4.6.2.2 rdtb - Read table

This option is available for all three types of scattering factors. It indicates that the corresponding scattering factor table is to be read from the master file. For an unknown element (not stored in the xd.bnk_* databank file) the rdtb option must be specified. The input should consist of \(n_{grd}\) values (8 entries/lines) of the function taken at an equidistant grid of \(\sin \theta/\lambda\) with a step size of \(grd\). \(n_{grd}\) and \(grd\) are parameters with default values of 40 and of 0.05, respectively, in the present version of XDLSM (See Table 4.1). The first grid point must be zero. The default setup requires the table to be given up to 1.95 in \(\sin \theta/\lambda\). The parameters \(n_{grd}\) and \(grd\) should be adjusted to the wavelength of the radiation used for the data collection. The scattering factor at an arbitrary scattering angle is interpolated and the derivatives with respect to the expansion-contraction parameters are numerically obtained. Accurate evaluation require a considerably fine grid size (not exceeding 0.06 Å\(^{-1}\)).

Example:

```
C    RDTB RDTB CSZD
2.00000   1.99642   1.98575   1.96816   1.94394   1.91349   1.87726   1.83581
1.78973   1.73965   1.68621   1.63006   1.57183   1.51212   1.45148   1.39046
1.32950   1.26904   1.20944   1.15100   1.09400   0.98506   0.93433
0.88381   0.83628   0.79085   0.74754   0.70632   0.67177   0.63004   0.59488
0.56163   0.53021   0.50055   0.47258   0.44621   0.42137   0.39798   0.37597
1.00000   0.93697   0.77692   0.58120   0.40061   0.25845   0.15714   0.08962
0.04686   0.02103   0.00626   -0.00155  -0.00512  -0.00622  -0.00596  -0.00381
-0.00381  -0.00256  -0.00140  -0.00037  -0.00173  -0.00216
0.00247   0.00269   0.00283   0.00291   0.00294   0.00294   0.00291   0.00285
0.00278   0.00269   0.00260   0.00250   0.00240   0.00230   0.00220   0.00210
```

### 4.6.2.3 cszd, rdsd - Single-zeta density parameters for defv

By default (cszd), the radial functions of the valence deformation density are of single Slater-type. The parameters of the radial functions \((n(l), \xi(l))\) are obtained from the corresponding single-\(\xi\) wavefunctions of Clementi & Roetti [14] stored also in xd.bnk_* files. In previous versions of XD, the option cszd in the SCAT table computed the exponents for the radial deformation functions by simply averaging the valence exponents of the “best” single-\(\xi\) orbitals (Clementi and Raimondi [15]) of the default configuration. The new version of the program now computes \(\xi\)’s by weighting the orbitals by their occupation. For noble gases and closed shell ions, \(\xi\) is computed thus:

- from the (weighted) exponents of the outermost shell of the core for noble gases and anions (e.g. the 2s and 2p orbitals for F, Ne etc.)
- from the exponents of the first empty orbital(s) for closed-shell cations (3s for Na\(^+\), Mg\(^{2+}\) etc.; 3s and 3p for Al\(^{3+}\), Si\(^{4+}\) etc.; 4s for K\(^+\), Ca\(^{2+}\); 3d for Sc\(^{3+}\), Ti\(^{4+}\) etc.).

The closed-shell configurations recognized are those of the noble gases (thus, 2, 10, 18, 36, 54 electrons), and those of some cations of the 4\(^{th}\) and 5\(^{th}\) row (like Cu\(^{+}\), Ga\(^{3+}\), Sb\(^{5+}\)), which may have 28 or 46 electrons (single-\(\xi\) exponents considered are those of 3d and 4d orbitals, respectively). All other configurations missing the valence electrons are not recognized by the program, which then stops.

The orbitals used to compute the average are directly linked to the SCAT table configuration. Thus, if the user modifies the number or the type of valence electrons (at his own risk!) in
the SCAT table, then $Z$ will change. Note that in the previous versions of XD, the SCAT table was intended to modify just the SPHV monopole, evaluated by the multi-exponent HF wave functions of Clementi and Roetti [14].

Warning messages will appear in the output if the configuration chosen is unusual or dangerous and severe stops are applied if the requested orbitals are not stored for a given atom.

The default values can be modified by using the option rdsd which makes it possible to input all $n(l)$ and $\zeta(l)$ in atomic units:

```
C    chfw chfw rdsd
n(0) zeta(0) n(1) zeta(1) n(2) zeta(2) n(3) zeta(3) n(4) zeta(4)
```

### 4.6.2.4 The chfw option for defv

An advanced feature of XDLSM is to allow for the use of HF radial functions for the deformation density. Such application needs each $\langle J \rangle$ to be attributed to a proper combination of orbital products. The Table given in the Introduction (Section 1.9) summarizes the different order of Fourier-Bessel transforms that occur for the different orbital products. An orbital product is given by the names of the comprising orbitals in brackets: (2s2s), (2p2p), (3d3d), etc. If more than one product contributes to $\langle J \rangle$ they should be connected by the plus "+" sign. The character string composed in such a way must contain no embedded blanks: (2s2s)+(2p2p) ... etc. A product or a sum of products contributing to $\langle J \rangle$ has to be specified for each $l$. Not all radial densities can be constructed from a given wavefunction. To make a complete set, all options available for defv can be combined, as explained below. If neither cszd nor rdsd is specified for sphv the program expects additional input lines with one of the following contents:

```
[cszd]
[chfw conf]
[rdsd n(l) zeta(l)]
[rdtb]
```

For each $l$ an option can be selected which determines any further input. If no line is given for certain $l$ values the default (cszd) applies. For chfw the configuration (conf) is to be given in terms of orbital products or their sum. For rdsd the parameters of the radial functions are needed. After rdtb a scattering factor table is to be read as described above.

Example:

```
C    chfw chfw chfw   2 -2  0  0 -2
0 chfw (2s2s)+(2p2p)
1 chfw (2s2p)
2 chfw (2p2p)
3 rdsd 3 3.71
4 rdtb
```

In the above example $\langle J_0 \rangle$ is the sum of the 0-th order transforms of ss and pp type radial functions and irrespective of the normalization it is equivalent to sphv. The $\langle J_1 \rangle$ and $\langle J_2 \rangle$ functions are related to sp and pp type orbital products, respectively. The scattering factor for octupoles is created from single-$\zeta$ radial functions while that for hexadecapoles is read in.
4.6.2.5 rhft - Relativistic Hartree-Fock scattering factors for sphv

If a spherical atom model is selected, the RHF scattering factors, as given in the International Tables [16] or in references [17,18] in the form of an expansion over Gaussian functions, can also be used. The rhft option for hydrogen selects the contracted scattering factors of Stewart, Davidson & Simpson [13].

4.6.2.6 Current Limitations

The calculation of the static electron density and of electronic properties requires the evaluation of the radial functions within an accuracy that can hardly be reached by numerical inverse Fourier transform of the scattering factors. The default choice, the use of Slater-type HF wavefunctions (chfw, cszd or rdsd), means analytical representation of both direct and reciprocal space functions. For a refinement to be consistent with the property calculation, it must be based on the wavefunctions stored in xd.bnk_. The corresponding scattering factors are certainly not the best available ones and can slightly differ from those found in the International Tables. Relativistic effects are important only for heavier elements - this can be seen by comparing the total chfw spherical scattering factors with those based on relativistic numerical wavefunctions. Efforts are being made to eliminate this limitation.

4.6.2.7 Anomalous scattering
delf’ delf" 
The defaults correspond to Mo radiation.

4.6.2.8 Neutron Scattering Length

nsctl
The last entry of a SCAT line is the neutron scattering length.

4.6.3 The ATOM table

For each atom included in the structure factor calculation the following entries are to be given:

ATOM atom0 ax1 atom1 atom2 ax2 r/l tp tbl kap lmx sitesym chemcon

4.6.3.1 Atom name conventions

The atom name is a continuous string of up to 8 characters, starting with a correct, case sensitive chemical symbol (e.g. ‘Na’ and not ‘NA’) used in the SCAT table and followed by further characters enclosed in parentheses (). Legal atom names are:

Cu(3) Ti3+(1a)

4.6.3.2 The local coordinate system

The entries in the first seven columns define the local coordinate systems. atom, atom0, atom1 and atom2 are atom names from the ATOM table list. ax1 and ax2 stand for different axis assignments, each being either x or y or z. The first axis (ax1) is given by the internuclear vector from atom to atom0 (v1). This together with the second vector from atom1 to atom2 (v2) define the (ax1,ax2) plane. The third vector (v3) is taken perpendicular to this plane.
\[ v_1 = (r_0 - r) \quad v_2 = (r_2 - r_1) \quad v_3 = v_1 \times v_2 \]

Finally, an orthonormal vector triplet \((e_{ax1}, e_{ax2}, e_{ax3})\) is formed which can be chosen to be either right (R) or left (L) handed

\[ e_{ax1} = v_1 / |v_1| \quad e_{ax2} = (v_3 \times v_1) / |(v_3 \times v_1)| \quad e_{ax3} = v_3 / |v_3| \]

where \(r, r_0, r_1\) and \(r_2\) are the position vectors of atom, atom0, atom1 and atom2, respectively (MOLLY).

### 4.6.3.3 tp - the Order of the Atomic Displacement Tensor

- **0** no thermal parameter is applied (static scattering model)
- **1** isotropic \(U\)
- **[2]** anisotropic \(U_{ij}\)
- **3** anharmonic 3\(^{rd}\) order Gram-Charlier expansion \(U_{ijk}\)
- **4** anharmonic 4\(^{th}\) order Gram-Charlier expansion \(U_{ijkl}\)

If this value differs from that in the input file the thermal displacement parameters will be converted:

**xd.mas** \(\rightarrow\) **xd.inp**

- **1** \(\rightarrow\) **2** from anisotropic to isotropic
- **2** \(\rightarrow\) **1** from isotropic to anisotropic

### 4.6.3.4 tbl - The Core Scattering Table

Refers to the core scattering table. It is the order number of the corresponding element given in the \textbf{SCAT} table.

### 4.6.3.5 kap - the Kappa Set

Defines the kappa set applied to the valence radial functions. \textit{If a new set is introduced or the previous arrangement is redefined, the corresponding changes must also be made in the parameter file \textit{xd.inp} and/or to the \textit{KEY} table.} If the values in the parameter file are not changed the refinement will start from the default value (1.0) for all kappa sets.

### 4.6.3.6 lmx - the maximal level of multipole expansion

- **0** Monopoles (sphv and defv)
- **1** Dipoles (default for hydrogen atoms)
- **2** Quadrupoles
- **3** Octupoles
- **4** Hexadecapoles (default for non-hydrogen atoms)

### 4.6.3.7 sitesym [no]

The point group number of the atomic site symmetry. It is not used in the present version.

### 4.6.3.8 chemcon

Refers to the atom to which the valence deformation density of the atom considered is constrained. \textit{If the same set of multipole populations are to be shared by two or more atoms, the definition of the local coordinate systems of the corresponding atoms must be consistent.}
4.6.3.9 Dummy Atoms

To enable one to define a local system of arbitrary orientation, dummy atoms can be used. These are to be specified after the atom list but within the atom table by giving a name, composed of the string "DUM" and a number, followed by the three coordinates (free format) in the crystal system.

Example:

```
ATOM  atom0  ax1 atom1  atom2  ax2 r/l tp tbl kap lmx sitesym  chemcon
O(1)  O(2)   X  O(1)   DUM0   Y   R   2  1   1   4  NO
O(2)  O(1)   X  O(1)   DUM1   Y   R   2  1   1   4  NO   O(1)
   .
   .
DUM0  0.  0.  0.
DUM1  -0.4800  0.5335  0.0973
END ATOM
```

4.6.4 GROUPn

GROUPn atom(1)... with n > 1

The GROUP command selects a set of atoms to be considered as a unit for special applications. Such applications available presently are the electroneutrality and rigid-body, rigid-link type constraints see (KEEP instruction). The first group (GROUP1) is, by definition composed of the atoms in the asymmetric unit. An atom is allowed to be part of more than one group. The atom list defining a group can be on more than one input line, but each line must start with the same GROUPn command.

4.6.5 KEEP

The KEEP instruction simplifies the application of certain constraints.

KEEP kappa set(1) ...

For each set defined \( \kappa \), the expansion-contraction parameter of defv, is kept the same for all \( l \). This is a default constraint that is suggested to apply, at least in the initial stages of a refinement.

Important! The fit is always very sensitive to \( \kappa \), even if a single parameter is refined for all \( l \) values. The results of \( \kappa \) refinement should always be critically examined and compared to those obtained with \( \kappa = 1.0 \). For HF radial functions, the chance of obtaining convergence with reliable estimates of different \( \kappa \) parameters is expected to be better than for single Slater orbitals. A separate \( \kappa \) refinement is worth trying for transition metals.

KEEP charge [group1] groupn ...

Each group defined by the GROUP command can be treated as a closed unit for which the total charge is kept fixed during the refinement. The total charge of the group is given by the sum of the starting monopole populations of the comprising atoms. The user is free to define any subset of atoms (even having common elements) which are excluded from charge transfer. Each group fixed in this way adds one new equation to the system of constraints. A zero singular value of the matrix of constraints means inadequate grouping and the redundancy found will be rejected.

KEEP rigid [group1] groupn ...

Each group defined by the GROUP command is kept rigid in the sense that the shifts in the ADP’s of the atoms comprising the group are constrained to satisfy Hirshfeld’s rigidity
postulate. To make such a restriction work, all ADP’s of all atoms in the group must be refined. The equation of constraint is set for all internuclear connections in the group and the linearly dependent equations are eliminated leading to the necessary reduction in the number of restrictions.

```
KEEP DIST atom1 atom2
KEEP ANGLE atom1 atom2 atom3
KEEP TORS atom1 atom2 atom3 atom4
KEEP SHAPE [group1] group
KEEP INTRA [group1] group
RESET BOND  atom H-atom distance (Å)
```

The distance between an H-atom and its bonded atom is reset to the designated (usually neutron determined) distance at the end of each cycle. The H-atom must be listed second, and its coordinates should not be refined.

### 4.6.6 The Weighting Scheme

```
WEIGHT a b c d e f
```

The weighting scheme as implemented in SHELXL for refinement on $F^2$ is used:

$$w_2 = q / \left[ s_2^2 + (ap)^2 + bp + d + e \times \sin(\theta) \right]$$

where

$$s_2 = \sigma(F_o^2)$$

$$p = f \times F_o^2 + (1-f) \times F_c^2$$

$$q = 1.0 \quad \text{if } c = 0$$

or

$$q = \exp[c \times (\sin\theta/\lambda)^2] \quad \text{if } c > 0$$

or

$$q = 1.0 - \exp[c \times (\sin\theta/\lambda)^2] \quad \text{if } c < 0$$

The weight ($w_1$) for refinement on $F$ is calculated as follows:

$$w_1 = \left[ F \times \sqrt{w_2} + a \right]^2$$

where

$$b = (F^2 \times w_2) + \sqrt{w_2}$$

$$a = 0.0 \quad \text{if } b < 0.0$$

$$a = \sqrt{b} \quad \text{if } b > 0.0$$

**Important!** This general weighting scheme has been developed for refinement based on a conventional, spherical atom model and thus may not be adequate for multipole refinement.

There are two special weighting schemes: when $a = -1.0$ and when $a < -1.0$. These two schemes are applied regardless of the input values of the other weighting parameters $b - f$. For a multipole refinement on $F$, it is suggested to set $a < -1.0$, when statistical weights will be applied.

$$w_2 = 1/s_2^2 \quad \text{and } w_1 = 1/s_1^2$$

To apply units weights, set $a = -1.0$. The parameters of the weighting form cannot be refined.

Two goodness of fit parameters are printed in `xd_lsm.out`, one (GOFw) based on the weighting scheme used in refinement, and one (GOF) based on statistical weights $w = 1/s^2$.

### 4.6.7 DMSDA

```
DMSDA rmin rmax
rmin [1.1] rmax [1.8]
```

The difference of the projections of the mean square amplitude tensors of two atoms to the corresponding internuclear vector are calculated if the interatomic distance falls in the range given by rmin and rmax. As discussed in the Introduction, Hirshfeld’s rigid-bond test [7] can
help to reveal model inadequacies and should always be a part of a careful analysis. The positional coordinates and the anisotropic displacement parameters in an orthogonal system are also printed.

4.6.8 Extinction refinement

This instruction line for extinction correction appears in \texttt{xd.mas}:

\texttt{EXTCN (*)iso (*)aniso (*)type\_1 (*)type\_2 (*)type\_3 (*)distr\_g (*)distr\_1 (*)msc\_0 (*)msc\_1}

By default the \texttt{EXTCN} command appears as a comment in \texttt{xd.mas}. To get it activated the exclamation mark (!) has to be removed. The extinction correction is based on the models proposed by Becker and Coppens [19-21], which can be summarized as follows:

4.6.8.1 Isotropic extinction (*iso, default):

\textit{extinction type:}
- \texttt{type 1} (*type\_1, default): mosaic spread, the $g'$ coefficient is refined (variable EXT11);
- \texttt{type 2} (*type\_2): particle size, the $\bullet$ coefficient is refined (variable EXT11);
- \texttt{type 3} (*type\_3, *type\_g is also accepted): generalized type, mosaic spread and particle size ($g'$ and $\bullet$) are simultaneously refined (variables EXT11 and EXT22, respectively);

\textit{mosaic spread distribution} (active only for \texttt{type\_1} and \texttt{type\_3}):
- Gaussian (*distr\_g, default): a Gaussian distribution is assumed;
- Lorentzian (*distr\_l): a Lorentzian distribution is assumed;

4.6.8.2 Anisotropic extinction (*aniso):

\textit{extinction type:}
- \texttt{type 1} (*type\_1, default): mosaic spread, the $Z$ tensor is refined (variables EXT11-EXT23) \((g(D) = (D^T Z D)^{1/2}; D \text{ is a unit vector perpendicular to the diffraction plane})\);
- \texttt{type 2} (*type\_2): particle size, the $W$ tensor is refined (variables EXT11-EXT23) \((\cdot(N) = \lambda (N^T W N)^{1/2}; N \text{ is a unit vector in the diffraction plane, perpendicular to the incident beam})\);
- \texttt{type 3} (*type\_3, *type\_g is also accepted): generalized type (mosaic spread and particle size), $Z$ tensor and $\bullet$ simultaneously refined (variables EXT11-EXT23 for $Z$; variable RHOEX for $\bullet$);

\textit{mosaic spread distribution} (active only for \texttt{type\_1} and \texttt{type\_3}):
- Gaussian (*distr\_g, default): a Gaussian distribution is assumed;
- Lorentzian (*distr\_l): a Lorentzian distribution is assumed;

\textit{mosaic orientation} (active only for \texttt{type\_1} and \texttt{type\_3}):
- Coppens and Hamilton (*msc\_0): the distribution proposed by Coppens and Hamilton [22];
- Thorney and Nelmes (*msc\_1, default): the distribution proposed by Thorney and Nelmes [23]. In this case the $Y$ tensor is refined instead of $Z$ \((g(D) = (D^T Y D)^{-1/2})\).

Given the expression for tensors $W$, $Y$, and $Z$, a switch from isotropic to anisotropic correction requires the following variables in \texttt{xd.inp}:

1. $g' = tZ$
   \begin{align*}
   \text{EXT11} &= \text{EXT22} = \text{EXT33} = (g'_\text{iso})^2 \\
   \text{EXT12} &= \text{EXT13} = \text{EXT23} = 0.0
   \end{align*}
2. $g' = tY$
   \begin{align*}
   \text{EXT11} &= \text{EXT22} = \text{EXT33} = 1/(g'_\text{iso})^2 \\
   \text{EXT12} &= \text{EXT13} = \text{EXT23} = 0.0
   \end{align*}
(3) to \( W \)

\[
\begin{align*}
\text{EXT11} &= \text{EXT22} = \text{EXT33} = 1/(\cdot \text{iso}^2) \\
\text{EXT12} &= \text{EXT13} = \text{EXT23} = 0.0
\end{align*}
\]

When a non-positive definite tensor (\( W, Y \) or \( Z \)) is obtained, the program stops if the automatic resetting of the tensor fails.

In the output file xd_lsm.out, the following parameters are reported:

- for isotropic extinction: mosaic spread, \( \eta (\eta = 1/g^\prime \text{, unit is seconds}) \) and domain size \( r (r = g^\prime \lambda 10^{-4} \text{, unit is centimeters}) \) are given, as derived from the refined \( g^\prime \) and/or •
- for anisotropic extinction: Principal axes of mosaic spread \( \eta(D) \) (if *type_1 or *type_3) or domain size \( r(N) \) (*type_2) distribution are given; the corresponding equivalent\( \eta \) and \( r \) scalars are computed (or the refined \( r \) is printed if *type_3 is applied).

**Important**: For any extinction refinement, the absorption weighted path length \( (t_{\text{bar}}) \) has to be stored in the reflection file xd.hkl. For an anisotropic extinction refinement, an additional six entries for each observation are required:

1-3: the direction cosines of vector \( D \) against the real crystal axes

4-6: the direction cosines of vector \( N \) against the real crystal axes

### 4.6.9 FOUR - Structure factor calculation

**FOUR fmod1 m1.1 m1.2 m1.3 m1.4 fmod2 m2.1 m2.2 m2.3 m2.4**

The **FOUR** command ensures that a Fourier file xd.fou is written after the last cycle. Structure factors based on two models but on the same set of parameters are calculated and saved together with \( F_{\text{obs}} \) and the phases. The latter quantities are corrected for anomalous dispersion as follows.

\[
\begin{align*}
A_{\text{obs}}^0 &= \frac{F_{\text{obs}} A_{\text{calc}}}{F_{\text{calc}}} - (A_{\text{calc}} - A_{\text{calc}}^0) \\
B_{\text{obs}}^0 &= \frac{F_{\text{obs}} B_{\text{calc}}}{F_{\text{calc}}} - (B_{\text{calc}} - B_{\text{calc}}^0) \\
F_{\text{obs}}^0 &= \sqrt{(A_{\text{obs}}^0)^2 + (B_{\text{obs}}^0)^2}
\end{align*}
\]

where \( A \) and \( B \) are the real and imaginary parts of the structure factor \( F \) and a zero superscript \( (0) \) designate terms without anomalous dispersion correction. The standard deviation of \( F_{\text{obs}}^0 \) is estimated as

\[
\sigma(F_{\text{obs}}^0) = \frac{\sigma(F_{\text{obs}})}{\sqrt{(A_{\text{obs}} A_{\text{calc}})^2 + (B_{\text{obs}} B_{\text{calc}})^2}}
\]

The phases are based on the model applied in the refinement (see instruction **MODEL**). Each of the structure factor model (\( f\text{mod1, fmod2} \)) is specified with four integers, in the same way as described above. The combination of \( f\text{obs, fmod1} \) and \( f\text{mod2} \) makes it possible to generate six different Fourier maps (see XDFOUR). By default, the **FOUR** command appears as a comment line in xd.mas. In the example below, a Fourier file is created with two calculated structure factors. The first one based on a multipole model \( (l_{\text{max}}=4) \), the second one on a neutral spherical atom model \( (l_{\text{max}}=-1) \). Both are free of anomalous dispersion and extinction.

**FOUR fmod1 4 2 0 0 fmod2 -1 2 0 0**
4.6.10 CON - General linear constraint

**CON** \( a_1 \text{ var1} \ a_2 \text{ var2} \ a_3 \text{ var3} \ldots = c \)

The **CON** instruction defines a linear combination among a set of variables used as a constraint. A list of the coefficients \((a_1, a_2, a_3, \ldots)\) and variable-symbols \((\text{var1}, \text{var2}, \text{var3}, \ldots)\) are to be given terminated by the equal sign ("="), which is followed by the last entry \(c\) to define the right side of the equation. The coefficients and the variable names are read as numeric and character fields, respectively. A variable name is composed from the corresponding symbol described before (Table 4.2) and from a number (if needed) referring to the atom (kappa set or scale group) to which the variable is assigned. The two components of the name are divided by the slash ("/"). The resulting character string must not contain blanks. The following are correct variable names:

\[ \text{X/1, U12/12, U333/1, M1/2, H4+/11, KS/3, K2/2, SC/1, EX12} \]

The list of coefficients and variable names must be terminated by the equal sign. More than one line can be input with the same **CON** command. A new line is read until the equal sign is found. Note, that here, what is meant by the term "variable" is actually, the shift in that variable and not the variable itself.

**Important!** The program does not check if a constraint is meaningful. Redundancies (linear dependencies between the constraints) are recognized, reported and eliminated.

Applications of practical importance are the constraints due to crystallographic site symmetries. In the present version of XDLSM these constraints are not treated automatically. The violation of a symmetry restriction leads to singular least squares matrix. If the solution is obtained via diagonalization, the singularity can be eliminated (the corresponding constraint is introduced). Although this procedure might work in most cases, it is not advisable to let a numerical procedure handle the symmetry. After several cycles, round-off errors are likely to break the symmetry in the shifts. Symmetry restrictions can easily be formulated. Either the variable itself or the combination of two variables are fixed. The former does not need extra **CON** card since the corresponding variable is simply not refined (see **KEY** table).

**Example:**
Formula KHF\(_2\), space group \(I 4/mcm\) (No. 140). The three atoms in the asymmetric unit occupy the following special positions:

<table>
<thead>
<tr>
<th>No.</th>
<th>atom</th>
<th>Wyckoff letter</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>K</td>
<td>a</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>h</td>
<td>x</td>
<td>x+1/2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>H</td>
<td>d</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
</tr>
</tbody>
</table>

The corresponding **CON** cards are:

CON 1 x/2 -1 y/2 = -0.5  
CON 1 u11/1 -1 u22/1 = 0  
CON 1 u11/2 -1 u22/2 = 0  
CON 1 u11/3 -1 u22/3 = 0  

Another example of practical importance is the use of Kubic harmonics. Let suppose that atom 1 is at a cubic site. The symmetrized spherical harmonics to be applied is composed as a linear combination of hexadecapoles, H0 and H4+. The corresponding constraint is

CON 1 H4+/1 -0.74048 H0/1 = 0

Site symmetry restrictions on thermal tensor elements and on spherical harmonics are given in references [24,25].
4.6.11 The KEY table

```
KEY  xyz -U2- ----U3---- ------U4------- M- -D- -Q- -O- ----H----
```

This input segment is to specify which parameter is to be refined and which is not. It is done by giving the **KEY**-integer array with values 0 or 1 for a fixed or for a refined parameter, respectively. The order of the parameters is as defined before in Table 4-2. First the atomic parameters \((x, y, z, U_1^i, U_2^i, U_3^i, U_4^i, P_{lm})\) have to be given for all of the atoms included in the structure model. These are followed by the "shared" \((\kappa, \kappa')\) and by the "global" parameters (extinction, overall thermal parameter, scale factors). The heading of the **KEY** table helps in keeping account of the variables. The different abbreviations are as follows:

- **xyz**: 3 positional coordinates
- **Un**: \(n\)th order displacement amplitude tensor components. There are 6, 10 and 15 for \(n=2,3\) and 4, respectively
- **M**: 2 monopole populations; the first for sphv and the second for defv
- **D**: 3 dipole
- **Q**: 5 quadrupole
- **O**: 7 octupole
- **H**: 9 hexadecapole populations
- **KAPPA**: 1 for sphv and 5 for defv \((\kappa', l=0,lmx)\). It should be given for each KAPPA set defined in the ATOM table
- **EXTCN**: 1, 6 or 7 extinction parameters
- **OVTHP**: 1 overall thermal parameter
- **SCALE**: \(NQ\) scale factors

As many atom entries are to be given as in the atom table. The atom names used here have to be identical to those in the atom table otherwise the program terminates with error message. Similarly, the number of kappa entries must be equal to the maximal number used in the atom table to refer to kappa sets (see 4.6.3.5). If the command **KEEP kappa** is applied to a set then all but the first kappa-integers for the corresponding defv \((\kappa', l=1,4)\) should be zero. The number of key integers for the scale factors should be less than or equal to the number given in xd.inp \((NQ)\). The key integers are interpreted according to the maximal level of GC and multipole expansion defined in the atom table for each atom by the parameters \(tp\) and \(lmx\), respectively.

**Important!** The multipole populations of the atoms involved in "chemical constraints" (those which are constrained) **must** be fixed. The populations of that atom to which the others are constrained are "free" variables. Parameters involved in any other constraint **must** be made variables. An example is:

```
KEY  xyz --U2-- ----U3---- ------U4------- M- -D- --Q-- ----O-- ----H----
O(1)  111 111111 0000000000 000000000000000 10 110 10011 0110011 100110011
  ...
KAPPA 110000
EXTCN 0000000
OVTHP 0
SCALE 111
END KEY -----------
```
Table 4-1: Dimensioning of XDLSM

<table>
<thead>
<tr>
<th>Name</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nat</td>
<td>2000</td>
<td>maximum number of atoms in the asymmetric unit</td>
</tr>
<tr>
<td>ntx</td>
<td>31</td>
<td>maximum number of displacement tensor components:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$6 U_{ij} + 10 U_{ijk} + 15 U_{ijkl} = 31$</td>
</tr>
<tr>
<td>lnx</td>
<td>4</td>
<td>maximum level of multipole expansion</td>
</tr>
<tr>
<td>nzz</td>
<td>30</td>
<td>maximum number of kappa sets</td>
</tr>
<tr>
<td>nto</td>
<td>1</td>
<td>currently not used</td>
</tr>
<tr>
<td>nsc</td>
<td>20</td>
<td>maximum number of scale factors</td>
</tr>
<tr>
<td>ntb</td>
<td>20</td>
<td>maximum number of core, valence scattering factor tables</td>
</tr>
<tr>
<td>nov</td>
<td>2500</td>
<td>maximum number of variables allowed</td>
</tr>
<tr>
<td>ncst</td>
<td>200</td>
<td>maximum number of constraints</td>
</tr>
<tr>
<td>nao</td>
<td>18</td>
<td>maximum number of atomic orbitals allowed in the wavefunction input for the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scattering factors: 1s, 2s, 3s, 4s, 2p, 3p, 4p, 3d, 4d, 4f, 5s, 5p, 6s,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6p, 5d, 7s, 6d, 5f</td>
</tr>
<tr>
<td>mgrd</td>
<td>40</td>
<td>maximum number of grids used to store scattering factors</td>
</tr>
<tr>
<td>grd</td>
<td>0.05</td>
<td>Step size in $\sin \theta/\lambda$</td>
</tr>
</tbody>
</table>

Related to these the following parameters are also in use:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>npop</td>
<td>$\text{lnx} \times \text{lnx} + 2 \times \text{lnx} + 2$</td>
<td>maximum number of multipole populations</td>
</tr>
<tr>
<td>nap</td>
<td>$3 + \text{ntx} + \text{npop}$</td>
<td>maximum number of atomic parameters</td>
</tr>
<tr>
<td>npp</td>
<td>$\text{nap} \times \text{nat} + (\text{lnx} + 2) \times \text{nzz} + \text{nsc} + 8$</td>
<td>total number of parameters</td>
</tr>
</tbody>
</table>

Table 4-2: Variable names and order numbers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbolic name</th>
<th>Order number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fractional Coordinates</td>
<td>X, Y, Z</td>
<td>1 - 3</td>
</tr>
<tr>
<td>Displacement Tensor Components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd order $U_{ij}$</td>
<td>U11, U22, U33, U12, U13, U23</td>
<td>4 - 9</td>
</tr>
<tr>
<td>3rd order $U_{ijk}$</td>
<td>U111, U222, U333, U112, U221, U113, U331, U223, U332, U123</td>
<td>10 - 19</td>
</tr>
<tr>
<td>4th order $U_{ijkl}$</td>
<td>U1111, U2222, U3333, U1112, U2221, U1113, U3331, U2223, U3332, U1122, U1113, U2223, U1123, U2213, U3312</td>
<td>10 - 34</td>
</tr>
<tr>
<td>Multipole Populations</td>
<td>M1, M2</td>
<td>35 - 36</td>
</tr>
<tr>
<td>Monopoles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dipoles</td>
<td>D1+, D1-, D0</td>
<td>37 - 39</td>
</tr>
<tr>
<td>Quadrupoles</td>
<td>Q0, Q1+, Q1-, Q2+, Q2-</td>
<td>40 - 44</td>
</tr>
<tr>
<td>Octupoles</td>
<td>O0, O1+, O1-, O2+, O2-, O3+, O3-</td>
<td>45 - 51</td>
</tr>
<tr>
<td>Hexadecapoles</td>
<td>H0, H1+, H1-, H2+, H2-, H3+, H3-, H4+, H4-</td>
<td>52 - 60</td>
</tr>
<tr>
<td>Radial Screening</td>
<td>KS, K0, K1, K2, K3, K4</td>
<td>61 - 66</td>
</tr>
<tr>
<td>Isotropic and Anisotropic Extinction</td>
<td>EX11,EX22,EX33,EX12,EX13,EX23,RHOEX</td>
<td>67 - 73</td>
</tr>
<tr>
<td>Overall U</td>
<td>OTP</td>
<td>74</td>
</tr>
<tr>
<td>Scale Factor</td>
<td>SC</td>
<td>75</td>
</tr>
</tbody>
</table>
Table 4-3: The model limits

<table>
<thead>
<tr>
<th>m1 static scattering models</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
</tr>
<tr>
<td>-3</td>
</tr>
<tr>
<td>-2</td>
</tr>
<tr>
<td>-1</td>
</tr>
<tr>
<td>lmax</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m2 thermal motion models</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>tmax</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m3 anomalous dispersion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m4 extinction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
### Table 4-4: Index Picking Rules of Site-Symmetric Spherical Harmonics [25]

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Choice of coordinate axes</th>
<th>Indices of symmetric $Y_{\ell m}$ ($\lambda, \mu$ are integers)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>any</td>
<td>all ($l,m,\pm$)</td>
</tr>
<tr>
<td>6</td>
<td>$\vec{z}$</td>
<td>($2\lambda,\mu,\pm$)</td>
</tr>
<tr>
<td>m</td>
<td>$\vec{m}\perp z$</td>
<td>($l,\pm 2\mu$)</td>
</tr>
<tr>
<td>2/m</td>
<td>$2</td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>$2</td>
<td></td>
</tr>
<tr>
<td>mm2</td>
<td>$2</td>
<td></td>
</tr>
<tr>
<td>mmm</td>
<td>$m \perp z$, $m \perp y$, $m \perp x$</td>
<td>($2\lambda,2\mu,+$)</td>
</tr>
<tr>
<td>4</td>
<td>$4</td>
<td></td>
</tr>
<tr>
<td>4/m</td>
<td>$4</td>
<td></td>
</tr>
<tr>
<td>422</td>
<td>$4</td>
<td></td>
</tr>
<tr>
<td>4/mm</td>
<td>$4</td>
<td></td>
</tr>
<tr>
<td>$\bar{4}$</td>
<td>$4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$3</td>
<td></td>
</tr>
<tr>
<td>$\bar{3}$</td>
<td>$3</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>$3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$2</td>
<td></td>
</tr>
<tr>
<td>3/m</td>
<td>$3</td>
<td></td>
</tr>
<tr>
<td>$\bar{3}$</td>
<td>$3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$m \perp x$</td>
<td>($2,\lambda,6\mu,+, (2,\lambda,6\mu,3,-)$)</td>
</tr>
<tr>
<td>6</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>$\bar{6}$</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>6/m</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>622</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>6/mm</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>$\bar{6}$</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td>m2</td>
<td>$6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$m \perp x$, $(2</td>
<td></td>
</tr>
<tr>
<td>6/mmm</td>
<td>$6</td>
<td></td>
</tr>
</tbody>
</table>
Bibliography

Chapter 5

XDGEOM – Geometry Functions, Errors and Tables

5.1 Overview

This program computes various functions of the atomic coordinates together with standard uncertainties (s.u.s) obtained from the variance-covariance matrix. The method is fully described by Busing, Martin and Levy [1], and is given in outline below. Errors in the unit cell parameters are applied if they are provided in the form of a CELLSD entry in the Master File (see example). Torsion angle s.u.s are calculated by the method of Stanford and Waser [2]. An algorithm due to Rollett [3] is used to generate space-group symmetry-related atoms automatically, where relevant.

Additionally, CIF (Crystallographic Information File [4]) listings of the functions and atomic parameters are produced in the file xd_geo.cif. The rhoCIF specifications [5] for multipole population coefficients are used. Note that in a future release of XD, it is planned to move listings of all refined parameters to xd_lsm.cif.

5.1.1 Mathematical method

A function \( l \) of the \( n \) nuclear positions may be defined as

\[
l = f(p_1, p_2, \ldots, p_n)
\]

and its estimated standard uncertainty as

\[
\sigma(l) = f(u_1, u_2, \ldots, u_n)
\]

where

\[
u_i = \frac{\partial l}{\partial p_i}
\]

The derivative of a function of several variables is given by

\[
dl = \frac{\partial l}{\partial p_1} dp_1 + \frac{\partial l}{\partial p_2} dp_2 + \ldots + \frac{\partial l}{\partial p_n} dp_n
\]

so that

\[
\sigma^2(l) = \left[ \sum_{i=1}^{n} u_i c_i \right]^2
\]

where \( c_i = \sigma(p) \). Hence

\[
\sigma^2(l) = \sum_{i=1}^{n} \sum_{j=1}^{n} u_i u_j c_i c_j
\]

which is a quadratic form. So the variance of \( l \) may be written

\[
\sigma^2(l) = \mathbf{u}^T \mathbf{C} \mathbf{u}
\]

and the matrix \( \mathbf{C} \) is the variance-covariance matrix. Its diagonal elements are the variance of the parameters:

\[
C_{ii} = c_i^2
\]
and its off-diagonal elements are covariances:
\[ C_{ij} = \text{cov}(p_i, p_j) = c_i c_j r_{ij} \]
where \( r_{ij} \) is a correlation coefficient.

5.2 Files used and created by XDGEOM

Input: xd.mas, xd.res, xd.cov
Output: xd_geo.cif, xd.tex, xd_geo.out

5.3 Input instructions for XDGEOM

5.3.1 SELECT

SELECT (*rmin rmin (*rmax rmax (*ato (*bon (*ang (*tor (*loc (*non

(*rmin rmin [0.01] (*rmax rmax [use radii from databank
Internuclear separations in Å defining bonded atoms, and atoms in van der Waals contact, may be given here. Bond and contact distances and angles are calculated according to the connectivity so defined. If non is selected, rmin and rmax define the range of van der Waals contacts. Distances less than \( r_{\text{min}} \) are then taken to be bonding distances.

Alternatively, the default action is to use the covalent and van der Waals radii from xd.bnk_* (the chosen system databank file) to define bond and contact distances. In this case rmin and rmax are not selected. A special case is the selection of an rmin value of exactly zero, in conjunction with non. Then bonds are defined by covalent radii and contacts by the selected value of rmax.

(*ato
Generates a list of atomic coordinates, displacement amplitudes, multipole population coefficients, and their estimated standard deviations.

(*bon
Generates a list of bond distances and their s.u.’s.

(*ang
Generates a list of bond angles and their s.u.’s.

(*tor
Generates a list of torsion angles and their s.u.’s.

(*loc
Lists the “local geometry” at each atom. This is a list of all bonds and angles and their s.u.s for each atom in turn, together with the symmetry operations used to generate the connected atoms.

(*non
Generates a list of van der Waals contacts and their s.u.s, with the symmetry operations used to generate the contacting atoms.
5.4 Example Master File segment for XDGEOM

TITL     HCONH2  Formamide 123K
CELL      3.6130     9.0570     6.9730   90.000  100.360   90.000
CELLSD .005 .003 .003 0.0 0.06 0.0
WAVE    0.71069
LATT C P
SYMM  1/2 – X, 1/2 + Y, 1/2 – Z
.
.
MODULE XDGEOM
SELECT *rmin 1.6 *rmax 4.0 *ato *bon *ang *tor *non
END XDGEOM

Bibliography

Chapter 6

XDPROP – One-Electron Properties based on the Multipole Representation of the ED

6.1 Overview

XDPROP is a program for analysing a STATIC ED which has been obtained in the form of a nuclear-centered multipole expansion. It is based on the earlier programs LSPROP (by Sean Howard & Paul Mallinson) and MOLPROP (by Zhengwei Su). XDPROP is under continuous development and now contains several new features which are discussed below.

Many of the features in the static properties package are not new. Existing programs such as VALRAY [1] (by Mark Spackman and Bob Stewart), POP (by Brian Craven), LSEXP (by Fred Hirshfeld), and MOLLY[2] by Niels Hansen and Philip Coppens) have the ability to compute electrostatic potentials, dipole moments and carry out critical point analyses of the total density. For accounts of the state-of-the-art of charge density applications, the reader is referred to pertinent literature [3].

6.1.1 Abbreviations

This chapter uses the following notation and abbreviations: \( \rho \) (total electron density); CP (Critical Point); \( \Delta \rho \) (Deformation Density); IAM (Independent Atom Model); \( V(\mathbf{r}) \) (electrostatic potential); \( \nu(\mathbf{r}) \) (nuclear potential); BP (Bond Path - a line of maximum charge density connecting two nuclei); PD charges (electrostatic Potential-Derived charges; \( \mathbf{E} \) (Electric field); EFG (Electric Field Gradient); ST (Slater-Type, as of a radial function); HF (Hartree-Fock); LSQ (Least Squares, as of a refinement procedure); ESP (ElectroStatic Potential).

6.1.2 Units

XDPROP output is in Å, electrons and degrees. Exceptions include the \( \zeta_l \)-values (exponents of Slater-type radial functions), which are conventionally reported in Bohr\(^{-1}\), and multipole moments (Debye and Debye-Å).

6.1.3 Files used and created by XDPROP

Input: xd.mas, xd.res, xd.bnk_*
Optional input: xd.inp (if xd.res does not exist)
xd.cov (if errors are to be calculated)
Output: xd_pro.out
Optional output: xd_property.cps, xd_property.grd, xd.pth,
xd_bubble.spf, xd_* .dat

6.1.4 Coordinate systems

Although XDLSM uses local coordinate systems on the pseudoatoms, XDPROP works mostly with a single, "global" system of Cartesian coordinates \( \mathbf{r} \) generated from the fractional coordinates \( \mathbf{r}_f \) in xd.res by the transformation

\[
\mathbf{r} = \mathbf{M} \mathbf{r}_f
\]  
(Eq. 6.1)

where \( \mathbf{M} \) is the matrix [4]
This is particularly important in considering dipole and quadrupole moments computed with XDPROP, which are computed in this frame.

6.1.5 Current Limitations

(a) The calculation of errors on \( \rho \) and \( -\nabla^2 \rho \) has some severe limitations at the moment, namely: (i) only contributions from the multipole populations are currently taken into account \( \text{i.e.} \) not \( \kappa \)'s or coordinates) (ii) contributions to the error from symmetry-generated atoms are not taken into account.

(b) The potential-derived charge fitting subroutine assumes that the fragment is molecular (neutral), so it will give nonsensical results for ions.

(c) Electric field and electric field gradient calculations are not yet available, although they are mentioned in some parts of the manual.

(d) It is only possible to compute errors on the dipole moment, \( \rho \), or \( -\nabla^2 \rho \) if Slater-type radial functions have been used for all multipoles.

6.2 Input Instructions for XDPROP

The Master file entry for XDPROP should begin with MODULE *XDPROP and be terminated by the END XDPROP line.

6.2.1 MODEL

MODEL (*)iam (*)multipole

Only two options are currently available - to use the multipole model (or some modification of it) which was applied in the refinement; or to generate the iam (independent atom model) in its place. In the latter case, multipole populations \( (l \geq 1) \) are set to zero, monopole populations are set to the free atom values, and the \( \kappa \) parameters are set to unity.

6.2.2 APPLY

APPLY symm is trans tx ty tz (atoms ...l all)

This command is used to apply a crystallographic symmetry operation to some or all of the pseudoatoms in the asymmetric unit. This can be useful if, for example, the asymmetric unit does not contain a whole molecule, or if the properties of two or more molecules are to be analysed, or for studying intermolecular hydrogen bonding. The symmetry operations are referenced according to the sequence in which they are listed in xd_pro.out. For example, the following entry

APPLY SYMM 3 TRANSLATIONS -2 -1 1 O(1) N(1) C(1) H(1) H(2) H(3)

applies the third symmetry operation, with extra lattice translations, to the six atoms whose labels are given. If you do not wish to supplement the symmetry operation with additional lattice translations, use TRANSLATIONS 0 0 0. If the symmetry operation is to be applied to all atoms in the asymmetric unit, then the keyword all may be used in place of the atom labels. More than one line beginning with APPLY may be present. There is a limit of a maximum of 7 atoms which may be specified individually on any single APPLY instruction.
One important point to note is the XDPROP convention for labelling symmetry-generated atoms. This label is of the general form "X"+sym.op number+"_"+original atom label. So for example, an atom generated from O(8) by symmetry operation 6 will have the label X6_O(8). This label should be used to refer to the symmetry generated atom in the other modules of XDPROP, for example if you want to use such an atom to define a plane, or in a CP search.

6.2.2 GROUP

GROUP (not) atoms ...

Whereas APPLY is concerned with expanding the number of pseudoatoms under study, GROUP has the opposite effect of selecting a subset of pseudoatoms to be studied. Henceforth all sums over pseudoatoms to compute properties will include only those pseudoatoms whose atom labels follow the GROUP keyword. Thus an example might be

GROUP H(1) O(1) H(2)

If the option not is given, all atoms specified in this GROUP instruction are not part of the active group.

GROUP NOT H(1) O(1) H(2)

The effect of two or more GROUP instructions is additive. It is possible to reset the active group using the keyword ALL GROUP NOT ALL or GROUP ALL.

There are effectively no lower or upper limits on how many atoms can make up the group. This command is useful, for example, when the asymmetric unit contains several moieties, and you want to compute the properties of just one of them. The negated form might be useful if you want to exclude only a small number of atoms.

6.2.4 DEFGROUP

DEFGROUP atoms ...

This is specialized option concerned only with the deformation density. It will have no effect if the active property is not DEFDEN. The purpose of DEFGROUP is to define a subset of atoms whose spherical atom density will not be subtracted from the total density in creating the deformation density. The normal option (the default) is to subtract the spherical atom density at the nuclear positions for all the atoms in the active group. The syntax is simply (for example)

DEFGROUP Ni(1) Ni(2)

6.2.5 SELECT

SELECT (*) flag ...

SELECT (keyword value) ...

So far as is possible, all parameters in XDPROP have some useful default values. However, you will doubtless find it necessary to change something, and a number of parameters may be altered using this option. More than one card beginning with SELECT may be present, and each may contain up to five of the keywords to be described shortly. The first five of these, namely local, numdx, check, esd and nocore are just flags, which require no further qualifiers. The remaining sub-options correspond to system variables, and the numerical value of the variable must follow the keyword.

Currently, you should not mix flags and options followed by a value on one SELECT line.
(**)local

The default is to use a set of multipole populations transformed into a global coordinate system for property computation. Selecting the local keyword with an asterisk will instead carry out these sums using the original, untransformed populations which came directly from the multipole refinement. The results should be identical - this is a diagnostic feature, mainly useful for further development of the program.

(**)numdx

CP searching a function requires a full set of first and second derivatives with respect to a global cartesian coordinate system. Analytic derivatives are available for many functions, and for others the derivatives will be computed numerically with finite differences. Placing an asterisk by this keyword forces the program to use numerical derivatives for CP searching any function, regardless of whether analytic derivatives are available.

(**)check

This turns on some extra output from the properties program, including lists of pseudoatom parameters and populations. It also checks the atomic density matrices for all atom types found in the input file xd.res, by numerical integration. This helps to verify that any new wavefunctions added to the file xd.bnk_* have been incorporated correctly. These normalization integrals can in fact be carried out analytically (a subroutine PMOM in XDPROP not called anywhere in this release computes analytic radial moments of the atomic charge distributions - the zero moment gives the total charge).

(**)esd

The default is not to compute any errors since this needs the variance-covariance matrix file xd.cov which, due to its large size, may have been deleted prior to running XDPROP. Selecting esd turns on computation of errors, in so far as they are available for different properties. Currently errors are only available for: the dipole moment, \( \rho \) (or \( \Delta \rho \)) and \( \nabla^2 \rho \). In least squares refinement, the variance in some property \( A \), derived from the \( N_{\text{var}} \) refined parameters \( \{ P \} \), may be estimated from the variance-covariance matrix \( \epsilon \) [5]:

\[
\sigma^2(A) = S^2 \sum_{i,j}^{N_{\text{var}}} \left( \frac{\partial A}{\partial P_i} \right) \left( \frac{\partial A}{\partial P_j} \right) \epsilon_{ij} \tag{Eq. 6-2}
\]

where \( S \) is the goodness-of-fit.

(**)nocore

This means that for the peripheral contributions to the potential, the core electrons and the part of nuclear charge equal in magnitude to the number of core electrons are assumed to cancel exactly. (Applies to the calculation of the esp only.)

cpcut dxcut [0.0001]

This is used in CP searching. A CP is considered to have been located at some point \( r_c \) if \(|\nabla f(r_c)| < \text{dxcut}\). The default value is 0.0001 Å, which is generally fine for \( \rho \), but may be quite inappropriate for searching other functions.

lmax lmax

This is a global maximum \( l \)-value applied to all pseudoatoms. The default procedure is to carry out multipole sums on each pseudoatom to the maximum \( l \)-value in place on that atom. This will be overridden by the lmax option (an integer variable must be given).
Although it is not possible to change $l_{\text{max}}$ on individual pseudoatoms using these master file instructions, it could of course be achieved by manually editing the \texttt{xd.res} file.

\textbf{nstep nstep}

Critical Points (CPs) are located by an iterative procedure - this parameter determines the maximum number of steps used in searching for a CP, before the search is abandoned.

\textbf{rcut rcut [4.0]}

Local properties $f(r)$ are computed as a sum over pseudoatom contributions: $f(r) = \Sigma_j f_j(r)$. For many properties, notably $\rho$, $\nabla^2 \rho$ and $\Delta \rho$, it is an excellent approximation to ignore contributions of pseudoatoms which lie further than a distance cut from the point $r$. The default distance of $4.0$ Å is generally reliable for the afore-mentioned properties, and organic compounds. It is not likely to be suitable (i) when ‘large’ atoms are present, e.g. transition metals (ii) for $V(r)$ or $\nu(r)$ (iii) when some pseudoatoms have rather small values of $\kappa$ and/or $\kappa^\circ$. Users are therefore urged to check that the property being computed is converged with respect to this parameter.

\textbf{scale scale [0.05]}

CP searching involves consecutive steps $\delta s$ towards stationary points. The direction of each step is determined by the local gradient vector and Hessian matrix for the property concerned. The step-length, on the other hand, is controlled by the variable \texttt{scale}, with a default of $0.05$ Å. This value works well for searching $\rho$, but other (most likely smaller) values should be considered in searching more rapidly-changing functions such as $\nabla^2 \rho$.

\textbf{dx xstep [0.001]}

The step-length for numerical derivatives (used in CP searching those properties for which analytical derivatives are not yet available).

\textbf{ds pstep [0.005]}

Bond paths are determined by numerical integration of the gradient vector $\nabla \rho(r)$ along the line of maximum density, and \texttt{pstep} is the (initial) step-length for this integration.

The following options apply to the calculation of the electrostatic potential (\texttt{esp}) only:

\textbf{rad1 rad1 [0.1]}

Calculations for points within a radius of \texttt{rad1} of any atom will not be performed. Instead, arbitrary big values will be assigned. The properties at the nuclei however, will be calculated.

\textbf{rad2 rad2 [200.0]}

Atoms outside a radius of \texttt{rad2} of a point will not be included in calculations.

\textbf{rad3 rad3 [10.0]}

Border between \texttt{zone1} and \texttt{zone2} (see below).

\textbf{zone1 idf1 [1]}
If $idf_1=1$, contribution of an atom whose nucleus is between $rad_1$ and $rad_2$ but less than $rad_3$ away from the point of interest is evaluated without neglect of those due to dipolar and higher multipolar densities: 'exact' formulae used for all the multipoles.

If $idf_1=0$, the contributions of dipoles and higher multipoles are neglected. "Exact" formulae used for monopoles only.

**zone2 idf2 [1]**

If $idf_2=1$, contribution of an atom whose nucleus is between $rad_1$ and $rad_2$ but greater than or equal $rad_3$ away from the point under question is evaluated in the following way: "Exact" formulae for dipolar and higher multipolar densities, point charges for monopolar densities.

If $idf_2=0$, point charges for monopoles; higher poles neglected.

### 6.2.6 DIPOLE

**DIPOLE *cmass (*)center (*)ucell**

The dipole moment can be calculated in one of three frames. **cmass** uses the center of mass as the origin, **center** the geometric center and **ucell** the origin of the unit cell.

The user is referred to the excellent review by Spackman on computing molecular multipole moments from X-ray diffraction data [6]. The computation of the molecular dipole moment requires some definition of a "molecule" in the crystal [7-9] since the positioning of the intermolecular boundary is arbitrary. XDPROP assumes that the density of a molecular fragment in the crystal is derivable from just the $n$ pseudoatoms that would describe the free molecule. This ignores any overlap between pseudoatom multipole functions on neighbouring molecules. Then the dipole moment is given by

$$
\mathbf{p} = \sum_{j=1}^{n} \mathbf{p}_j \quad \text{(Eq. 6-3)}
$$

where

$$
\mathbf{p}_j = Z_j \mathbf{R}_j - \int \mathbf{r} \rho_j (\mathbf{r}_j) d\mathbf{r} \quad \text{(Eq. 6-4)}
$$

$\mathbf{R}_j$ is the nuclear position vector, and $\mathbf{r} = \mathbf{r} - \mathbf{R}_j$. In a monopole-only refinement, the pseudoatom dipole contribution is given by

$$
\mathbf{p}_j = Z_j \mathbf{R}_j - \int \mathbf{r} \rho_j (\mathbf{r}_j) (\mathbf{r}_j + \mathbf{R}_j) d\mathbf{r} = q_j \mathbf{R}_j \quad \text{(Eq. 6-5)}
$$

which follows from the normalization of the monopole functions ($\int \rho_j (\mathbf{r}_j) d\mathbf{r}_j = 1$), and the odd symmetry of the second integrand ($\int \rho_j (\mathbf{r}_j) \mathbf{r}_j d\mathbf{r}_j = 0$). Only the monopoles and dipoles contribute to $\mathbf{p}_j$ due to the symmetries of the $d_{lm}$ and the dipole operator. If the radial functions are normalized Slater-type functions

$$
\mathbf{R}_j (r) = \frac{\sum_{l=0}^{n+3} \rho_j^m l^m e^{-z_j r}}{(n + 2)!} \quad \text{(Eq. 6-6)}
$$

then the remaining integrations may be performed analytically, to give

$$
\mathbf{p}_j = q_j \mathbf{R}_j - P_{l=1}^{1+l} \int d_{l=1} \mathbf{R}_j x d\mathbf{r} \quad \text{(Eq. 6-7)}
$$

$$
- P_{l=1}^{1+l} \int d_{l=-1} R_j y d\mathbf{r} - P_{l=0}^{1} \int d_{l=0} R_j z d\mathbf{r}
$$

59
The quadrupole moments are computed from the following formulae:

\[
Q_{ij} = \sum_j x_j y_j (Z_j - P_{j\,\xi} - P_{j\,\nu}) + q_{i\,\xi y} + y_j p_{j\,\xi} + x_j p_{j\,\nu} \quad \text{(Eq. 6-8)}
\]

\[
Q_{iz} = \sum_j x_j z_j (Z_j - P_{j\,\xi} - P_{j\,\nu}) + q_{i\,\xi z} + x_j p_{j\,\xi} + z_j p_{j\,\nu} \quad \text{(Eq. 6-9)}
\]

\[
Q_{yz} = \sum_j y_j z_j (Z_j - P_{j\,\xi} - P_{j\,\nu}) + q_{i\,y z} + y_j p_{j\,\xi} + z_j p_{j\,\nu} \quad \text{(Eq. 6-10)}
\]

\[
Q_{i\,z^2-y^2} = \sum_j \frac{1}{2} (x_j^2 - z_j^2) (Z_j - P_{j\,\xi} - P_{j\,\nu}) + q_{j\,x\,z^2-y^2} + x_j p_{j\,\xi} - y_j p_{j\,\nu} \quad \text{(Eq. 6-11)}
\]

\[
Q_{i\,3z^2-r^2} = \sum_j \frac{1}{2} (2z_j^2 - x_j^2 - y_j^2) (Z_j - P_{j\,\xi} - P_{j\,\nu}) + q_{j\,3z^2-r^2} + 4z_j p_{j\,\xi} - 2x_j p_{j\,\nu} - 2y_j p_{j\,\nu} \quad \text{(Eq. 6-12)}
\]

where \( P_{\xi} \) and \( P_{\nu} \) are the core and monopole populations of the \( j \)-th pseudoatom. The dipole terms are given by

\[
P_{j\,\xi
\nu} = -\frac{4P_{j\,1\,\xi\,\nu}(n_j + 3)!}{3k_j^{\xi\nu}(n_j + 2)!} \quad \text{(Eq. 6-13)}
\]

(for \( P_{ii} \) and \( P_{ii} \) replace \( P_{ji\,\xi\,\nu} \), with \( P_{ji\,\xi\,\nu} \) and \( P_{ji\,\xi\,\nu} \) respectively). The pseudoatom quadrupole terms are:

\[
q_{j\nu} = -\frac{2\pi}{15(k_j^{\nu})(n_j + 3)(n_j + 4)} P_{2\nu\,i\,2\nu\,i\,2\nu\,i\,2\nu\,i} \quad \text{(Eq. 6-14)}
\]
(for \(q_{jxx}, q_{jyy}\) or \(q_{jxz}, q_{jyz}\) replace \(P_{jz2}\), with \(P_{jz1}\), \(P_{jz1}\) or \(P_{jz2}\), respectively). The remaining quadrupole is

\[
q_{j,3^2-r^2} = -\frac{6\sqrt{3}}{5(k_j^2r_j)^2} P_{20}(n_l + 3)(n_l + 4) \tag{Eq. 6-15}
\]

\(\text{xd}\.\text{pro}\).out gives a complete "breakdown" of pseudoatom contributions to each of the five molecular components in terms of monopole, dipole and quadrupole functions. The main number likely to be of interest is the total moment, appearing at the bottom right of the output for each component.

### 6.2.8 EXPORT

**EXPORT *orient *min16 lmax nmo nmol natmol n1 n2 ...**

This command enables XDPROP to write out atomic moments in spherical tensor notation [10] in ORIENT [11] (when *orient* is flagged) and MIN16 [12] (when *min16* is flagged) input formats. *lmax* defines the maximum level of atomic moments to be written. *nmo* defines the "number of molecules" present in the atom list (including atoms generated with APPLY and GROUP instructions). *n1*, *n2* .. define the number of atoms in each of the molecules. By default there is only 1 molecule, which includes all atoms in the list.

### 6.2.9 STOCKHOLDER partitioning and UNABRIDGED MOMENTS

Integration of the electron density partitioned according to the Stockholder method [13] and corresponding unabridged moments can be performed.

The following lines are necessary in the xd.mas file:

```
STOKMOM (*)defden lmin lmax *cmass (*)center (*)ucell (*)debug
STOKMOM minlim xmin ymin zmin maxlim xmax ymax zmax epsa epsa epr epr
STOKMOM *orient *min16 lmax nmol natmol n1 n2 ...
STOKMOM atoms (*)all (*)select ato(1) ato(2) ...
```

(*defden lmin lmax [0] lmax [4] *cmass (*)center (*)ucell (*)debug

defden defines the type of the density on which the integration is done. If flagged the integration of Stockholder moments is done on deformation density, otherwise on the total density (the default). *lmin* and *lmax* define the limits of the moments to be integrated. *cmass*, *center* and *ucell* define the origin of the coordinate system (same as for DIPOLE and QUADPOLE): *cmass* uses the center of mass, *center* the geometric center and *ucell* the origin of the unit cell. *debug*, if flagged, enables more verbose output.

```
```

Parameters *xmin* *ymin* *zmin* *xmax* *ymax* *zmax* define the integration limits (in Å) around each atom in the Cartesian frame. Parameters *epsa* and *epr* define the accuracy of numerical integration, i.e. absolute (*epsa*) and relative (*epr*) errors.

*orient *min16 lmax [4] nmol nmol [1] natmol n1 n2 ...

If flagged, *orient* or *min16* write out Stockholder atomic moments in spherical tensor notation [10] in ORIENT [11] and MIN16 [12] input formats. *lmax* defines the maximum level of atomic moment expansion to be written. *nmol* defines the "number of molecules" present...
in the atom list (including atoms generated with APPLY and GROUP instructions). \( n1, n2 \ldots \) define the number of atoms in each of the molecules. By default there is only 1 molecule, which includes all atoms in the list.

**atoms *all (*)select ato(1) ato(2) ...**

If select is flagged, only the specified atoms \( ato(1) ato(2) \ldots \) will be integrated, otherwise all atoms in the atom list will be integrated (default).

### 6.2.10 D-POP

Occupancies of the \( d \) orbitals can be derived (in the approximation of low overlap) according to the approach suggested by Holladay, Leung and Coppens [14]. The option D-POP will give the calculated the \( d \) orbital occupancies in the output, for example:

**MULTIPOLE POPULATIONS**

\[
\begin{align*}
P00 & = 2.376 \\
P20 & = 0.094 \\
P21^+ & = 0.000 \\
P21^- & = 0.000 \\
P22^+ & = 0.000 \\
P22^- & = 0.000 \\
P40 & = 0.213 \\
P41^+ & = 0.000 \\
P41^- & = 0.000 \\
P42^+ & = 0.000 \\
P42^- & = 0.000 \\
P43^+ & = 0.001 \\
P43^- & = 0.011 \\
P44^+ & = 0.000 \\
P44^- & = 0.000 \\
\end{align*}
\]

**ORBITAL POPULATIONS**

\[
\begin{align*}
z^2 & = 0.87065 \\
xz & = 0.32579 \\
yz & = 0.32579 \\
x^2-y^2 & = 0.42701 \\
xy & = 0.42701 \\
z^2/xz & = 0.00000 \\
z^2/yz & = 0.00000 \\
z^2/x^2-y^2 & = 0.00000 \\
z^2/xy & = 0.00000 \\
xz/yz & = 0.00000 \\
xz/x^2-y^2 & = 0.00126 \\
xz/xy & = 0.02387 \\
yz/x^2-y^2 & = 0.00126 \\
yz/xy & = -0.02387 \\
xz-y^2/xy & = 0.00000 \\
tot d-pop & = 2.37625 \\
\end{align*}
\]

If (*)esd in the xdprop heading is flagged, estimated errors are also tabulated.

### 6.2.11 PROPERTY

**PROPERTY *prop**

XDPREP will analyse the property which is flagged with an asterisk (*) on the PROPERTY card (there should be only one such a card). The following are currently available:

- **core** Core density
- **valence** Valence density
- **rho** Total density \( \rho \)
- **defden** Deformation density \( \Delta \rho \)
- **gradrho** magnitude of gradient vector of \( \rho \) \( |\nabla \rho| \)
- **d2rho** Laplacian of \( \rho \) \( \nabla^2 \rho \)
- **elpot** Electrostatic Potential \( V(\mathbf{r}) \) using a crude approximation (now deprecated)
- **esp** Electrostatic Potential \( V(\mathbf{r}) \) using the method of Su and Coppens [15]
- **nucpot** Nuclear Potential \( V(\mathbf{r}) \)
The core/valence decomposition of the electron density is based on the orbital occupations given in xd.bnk_* or in the SCAT table in the master file (see Section 4.6.2).

$\Delta \rho$ is the standard deformation density, i.e. the difference $\rho - \rho_{\text{IAM}}$. $V(r)$ is defined as

$$V(r) = \sum_j \frac{Z_j}{|r - R_j|} - \frac{1}{|r - r'|} \int \frac{\rho(r)}{|r - r'|} dr'$$

(Eq. 6-16)

where $R_j$ and $Z_j$ are the position and charge of the $j$-th nucleus, respectively. Outside the Van der Waals surface, it may be computed from an expansion in the multipole moments of individual pseudoatoms [16,17]

$$V(r) = \sum_j \frac{Z_j}{|r - R_j|} + \sum_j \left[ \frac{q_j}{r_j} + \frac{p_j \cdot e_j}{r_j^2} + \frac{1}{2} \sum_{k} \sum_{\ell} Q_{\ell j} \frac{r_{\ell j}^2}{r_j^2} + \ldots \right]$$

(Eq. 6-17)

where $q,p,Q...$ represent pseudoatom moments. A number of experimental charge density analyses using this type of multipolar representation of the potential have been reported [18-21]. In XDPROP, $V(r)$ is computed using the method of Su and Coppens [15], which does not rely on a multipole expansion. Thus the potential may be computed at points inside the charge distribution, permitting the calculation of $V$, $E$, and the EFG at the nuclei, and exploration of the topology of $V(r)$ [22].

The nuclear potential is just the contribution of the nuclei alone to (3.15), i.e.

$$V(r) = \sum_j Z_j |r - R_j|$$

(Eq. 6-18)

$sigrho$ and $siglap$ are computed with equation 6.2: *currently only the contributions from population parameters and $\kappa$'s are taken into account*. Hence these should be considered lower-limits. Note that $sigrho$ and $siglap$ may be very tedious to compute, so you might want to think twice before trying to generate a cube of $siglap$ data!

The electron localization function elf is computed based on the approximation for the kinetic energy density of Kirzhnits [23] as recently applied by Tsirelson [24]. **IMPORTANT**: this approximation for the elf is very poor at the nuclear positions, so it is completely inappropriate to analyse the topology of this scalar field using XDPROP.

The one-electron-potential function oep [25-27] is defined as:

$$\text{OEP} = \nabla^2 \sqrt{\rho} / (2 \sqrt{\rho}) = 1/4(\nabla^2 \rho / \rho) - 1/8(\nabla \rho / \rho)^2$$

The following four options are for computing values of the active property (i.e. flagged on the PROPERTY card) at specified points, along lines, or over 2-D and 3-D grids of points. An unlimited number of POINT, LINE, MAP and CUBE entries may be present. However, depending on the computer operating system, the various map files which are produced may overwrite one another. Thus it may be necessary to re-run XDPROP every time you want to produce a new map file, or rename the files between runs.
6.2.12 POINT (property at a point)

POINT x y z

This keyword followed by three coordinates in the global frame specifies a point at which the active property will be computed. No map file is created by this option, the result only appears in xd_pro.out.

6.2.13 LINE (property along a line)

LINE atom1 atom2 npts npts

This option computes the value of the active property along a line between two nuclei, specified by their atom labels. A final parameter which must be supplied is the number of (equidistant) points at which the property will be computed. So an example could be

LINE C(1) O(2) NPTS 21

Alternatively, the user may request a property along a line defined by points.

LINE POINTS x1 y1 z1 x2 y2 z2 npts 50

Note that x1, y1, z1 and x2, y2, z2 are the global Cartesian coordinates of the two points.

6.2.14 MAP (property over a 2-D grid of points)

MAP atoms atom1 atom2 atom3 npts npts step step (*)trans tx ty tz

MAP b vect1 x1 y1 z1 b vect2 x2 y2 z2 cen x0 y0 z0 npts npts step step

There are two ways of specifying the plane with MAP. The first uses three atom labels, and the map center is the centroid of the three atoms. The grid will be square (npts x npts), with a gridspacing of step in Å. You can also specify a translation of the origin, with respect to the grid axes. Translations of 1.0, 0.0, 0.0 or 0.0, 1.0, 0.0 will shift the origin 1 Å along the x or y grid axis directions (+x leftwards in the horizontal, +y downwards in the vertical directions), respectively. The trans directive must be starred to effect this. A translation of 0.0, 0.0, 1.0 selects the plane parallel to and 1 Å above the plane containing the three atoms (change the sign of the z translation to select planes below the plane of the atoms). The shifts in x, y and z can be non-integral. Thus an example is:

MAP ATOMS C(1) C(2) O(1) NPTS 61 STEP 0.1 TRANS 0.5 0. 0.

The second method is to supply two basis vectors, in the global Cartesian coordinate system. The map center must be given in the global Cartesian frame. npts and step have the same meaning as before. A complete example is:

MAP BVECT1 1. .0 .0 BVECT2 .0 .0 1. CEN 1.3 .2 -.1 NPTS 13 STEP 0.2

A map file is produced with the filename structure xd_mid_prop.grd. The maximum value of npts is 150.

6.2.15 CUBE (property over a 3-D grid of points)

CUBE center x0 y0 z0 npts npts step npts

CUBE atom1 atom2 npts npts step npts
This generates an \( npts \times npts \times npts \) cube of the active property, with a grid spacing of \( \text{step} \) (in Å). There are two ways of specifying the cube center: in global Cartesian coordinates, or as the midpoint of two atoms. Thus two examples are:

\begin{verbatim}
CUBE CENTER 0.3 9.82 1.5 NPTS 21 STEP 0.3
CUBE C(1) O(3) NPTS 21 STEP 0.3
\end{verbatim}

A map file is produced with the filename structure \( \text{xd}_\text{mid}_\text{prop}.\text{grd} \). The maximum value of \( npts \) is \( \approx 75 \).

### 6.2.16 CPSEARCH

**CPSEARCH bond** \( \text{atom1} \text{ atom2} \left( \frac{\text{frac}}{\text{frac}} \right) \left( \text{scan} \text{ delta} \right) \)

**CPSEARCH bond** \( \text{rmin} \text{ rmax} \)

**CPSEARCH ring** \( \text{atom1} \text{ atom2} \ldots \)

**CPSEARCH point** \( x \ y \ z \)

**CPSEARCH shell** \( \text{atom} \text{ rmin} \text{ rmax} \text{ nrad} \text{ nrad} \text{ nang} \text{ theta} \text{ phi} \text{ cutoff} \text{ cutoff} \)

**CPSEARCH start** \( \text{filename} \)

**CPSEARCH bubble** \( \text{atom} \text{ rmin} \text{ rmax} \text{ curv} \text{ icurv} \text{ ncps} \text{ npoints} \)

Topological analysis of the density is concerned with the scalar fields \( \rho \) and \( \nabla^2 \rho \) but XDPROP allows the user to CP search any of the properties listed previously. In the case of the nuclear and electrostatic potentials, these have very similar topologies to \( \rho \). The usefulness of CP searching the defden is probably restricted to finding local \((3,+3)\) minima or \((3,-3)\) maxima, the latter (for example) would be bonded or non-bonded peaks in the defden.

The search for CPs employs a 3-dimensional Newton-Raphson technique, which requires both the gradient vector and the Hessian \( H \) (i.e. the \( 3 \times 3 \) matrix of partial second derivatives of \( f \) with respect to \( \{x,y,z\} \)). Starting from some point, such as the midpoint between two nuclei, an improved estimate of a CP position \( r_c \) is found from iterative application of

\[
r_{c}^{(n+1)} = r_{c}^{(n)} - H^{-1} \cdot \nabla f
\]

(Eq. 6-19)

Each card beginning with CPSEARCH initiates a search of the property flagged with an asterisk on the PROPERTY card. Following CPSEARCH is a string which indicates how the searching is to be done. There are five modes of searching, which differ in how the starting coordinates are chosen, and how many times the CP search algorithm will be called.

(i) **bond** can initiate a CP search between the two atoms whose labels follow, e.g.

\begin{verbatim}
CPSEARCH BOND C(1) O(1)
\end{verbatim}

This type of search starts from the midpoint of the two nuclei. The **frac** option can be used to change this.

\begin{verbatim}
CPSEARCH BOND C(1) O(1) FRAC 0.7
\end{verbatim}
This starts the CP search at a fractional distance of 0.7 times the C-O bondlength, i.e. nearer to the oxygen. For bonds CPs which are tricky to find, a more thorough alternative is

\[
\text{CPSEARCH BOND C(1) O(1) SCAN 0.3}
\]

This carries out seven CP searches starting from (i) the midpoint of the nuclei \( \{x_o, y_o, z_o\} \) as before, then at \( \{x_o + \delta y_o, z_o\} \) and \( \{x_o, y_o + \delta, z_o + \delta\} \), where delta is the step (in Å) which appears after scan.

Alternatively if \textit{bond} is followed by \textit{rmin} instead of an atom label, then the program will search for all CP’s between all pairs of nuclei with internuclear separations between \( \text{rmin} \) and \( \text{rmax} \) (in Å). So typically, to locate all CP’s between bonded first-row atoms one might use:

\[
\text{CPSEARCH BOND RMIN 1.15 RMAX 1.6}
\]

(ii) \textit{ring} carries out a CP search starting at the centroid of the atomic coordinates of the atom labels which follow, so typically for a phenyl one could use something like:

\[
\text{CPSEARCH RING C(1) C(2) C(3) C(4) C(5) C(6)}
\]

At least two atom labels should be supplied, and no more than eight.

(iii) \textit{point} starts a search from the coordinates (in Å) which follow. These must be coordinates in the global Cartesian system, which is defined near to the top of the xd_pro.out file. So the form of the command is:

\[
\text{CPSEARCH POINT 1.2 -4.3 9.3}
\]

(iv) \textit{shell} is the most complex searching option, primarily intended for locating non-bonded charge concentrations in \( \nabla^2 \rho \), \( \Delta \rho \), or \( V(r) \). An example would be:

\[
\text{CPSEARCH SHELL O(1) MIN .5 MAX .7 NRAD 3 NANG 11 11 CUT 20.}
\]

This searches in concentric spherical shells around O(1), with radii 0.5, 0.6 and 0.7 Å (since the number of shells \textit{nrad}=3), over an \( 11 \times 11 \) angular grid of points in \( \theta \) and \( \phi \) (polar coordinates, referred to the global cartesian system). The minimum number of points in both \( \theta \) and \( \phi \) \text{[nang]} is two, and the minimum number of radial shells \( \text{[nrad]} \) is one. The algorithm operates in the following manner. Denoting the property being searched by \( f(r) \), \( |\nabla f(r)| \) is computed at each point \( \{r, \theta, \phi\} \) in the shell (\( 3 \times 11 \times 11 = 363 \) points, in the above example). If a point is found where \( |\nabla f(r)| < 20.0 \), then the Newton-Raphson CP search algorithm is initiated at this point, to see if a nearby CP can be located precisely. This routine often finds the same CP many times. Useful values of \textit{cutoff} vary so much with the property to be searched, the radial distance from the nucleus, and the atomic number of the atom, that it is difficult to give guidelines on sensible values (currently there are no default values in the program). Subsequent releases of the program may be more helpful in this respect, but for the moment it is a matter for trial and error, and perseverance! The example given above has successfully found both carbonyl oxygen lone pair (3.-3) critical points in \( \nabla^2 \rho \) in low-temperature data sets for formamide and acetamide.

(v) \textit{start} reads in a data file generated by a previous run of the properties program, and carries out a sequence of CP searches using the CP positions given in that file as starting coordinates. This may be useful (i) if the refinement model has been altered, but a set of CP positions have been computed for a previous model (ii) since it allows
CP positions in one property to be used as starting points for searching another property. For example, the CP’s in $V(\mathbf{r})$, which are very rapidly computed, may provide useful starting points for searching $\rho$. An example of the command would be:

```
CPSEARCH START XD_FORMAMIDE.CPS
```

where the last string is a filename.

vi) **bubble** allows a complete search of critical points around a given atom (useful especially when searching critical points of the Laplacian):

```
CPSEARCH BUBBLE C(1) rmin 0.3 rmax 0.5 curv -3 ncps 2
```

**curv** is the signature of the critical point to be searched. **ncps** is the expected number of CP’s of this type around atom ato(1). Setting **ncps** to zero allows the search to continue until finished. Enlarge **nstep** (in the heading of xdprop) if a more thorough search is needed. A file **xd_bubble.spf** is created with all CP’s found, suitable for rendering in PLATON [29].

### 6.2.17 QFIT

**QFIT grid npts length length width width constrain [true | false]**

**CONSTRAIN keys ...**

This activates the potential-derived (PD) charges algorithm, which follows the procedure described by Williams & Cox [30,31]. A cubic grid of ($npts)^3$ points is centered at the centroid of coordinates of the active molecule. This grid is of size $\text{length} \ \text{Å}$. $V(\mathbf{r})$ is computed over this grid, and then a subset of these points in a shell of thickness $\text{width} \ \text{Å}$ is selected for the fit. This corresponds to a shell of points, whose inner surface is taken as the Van der Waals radius of the nearest atom, plus the Van der Waals radius of hydrogen. The residual

$$
\chi^2 = \sum_i^m \omega_i \left[V_i^o - \sum_j^{n-1} q_j r_{ij}^{-1} + \left(\sum_j^{n-1} q_j - Z \right) r_{im}^{-1}\right]^2
$$

(Eq. 6-20)

($m$ is the number of grid points used in the fit; $V_o$ is the exact potential from the multipole model at the $i$-th grid-point; $q_j$ is the PD charge for the $j$-th nucleus; $r_{ij}$ is the distance from the $j$-th charge to the $i$-th grid point; $Z$ is the net charge on the molecule; and $\omega_i$ is the weight for the $i$-th point) is minimized by a least squares fit. In this version of **QFIT**, all grid-points have unit weights. The root-mean square fit parameter

$$
\sigma = \left[\frac{1}{m} \sum_i^m \left(V_i^o - V_i^{\text{mod} \cdot d}\right)^2\right]^{1/2}
$$

(Eq. 6-21)

is reported. Chemical symmetry constraints may be applied to the fitted charges. A full example is:

```
QFIT  grid 11 length 7.0 width 1.0 constrain true
CONSTRAIN   1 2 3 4 4 4
```

If **constrain** is **false** then the **CONSTRAIN** line need not be present. The example above, which refers to a six-atom molecule, will fit independent charges for the first three atoms, and then a single (constrained) PD charges. This might be used e.g. in formamide where the last three atoms are hydrogens. A second example is for acetamide, where the amine hydrogens (atoms 4 & 5) and methyl hydrogens (atoms 6-8) could be constrained to have equal PD charges within each group:
It should be noted that there are many different ideas and algorithms for obtaining PD charges, and the routine in XDPROP is one of the more primitive. Algorithms which additionally constrain the point-charge dipole moment to reproduce some ‘exact’ moment have been available for some time [32]. A recent idea, based on reproducing \( V(\mathbf{r}) \) at the nuclear sites, rather than in a volume around the molecule, has been developed by Su [33]. This gives charges which have some internal chemical significance for the molecule, rather than optimally predicting its intermolecular interactions. This code may be incorporated in further releases of XDPROP.

6.2.18 BPATH (Bond Path analysis)

**BPATH** atom1 atom2 algorithm (2|6) (profile)

The bond path (BP) is the line of maximum charge density joining two nuclei. It passes through the CP, and the technique for computing the BP trajectory is to carry out separate line integrations of \( \nabla \rho \) from the CP towards the two nuclei. The initial direction for this integration is determined by the CP eigenvector with the positive eigenvalue, approximately parallel to the internuclear vector. An analysis of the BP is mainly useful for detecting strain in bonds [34], since in such bonds the BP is significantly longer than the internuclear vector. Other parameters of interest are the take-off angles of the BP at the CP and the nuclei. These are angles between the BP and the internuclear vector.

There are two BP-determining algorithms in XDPROP, with different levels of accuracy. The simplest uses an Euler second-order numerical integration technique, requiring only \( \nabla \rho \) at a point to extrapolate to the next step. The second is a sixth-order algorithm, which uses a number of prior points to extrapolate to the next step, and ought to be rather more accurate. BP files may be produced, which contain the coordinates of BPs at regular intervals along the path, so that they may be plotted using the graphics software supplied in XD.

The command format is simple:

```
BPATH C(1) O(1) algorithm 2
```

computes the BPATH between C(1) and O(1) using the second-order algorithm, simultaneously producing a BP file for plotting. The more accurate sixth-order algorithm is selected by replacing the "2" with a "6" in the above example. An experimental modification of the sixth-order algorithm is also available by replacing "6" with "-6".

If the option **profile** is added to the command, a complete Bond Path Analysis [35] (\( \rho(\mathbf{r}) \); \( \nabla^2 \rho(\mathbf{r}) \) and \( \epsilon \)) is performed and reported in the file xd_profile.dat.

```
BPATH a(1) a(2) algorithm 2 profile
```

Here is an example of xd_profile.dat

```
#
# bond path 1 between atoms FE and C(1)
#
# r (A) ellip rho (e/A^3) Laplacian (e/A^5)
0.0000 0.1249 1.0201 7.0729
-0.0060 0.1275 1.0205 7.5554
-0.0110 0.1296 1.0212 7.9684
```

68
The origin is fixed at the bcp, two "legs" are printed (one for each atom). If more than one BPATH command is requested, then the following bond path analyses are simply appended.

If the option check is flagged, the xdprop output file xd_pro.out contains full analysis of each point along the bond path integration (density, Laplacian, Hessian, diagonalisation of the Hessian matrix, direction of the eigenvectors, etc.).

Bibliography

1. R.F. Stewart and M.A. Spackman, VALRAY users manual, Department of Chemistry, Carnegie-Mellon University, Pittsburgh, PA, U.S.A.
Chapter 7

XDFOUR – A General 2-D And 3-D Fourier Synthesis Program

7.1 Overview

The aspherical atom model used in multipole refinement gives structure factor phases closer to the true phases for non-centrosymmetric crystals than does the spherical or independent atom model (SPH). This permits mapping of the density by Fourier synthesis in various ways. The experimental deformation map is obtained using the calculated multipole phases with the observed structure factors \( F_o \): 

\[
\delta \rho^\text{exp} (r) = \frac{1}{V} \sum \left[ F_o(h) e^{i \phi_{mul}} - F_{sph}(h) e^{i \phi_{sph}} \right] e^{-2\pi i h \cdot r}
\]

\( F_{sph}(h) \) is computed with atomic positions and thermal parameters obtained from the multipole refinement. The dynamic model map is obtained from the calculated multipole structure factors, i.e. the Fourier coefficients are the difference of two values of \( F_c \): 

\[
\delta \rho^\text{dyn} (r) = \frac{1}{V} \sum \left[ F_{mul}(h) e^{i \phi_{mul}} - F_{sph}(h) e^{i \phi_{sph}} \right] e^{-2\pi i h \cdot r}
\]

(temperature factors are included in \( F_{mul} \) and \( F_{sph} \)). This density distribution is free of experimental noise. The use of multipole phases makes the maps slightly model-dependent; to check that all significant density features of the experimental data are included in the model we compute the residual map:

\[
\delta \rho^\text{res} (r) = \frac{1}{V} \sum \left[ F_o(h) - F_{mul}(h) \right] e^{i \phi_{mul}} e^{-2\pi i h \cdot r}
\]

For good data this should be a flat, featureless map.

Crystallographic Fourier synthesis programs (except the FFT type) compute the density distribution as 

\[
\rho(r) = \frac{1}{V} \sum h \left[ A(h) \cos 2\pi h \cdot r + B(h) \sin 2\pi h \cdot r \right]
\]

where \( A(h) + i B(h) = F(h) \). For the total density map the Fourier coefficients are 

\[
A(h) = \left| F_o(h) \right| \cos \phi_{mul} \\
B(h) = \left| F_o(h) \right| \sin \phi_{mul}
\]

For the experimental deformation map 

\[
A(h) = \left| F_o(h) \right| \cos \phi_{mul} - \left| F_{sph}(h) \right| \cos \phi_{sph} \\
B(h) = \left| F_o(h) \right| \sin \phi_{mul} - \left| F_{sph}(h) \right| \sin \phi_{sph}
\]

For the dynamic model map 

\[
A(h) = \left| F_{mul}(h) \right| \cos \phi_{mul} - \left| F_{sph}(h) \right| \cos \phi_{sph} \\
B(h) = \left| F_{mul}(h) \right| \sin \phi_{mul} - \left| F_{sph}(h) \right| \sin \phi_{sph}
\]

For the residual map
Chapter 7 - XDFOUR A General 2-D and 3-D Fourier Synthesis Program

\[
A(h) = \left| F_o(h) \right| - \left| F_{\text{mul}}(h) \right| \cos \phi_{\text{mul}} \\
B(h) = \left| F_o(h) \right| - \left| F_{\text{mul}}(h) \right| \sin \phi_{\text{mul}}
\]

XDFOUR computes a 2-D or 3-D Fourier summation on a grid oriented either with respect to a general (non-rational) plane, without interpolation errors, or with axes parallel to the crystallographic axes.

### 7.2 Files used and created by XDFOUR

**Input:** xd.mas, xd.res, xd.fou  
**Output:** xd_fou.grd, xd_fou.out

### 7.3 Input instructions for XDFOUR

#### 7.3.1 SELECT

**SELECT (*)fobs (*)fmod1 (*)fmod2 (*)print (*)snlmin snlmin (*)snlmax snlmax**

**fobs, fmod1, fmod2**

The coefficients for the summation are defined here. The reflection file contains the observed, dispersion-removed structure factors \(F_o\), and two sets of calculated structure factors. \(F_c\) may be computed by the least squares program according to various alternative density models, e.g. independent atom, multipole, static, anharmonic, and any two of these may be selected for output of the corresponding \(F_c\). The starred options in the **SELECT** line signify the type of coefficient to be used. If two are starred then the coefficients are formed from the difference of the corresponding \(F_c\) sets. If only one is starred, it forms the coefficients. In the example below, the **SELECT** line specifies a residual map with coefficients \(F_o - F_{\text{multipole}}\). In this case the \(F_c\) set labelled fmod1 has been defined in the least squares program as \(F_{\text{multipole}}\).

**SELECT *fobs *fmod1 fmod2**

Note that \(F_c\) must be calculated without anomalous dispersion (so that the scattering factors are real) as signified by zero as the third parameter after fmod1 in the XDLSM section. Fourier maps are computed without dispersion, hence it is removed from \(F_c\) also.

As a further example, one would obtain a dynamic model map by including

**FOUR fmod1 4 2 0 0 fmod2 -1 2 0 0**

in the XDLSM section, and

**SELECT fobs *fmod1 *fmod2**

in the XDFOUR section. The Fourier coefficients are then formed as the difference between the dispersion-removed multipole \(F_c\) and the dispersion-removed independent atom \(F_c\). Note that in this example no extinction correction is applied. This is signified by the final zero in the fmod1 and fmod2 options of XDLSM.

**print**

The results listing file will include the grid of density values if this option is starred. A grid file (xd_fou.grd) suitable for input to the graphical programs is always written.
Chapter 7 - XDFOUR A General 2-D and 3-D Fourier Synthesis Program

**snlmin snlmin snlmax snlmax**

These options define the sin(theta)/lambda range for which Fourier coefficients are included in the calculation. The default values are snlmin 0.0, snlmax 2.0. Note that the specified ranges are only applied if the corresponding items are starred - otherwise the default ranges are used.

### 7.3.2 APPLY

**APPLY symm is trans tx ty tz (atoms ... | all)**

This command is used to apply a symmetry operation to the pseudoatoms in the asymmetric unit. This is only used to include the additional atoms in the gridfile for plotting purposes. The symmetry operations are referenced according to the sequence in which they are listed at the start of the program output. For example,

```
APPLY SYMM 3 TRANSLATIONS -2 -1  1  O(1) N(1) C(1) H(1) H(2) H(3)
```

applies the third symmetry operation, with the lattice translations shown, to the six atoms whose labels are given. If a symmetry operation is to be applied to all atoms in the asymmetric unit, then the keyword "all" may replace the pseudoatom labels. More than one line beginning with APPLY may be present.

### 7.3.3 GRID

**GRID (*)3-points (*)perp (*)cryst**

The GRID line should specify one of the following options: **3-points**, perp, or cryst preceded by an asterisk, e.g.

```
GRID *3-POINTS PERP CRYST
```

This means that the option **3-points** is selected.

#### 3-points and perp

When either of the options **3-points** or perp are chosen, at least 3 points must be specified. There are 3 types of formats.

1. **ATOM label atom (symm symop) (trans tx ty tz) (*)mark**
   
   atom must be identical to an atom label given in the parameter file.

2. **ATOM no no (symm symop) (trans tx ty tz) (*)mark**
   
   no is the sequence number of an atom in the parameter file.

3. **XYZ (label label) x y z (symm symop) (trans tx ty tz) (*)mark**

   label is a label which may marked on the plot. The fractional coordinates x y z are free format real numbers.

Common for the three formats are the options:

- **symm symop** where symop is the sequence number of an operation in the list of space group operations which appears at the start of the program output.
- **trans tx ty tz** indicating three lattice translations (positive or negative integers).
- **(*)mark** If flagged with a star, the position is marked on the plot.
The first 3 points are used to define a right-handed orthonormal coordinate system in the following way. The origin of this coordinate system is the centroid of the three points in the list. For the option 3-points, the points define the $xy$ plane; the $x$-axis is parallel to the vector from point 1 to point 2 and the third point is in the half-plane $y>0$. For the option perpendicular, the $xy$ plane is perpendicular to the vector from point 1 to point 2; the projection of the third point onto the $xy$ plane defines the direction of the $x$-axis, $x>0$.

cryst

In this case the grid has oblique axes parallel to the crystallographic axes $a$, $b$ and $c$. Of the commands described below, only LIMITs, TRANslate and PERMute should be used in this case.

7.3.4 TRAN

Having defined a first orthonormal coordinate system, we may define the final grid-coordinate system by any of the following operations on it, in any order, and as many of them as you like. The operations are:

- Translation of the coordinate system origin, command TRAN
- Rotation around axes through the origin, command ROTA
- Permutation of the axes, command PERM

As soon as a line is read, the operation is performed on the coordinate system, and the next operation acts on this new coordinate system with respect to its axes. The command for translation is

TRAN $tx$ $ty$ $tz$

The interpretation of $tx$ $ty$ $tz$ depends on the option chosen on the GRID line:

- *3-points*: $tx$ $ty$ $tz$ are in Ångstroms
- *perp*: $tx$ $ty$ are in Ångstroms, $tz$ in fractions of the vector from point 1 to point 2.
- *cryst*: $tx$ $ty$ $tz$ are in fractional coordinates

7.3.5 ROTATE

ROTA eulerian $alpha$ $beta$ $gamma$

ROTA axis angle

$alpha$, $beta$, $gamma$ are the Eulerian rotation angles: first a rotation of $alpha$ degrees about the $z$-axis, then $beta$ degrees about the new $y$-axis, and finally $gamma$ degrees about the new $z$-axis.

axis: $x$, $y$ or $z$

angle equals the angle (degrees) which the coordinate system is rotated about the coordinate axis given by axis. An example: the result of these four instructions is to leave the coordinate system unchanged.

ROTA EULERIAN  45 -54.5 90
ROTA  Z  -90
ROTA  Y  54.5
ROTA  Z  -45.

7.3.6 PERM

PERM new-$x$ new-$y$ new-$z$

new-$x$, new-$y$, new-$z$: some permutation of $x$, $y$ or $z$. An example: How to generate a left-handed system by turning $z$ into -$z$? Here is one way of doing it:
Chapter 7 - XDFOUR A General 2-D and 3-D Fourier Synthesis Program

ROTAtion Y 90
PERMute Z Y X

7.3.7 LIMITS

This command defines the limits of summation. A 3-dimensional grid is obtained when all three axes have more than one grid point.

**LIMI (keyword value) ...**

Where *keyword* is any of the following (default values in brackets):

```
xmin [0.] xmax [1.] nx [41]
ymin [0.] ymax [1.] ny [41]
zmin [0.] zmax [0.975] nz [40]
```

?min and ?max are the limits of the box dimensions along the respective coordinate axis. n? is the number of grid points in this direction. It is stressed that the limit information is only applied to the final grid-coordinate system. Example: a 2-D grid is defined. The sense of the y-axis is inverted since ymin>ymax.

LIMITS XMIN -2. XMAX 2. NX 41
LIMITS YMIN 2. YMAX -1. NY 31
LIMITS ZMIN 0. ZMAX 0. NZ 1

Default values assumed by the program divide the cell up into 40 sections, each having \(41 \times 41\) points.
Chapter 8

XDFFT - A Fast Fourier Transform program

8.1 Overview

XDFFT is a 3-D fast Fourier Transform program operating over the whole unit cell, and using the algorithm and code of Ten Eyck [1]. It includes a peak searching routine, and is adapted from the GX programs FFT and SEARCH [2]. Since the calculation time scales as $N \log N$ rather than $N^2$, it is at least an order of magnitude faster than a corresponding calculation using XDFOUR. It will probably be most useful for determining the extrema of the residuals after refinement, or the extrema of the deformation density.

8.2 Files used and created by XDFFT

Input: xd.mas, xd.res (or xd.inp), xd.fou
Output: xd_fft.out, xd_fft.cif (also xd_fft.grd, xd_fft.pks)

8.3 Input instructions for XDFFT

8.3.1 SELECT

SELECT *fobs *fmod1 fmod2 (*)snlmin snlmin (*)snlmax snlmax (*)sig sigcut (*)phase phasecut

fobs, fmod1, fmod2

The selection of these coefficients for the Fourier calculation is exactly as described for XDFOUR (see Chapter 7). The default calculation is a difference Fourier. If a difference Fourier is selected, then the program writes out a CIF called xd_fft.cif, containing the maximum and minimum and RMS residual densities.

snlmin snlmin snlmax snlmax

These options define the $\sin(\theta)/\lambda$ range for which Fourier coefficients are included in the calculation. The default values are $snlmin \ 0.0$, $snlmax \ 2.0$. Note that the specified ranges are only applied if the corresponding items are starred - otherwise the default ranges are used.

sig sigcut phase phasecut

These options define cut-offs for which Fourier coefficients are included in the calculation. Only those reflections with $|F_o|/\sigma(F_o) > sigcut$, and with $|F_c| > phasecut |F_o|$ will be included. The default values are $sigcut \ 3.0$, $phasecut \ 0.0$. Note that the cut-offs are only applied if the corresponding items are starred - otherwise the default cut-offs are used.
SELECT gridsize gridsize scale scale npeak peaks nhole holes (*)neutron (*)gridf
(*)peakf

gridsize gridsize

The grid spacing in Angstroms. The default value is 0.2 Å. The maximum number of grid points in any direction is 200. If the chosen grid spacing results in more than 200 points along any axis, the spacing is automatically increased by the program.

scale scale

The electron density scale factor. Currently not in use.

npeak peaks / nhole holes

The number of peaks and holes required in the peak searching routine. Default values are 10 for both. Input atomic positions are read from the xd.res (or xd.inp) files, and all details and interpretation of the map is listed in the file xd_fft.out.

neutron

If this item is starred, then holes will also be included in the peaks interpretation. The default is not to include holes in the interpretation.

gridf

If this item is starred, then an XD gridfile xd_fft.grd is written. Users should note that the planes are always calculated along the y direction, with the z direction varying fastest, which is different from XDFOUR. Since this file can be very large and is probably of little use, the default action is not to write a gridfile.

peakf

If this item is starred, the peaks selected by the search routine are written to the file xd_fft.pks.

Bibliography

Chapter 9

XDGRAPH - Visualising the Results

9.1 Overview

The graphics program differs from the rest of the package in one major way. To account for its interactive nature, it is not driven by the master file, but instead is controlled by a command language. The Tool Command Language (Tcl, by J.K.Ousterhout) was chosen because it provides a general scripting language in which special application-defined commands are easily integrated. The Tcl based toolkit (Tk) for the X11 Window System was then used to add a graphical user interface on top of the existing commands.

9.2 The Command Line Interface

XDGRAPH roughly follows the concept of Tk with its commands. For each type of high-level "object" one can work with (examples of objects are datasets, contour levels, etc.), a command exists to create this object (e.g. dataset, contour). This, in turn, creates a new command with the same name as the object. Actions on the object (apart from creation) are performed using that "object command". The different actions available for an object are called subcommands.

The name of an object must follow certain rules:

- All names start with a ":" (colon).
- Objects that are derived from other objects (i.e. from datasets) start with the name of that object, separated by a colon.

For example a valid name for a dataset would be :set1, and a possible name for a contour level group derived from that dataset would be :set1:plus. In addition to the "object oriented" commands, more "action oriented" commands are available. They are mostly implemented as Tcl-procedures.

XDGRAPH distinguishes between the creation of an object and the actual graphics output. The latter is considered an action you perform on this object by using the subcommand plot of the object.

So the general scheme to create a plot looks like this:

Load the data you want to visualize from a file

dataset :set001 -load xd_defden.grd

Create some graphical objects

contour :set001:plus -val {0.1 0.2 0.3 0.4 0.5}
contour :set001:zero -val 0. -style dash
molecule :set001:mol

Display those objects

:set001:plus plot
:set001:zero plot
:set001:mol plot
9.3 The Graphical User Interface

The graphical user interface to XDGRAPH reflects the command structure described in the previous section. A dialog box exists for each object which allows to enter the options the respective command takes. Online help, access to reasonable default values, menu lists, file and colour browsers support the user entering required data.

Figure 9-1 : A sample dialog box from the graphical user interface

Here is the example from the previous section, this time using the GUI:
(Menu entries are written as 'Menu::Submenu' )

Load the data you want to visualize from a file
Choose File::Load Dataset and fill in the necessary fields. You can get support to enter the file name from a directory browser. Which file selector is actually used depends on the Tcl/Tk release. Starting with 8.0, the internal file selector which comes with Tk is used. For older versions XD’s own selector is used. You can force its use by setting the environment variable XD_USE_PRIVATE_FSBOX. A list of allowed file types is available as a menu. Use ‘Ok’ to finish this step.

Create and plot a simple graphical object
Choose Create::Contour. Again, a dialog box appears. See Figure 9-1 for details. Use ‘Plot’ to finish this step.

Create and plot the second group of contour lines
Same procedure as above. Open the ‘Style Options’ part of the dialog box and use the ‘Linestyle’ menu to specify dashed lines.

Finally, add a representation of the molecule to the graphics
Use Create::Molecule. Again, use ‘Plot’ to finish this step.
9.4 Running XDGRAPH

SYNOPSIS

xdgraph (options) (tcl-script)

OPTIONS

-d driver
    select a driver (tk or gt)

9.5 Commands

Some of the options and subcommands are marked with a star (*). They are common to
more than one command. Their descriptions can be found in Sections 9.6 & 9.7.

9.5.1 dataset

Create a dataset or list existing datasets.

SYNOPSIS

dataset ?dataSetPattern?

dataset name -load fileName options

dataset name -slice sourceDataSet options

DESCRIPTION

This command has different uses, depending on the number of arguments. The first form of
the command, with at most one argument, returns a list of existing datasets. If no pattern is
specified all datasets are listed, otherwise only those matching the given pattern are
returned. A list of special character sequences for pattern matching can be found in Section
9.11.3

The second form of the dataset command, which requires at least two arguments, creates a
dataset. This can either be done by reading a file (requires -load as the first option) or by
interpolating another dataset (if -slice is given as the first option). In either case, the first
argument is the name of the new dataset.

The following set of options is available for the file reading version of the dataset command:

OPTIONS

-load fileName
    The name of the file to be read. The GUI provides a file selector box for Unix users.
    Note for VMS users: Please read the section about file name syntax in Tcl in Section
    9.11.4. Remember that square brackets [] and dollar signs have special meanings in
    Tcl.

-type fileType
    The format (and to some extent the content type) of the file to be read. Valid file types
    are:
    aim
        A grid file format as used by some versions of the AIMPAC package.
    xddata
        This is used for xy-diagrams. It contains a list of arbitrary data points, with
        possibly multiple values per point. Still experimental!
**xdgrid**

Grid files contain data on a rectangular grid, together with a list of objects (atoms and critical points). Either 2- or 3-dimensional grids are possible. XDPROP and XDFour write grid files using this format.

**xdpath**

This type of file is written by XDPROP if a bond path calculation with algorithm 2 was done.

The following set of options is only available for the slicing version of the `dataset` command. This option is not yet available in the GUI.

**OPTIONS**

- `slice` sourceDataSet
- `3point*` p1 p2 p3
- `nx` numXPoints
- `ny` numXPoints
- `nz` numXPoints
- `xmin` minXCoord
- `xmax` maxXCoord
- `ymin` minYCoord
- `ymax` maxYCoord
- `zmin` minZCoord
- `zmax` maxZCoord

**SUBCOMMANDS**

configure* options

connections –auto

  Generate a list of connections between atoms based on their distance.

contour

  Returns a list of all contour level groups created from this dataset.

heightfield

  Returns a list of all height field objects created from this dataset.

info*

  Output some information about the dataset.

interpolate

  *Only available for slices.* Re-calculate the data values by interpolating in the original dataset given when the slice was created. This subcommand must be used after `rotate` and `translate` subcommands.

isosurface

  Returns a list of iso-surface objects created from this dataset.

molecule

  Returns a list of molecule objects created from this dataset.

path

  Returns a list of bond path objects created from this dataset.

property

  Returns the property this dataset maps.

relief

  Returns a list of relief objects created from this dataset.

remove*

  Remove this dataset.

rotate options

  *Only available for slices.* Rotate the slice. Note, that this command does not re-calculate the data values. You have to call the `interpolate` subcommand explicitly. Different options are available:

  - `eulerian` angle1 angle2 angle3
Chapter 9 - XDGRAPH - Visualising the Results

-x angle
-y angle
-z angle
translate \( \{x \ y \ z\} \)

Only available for slices. Translate the slice. Note that this command does not recalculate the data values. You have to call the **interpolate** subcommand explicitly.

**xydiagram**

Returns a list of xy-diagram objects created from this dataset.

**colorbg** min max

Draw a filled rectangle on each grid point. The colour is chosen by mapping the given range (defaults to the whole range of values in the grid) to 256 colours in a colour map (Currently the colour map can't be changed by the user. A fixed map $xd_datadir/default.cmap is used.Values outside the given range are not filled.

**EXAMPLES**

> dataset :ox -load xd_drho.grd
> dataset < :ox

9.5.2 contour

This is used to visualize data on a 2-dimensional grid by drawing smooth lines connecting points of equal value. XDGRAPH handles groups of lines for different data values together as single objects. To get a complete contour map you usually create a few groups with different style options to distinguish different data ranges (for example positive and negative values.)

**SYNOPSIS**

```
contour name ?options?
```

**OPTIONS**

**Specific Options**

-**plane*** plane

This is used to select a plane from a 3-dimensional grid.

-**values** valueList

A list of values, for which contours are to be drawn. When used with a **configure** subcommand, the list overwrites the previous one, while when used with **append** the new values are appended to the existing list (appended, not merged!). Note that this is a 'list' in the Tcl sense:

```
contour :d1:clg -values  0. 1. 2.   is not correct !!
contour :d1:clg -values {0. 1. 2.}       this is the correct usage using curly braces {}
```

-vertac vertAcc

Vertical accentuation - this scales the data values to z coordinates. It is useful to prevent clipping and when adding contour lines to a height field. Currently only used when the OpenGL driver is in effect.

-view view

The view this contour group should use. See Section 9.5.9 for details about the default behaviour, which should be reasonable for most simple applications.

**Style Options**

-**foreground** | -**fg** colour | {colour1 colour2}.

This option takes either a single colour (see Section 9.6 on different ways to specify a colour) or a list consisting of two colour values. When two colours are given, the first one is assigned to the first contour level and the second one to the last contour level

81
in the group. Intermediate values are interpolated. Note that the way the colours are
specified influences intermediate colours. (See Section on common options).

- **style** lineStyle
  The line style (solid, dotted, etc.) used for this contour group.

**View Options**

See Section 9.6 for a list of possible options.

**SUBCOMMANDS**

- **configure** options
  Change options for an existing contour line group. See previous section for a list of
  possible options.

- **append** options
  This is like **configure**, except that any **-values** given with **append** are added to the
  contour group, while those given with **configure** overwrite the previous list.

- **clear**
  Remove the contour group from its view.

- **info**
  Output some information about the contour group.

- **plot**
  Display this contour group, adding it to its view.

- **remove**
  Delete the contour group.

**9.5.3 height field**

This is used to visualize data on a 2-dimensional grid by drawing an open surface on the
grid which is elevated according to the data values. Creates a smooth surface from data on a
rectangular grid where the height corresponds to the data values. *Only available with OpenGL.*

**SYNOPSIS**

```
heightfield fieldheightField ?options?
```

**OPTIONS**

**Specific Options**

- **-cutoff** cutoff | (lowCut highCut)
  Limits the maximum and minimum elevation. If only one value is given, it specifies
  the upper cutoff limit. The lower cutoff value is - cutoff in this case.

- **-map** name
  For future use.

- **-plane** n
  This is used to select a plane from a 3-dimensional grid.

- **-vertac** vertAcc
  This scales the data values to z coordinates. Reasonable values depend on the
  mapped property.

- **-view** view
  The view this contour group should use. See Section 9.5.9 for details about the
  default behaviour, which should be reasonable for most simple applications.

**Style Options**

- **-foreground** | -**fg** colour | (colour1 colour2)
  The colour of the surface, changed by lighting calculations. Currently only one colour
  is used.
Chapter 9 - XDGRAPH - Visualising the Results

-polygon* polygonMode

View Options
See section 9.6 for a list of possible options.

SUBCOMMANDS
configure* options
Change options for an existing height field object. See previous section for a list of possible options.
clear*
Remove the height field from its view.
info*
Output some information about the height field.
plot*
Display this height field, adding it to its view.
remove*
Delete the height field.

9.5.4 iso-surface
This is used to visualize data on a 3-dimensional grid by drawing smooth surfaces connecting points of equal value. Surfaces are represented by triangles which can be rendered using solid planes, lines or points. The later two options make it possible to see surfaces inside of one another. Only available with OpenGL, i.e. not available in the Windows version of XDGRAPH.

SYNOPSIS
isosurface name ?options?

OPTIONS
Specific Options
-values valueList
A list of values for which iso-surfaces are to be drawn. When used with a configure subcommand, the list overwrites the previous one, while when used with append the new values are appended to the existing list (appended, not merged!).
-view view
The view these iso-surfaces should use. See Section 9.5.9 for details about the default behaviour, which should be reasonable for most simple applications.

Style Options
-foreground | -fg* colour
The colour used two draw this iso-surface.
-style* lineStyle
The line style (solid, dotted, etc.) used in case the polygonMode is set too line.
-polygon* polygonMode
How to render this iso-surface - fill (solid), line (lines) or point (points)

View Options
See Section 9.6 or a list of possible options.

SUBCOMMANDS
configure* options
Change options for an existing iso-surface. See previous section for a list of possible options.
append options
   This is like configure, except that any -values given with append are added to the iso-surfaces, while those given with configure overwrite the previous list.

clear*
   Remove the iso-surface from its view.

info*
   Output some information about the iso-surface.

plot*
   Display this iso-surface, adding it to its view.

remove*
   Delete the iso-surface.

9.5.5 molecule

The molecule as read from a grid or bond path file is visualized according to the view type used. For a contour or bond path view a line drawing is used, while for an iso-surface view a ball-and-stick model is used.

SYNOPSIS
molecule name ?options?

OPTIONS

Specific Options

-Atoms drawAtoms
   Include or exclude atoms from the display. drawAtoms is a boolean (on, off, yes, no).

-Bonds drawBonds
   Include or exclude bonds from the display. drawBonds is a boolean

-Label drawLabels
   Include or exclude labels from the display. drawLabels is a boolean

-View view
   The view this molecule should use.

-Zlimit zlimit
   Exclude atoms further away from the plane than zlimit Angstrom.

Style Options
   Not yet implemented.

View Options
   See section 9.6 for a list of possible options.

SUBCOMMANDS

configure* options
   Change options for an existing molecule. See previous section for a list of possible options.

clear*
   Remove the molecule from its view.

info*
   Output some information about the molecule.

plot*
   Display this molecule, adding it to its view.

remove*
   Delete the molecule.
9.5.6 path

This is used to visualize data from a bond path calculation.

SYNOPSIS
path name ?options?

OPTIONS

Specific Options
-view view
The view this bond path plot should use. See section 9.6 for a list of possible view options.

Style Options
-foreground | -fg* colour
The colour used for this bond path plot.
-style* lineStyle
The line style (solid, dotted, etc.) used for this bond path plot.

SUBCOMMANDS
configure* options
Change options for an existing bond path plot. See previous section for a list of possible options.
clear*
Remove the bond path plot from its view.
info*
Output some information about the bond path plot.
plot*
Display this bond path plot, adding it to its view.
remove*
Delete the bond path plot.

9.5.7 relief

Create a relief plot, i.e. the data is visualized as view of a landscape, using the value on each grid point as its height. The transformation into the display plane is chosen by giving a viewpoint. Currently, a number of restrictions apply:
- you can’t select a plane from a 3-dimensional grid and
- you can’t choose in which direction lines are drawn (currently always along x and y).

SYNOPSIS
relief name ?options?

OPTIONS

Specific Options
-cutoff cutoffValue | {highCutoff lowCutoff}
When used with one value, this option limits the absolute value of any data point to cutoffValue. When a list with two values is given, the high and low cutoff values can be given separately.
-eye [xyz]
The eye-point is a point in 3d-space from where the relief is viewed. The viewer is always looking across the map to the corner 0., 0., 0. This is a parallel projection, so only the ratio of the three numbers is used. The default value is (1. 1. 0.6).
-size \{hSteps vSteps\}
  The number of lines to draw parallel to x and y, respectively. Defaults to the number of grid points.

-vertac scaleFactor
  Gives a scale factor from data values to y-plot coordinates. This defaults to fitting the data to the plot size. Note this option is probably required if any of the cutoff options were used.

-view view
  The view this contour group should use. See Section 9.5.9 for details about the default behaviour, which should be reasonable for most simple applications.

Style Options

-foreground | -fg* colour
  The colour used for this relief plot.

-style* lineStyle
  The line style (solid, dotted, etc.) used for this relief plot.

View Options

See section 9.6 for a list of possible options.

SUBCOMMANDS

configure* options
  Change options for an existing relief plot. See previous section for a list of possible options.

clear*
  Remove the relief plot from its view.

info*
  Output some information about the relief plot.

plot*
  Display this relief plot, adding it to its view.

remove*
  Delete the relief plot.

9.5.8 xydiagram

Create an xy-diagram, experimental.

SYNOPSIS
xydiagram name ?options?

OPTIONS

Specific Options

-type point | line

-view
-x i | varName
-y i | varName

View Options

See Section 9.6 for a list of possible options.

SUBCOMMANDS

configure* options

-configure* options
  Change options for an existing relief plot. See previous section for a list of possible options.

clear*
  Remove the relief plot from its view.

info*
  Output some information about the relief plot.

plot*
  Display this relief plot, adding it to its view.

remove*
  Delete the relief plot.
9.5.9 view

A view is used to create a connection between graphical objects (like contour lines or molecules) and the screen. It is an abstract object which manifests itself as an X11 window.

name: The name of the view object to create. Please note, that view names do not follow the rules for other objects in XDGRAPH. They do not have to start with a colon and they are not derived from any object.

type: The type of the view corresponds to the kind of objects shown. There is no separate view for molecules. Molecules may use any of the other view types (except \texttt{xydiag} and \texttt{relief}). The way molecules are represented depends on the type of view. E.g. in a \texttt{contour} view it is a simple line drawing whereas in a \texttt{surface} view a 3D ball and stick model is used.

Whenever you create a graphical object using a non-existing view (explicitly or implicitly) a view of the appropriate type is created automatically. Its name is derived from the dataset the object belongs to and the type of the object. For molecules a contour view is used. If you want to draw a molecule in another type of view, you must specify the name explicitly using the \texttt{-view} option. You can either create the view by hand or create the other object first and use the automatic name.

The following view types are available:

\begin{itemize}
  \item \texttt{contour}
  \item \texttt{height}
  \item \texttt{path}
  \item \texttt{relief}
  \item \texttt{surface}
  \item \texttt{xydiag}
\end{itemize}

SYNOPSIS

\texttt{view name -type type ?options?}

OPTIONS

View Options
See Section 9.6 for a list of possible options.

SUBCOMMANDS

\begin{itemize}
  \item \texttt{configure* options}
    Change options for an existing view. See previous section for a list of possible options.
  \item \texttt{matrix}
    Print, the 4 by 4 transformation matrix. This can be used to restore an orientation obtained using the mouse. This is currently implemented as \texttt{ddr}
  \item \texttt{objects}
    Returns a list of objects connected to this view.
  \item \texttt{rotate options}
    Rotate the view. \texttt{Not yet implemented}!!!
    \begin{itemize}
      \item \texttt{-x angle}
      \item \texttt{-y angle}
      \item \texttt{-z angle}
    \end{itemize}
  \item \texttt{translate \{x y z\}}
    Translate the view. \texttt{Not yet implemented}!!!
\end{itemize}
9.6 Common Options

View Options

These options are available for all graphics objects as well as for views. When used with graphics objects, they are applied to the related view, however. You cannot transform one object separate from another in the same view using these options.

Size and Scaling

- **-width width**
- **-height height**
- **-scale scaleFactor | auto**

The first two options set the size of the view (in cm). If negative or not specified, this is calculated from *vrange* and *scaleFactor*. If no scale factor is given either, a default of 18cm is used. The scale factor is used to transform Ångstrom to cm, a value of one meaning that 1cm in the plot corresponds to lÅ in the data. If the scale factor is less than or equal to zero or specified as *auto*, the data is scaled to fit into the view. This requires -vrage to be specified.

If all three options are given, the view might not be fully used or clipping might occur.

- **-vrange {xrange yrange}**
  For a grid file, this gives the length of the x and y axis of the grid (in Å).

Translation and Rotation

- **-matrix matrix**
  Transformation matrix, *matrix* is either a 3 by 3 or a 4 by 4 transformation matrix. This is mainly useful to restore a matrix from a previous run. (See section 9.5.9).
  **This option is not yet fully implemented !!!**
- **-origin {x y z}**
  Shift the objects before rotation.
- **-translate {x y z}**
  Shift the plot after rotation but before scaling. (So this is in Ångstrom.)

- **-3point p1 p2 p3**
  An easy way to give origin and orientation. \( \vec{p}_i \) is the origin. \( \vec{p}_2 - \vec{p}_1 \) gives the positive x axis \( \vec{x} \). The z axis \( \vec{z} \) is given by \( \vec{x} \times (\vec{p}_3 - \vec{p}_1) \), and \( \vec{z} \times \vec{x} \) is the y axis. Each of the points may be specified in one of the following ways:
  \( \{p_1, p_2, p_3\} \)
  A list, giving the coordinates directly.
  **label**
  The object label.

Style Options

- **-foreground or -fg colour**
  Sets the foreground colour(s) used to draw an object. The colour can be given in a number of ways.
  - **name**
    XD’ s own database is used to convert colour names to RGB values when necessary, (i.e. not for Tk)
  - **RGBtriple**
    X Window System style #rgb, #rrggbb #rrrgggbbb #rrrrggggbbbb with ‘r’, ‘g’, ‘b’ being hex digits
Chapter 9 - XDGRAPH - Visualising the Results

{RGB red green blue}
    red, green, blue: [0., 1.]

{HSV hue saturation value}
    hue: [0..360.), 0. is red, saturation: [0.. 1.], value: [0..1.]

{YUV luminance u v}
    luminance, u, v: [0..1.], u, v: [-0.5,0.5]

-polygon polygonMode
    How to draw polygons. One of the following:
    solid
        Draw solid, shaded faces.
    line
        Draw the shaded edges of the polygons.
    point
        Draw only the vertices of the polygons.

-style lineStyle
    Sets the line style. lineStyle is either the keyword solid or a string build from the following elements.
    
    " " (blank)
        Empty space. You can use multiple blanks to add more space. For example the string
    "dot  dot dot" would draw a line with three dots close to each other, separated by a
        larger space. If there are no trailing blanks given, a single one is added automatically.
    dot
        A dot.
    dash
        A dash.
    long
        A long dash.

    Valid examples are: ' solid' (default), ' dot' or ' dot dash'.

Other Common Options
-plane n
    This is used to select a plane from a 3-dimensional grid. For those grids, an xy-plane
    is plotted and the parameter n selects the nth section along the z-axis. The first
    plane is numbered 1, which is also the default value for this option. Rarely used.

-view view
    The view the object will use. See Section 9.5.9 for details about the default behaviour,
    which should be reasonable for most simple applications. This option is only
    meaningful while objects are created. Currently, the behaviour when used with the
    configure subcommand is undefined.

9.7 Common Subcommands

configure options
    Change options for an existing object. The options that can be used are the same as
    for the command that is used to create the respective object type.

info
    Show some information about the object.

plot
    Plot the object.

clear
    The object is removed from the display

remove
    Delete the object. The object command is removed and the associated memory is
    released.
view
Return the view this object is using.

9.8 Toolbox

9.8.1 sleep
Sleep (do nothing) for seconds seconds.

SYNOPSIS
sleep seconds

9.8.2 plot
Plot all objects derived from the given dataset(s). Default are all datasets.

SYNOPSIS
plot ?datasets?

9.8.3 clear
Clear all plotted objects derived from the given dataset(s). Default are all datasets.

SYNOPSIS
clear ?datasets?

9.8.4 hardcopy
Dump the contents of the currently active display to file. The actual semantics of this command depend on the display driver in use.
When used with the Tk driver, the output will be a PostScript file. Width and height are standard Tk measures, they default to the size of the window. The default filename is xdgraph.ps.

Unfortunately, the OpenGL driver is only capable of outputing pixel oriented data. Currently, the file is written as a PBM file (Portable BitMap file). Conversion utilities to other pixel file formats are available from many ftp servers around the world. The default file name is xdgraph.ppm. Width and height are given in pixels. The default of 500 is only suitable for test purposes. The optimal value depends on your printer and the type of graphics shown. Start with values around 1500 for serious work. Use the Tk driver for line drawings such as contour maps. The PostScript output is much better suited for this purpose.

Linux users may find that the hardcopy option for the OpenGL driver is not functioning (an error message about no "visual for dump" is given). In this instance, hardcopy may be obtained with the Linux utility program import, by screen-grabbing the window. The image may be saved in several formats include PostScript and GIF, and may also be resized. See "man import" for further details of command syntax for import. The command line is given in any terminal window and the actual image is grabbed by then clicking on the window displaying the desired graphic with the mouse. For example, to save as a PostScript file
import  image.ps <CR> then click desired window with mouse
or to save as a GIF with 150% expansion in image size
import image.gif -geometry 150% <CR> then click desired window with mouse
Chapter 9 - XDGRAPH - Visualising the Results

SYNOPSIS

hardcopy ?-file filename? ?-width width? ?-height height?

9.8.5 generate

Return a list of values. Very useful where values for contour levels have to be given. type
maybe one of

lin
  Create a linear range of nsteps values, starting with start, adding increment.

geo
  Create a geometric range of nsteps values, starting with start, multiplying by
  increment.

aim
  { .001 .002 .004 .008 .04 .08 .2 .4 .8 2. 4. 8. 20. 40. 80. 200. 400. 800.}

maim
  As aim, but with negative sign.

SYNOPSIS

generate type ?nsteps ?start ?increment???

EXAMPLES

contour :d1:plus -values [generate lin 10 0.1 0.1]
  is the same as
contour :d1:plus -values {0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0}

9.9 Display Driver

Currently, XDGRAPH is interfaced to two different libraries for the actual display/plotting.
One is the X11 Window System toolkit Tk already mentioned. It is the preferred display
driver on any platform with X11 available (note that this includes VMS systems with
DECwindows). It requires the installation of Tk.

The other driver uses the OpenGL (or the Linux-clone Mesa) libraries. This driver is not
available under Windows.

The Tk driver has some special features:

- The atomic labels in contour and path plots can be moved using the mouse. Move the
cursor over the text and wait until the colour changes. Than press the left mouse button
and drag the label over the display. Release the mouse button when you are satisfied
with the placement. This does not move the marker of the atom, just the label.

9.10 Examples

The following examples can be found in the source directory $TOP/xdgraph/examples.

9.10.1 bw.tcl

This creates a contour plot from the data in file xd_defden.grd. The dataset is called :1.
For positive and negative contour levels the generate command is used. The -style option
is used to draw the zero level and negative levels with a different line style. The command
plot is used to plot the three contour level groups together with a title, atom labels and some
further info.

dataset   :1  -load xd_defden.grd
contour :1:plus -val [generate lin 10 0.1 0.1]
contour :1:zero -val 0. -style dash
contour :1:minus -val [generate lin 10 -0.1 -0.1] -style dot
plot
This creates a contour plot with coloured contour lines. Because two different colours are given for positive and negative contour level groups each contour level has a different colour. The YUV colour model is used to specify the colours because it is especially useful for interpolation. Unfortunately, it is not easy for humans to relate a specific colour to an YUV triple.

```tcl
dataset :1 -load xd_defden.grd
contour :1:plus -val [generate lin 15 0.1 0.1] \ 
-foreground {{YUV 0.6 0.6 0.1} {YUV 0.1 0.7 0.44}}
contour :1:zero -val 0. -style dash
contour :1:minus -val [generate lin 10-0.1-0.1] \ 
-fore {{YUV 0.9 0.2 0.5} {YUV 0.2 0.4 0.88}}
plot
```

**9.11 A few words about Tcl**

**9.11.1 Syntax**

The following list is derived from the Tcl man page. It gives almost all rules that define the syntax of the Tcl language. The examples mainly make use of two of the features: Quoting strings with curly braces ({} ) and command substitution with square brackets ([ ] ), which executes the enclosed string and substitutes it with the result.

- A Tcl script is a string containing one or more commands. Semi-colons and newlines are command separators unless quoted as described below. Close brackets are command terminators during command substitution (see below) unless quoted.

- A command is evaluated in two steps. First, the Tcl interpreter breaks the command into *words* and performs substitutions as described below. These substitutions are performed in the same way for all commands. The first word is used to locate a command procedure to carry out the command, then all of the words of the command are passed to the command procedure. The command procedure is free to interpret each of its words in any way it likes, such as an integer, variable name, list, or Tcl script. Different commands interpret their words differently.

  - Words of a command are separated by white space (except for newlines, which are command separators).

  - If the first character of a word is double-quote (" ") then the word is terminated by the next double-quote character. If semi-colons, close brackets, or white space characters (including newlines) appear between the quotes then they are treated as ordinary characters and included in the word. Command substitution, variable substitution, and backslash substitution are performed on the characters between the quotes as described below. The double-quotes are not retained as part of the word.

  - If the first character of a word is an open brace ("{") then Tcl performs command substitution. To
do this it invokes the Tcl interpreter recursively to process the characters following the
open bracket as a Tcl script. The script may contain any number of commands and
must be terminated by a close bracket ("]"). The result of the script (i.e. the result of
its last command) is substituted into the word in place of the brackets and all of the
characters between them. There may be any number of command substitutions in
a single word. Command substitution is not performed on words enclosed in
braces.

• If a word contains a dollar-sign ("\$") then Tcl performs variable substitution: the
dollar-sign and the following characters are replaced in the word by the value of a
variable.

• If a backslash ("\") appears within a word then backslash substitution occurs. In all
cases but those described below the backslash is dropped and the following
character is treated as an ordinary character and included in the word. This allows
characters such as double quotes, close brackets, and dollar signs to be included in
words without triggering special processing. The following table lists the backslash
sequences that are handled specially, along with the value that replaces each
sequence.

<table>
<thead>
<tr>
<th>Backslash</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>\a</td>
<td>Audible alert (bell) (0x7).</td>
</tr>
<tr>
<td>\b</td>
<td>Backspace (0x8).</td>
</tr>
<tr>
<td>\f</td>
<td>Form feed (0xc).</td>
</tr>
<tr>
<td>\n</td>
<td>Newline (0xa).</td>
</tr>
<tr>
<td>\r</td>
<td>Carriage-return (0xd).</td>
</tr>
<tr>
<td>\t</td>
<td>Tab (0x9).</td>
</tr>
<tr>
<td>\v</td>
<td>Vertical tab (0xb).</td>
</tr>
<tr>
<td>&lt;newline&gt;</td>
<td>newline</td>
</tr>
<tr>
<td>\vwhitespace</td>
<td>A single space character replaces the backslash, newline, and all white space after the newline. This backslash sequence is unique in that it is replaced in a separate pre-pass before the command is actually parsed. This means that it will be replaced even when it occurs between braces, and the resulting space will be treated as a word separator if it is not in braces or quotes.</td>
</tr>
<tr>
<td>\</td>
<td>Backslash (&quot;&quot;).</td>
</tr>
<tr>
<td>\ooo</td>
<td>The digits {one, two, or three of them} give the octal value of the character.</td>
</tr>
<tr>
<td>\xhh</td>
<td>The hexadecimal digits hh give the hexadecimal value of the character. Any number of digits may be present.</td>
</tr>
</tbody>
</table>

Backslash substitution is not performed on words enclosed in braces, except for
backslash-newline as described above.

• If a hash character ("#") appears at a point where Tcl is expecting the first character
of the first word of a command, then the hash character and the characters that
follow it, up through the next newline, are treated as a comment and ignored. The
comment character only has significance when it appears at the beginning of a
command.

• Each character is processed exactly once by the Tcl interpreter as part of creating the
words of a command. For example, if variable substitution occurs then no further
substitutions are performed on the value of the variable: the value is inserted into
the word verbatim. If command substitution occurs then the nested command is
processed entirely by the recursive call to the Tcl interpreter: no substitutions are
performed before making the recursive call and no additional substitutions are
performed on the result of the nested script.

• Substitutions do not affect the word boundaries of a command. For example, during
variable substitution the entire value of the variable becomes part of a single word,
even if the variable’s value contains spaces.
9.11.2 Some built-in commands

A very handy Tcl command is `source`. This is used to begin reading further commands from a file, switching back to stdin at the end of the file. For example, you could copy the file bw.tcl (see above) into your current directory and use it by typing "source bw.tcl" inside XDGRAPH.

9.11.3 Pattern Matching

* Matches any sequence of characters including a null string.

? Matches any single character.

[char*] Matches any character in the set given by `chars`. If a sequence of the form `x-y` appears in `chars`, then any character between `x` and `y`, inclusive, will match.

\x Matches the single character `x`. This provides a way of avoiding the special interpretation of the characters `*`? `[]` in the pattern.

9.11.4 Notes for VMS users

Because of the special use of square brackets `[]` in Tcl you can’t give filenames with directory specifiers without precautions. For example, if you want to read in a file `[datadir.pro]xd_fou.grd`, you have to specify the filename in one of three ways:

- Quote the whole filename:
  ```
  dataset :d1 -load {[datadir.pro]xd_fou.grd}
  ```

- Quote the brackets only:
  ```
  dataset :d1 -load \[datadir.pro\]xd_fou.grd
  ```

- Give the filename in Unix-style. You must give a device name in this case:
  ```
  dataset :d1 -load /device/datadir/pro/xd_fou.grd
  ```

Similar considerations apply to filenames with dollar signs. Use one of the first two methods to overcome this problem. In some cases it might be better to define a DCL logical name. This is especially true for the initialization files Tcl, Tk and XDGRAPH automatically include at startup.

Let’s assume the startup file for XDGRAPH can be found in the directory `SYS$SYSDEVICE`:

```
[XD. LIB. TCL]. Using the following logical names will hide the dollar sign from Tcl.
```

```
$ define TCL.DISK SYS$SYSDEVICE
$ define XD_TCLDIR /tcl_disk/xd/lib/tcl
```

Note however, that this particular example should only concern the person responsible for installation.

9.11.5 Notes for Windows users

The Windows release of XD contains an executable version of XDGRAPH which is linked with version 8.3 of the Tcl/tk library. The necessary runtime libraries TCL83.DLL and TK83.DLL are supplied, as well as all the Tcl/tk system files. Unfortunately, the OpenGL driver is not available for this version of XDGRAPH, so several functions like iso-surface plots are not working. See Section 13.4 on installation.
Chapter 10

TOPXD – Full Topological Analysis

10.1 Overview

TOPXD [1] has been developed in order to fully incorporate the Quantum Theory of Atoms in Molecules [2] (QTAM) into routine X-ray charge density studies. The program TOPOND98 [3], originally written for the CRYSTAL98 package [4] has been adapted for the experimental charge-density package XD. While the evaluation of several charge density properties at the critical points is already included in the XDPROP program, the TOPXD program provides several additional features. The main ones are its capability to define atomic basin boundaries and to integrate density functions within the basins, thus producing an extensive set of atomic properties, including net charges, dipole and higher electrostatic moments.

TOPXD allows the user to undertake:

- fully-automated chain-like searching for critical points in the $\rho$ and $\nabla^2 \rho$ scalar fields, using either conventional Newton-Raphson techniques or the eigenvector following method [5,6]
- grid searching of critical points in the asymmetric unit
- evaluation of atomic properties
- finely tuned algorithms for the evaluation of atomic interaction lines or of atomic graphs
- extensive 2D and 3D graphical representations.

The experimental electron density and its analytical derivatives up to order 2 are calculated using the same subroutines as XDLSM. However, derivatives of a higher order (up to 4) are required when searching for Laplacian critical points in the field of the Laplacian of the electron density. Derivatives of the third and fourth order are evaluated in TOPXD as a numerical finite-difference approximation of the first and the second order analytical derivatives. The numerical derivative approach has been described before [7] and was shown to be extremely useful when no analytical derivatives are available. For that purpose well known central-difference expressions with fourth-order error ($O(h^4)$) have been used [8]:

$$f_2'' = -f_{x+2h} + 8f_{x+h} - 8f_{x-h} + f_{x-2h},$$

$$f_4'' = -f_{x+2h} + 16f_{x+h} - 30f_x + 16f_{x-h} + f_{x-2h},$$

where $x$ is the point at which the numerical derivative is evaluated and $h$ is the step size. Higher order numerical derivatives (or partial derivatives) are not needed, because every derivative of order 3 to 4 can be represented as first or second order finite-difference numerical approximation of the first or second order analytical derivative using a simple chain rule, for example:

$$\frac{d^3 \rho}{dx^2 dy} = \frac{d}{dx} \left[ \frac{d^2 \rho}{dy dx} \right] = \frac{d}{dy} \left[ \frac{d^2 \rho}{dx^2} \right] = \frac{d^2}{dx^2} \left[ \frac{dp}{dy} \right],$$

in which expressions in square brackets are analytical derivatives while the outer part is evaluated numerically.
The accuracy of the numerical differentiation of the electron density has been extensively tested by comparison of the numerical first and pure second derivatives with those obtained analytically for a number of (3,-1) critical points and for some arbitrarily selected points. With a step size of \( h=5\times10^{-3} \), the expected error in the numerical derivatives is only \( O(h^4)=6.25\times10^{-10} \). Numerical examples show the actual error to be less than \( 1\times10^{-9} \) and practically nonexistent when double precision variables are used (as is the current default for TOPXD). A comparison of analytical mixed second derivatives with those obtained by numerical finite-difference differentiation of the first analytical derivative shows the difference to be less than \( 1\times10^{-9} \). A drawback of numerical differentiation is that in order to approximate one derivative, several evaluations of the function are required. Indeed, in order to obtain a numerical approximation of a pure second derivative, for example \( \frac{d^2 \rho}{dx^2} \), evaluation of density is required at 5 different points. Fortunately, due to the exceptional computational power of modern computers, such evaluations are only slightly more costly than using pure analytical expressions.

TOPXD works in the XDPROP-like "global" Cartesian system. All input and output atomic Cartesian atomic coordinates are assumed to be in Ångstroms. Also, in some cases, fractional atomic coordinates can be used. The charge density and its derivatives can be in either atomic units (au) or electrons/Ångstroms. Internally, TOPXD uses only atomic units.

10.1.1 Input Files and Running TOPXD

TOPXD requires only two input files:

- xd.mas – XD master file
- xd.res(xd.inp) – XD parameter file

The standard XD parameter file with atomic positional and multipole parameters is used by TOPXD.

The XD master file (xd.mas) should contain the TOPXD specific instructions described in the next section of this manual. The current version of XDINI provides a default mask for TOPXD.

Once the desired TOPXD instructions are included and activated in XD master file, TOPXD can be run via command line:

```bash
topxd >& topxd-output-file & (Unix/Linux as background process) or
topxd topxd-output-file (Windows console version)
```

where `topxd` is the name of the TOPXD executable file and `topxd-output_file` is a legal filename such as `topxd.out`. Both XD master and parameter files must be present in the current directory, otherwise the program stops and the error message is printed.
10.1.2 Description of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTAM</td>
<td>Quantum Theory of Atoms in Molecules (R.F.W. Bader)</td>
</tr>
<tr>
<td>$\rho(r)$</td>
<td>Electron density</td>
</tr>
<tr>
<td>$\nabla^2 \rho(r)$</td>
<td>Laplacian ($\nabla \cdot \nabla$) of $\rho(r)$</td>
</tr>
<tr>
<td>$H(\rho(r))$</td>
<td>The Hessian (second derivatives) matrix of $\rho(r)$</td>
</tr>
<tr>
<td>$\lambda_k$</td>
<td>Eigenvalues ($\lambda_1 \cdot \lambda_2 \cdot \lambda_3$) of $H(\rho(r))$</td>
</tr>
<tr>
<td>CP(s)</td>
<td>Critical Point(s) : a point $r$ where a given scalar $f$ has $\nabla f(r) = 0$</td>
</tr>
<tr>
<td>(m,n) CP</td>
<td>A critical point with rank $m$ and signature $n$. The rank is the number of non-zero eigenvalues, the signature is the difference between the number of positive and negative eigenvalues of $H(\rho(r))$ at the CP.</td>
</tr>
<tr>
<td>BCP</td>
<td>Bond Critical Point : a (3,-1) CP in the $\rho(r)$ scalar field</td>
</tr>
<tr>
<td>ZFS</td>
<td>Zero-Flux-Surface : $\nabla \rho(r) \cdot n(r) = 0 \ \ \forall \ r \in \text{surface}$</td>
</tr>
<tr>
<td>Atomic Basin</td>
<td>The space traversed by all the uphill $\nabla \rho(r)$ paths which terminate at a nucleus, which acts as a 3D attractor for its basin. The atomic basin is also the portion of space enclosed by a ZFS and including a nucleus.</td>
</tr>
<tr>
<td>NNA</td>
<td>Non-Nuclear Attractor: a (3,-3) CP of $\rho$ at a position other than nuclei</td>
</tr>
<tr>
<td>NEA</td>
<td>Non-Equivalent (unique) Atom</td>
</tr>
<tr>
<td>AIL</td>
<td>Atomic Interaction Line : a curved path joining two nuclei along which $-\nabla^2 \rho(r)$ is a maximum with respect to any lateral displacement</td>
</tr>
<tr>
<td>&quot;Crystal Graph&quot;</td>
<td>The network of AIL's for a given crystal geometry. It is the crystal correspondent of the QTAM molecular graph for an isolated molecule</td>
</tr>
<tr>
<td>AGL</td>
<td>Atomic Graph Line : a curved path joining two -$\nabla^2 \rho(r)$ (3,-3) CPs along which -$\nabla^2 \rho(r)$ is a maximum with respect to any lateral displacement</td>
</tr>
<tr>
<td>VSCC</td>
<td>Valence Shell Charge Concentration : the atomic valence shell region where $\nabla^2 \rho(r)$ is negative</td>
</tr>
<tr>
<td>IR</td>
<td>Integration Ray : in atomic properties evaluation</td>
</tr>
<tr>
<td>EF</td>
<td>Eigenvector Following method (CP search)</td>
</tr>
</tbody>
</table>

10.2 TOPXD Instructions

TOPXD reads both general XD instructions (CELL, LATT, SYMM and SCAT) and TOPXD-specific instructions from the XD master file (xd.mas). The TOPXD specific instructions begin with *MODULE TOPXD and be terminated by the END TOPXD line, in the same manner as other XD modules. Input is in free format and not case sensitive, as it is internally converted to upper case. Blank lines and lines beginning with the exclamation mark (!) are treated as comments and are ignored. Most of the XD conventions regarding the format and style of the xd.mas file are also be valid for TOPXD instructions.

All TOPXD instructions begin on a new line, usually with a keyword (normally a four-character keyword) at the start. Curly brackets "{" denote an optional input. If in doubt as to the required syntax, see the example xd.mas file shown in Chapter 12.

There are several sections in the TOPXD program input, each identified by a special keyword:
10.2.1 General Instructions

General instructions are the first to be specified in `*MODULE TOPXD` part of the xd.mas file and apply to all TOPXD sections that follow. All these instructions are optional, i.e. the defaults are provided internally for all of these options, yet it is recommended to always include these instructions in the xd.mas file.

The following general instructions can be specified:

**10.2.1.1 COMT**

```
COMT comment-string
```

*comment-string* is a comment for TOPXD run. It is read in free format as an eighty character string, starting from fifth character in the `COMT` line. The default is no comment.

**10.2.1.2 DEBG**

```
DEBG (*)symeqv (*)derive *check
```

(*)symeqv
When starred, this option creates the file `gen_eq.log` with extensive information about symmetry-equivalent atoms generated using `SYMM` (see Section 2.2.1.5) and `CGEN` (see below) instructions. This file can be large, so generally this option would only be used for a first time run of TOPXD for a particular compound, in order to check if symmetry-equivalent atoms have been generated properly. The default is not to create the `gen_eq.log` file.

(*)derive
When starred, it enables the debugging printout of the charge density and its derivatives to the file `debug_rho.log` for each `xyz` point. When using this option, please make sure that plenty of the disk space is available. In general, one should not use this option at all. The default is not to print this file.

(*)check
This option prints extra (very voluminous !) debugging information when starred.

**10.2.1.3 CGEN**

```
CGEN alim xmin xmax blim ymin ymax clim zmin zmax
```

*xmin xmax*
Minimum and maximum allowed fractional coordinates of atoms along the unit cell axis \( a \) for generating of symmetry equivalent molecules (atoms). The default is \(-1.0 \, 2.0\), i.e. only atoms with \(-1. < x < 2.\) will be generated. \( y_{\text{min}} \, y_{\text{max}} \, z_{\text{min}} \, z_{\text{max}} \) are similarly defined along the unit cell axes \( b \) and \( c \) respectively and have the same defaults.

10.2.1.4 MPAR

MPAR \( r_{\text{cut}} \) \( r_{\text{cut}} \) \( d_{\text{step}} \) \( d_{\text{step}} \) (*)au (*)iam

\( r_{\text{cut}} \)
As in XDPROP (see Section 6.2.5), the maximum distance in Å from \( xyz \) point to contributing pseudoatom. The default is 4.0 Å.

\( d_{\text{step}} \)
As in XDPROP (see Section 6.2.5), the numerical differentiation step in Å. This parameter will only be applied to numerical evaluation of the third and fourth derivatives of \( \rho(\mathbf{r}) \) (see Section 10.1) since first and second derivatives are always evaluated analytically. The default is 0.005 (Å).

(*)au
If starred, all output parameters related to the charge density and its derivatives will be in atomic units, otherwise the units are electrons and Ångstroms. This keyword has no effect on ATBP section of TOPXD, in which the output is always in atomic units. The default (except for the ATBP section) is electrons and Ångstroms.

(*)iam
If starred, the independent atom model (IAM) will be used in calculation of charge density and its derivatives, i.e. all multipole populations \( (l \cdot 1) \) are set to zero; monopole populations are set to free atom values and \( \kappa \) parameters are set to unity. Otherwise the multipole model specified in the \( \text{xd.mas} \) file is used.

10.2.1.5 DGRD

DGRD (*)use (*)gen (*)fra gstep dx dy dz (*)read (*)ascii filename

(*)use
If starred, TOPXD will use the "density on the grid" method. The default is not to use the "density on the grid" method.

(*)gen
If starred, TOPXD will generate the grid based on the multipole parameters in the \( \text{xd.mas} \) file.

(*)fra
If starred, the grid spacing parameters \( dx \) \( dy \) and \( dz \) specified after gstep are in fractional units. Otherwise \( dx \) \( dy \) and \( dz \) are in Angstrom units.

(*)read
If starred, TOPXD will read the external grid file and ignore whatever multipole parameters are specified in \( \text{xd.mas} \).

(*)ascii
If starred, the gridfile specified by the \( \text{filename} \) is in ASCII (text) format, otherwise it is a binary file.
10.3 Topological Analysis of Electron Density

In TOPXD, the search for (3,-3) critical points associated with the nuclear maxima is skipped, since the Slater type basis functions used in the multipolar expansion correctly yield a cusp at the nuclear position (and hence no critical point). The non-nuclear attractor (3,-3) critical points can be recovered either as termini of a bond path associated with a (3,-1) critical point or in the grid search for critical points (see Section 10.3.6).

The construction of the clusters of atoms used in the TOPXD calculations is based around the idea of adding "coordination shells" of (symmetry related) atoms at a specific distance from the starting or seed atom. If this starting or seed atom is in a general position in a low symmetry space group, then normally there will only be one atom per "coordination shell". It is important that enough "coordination shells" are specified to generate all neighbours of the starting or seed atom. The "coordination shell" was called a "symmetry related star of atoms" in previous versions of the manual, but the former term is less confusing and is used here.

10.3.1 Auto critical point search within molecular clusters built around "seed" point(s)

This is a fully-automated and chain-like search strategy for all kinds of critical points using at each stage the eigenvector-following (EF) step suitable for the kind of critical point searched for. The search is performed within a finite region of space, which encloses a finite molecular cluster built-up around a specified "seed point" A. The size and origin of the cluster are given in following input.

TRHO (*)seed (*)all (*)ail (*)debug nstep nnb nmb rmax th th
(*)fra x y z (several of either of these lines may be present) or
(*)car x y z (several of either of these lines may be present)

(*)seed
If starred, the search is performed. Otherwise no search is undertaken.

(*)all
If this keyword is starred, all kinds of critical points are searched for, otherwise the chain-like search is stopped after the (3,-1) stage. This option saves the largest part of the CPU time (if the keyword ail is not activated) required by the automatic search. It is useful when very large clusters are defined around the seed point.

(*)ail
If this keyword is starred, atomic interaction line (AIL) lengths and termini are evaluated numerically for each unique (3,-1) critical point. Otherwise atomic interaction line lengths and termini are not evaluated. NOTE this option is VERY compute-intensive !

(*)debug
Starring this keyword enables the debug printing during the critical point search

nstep nstep
nstep determines the maximum number of EF steps along each search

nnb nmb
The value of nmb determines the maximum number of "coordination shells" of atoms to be included in the cluster generated around the "seed point". nmb also defines the number of neighbours in the nearest neighbour analysis around each unique critical point (of any kind) found. In general, nmb should be set to be as large as is required to generate all neighbours around a "seed point"
Chapter 10 - TOPXD - Full Topological Analysis

**rmax**

$rmax$ is maximum radius (Å) of the clusters. Each cluster includes all atoms within a sphere of radius $rmax$, centered on the "seed-point" A. $rmax$ may locally reduce the actual value of $nnb$

**th**

If $th$ is not zero, the (3,-1) critical point search is only performed among all the unique atom pairs whose internuclear distance falls below $th$ (Å). Otherwise ($th=0.0$) the default value is used (5Å)

(*)$fra$  $x$  $y$  $z$

(*)$car$  $x$  $y$  $z$

*Each of these commands must begin on a new line!*

Either the fractional ($fra$) or Cartesian (Å) ($car$) coordinates ($x$ $y$ $z$) of the "seed point" A (i.e. the center of the cluster) are given. These lines may be repeated $n$ times for $n$ "seed points" (one "seed point" per line). A search for a particular "seed point" will only be performed if the corresponding keywords $fra$ or $car$ are starred.

### 10.3.2 Auto critical point search within molecular clusters built around each of the unique atoms

This is a fully automated and chain-like search strategy for all kinds of critical points, using at each stage the eigenvector following (EF) step suitable for the kind of critical points searched for. The search is performed within a finite region of space, which is defined by building-up finite molecular clusters centered on each of the unique atoms of the unit cell. The size of clusters is given in input.

**TRHO** (*)$cluster$ (*)$all$  (*)$ail$  (*)$debug$  $nstep$  $nnb$  $nnb$  $rmax$  $rmax$  $th$  $th$

(*)$cluster$

If starred, the search is performed. Otherwise no search is undertaken.

The usage of keywords (*)$all$, (*)$ail$, (*)$debug$, $nstep$, $nnb$, $rmax$, $th$ is as described in Section 10.3.1

### 10.3.3 Auto critical point search between unique atom pairs

(3,-1) critical points are searched for, among all the unique pairs generated from a set of nuclei. The set is generated by constructing clusters of atoms around each unique atom of the unit cell. The use of the eigenvector following method allows the user to select other types of critical point.

**TRHO** (*)$pairs$  $meth$  (*)$ail$  (*)$debug$  $nstep$  $nnb$  $nnb$  $rmax$  $rmax$  $th$  $th$  $pc$  $pc$

(*)$pairs$

If starred, the search is performed. Otherwise no search is undertaken. $meth$ is the method for a critical point search and can be specified in one of the following formats:
nr  The Newton-Raphson (NR) algorithm is used in the critical point search.

ef type  The eigenvector-following (EF) algorithm is used in the critical point search. The value of variable type depends on the kind of critical point to be searched for (only one type can be specified per instruction). It is a three-character string, either ncp/bcp/rcp/ccp for (3,-3), (3,-1), (3,+1) or (3,+3) critical points respectively. Use the ncp type only when a non-nuclear attractor is searched for since the (3,-3) critical points is not found in TOPXD for nuclear positions.

an  Cioslowski’s analytical determination [9] of atomic interaction lines

The usage of keywords (*)ail, (*)debug, nnb, rmax and nstep is exactly as described in Section 10.3.1 Note that (*)ail cannot be used if meth=an.

th  t
If th is not zero, the (3,-1) critical point search is performed among all the unique atom pairs whose internuclear distance falls below th (Å). Otherwise (th=0.0) the default value is used (4Å)

\{ pc \}
should only be specified if meth=nr

pc=0 : if a critical point is not found between A-B atom pair, the starting point of the NR search is displaced along the internuclear axis from the mid-point of the axis to the following two positions:

- \( r'_{\text{start}} = r_A + pc \cdot (r_B - r_A) \)
- \( r''_{\text{start}} = r_A + (1 - pc) \cdot (r_B - r_A) \),

pc=0 : specifies the default value of pc (0.4).

10.3.4 Critical point search from a starting set of points

TRHO (*)points meth (*)ail (*)debug nstep nstep nnb nnb rmax rmax [fra | car ]
x y z (several of these lines may be given)

(*)points
If starred, the search is performed. Otherwise no search is undertaken. meth is the method for a critical point search and can be either nr or ef type as specified in Section 10.3.3

The usage of keywords (*)ail, (*)debug, nstep, nnb, rmax is as described in Section 10.3.1

[ car | fra ]
The coordinates of starting points are given in fractional (fra) or Cartesian (Å) (car) coordinates

x y z
This command must begin on a new line!

Coordinates of the starting point (units depend on the car|fra keyword above ). This line may be repeated n times for n starting points (one set of coordinates per line)
10.3.5 Critical point search along the line joining two nuclei or two general points

TRHO (*)line meth (*)ail (*)debug nstep nnb nnb rmax rmax
<line specification>

(*)line
If starred, the search is performed. Otherwise no search is undertaken. meth is the method for a critical point search and can be either nr or ef type as specified in Section 10.3.3. In this case, the Newton-Raphson method is strongly recommended unless looking for a specific type of critical point along the line to the exclusion of all others.

The usage of keywords (*)ail, (*)debug, nstep, nnb, rmax is as described in Section 10.3.1

This next command begins on a new line!
<line specification> can be given in one of the following two formats:

(*)atom label toneighbor i1...in
Critical point search along the line(s) joining the unique atom A with label label and its i1...in neighbour(s) (atom B), where i is the "NEW" number in the "Clusters around each of the unique atoms" printing at the beginning of the TOPXD output. The search will only be performed if the keyword atom is starred. Repeat this line n times for n unique atoms.

(*)point [car|fra] x1 y1 z1 x2 y2 z2
Critical point search along the line joining two points a and b with coordinates (x1 y1 z1) and (x2 y2 z2), respectively. The coordinates are in Cartesian (Å) (car) or fractional (fra) units. The search will only be performed if the keyword point is starred. Repeat this line n times for n point pairs.

10.3.6 Grid search for critical points

TRHO (*)grid meth (*)ail (*)debug nstep nstep nnb nnb rmax rmax xmin xmin xmax xmax xstep xstep ymin ymin ymax ymax ystep ystep zmin zmin zmax zmax zstep zstep

(*)grid
If starred, the search is performed. Otherwise no search is undertaken. meth is the method for a critical search and can be either nr or ef type as specified in Section 10.3.3. For this command, the Newton-Raphson method is strongly recommended unless looking for a specific type of CP in the cell volume to the exclusion of all others. WARNING ! The grid search is VERY costly if the entire asymmetric unit is explored.

The usage of keywords (*)ail, (*)debug, nstep, nnb, rmax is as described in Section 10.3.1

<grid specification>
xmin xmin xmax xmax xstep xstep (fractional units)
xmin xmax xstep determine the minimum, maximum and grid interval along crystal a-axis.
ymin ymin ymax ymax ystep ystep (fractional units) and
zmin zmin zmax zmax zstep zstep (fractional units)
are similarly defined with reference to the crystal b and c-axes respectively.

All three MUST be given and they must all start on a new line.
10.3.7 Profiles of $\rho(r)$, $\nabla^2 \rho(r)$ and $\lambda_3$ along the line joining two nuclei or two general points

Profiles of $\rho(r)$, $\nabla^2 \rho(r)$ and $\lambda_3$ are written to Fortran units 95, 96, 97, respectively. The units of $\rho(r)$, $\nabla^2 \rho(r)$ and $\lambda_3$ are determined by the keyword (*)au (see Section 10.1).

**TRHO** (*)profile perstep $n$
<profile specification> (Several of these may be given)
(*)profile
If starred, the search is performed. Otherwise no search is undertaken.

**perstep** $n$
Determines the percentage step $s$ along $A$-$B$ or $a$-$b$
if $n=1$, $s = 0.01 \times R_{A-B}$ (or $R_{a-b}$)
if $n=100$, $s = 1 \times R_{A-B}$ (or $R_{a-b}$)

<profile specification> On a new line ! It can be given in one of the following formats:

(*)atom label toneighbor $i1...i(n)$
Profile along the line(s) joining the unique atom $A$ with label $label$ and its $i1..i(n)$ neighbour(s) (atom $B$), where $i$ is the "NEW" number in the "Clusters around each of the unique atom" printing at the beginning of the TOPXD output. The profiling will only be performed if keyword atom is starred. Repeat this line $m$ times for the $m$ unique atoms to be considered.

(*)point | car | fra | $x1$ $y1$ $z1$ $x2$ $y2$ $z2$
Profile along the line joining two points $a$ and $b$ with coordinates ($x1$ $y1$ $z1$) and ($x2$ $y2$ $z2$), respectively. The coordinates are in Cartesian (Å) (car) or fractional (fra) units. The profiling will only be performed if the keyword point is starred. Repeat this line $n$ times for $n$ point pairs.

10.4 Topological Analysis of the Laplacian of Electron Density

10.4.1 Auto critical point search within the concentration (or depletion) shells of unique atoms and/or non-nuclear attractors

Usually the search is performed for critical points in the valence shell charge concentration (VSCC) of each selected unique atom. Nevertheless, a suitable choice for the sphere radius ($rstar$ parameter, see below) allows for a critical point search in (any of) the depletion shells of the selected unique atom.

**TLAP** (*)auto meth (*)ail (*)debug nstep $nstep$ nnb $nnb$ rmax $rmax$ ntheta $ntheta$ nphi $nphi$ <atom(s) specifications> (Several of these may be given)
<NNA specifications>

(*)auto
If starred, the search is performed. Otherwise no search is undertaken. $meth$ is the method for a CP search and can be either nr or ef type as specified in Section 10.3.3.

(*)ail
If this keyword is starred, atomic graph line (AGL) lengths and termini are evaluated numerically for each unique (3,4) critical point. This is a compute-intensive option ! The
AGL is the union of the unique pair of $\nabla^{2}\rho$ trajectories that originate at the \((3,-1)\) critical point and terminate at neighbouring \((3,-3)\) $\nabla^{2}\rho$ critical points.

The usage of keywords \((*)\)debug, nstep, nnb, rmax is as described in Section 10.3.1

\textbf{ntheta} ntheta nphi nphi
The critical point search is started from points located on the surface of a sphere, centered on the nucleus of a given unique atom or at the NNA location. The number of starting points is determined by the intervals ntheta, nphi chosen for the polar coordinates $\theta$ and $\varphi$, respectively

\textbf{atom(s) specifications} are given in the following format. Note that this record may be repeated as many times as needed for unique atoms for which the critical point search is desired. One may group in a single record those unique atoms that are characterized by equal rstar and nmax values)

\textbf{(*)atoms label1...label(n) nmax nmax rstar rstar}
The critical point search will only be performed if the keyword atoms is starred.

\textbf{label1...label(n)}
Labels of unique atoms for which the critical point search will be performed.

\textbf{nmax}
If nmax is nonzero and if the EF method is used, the search for each atom is stopped when nmax different critical points of the required type are found. Otherwise a normal search is undertaken.

\textbf{rstar}
If rstar is zero, the default sphere radius is adopted in the critical point search. The radius is taken to be equal to the distance from the nucleus to the spherical surface where $-\nabla^{2}\rho$ attains its maximum value in the valence shell of the isolated atom. If rstar is nonzero, then the sphere radius is taken to be equal to rstar value (Å).

\textbf{NNA specifications} given in the following format. Insert one record for each NNA :

\textbf{(*)nna x x y y z z nmax nmax rstar rstar}
The critical point search for this NNA will only be performed if the keyword nna is starred.

\textbf{x y z} - Cartesian coordinates of the current NNA (Å)

\textbf{nmax , rstar}
As above for (*)atoms

\textbf{10.4.2 Critical point search started from a given set of points}

\textbf{TLAP (*)points meth (*)ail (*)debug nstep nnb nnb rmax rmax nmax nmax}
\textbf{(*)car x y z} (Several of either of these records may be given)
\textbf{(*)fra x y z}

\textbf{(*)points}
If starred, the search is performed. Otherwise no search is undertaken. meth is the method for a critical point search and can be either nr or ef type as specified in Section 10.3.3.

The usage of keywords \((*)\)ail, \((*)\)debug, nstep, nnb, rmax, nmax is as described in Section 10.4.1 Note that for nstep nstep, use a very small number of steps, say no more than 5-8,
since the search is repeated 40 times, starting from 40 evenly distributed points along the line.

\(*)\text{car} \ x \ y \ z \\
\(*)\text{fra} \ x \ y \ z \\

*These commands must each start on a new line!* Starting point coordinates \((x \ y \ z)\) in either Cartesian (Å) \((\text{car})\) or fractional \((\text{fra})\) units are given. The search will only be performed if the corresponding keyword is starred. Insert this record \(n\) times to start CP search from \(n\) starting points.

10.4.3 Critical point search along the line joining two nuclei or two general points

**TLAP (*)line \(\text{meth} \ (*)\text{ail} \ (*)\text{debug} \ n\text{step} \ n\text{nb} \ n\text{nb} \ r\text{max} \ r\text{max} \ n\text{max} \ n\text{max}**

\(<\text{line specifications}>\) (Several of these line may be given)

\(*)\text{line}

If starred, the search is performed. Otherwise no search is undertaken. \(\text{meth}\) is the method for a critical point search and can be either \(\text{nr}\) or \(\text{ef}\) type as specified in Section 10.3.3. It is recommended to use the Newton-Raphson search \((\text{nr})\) unless a specific type of critical point is being sought to the exclusion of all others.

The usage of keywords \(*)\text{ail}, \(*)\text{debug}, \ n\text{step}, \ n\text{nb}, \ r\text{max}, \ r\text{max}, \ n\text{max} \ n\text{max} \) is exactly as described in Section 10.4.1

\(<\text{line specifications}>\) can be given in one of the following formats:

\(*)\text{atom} \ \text{label} \ \text{toneighbor} \ i1 \ldots i(n)

Critical point search along the line(s) joining the unique atom \(A\) with label \(\text{label}\) and its \(i1..i(n)\) neighbour(s) \(\text{atom} B\), where \(i\) is the "NEW" number in the "Clusters around each of the unique atom" printing at the beginning of the TOPXD output. The search will only be performed if the keyword \(\text{atom}\) is starred.

\(*)\text{point} \ [\ \text{car}\ | \ \text{fra}\ ] \ x1 \ y1 \ z1 \ x2 \ y2 \ z2

Critical point search along the line joining two points \(a\) and \(b\) with coordinates \((x1 \ y1 \ z1)\) and \((x2 \ y2 \ z2)\), respectively. The coordinates are either in Cartesian (Å) \((\text{car})\) or fractional \((\text{fra})\) units. The search will only be performed if the keyword \(\text{point}\) is starred. Repeat this line \(n\) times for \(n\) point pairs.

10.5 Evaluation of atomic and/or NNA basin properties

The atomic and NNA basin integration part of TOPXD consists of the following five types of instructions:

- **General parameters**
  \(\text{ATBP params} \ldots\)

- **Alternative method for the Zero Flux Surface (ZFS) search**
  \(\text{ATBP altguess} \ldots\)
• Capture sphere specifications for all unique atoms (optional)
  ATBP spheres ......

• Instructions for integration of unique atoms (required)
  ATBP (*)atoms ......

• NNA(s) specifications, if present (optional)
  ATBP NNA ......

10.5.1 General Parameters

ATBP params PhInSph phi ThInSph theta *SavSurf

The phi and theta parameters define the angular integration parameters which are used INSIDE the β-sphere, i.e. the number of φ (phi) and θ (theta) grid points. SavSurf, when starred, enables TOPXD to write out the lengths and coordinates of integrations rays to file rays.dat for visualization in VZ3D (see section 10.6.1).

10.5.2 Alternative method for Zero Flux Surface (ZFS) search

Activating this set of (optional) instructions enables an alternative method for the ZFS search, based on algorithms described in references [10-11]. In addition, a second-order Runge-Kutta method is used when tracing the gradient paths instead of a Predictor-Corrector method. The advantage of this method is that it can significantly improve the speed of the ZFS search, but can result in a less accurate ZFS if incorrect parameters are specified.

ATBP altguess bigstep bigstep accur accur maxrint maxrint rmax step0 step0 A a B b

bistep (au) defines the size of the step along the integration ray with which the search for intersection of each integration ray and ZFS is performed. accur (au) is the final precision in ZFS determination in the bisection method (NOTE: it overrides the accur parameter specified in ATBP atoms instructions). maxrint (au) defines the maximum distance along the integration ray which can be reached during the search for ZFS intersection. rmax (au) – radius which defines the size of the cluster of neighbours when tracing the gradient paths. Parameters step0, a and b determine the step size for tracing the gradient paths at each point according [11] to the formula:

\[
  \text{step} = \text{step0} \cdot \exp(a \cos \omega)^b
\]

where step0 (au), a and b are input parameters and ω is the angle between two vectors: integration ray and the gradient of the density.

10.5.3 Capture sphere specifications for unique atoms

Although these instruction(s) are optional, it is strongly recommended to specify the capture spheres for all atoms as it should considerably reduce the program runtime. There is no limit on the number of ATBP spheres instructions.

ATBP spheres label1 rad1 .... label(n) rad(n)

rad is the radius of a capture sphere (Å) for unique atom with label label. rad should be generally taken to be equal to the distance from the nucleus to the nearest of the p(r) bond critical points which lie on the Vp(r) ZFS, enclosing the atomic basin of the unique atom with
10.5.4 Instructions for integration of unique atoms

This required instruction requests the integration of the atom basin(s) of unique atom(s). There can be as many lines with ATBP (*)atoms .... instructions as the number of unique atoms. 
*Note that this command MUST be entered all on a single line, NOT as shown below (see Chapter 12 for an example).

\[
\text{ATBP} \ (*\text{atoms}) \ label \ [ \ izfs \ | \ zfs ] \ nvi \ nvi \ irsur \ irsur \ (*\text{irsav} \ (*\text{rest} \ (*\text{debug} \ phi \ nphi \ th \ nth \ rad \ rad \ accur \ accur \ \{ \ nbcp \ nbcp \ | \ car \ | \ fra \ \})
\]

\{ \ldots \text{if } nbcp > 0 \text{ insert } nbcp \text{ records with } BCP \text{ } x \text{ } y \text{ } z \text{ coordinates } \ldots \ldots \} \]

\text{(*)atoms}

If starred, this requests the integration of unique atoms specified with label instruction, otherwise no integration will be performed.

\text{label}

Specifies which atoms will be integrated if the \textbf{atoms} keyword is starred. There are several possible format specifications for \text{label},

1. Labels of unique atoms, for example: 
\begin{verbatim}
ATBP *atoms O(1) \ C(2) \ N(10) \ H(2A) \ldots
dd
\end{verbatim}

2. Keyword \textbf{all} for all unique atoms to be integrated, for example: 
\begin{verbatim}
ATBP *atoms all \ldots
\end{verbatim}

3. Atomic symbols – all atoms with the same atomic symbols will be integrated, for example: 
\begin{verbatim}
ATBP *atoms O \ H \ldots
\end{verbatim}

\text{izfs}

Indirect determination of the ZFS [12]. This is the \textbf{recommended} and more accurate method for determination of ZFSs, but is computationally rather demanding.

\text{zfs}

With this method the determination of the ZFSs is achieved in two steps: 
- direct determination [13] 
- indirect determination [12] for those integration rays (IRs) whose length was not correctly recovered in the first step 

\text{NOTE: The ZFS method is still experimental and has not been thoroughly tested. It may well fail!}

\text{nnvi nvi}

\text{nvi} is the number of “coordinination shells” of neighbours of the current unique atom(s) which have to be considered as possible attractors of the $\nabla \rho(r)$ paths launched from points along the integration rays.

\text{irsur irsur}
\( \text{irsur} = 0 \) - normal run
\( \text{irsur} = 1 \) - the lengths of the integration rays obtained in a previous run are read from Fortran unit 97 (file \texttt{fort.97}) and used as an initial guess for the IR lengths
\( \text{irsur} = -1 \) - the lengths of the integration rays are kept fixed to those obtained in a previous run and read from Fortran unit 97 (file \texttt{fort.97}).

\((*)\text{irsav}\)
When starred the lengths of the integration rays are saved in Fortran unit 98 (file \texttt{fort.98}). \textbf{NOTE:} The use of \( \text{irsur} \neq 0 \) requires that \texttt{irsav} was set in a previous ATBP run. The ZFS thus saved on Fortran unit 98 (file \texttt{fort.98}) may be used in a following run (\(\text{irsur} \neq 0\)), by copying it on Fortran unit 97 (file \texttt{fort.97}). Use \( \text{irsur} = -1 \) to run the integration step separate from the ZFS determination step. Put \( \text{irsur} = 1 \) to use the ZFS obtained in a previous run for a given unique atom (obtained, for example, using a different multipole model) as a starting guess for the new ZFS determination.

\((*)\text{rest}\)
When starred, the run is restarted from (partial) surface data stored in Fortran unit 96 (file \texttt{fort.96}) from a previous aborted run

\((*)\text{debug}\)
Activates the debug printing during the ZFS determination and integration

\(\text{phi} \ \text{nphi} \ \text{th} \ \text{nth}\)
Angular integration parameters used for OUTSIDE the \(\beta\)-sphere: number of \(\phi\) (\text{nphi}) and \(\theta\) (\text{nth}) grid points (see also FAQ section)

\(\text{rad} \ \text{rad} \ \text{rad}\)
\text{rad} is the number of radial integration points inside \(\beta\)-sphere

\(\text{accur} \ \text{accur}\)
Parameter \text{accur} (au) determines the numerical accuracy of each IR length and thus of the overall ZFS determination. The default value is 0.001 au. A larger \text{accur} value reduces the computational time at the expense of accuracy (see also FAQ section).

\{ \text{nbcp} \ \text{nbcp} [ \text{car} | \text{atom} ] \}
These optional parameters should only be used for a two-step procedure in ZFS determination (see keyword \texttt{zfs} above). \text{nbcp} is number of (3,-1) critical points associated with the atomic interaction lines (AIL) [see \texttt{TRHO} section] linking the current unique atom to other atoms and/or NNAs. The keyword \texttt{car|atom} determines the format for specification of (3,-1) critical points to be read in the following \text{nbcp} records (if \text{nbcp} > 0).

1. Keyword \texttt{car} specifies that the Cartesian coordinates of a (3,-1) critical point should be given:
   \{ \texttt{x} \ \texttt{y} \ \texttt{z} \}
2. Keyword \texttt{atom} specifies the atom linked to the current unique atom
   \{ \texttt{inum} \ \texttt{itx} \ \texttt{ity} \ \texttt{itz} \}

\text{inum} is the sequence number of a linked atom in the TOPXD printing of all atoms in the unit cell, while \texttt{itx} \ \texttt{ity} and \texttt{itz} specify the indices (direct cell) of the cell where the linked atom \text{inum} is located.
10.5.5 NNA(s) specifications

Use this optional instruction if non-nuclear attractors (NNAs) are present in the structure.

**ATBP NNA nna**

*nnn* is the number of NNAs in the structure. The default is zero, i.e. no NNAs.

If *nnn* > 0 insert *nnn* records (starting on a new line) with NNA specifications in the following format:

```
x x y y z z (*integ sphere rad { [ izfs | zfs ] nvi nvi isur isur (*)irsav (*)rest (*debug phi npht nth nth rad rad accur accur } )
```

There should be as many lines with NNA specifications as the number of NNAs in the structure. The entire command should be entered one ONE LINE (i.e. not as above but as in the example in Chapter 12)

```
x x y y z z
```

Cartesian coordinates of this NNA (Å)

(*integ)

When starred, the integration of the basin of this NNA will be performed

**sphere rad**

*rad* is the radius of a capture sphere for this NNA (Å). *rad* should be generally taken equal to the distance from the NNA to the nearest of the ρ(r) bond critical points which lie on the ∇ρ(r) zero-flux surface (ZFS), enclosing the NNA basin.

**NOTE:** The following keywords should only be used if keyword *integ* is starred: *izfs|zfs, nvi, isur, (*)irsav, (*)rest, (*)debug, phi, th, rad, accur*. Their usage is as described in Section 10.5.4

```
{ nbcp nbcp [ car | atom ] }
```

These optional parameters should only be used for the two-step procedure in the ZFS determination. Their usage is as described in Section 10.5.4

10.6 2-Dimensional (2D) Plots

The 2D (and 3D) plot instructions have a slightly different format than others. The general format consists of the following sections:

- General instructions
- Specific plot instruction(s) (one instructions per each specific plot type)
- Instructions for creating HPGL graphics files from the plot data
- Visualization with **hp2xx** program

PL2D general instructions apply to all the specific plot instructions PL2D plot until the next PL2D general is given and so on. There is no limit on neither the number of PL2D general nor PL2D plot instructions. Some of the TOPXD 2D plot files can also be visualized with program SURFER (a PC-DOS program for 3D plots) and XDGRAPH.
10.6.1 2D plot general instructions

This section MUST precede specific plot instructions. All parameters specified in this section will apply to the following specific plot instructions until the next general instruction section is given.

**PL2D general**

*point/atom A specification*

*point/atom B specification*

*point/atom C specification*

**plotdim xmin xmax xstep ymin ymax ystep**

**origin shift ishift origin xo yo zo vmod vmod**

**misc size ax scale name ‘name’ title ‘title’**

The format is exactly as laid out above, i.e. each command starts a new line. See Chapter 12 for an example.

**point/atom A/B/C specification**

Three atoms or points (A,B,C) must be given to define the plot plane. The specification format is different for atoms and points:

1. Atoms can be specified using the following format

   **atom inum itx ity itz**

   *inum* is the serial number of atom in TOPXD printout of all atoms in unit cell
   
   *itx ity itz* – translations applied to fractional coordinates of atom with number *inum* along X, Y and Z-crystal axis, respectively

2. General points can be specified using the following format:

   **point | car | fra | x y z**

   where *x y z* are fractional (fra) or Cartesian (Å) (car) coordinates

**NOTE:** If the evaluation of a molecular/crystal graph or, generally, of $\nabla \rho (r)$ trajectories is required, it is important to define the three atom/points in such a way that their associated clusters of neighbouring atoms (see below) adequately spans the plot plane.

**plotdim xmin xmax xstep ymin ymax ystep**

The plot plane is *XY*. *xmin* and *xmax* define the minimum and maximum values along the plot X-axis, respectively, while *xstep* defines the grid interval along X. *ymin ymax* and *ystep* have the same meaning but for Y-axis of the plot. All these parameters must be given in Ångstroms.

**origin shift ishift origin xo yo zo vmod vmod**

*ishift = 0* - origin as in the original Cartesian frame. A warning message is issued if, as a consequence of a given choice of the origin, the atoms/point A,B,C do not longer lie in the XY plot plane.

*ishift = 1* – the origin of the plot is translated to a point lying on ABC plane (must specify the xo yo and yo coordinates, see below)

*ishift = 2* – the origin of plot is put at mass-weighted centroid of the atoms/points, which define the ABC plane. A mass equal to 1 is assigned to any point in ABC plane definition.

*ishift = 3* – the origin of the plot lies along the A-B axis and its actual position is defined by the value of *vmod* variable (see below)

*ishift = 4* – the origin of the plot is at atom/point A

*ishift = 5* – the origin of the plot is in the centroid of the three atoms/points (as in XDGRAPH).
**xo yo zo** – Cartesian (Å) coordinates of the origin of the plot

**vmod** – the vmod value shifts the origin along A–B, so if vmod=0 the origin is at A and if vmod=1 the origin is at B; Negative vmod values as well as values greater than 1 are also allowed.

**misc size ax scale scale name name title title**

**ax** = a4 – plot size is A4

**a3** – plot size is A3

**scale** – Plot scale (Å/cm)

**name**

Suffix name enclosed in single quotes " " (maximum 24 characters, no blanks and no special symbols like ' , " , ^ etc) for the files which contain the values of each computed function func (see below) and the common information for the XYZ plot (system geometry, Euler rotation angles from crystal to plot frame, plot size, etc.). These files form the input for the **P2DCRY** section. Full names of the files are listed in **Table 10-1** below ("//" means character concatenation). Common information for the XY plot is saved in file: P2DCRYIN//name. The files prefixed with "SURF" may also be read from the SURFER program (a PC-DOS program for representing the scalar function in the plot plane as a 2D surface in 3D space, something that it is often referred to as a 3D graph).

**title** – Plot title enclosed in single quotes " " (maximum 80 symbols, blanks are allowed, no special symbols like ' , ", ^ ).

### Table 10-1

<table>
<thead>
<tr>
<th>Scalar function (or $\nabla \rho(r)$ plot) type</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electron density $\rho(r)$</td>
<td>$\rho(r)$</td>
</tr>
<tr>
<td>Laplacian of $\rho(r)$</td>
<td>$\nabla^2\rho(r)$</td>
</tr>
<tr>
<td>Negative of the Laplacian $\rho(r)$</td>
<td>$-\nabla^2\rho(r)$</td>
</tr>
<tr>
<td>Magnitude of the gradient of $\rho(r)$</td>
<td>$</td>
</tr>
<tr>
<td>$\nabla\rho(r)$ trajectories only</td>
<td></td>
</tr>
<tr>
<td>Molecular/crystal graph (and atomic basin boundaries)</td>
<td></td>
</tr>
<tr>
<td>$\nabla\rho(r)$ trajectories and molecular/crystal graph</td>
<td></td>
</tr>
</tbody>
</table>

#### 10.6.2 2D plot specific instructions

These instructions must follow the general 2D plot instructions.

**PL2D (\*)func nstar nstar rmax rmax (\*)test (\*)cut cutl {+func–dependent instructions}**

**Note:** There is no limit for the number of **PL2D** plot instructions (one instruction per line).

**(\*)func**

When starred, the function of one of the following types is plotted. The names of output files created are given in the description of 2D plot general instructions and in Table 10-1.

- ***rhoo** – electron density ($\rho(r)$)
- ***lapp** – Laplacian of the electron density ($\nabla^2\rho(r)$)
- ***lapm** – negative Laplacian of the electron density ($-\nabla^2\rho(r)$)
- ***grho** – magnitude of the gradient of electron density ($|\nabla\rho(r)|$)
- ***trajgrad** – $\nabla\rho(r)$ trajectories only
Chapter 10 - TOPXD - Full Topological Analysis

*molgraph – molecular/crystal graph (and atomic basin boundaries)
*trajmolg – \nabla \rho(\mathbf{r}) trajectories and molecular/crystal graph

\textbf{nstar} nstar

nstar is number of “coordination shells” of neighbours used in cluster construction around A, B and C atoms/points. From these clusters the atoms lying (not lying) in the ABC plane are selected and their position denoted with *(+ in the plots. The nstar variable is also used in and determines:

- Atom pairs considered in the evaluation of the molecular/crystal graph on the ABC plane
- Number of origins (nuclei) for the downhill \nabla \rho(\mathbf{r}) trajectories (gradient paths) on the ABC plane.

\textbf{rmax} rmax

rmax determines the maximum radius (Å) of each cluster (see nstar variable) and may (locally) reduce the actual value of nstar.

(*)test

If this keyword is starred, the program stops after printing the coordinates of A, B, C and corresponding clusters of atoms in the plot frame. Use this option to check if the choice of the plane ABC was correct at the first run.

(*)cut cutr cutl

If the keyword cut is starred, the scalar functions of \rho(\mathbf{r}) and |\nabla \rho(\mathbf{r})| are cut at the value of cutr and/or values of \nabla^2 \rho(\mathbf{r}) and -\nabla^2 \rho(\mathbf{r}) are cut at \pm cutl, according to their sign (cutr cutl must be given electrons and Angstroms). Cutting of the scalar function is generally required, especially in the case of the Laplacian, for representing the function in the plot plane as a 3D graph.

{+function–dependent parameters in Plot 2D specific instructions }

For certain plotted functions, some extra instructions may be necessary. These extra instructions are outlined in Sections 10.6.2.1 - 10.6.2.3.

\textbf{10.6.2.1 func = trajgrad}

Add the following parameters to the \textbf{PL2D plot trajgrad} instruction:

toler tol1 tol2 (*)plane npath npath nextr nextr

{ if nextr > 0 add nextr lines with Cartesian (Å) x y z of attractors \nabla \rho trajectories (one set per line) }

An atom in the XY plot plane is considered as an origin of downhill gradient paths or as a terminus of an atomic interaction line if its distance from plot edge is less than tol1 (Å) and the absolute value of the z-coordinate is less than tol2 (Å).

\textbf{NOTE:} A “correct” tracing of \nabla \rho(\mathbf{r}) trajectories in the plot plane would require this plane to be a mirror plane. However, this is not always the case. For example, in the study of molecular crystals at experimental geometries, it is common practice to deal with quasi-mirror planes. To cover such cases, \nabla \rho(\mathbf{r}) trajectories are projected on the plot plane whenever their absolute value of the z-coordinate is less than tol2 (Å).

If keyword plane is starred, the starting point of each \nabla \rho(\mathbf{r}) trajectory segment (which are typically 10^{-1} to 10^{-2} Å long) is forced on the XY plane, while the default is the normal tracing of \nabla \rho(\mathbf{r}) trajectories (with projection regulated by tol2).

The number of downhill \nabla \rho(\mathbf{r}) trajectories is defined by variable npath. The recommended value for npath is 36.

Variable nextr defines the number of other attractors, like non-nuclear (NNAs) attractors or the 2D attractors associated with a BCP, to be considered as origins of downhill \nabla \rho(\mathbf{r}) trajectories. A bond critical point is seen as a (2,-2) attractor whenever its associated ZFS
lies in the plot plane. If \( \text{nextr} \) is greater than zero, add \( \text{nextr} \) lines with Cartesian (Å) \( x \ y \ z \) coordinates of attractors of \( \nabla \rho(\mathbf{r}) \) trajectories (one set of \( x \ y \ z \) coordinates per line)

### 10.6.2.2 \texttt{func = molgraph}

Add the following parameters to the \texttt{PL2D plot molgraph} instruction:

\texttt{tol tol1 tol2 (*plane thr thr (*tr1 (*tr2 (*tr3 \( \)}

An atom in the XY plot plane is considered as an origin of downhill gradient paths or as a terminus of an atomic interaction line if its distance from plot edge is less than \( \text{tol1} \) (Å) and its absolute \( z \)-plot coordinate is less than \( \text{tol2} \) (Å). See note in Section 10.6.2.1 regarding “correct” tracing of molecular/crystal graphs.

If the keyword \texttt{plane} is starred, the starting point of each \( \nabla \rho(\mathbf{r}) \) trajectory segment is forced on the XY plane, while the default is the normal tracing of \( \nabla \rho(\mathbf{r}) \) trajectories (with projection regulated by \( \text{tol2} \)).

The maximum distance between atomic pairs, which is taken into account during the evaluation of bonded pairs and the automated tracing of the molecular/crystal graph, is defined by the value of \( \text{thr} \) (Å).

The \( \nabla \rho(\mathbf{r}) \) trajectories that originate at bond critical points and have as the initial direction the \( K \)-th eigenvector of Hessian of \( \rho(\mathbf{r}) \) at BCPs, are traced out (*tr\( K \)) or skipped (tr\( K \)). The \( K \)-th eigenvector is associated with the \( K \)-th eigenvalue of the Hessian \( \lambda \) (\( \lambda_1 \leq \lambda_2 \leq \lambda_3 \)).

### 10.6.2.3 \texttt{func = trajmolg}

Add the following parameters to the \texttt{PL2D plot trajmolg} instruction:

\texttt{tol tol1 tol2 (*plane thr thr (*tr1 (*tr2 (*tr3 npath npath nextr nextr \( \)}

\{ if nextr is greater than zero, add nextr lines with Cartesian (Å) \( x \ y \ z \) of attractors \( \nabla \rho \) trajectories (one set per line) \}

An atom in the XY plot plane is considered as an origin of downhill gradient paths or as a terminus of an atomic interaction line if its distance from plot edge is less than \( \text{tol1} \) (Å) and its absolute \( z \)-plot coordinate is less than \( \text{tol2} \) (Å). See note in Section 10.6.2.1 regarding “correct” tracing of molecular/crystal graphs.

The keywords (*tr\( 1 \) (*tr\( 2 \) (*tr\( 3 \) are as described in Section 10.6.2.2

The number of downhill \( \nabla \rho(\mathbf{r}) \) trajectories is defined by variable \( \text{npath} \) (the recommended value for \( \text{npath} \) is 36).

Variable \( \text{nextr} \) defines the number of other attractors, like non-nuclear (NNAs) attractors or the 2D attractors associated with a bond critical point, to be considered as origins of downhill \( \nabla \rho(\mathbf{r}) \) trajectories. A bond critical point is seen as a (2.-2) attractor whenever its associated ZFS lies in the plot plane.

If \( \text{nextr} \) is greater than zero, add \( \text{nextr} \) lines with Cartesian (Å) \( x \ y \ z \) coordinates of attractors of \( \nabla \rho(\mathbf{r}) \) trajectories (one set of \( x \ y \ z \) coordinates per line)

### 10.6.3 Converting TOPXD 2D plot data to HPGL graphics format

\texttt{P2DCRY (*\text{diff} var1 var2 var3 \} (*\text{diff} - starred for difference plots only}
For a difference plot:

`var1` - function type to plot, i.e. `rhoo`, `lapp` or `lapm`  
`var2` - name of the first file (from PL2D general instruction)  
`var3` - name of the second file (from PL2D general instruction)

Otherwise:

`var1` - name of the file (from PL2D general instruction)  
`var2` - function type to plot, i.e. one of the following: `rhoo (ρ)`, `lapp (∇²ρ)`, `lapm (−∇²ρ)`, `grho (|∇ρ|)`, `trajgrad (Vρ(r)) trajectories)`, `molgraph (molecular graph)`, `trajmolg (molecular graph and Vρ(r) trajectories)`, `rhoomolg (ρ and molecular graph)`, `lappmolg (∇²ρ and molecular graph)`, `lapmmolg (−∇²ρ and molecular graph).

### 10.6.4 Visualisation of 2D plots with program hp2xx or PRINTGL

The HPGL graphics files created by TOPXD can be visualized or converted to some other graphics formats by the UNIX program `hp2xx` (part of GNU software), which can be downloaded from:


At the time of writing, the latest version of `hp2xx` is 3.4.4 (23/06/2003).

**NOTE: Some of the newer versions of hp2xx have been reported to have problems with TOPXD files !!!** In this case, please download one of the older versions.

Once the `hp2xx` is installed, the HPGL file created by P2DCRY2000 can be visualized in graphical display using command:

```
hp2xx <name of P2DCRY2000 HPGL graphics file>
```

For better resolution one can use `-d dpi_value` option, i.e. the command:

```
hp2xx <name of P2DCRY2000 HPGL graphics file> -d 300
```

will display an image with 300 DPI rasterization.

In order to change the thickness of the lines in the image, one can use the `-p` option, which controls size (in pixels) of the virtual plotting pen. There are total 8 pens simulated. Each pen can be assigned a different size. Thus the command:

```
hp2xx <name of P2DCRY2000 HPGL graphics file> -p 43568111
```

requests size 4 for pen 1, size 3 for pen 2, size 5 for pen 3, size 6 for pen 4, size 8 for pen 5, and size 1 for pens 6,7 and 8.

It should be noted that TOPXD uses different pens to draw different objects in the 2D drawing. The assignment of pens is as follows:

<table>
<thead>
<tr>
<th>Pen number</th>
<th>Corresponding object in the drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>contours (and Vρ(r) trajectories)</td>
</tr>
<tr>
<td>2</td>
<td>dash lines</td>
</tr>
<tr>
<td>3</td>
<td>bond critical point position</td>
</tr>
<tr>
<td>4</td>
<td>bond path</td>
</tr>
</tbody>
</table>

115
Each pen can also be assigned a different colour using the option `-c`. The use of this option is similar to that of `-d`, i.e. for each pen one has to specify a colour number instead of a size. The colour-coding scheme is as follows: 0=off, 1=black, 2=red, 3=green, 4=blue, 5=cyan, 6=magenta, 7=yellow. Thus, the command:

```
hp2xx <name of HPGL graphics file> -c 276431
```

will produce a plot with pen 1 (contours, if any) being drawn in red colour, pen 2 (dashed lines, if any) in yellow colour, pen 3 (BCP position, if any) in magenta, pen 4 (bond path, if any) in blue, pen 5 (nuclei positions) in green, pen 6 (plot info and border) in black colour.

`hp2xx` also provides an option (`-r rotation_angle_in_degrees`) to rotate the object (image) prior to all scaling operations. Thus, the command `-r 90` rotates the entire picture on 90°, converting from portrait to landscape format and vice versa. In general, any reasonable rotation angle is valid.

In addition to visualization, `hp2xx` also provides an option to convert HPGL graphics file to more common and more supported vector and/or raster formats. There two options which control the output format type (`-m format`) and output filename (`-f filename`).

From vector formats the most useful is the PostScript (`-m eps`) and

From raster formats the most useful is probably PCX (`-m pcx`) since the image in this format can be easily inserted into Microsoft Word documents without any modifications. An example of a PCX image is shown in Figure 10-1

```
where PL2Dform is the name of the P2DCRY output file and form.pcx is the hp2xx output PCX image.
```

There are many other useful options provided in `hp2xx` program. Please refer to `hp2xx` manual pages for more information.

For Windows users, the free software PRINTGL is available for viewing and converting HPGL files. See

```
http://www.concentric.net/~ravitz/
```

This program has many of the features discussed above for hp2xx

<table>
<thead>
<tr>
<th>Pen</th>
<th>purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>nuclear positions</td>
</tr>
<tr>
<td>6</td>
<td>plot info and border</td>
</tr>
<tr>
<td>7</td>
<td>atomic basin boundaries</td>
</tr>
<tr>
<td>8</td>
<td>not used</td>
</tr>
</tbody>
</table>
Figure 10-1 $\rho(r)$ trajectories and molecular graph of formamide molecule in the crystal created from TOPXD/P2DCRY2000 data with hp2xx. The $\nabla \rho(r)$ trajectories are shown with red lines, nuclear positions are marked with + (black), BCPs are marked with * (magenta), bond paths are shown with blue lines.

10.7 3-Dimensional (3D) plots

The 3D plot section concerns the evaluation of a number of scalar functions on a 3D grid. The data so obtained may be used for 3D representations of specific envelopes of the scalar function.

The 3D plot instructions have the format, which is quite similar to 2D plots. The general format consists of the following two sections:

- General instructions
- Specific plot instruction(s)

The PL3D general instructions apply to all the specific plot instructions PL3D plot until the next PL3D general is given and so on. There is no limit on neither the number of PL3D general nor PL3D plot instructions.
As of this version, TOPXD writes all 3D grid files in XDGRAPH format if plot is defined in Cartesian coordinate system. It also writes output files in the SciAn [14] input format, but unfortunately this program appears to be no longer available.

10.7.1 3D plot general instruction section

This section MUST precede specific plot instructions. All parameters specified in this section will apply to the following specific plots until the next general instruction section is given.

**PL3D general [car | fra]**

- \( \text{xmin} \) \( \text{xmax} \) \( \text{xstep} \)
- \( \text{ymin} \) \( \text{ymax} \) \( \text{ystep} \)
- \( \text{zmin} \) \( \text{zmax} \) \( \text{zstep} \)
- **name** ‘name’

The format is exactly as laid out above, i.e. each command starts a new line. See Chapter 12 for an example.

\( \text{xmin} \) \( \text{xmax} \) \( \text{xstep} \)

Minimum and maximum values and grid interval along the crystal \( \text{X} \)-axis (fra) (in fractional units) or cartesian \( \text{X} \)-axis (car) (in Ångstroms).

\( \text{ymin} \) \( \text{ymax} \) \( \text{ystep} \)

These are similarly defined along the \( \text{Y} \)-axis and \( \text{Z} \)-axis respectively.

**name** **name**

Character variable name included in single quotes " " (maximum 24 characters, no blanks and no special symbols like ',',"^ etc ) defines the part of the name of files containing the values of each computed function func (see 3D plot Specific Instructions ). Full names of the files are listed in the table below ("//" means character concatenation).

<table>
<thead>
<tr>
<th>Scalar function type</th>
<th>File names</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SciAn SY format</td>
</tr>
<tr>
<td>Electron density ( \rho (r) )</td>
<td>3DRHOO//name</td>
</tr>
<tr>
<td>Laplacian ( \nabla^2 \rho (r) )</td>
<td>3DLAPP//name</td>
</tr>
<tr>
<td>Negative of the Laplacian ( -\nabla^2 \rho (r) )</td>
<td>3DLAPM//name</td>
</tr>
<tr>
<td>Magnitude of the gradient ( \nabla \rho (r) )</td>
<td>3DGRHO//name</td>
</tr>
</tbody>
</table>

10.7.2 3D plot specific instructions

These instructions must follow the general 3D plot instructions.

**PL3D \((*)\text{plot } \text{func}(1)\ldots \text{func}(n)\)**

**NOTE** There is no limit for **PL3D plot** instructions and number of \text{func} instructions in each line, except the 256 character string limitation in the latter case.

\((*)\text{plot} \) – If starred, then plot the specified functions (see below)
func(1)…func(n) - function type(s). The following function types are recognized:

- **rhoo** - electron density ($\rho(r)$)
- **lapp** - laplacian of the electron density ($\nabla^2\rho(r)$)
- **lapm** - negative laplacian of the electron density ($-\nabla^2\rho(r)$)
- **grho** - magnitude of the gradient of electron density ($|\nabla\rho(r)|$)

10.8 3-Dimensional (3D) visualization of atomic basins

As of this version, TOPXD contains options for 3D visualization of atomic basins in XDGRAPH from the results of atomic basin integration (see Section 10.4). If the keyword `SavSurf` in `ATBPParams` directive (see Section 10.4) is starred, then during the determination of the ZFS, the integration ray data are saved to the file `rays.dat` for each integrated atom. The `VZ3D` section provides the interface to XDGRAPH for the visualization of that data.

```markdown
VZ3D (*`plot
files  file(1) file(2) … file(n)
basins label(1) label(2) ... label(n)
range  (*default  xmi xni xma xma ymi ymi yma yma zmi zmi zma zma
grid   (*default  dx dy dz dz rvec (*`default  rvec END VZ3D
```

The format is exactly as laid out above, i.e. each command starts a new line. See Chapter 12 for an example.

```markdown
files  file(1) file(2) … file(n)
```
Names of files from which the integration ray data will be read. If integration of all atoms for which the basins will be plotted was done in one directory, then only one `rays.dat` file should be specified. If integration of atoms was done in separate directories and/or on different computers, then files `rays.dat` should be renamed and specified one by one in this instruction.

```markdown
basins label(1) label(2) ... label(n)
```
Labels of unique atoms for which the basins will be plotted. The integration ray data for these atoms should be present in one of the files read with `files` directive, otherwise the atom will be ignored.

```markdown
range  (*default  xmi xmi xma xma ymi ymi yma yma zmi zmi zma zma
```
Definition of limits of the Cartesian grid (in Å) generated for visualization in XDGRAPH. If keyword `default` is starred, these limits will be automatically determined based on the data read from all files specified in `files` directive.

```markdown
grid   (*default  dx dy dy dz dz rvec (*`default  rvec
```
Definition of the grid spacings `dx dy` and `dz` of the Cartesian grid (in Å). If keyword `default` is starred, the default value of 0.1 Å will be used for `dx dy` and `dz`. The parameter `rvec` specifies the radius of the sphere (in Å) centered on each grid point. If the boundary of the atomic basin lies within the `rvec` Å from the grid point, it is assumed that this grid point also belongs the atomic basin boundary.

The output 3D grid file from the VZ3D section of TOPXD (`basin.grd`) can be readily visualized in XDGRAPH (see example below).
Atomic basin of O(1) atom in methyl carbamate
10.9 Description of the atomic properties evaluated by TOPXD

### Populations

<table>
<thead>
<tr>
<th>Property</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic population</td>
<td>$N$</td>
<td>$N(\Omega) = \int_\Omega \rho(r) , d\tau$</td>
</tr>
<tr>
<td>Net charge</td>
<td>$\Omega$</td>
<td>$q(\Omega) = Z(\Omega) - N(\Omega)$</td>
</tr>
</tbody>
</table>

### Energies

<table>
<thead>
<tr>
<th>Energy</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic Lagrangian</td>
<td>$L$</td>
<td>$L(\Omega) = -1/4 \int_\Omega \nabla^2 \rho , d\tau$</td>
</tr>
<tr>
<td>Atomic value of nuclear-electron potential energy</td>
<td>$V_{NEO}$</td>
<td>$V_{NEO}(\Omega) = -\int_\Omega (Z(\Omega)/r(\Omega)) \rho(r) , d\tau$</td>
</tr>
</tbody>
</table>

### Atomic Forces

<table>
<thead>
<tr>
<th>Force</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic force components</td>
<td>$F_{AXA}$</td>
<td>$F_{AXA}(\Omega) = \int_\Omega (Z(\Omega)/r(\Omega)^3) x(\Omega) \rho(r) , d\tau$</td>
</tr>
<tr>
<td>Force on nucleus of atom $\Omega$ by the electron density of atom $\Omega$</td>
<td>$F_{AYA}$</td>
<td>$F_{AYA}(\Omega) = \int_\Omega (Z(\Omega)/r(\Omega)^3) y(\Omega) \rho(r) , d\tau$</td>
</tr>
<tr>
<td>Force on nucleus of atom $\Omega$ by the electron density of atom $\Omega$</td>
<td>$F_{AZA}$</td>
<td>$F_{AZA}(\Omega) = \int_\Omega (Z(\Omega)/r(\Omega)^3) z(\Omega) \rho(r) , d\tau$</td>
</tr>
</tbody>
</table>

### 'Radial' Atomic expectation values

<table>
<thead>
<tr>
<th>Expectation</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic expectation value of the operator $r^\alpha_{\Omega}$</td>
<td>$R(-1)$</td>
<td>$R(-1)(\Omega) = \int_\Omega \rho(r) , d\tau$</td>
</tr>
<tr>
<td>Atomic expectation value of $x^\alpha_{\Omega}$ averaged over $r^\alpha_{\Omega}$</td>
<td>$GR(-1)$</td>
<td>$GR(-1)(\Omega) = \int_\Omega r^\alpha_{\Omega} \rho(r) , d\tau$</td>
</tr>
</tbody>
</table>

### Atomic volumes and related populations

<table>
<thead>
<tr>
<th>Volume</th>
<th>Symbol</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume of the region of the atomic basin where $\rho(r)$ is greater or equal 0.001 au</td>
<td>$V_{001}$</td>
<td>$V_{001}(\Omega) = \int_\Omega d\tau_{0.001}$</td>
</tr>
<tr>
<td>Electron population in the $V_{001}$ region</td>
<td>$N_{001}$</td>
<td>$N_{001}(\Omega) = \int_\Omega \rho(r) , d\tau_{0.001}$</td>
</tr>
<tr>
<td>Ratio of electron populations in $V_{001}$ and in the atomic basin</td>
<td>$R_{001}$</td>
<td>$R_{001} = N_{001}/N$</td>
</tr>
<tr>
<td>Volume of the region of the atomic basin where $\rho(r)$ is greater or equal 0.002 au</td>
<td>$V_{002}$</td>
<td>$V_{002}(\Omega) = \int_\Omega d\tau_{0.002}$</td>
</tr>
<tr>
<td>Electron population in the $V_{002}$ region</td>
<td>$N_{002}$</td>
<td>$N_{002}(\Omega) = \int_\Omega \rho(r) , d\tau_{0.002}$</td>
</tr>
<tr>
<td>Ratio of electron populations in $V_{002}$ and in the atomic basin</td>
<td>$R_{002}$</td>
<td>$R_{002} = N_{002}/N$</td>
</tr>
<tr>
<td>Total atomic volume</td>
<td>$V_{TOT}$</td>
<td>$V_{TOT}(\Omega) = \int_\Omega d\tau$</td>
</tr>
<tr>
<td>Atomic unambiguously moments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic dipole components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DX, DY, DZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DX(\Omega) = \int_{\Omega} \rho(r)x_\Omega d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DY(\Omega) = \int_{\Omega} \rho(r)y_\Omega d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DZ(\Omega) = \int_{\Omega} \rho(r)z_\Omega d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Components of atomic displacement vector</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCX, DCY, DCZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DCX = -DX/N(\Omega) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DCY = -DY/N(\Omega) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( DCZ = -DZ/N(\Omega) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinates of the centroid of negative charge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CX, CY, CZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( CX = DCX + X_\Omega )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( CY = DCY + Y_\Omega )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( CZ = DCZ + Z_\Omega )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic 2\textsuperscript{nd} moment tensor components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX, QXY, QXZ, QYY, QYZ, QZZ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QXX(\Omega) = \int_{\Omega} x_\Omega^2 \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QXY(\Omega) = \int_{\Omega} x_\Omega y_\Omega \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QXZ(\Omega) = \int_{\Omega} x_\Omega z_\Omega \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QYY(\Omega) = \int_{\Omega} y_\Omega^2 \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QYZ(\Omega) = \int_{\Omega} y_\Omega z_\Omega \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( QZZ(\Omega) = \int_{\Omega} z_\Omega^2 \rho(r) d\tau )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic Shannon information entropy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atomic information (missing information function)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \text{IUN} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I(\Omega) = \int_{\Omega} \rho \ln \rho^* d\tau )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 10.10 Frequently Asked Questions

**Q:** When should I use the Newton-Raphson (NR) and when the Eigenvector Following (EF) method?

**A:** In general, the NR method is only suitable for the location of a critical point if one is already in a region where the Hessian of \( \rho(r) \) has the correct structure [6]. Thus, the NR method will fail to find a ring critical point if the starting point has the same eigenvalue signs as a bond critical point. The EF method has proved to be much less sensitive to the choice of good starting search points. The EF method, in practice a NR method with a suitable and locally defined shift for the NR step, seeks for the critical points of a given type, independently of the structure of the Hessian at the starting point. This is particularly important in the case of the \( \nabla^2 \rho(r) \) field, since this scalar function varies quite rapidly.
Q: When integrating an atomic basin with TOPXD I get the following error message:

**PATHE2: OSCILLATION OF PATHS**

**PATHE2: THE ATTRACTOR OF THIS PATH WAS PROBABLY NOT INCLUDED IN THE CLUSTER**

A: Check the list of atoms reached in the feeler rays determination step. If you think that some neighbouring atoms were missed, you may have to increase parameter \( nvi \) in order to include the missing atoms into the list of possible attractors of the \( \nabla \rho(\mathbf{r}) \) trajectories. Once you have used a very large \( nvi \) value, leave your calculation to try to end its task (even if the message appears many times).

The OSCILLATION OF PATHS message may also appear in some cases where the integration will be anyhow successful. In many instances it represents just a warning. Especially, if you noticed that the list of neighbouring atoms (after the feeler ray step) corresponds to your expectations.

Q: What grid should be used for integration of atomic basins and how does it affect the computing time?

A: In order to obtain satisfactory results you should use something like:

- \( 64 \times 48 \times 120 \) (\( \Phi \times \Theta \times \)radial) for non-H atoms
- \( 32 \times 24 \times 96 \) (\( \Phi \times \Theta \times \)radial) for hydrogen atoms (if not involved in H-bond)
- \( 48 \times 32 \times 96 \) (\( \Phi \times \Theta \times \)radial) for hydrogen atoms (if involved in H-bond)

Note that the number of angular points and of radial points refers to the integration within beta sphere and outside beta sphere, respectively. The computational time is roughly proportional to \( n\Phi \times n\Theta \). The number of radial points is very important for the precision, but hardly affects the total integration time, as it is operative only in the integration step and NOT in the ZFS determination (which takes about 95% of the total time).

Q: Integration of an atomic basin takes a very long time. What options do we have to speed up the calculation?

A: Unfortunately the integration step is very long (especially the ZFS determination which takes about 95% of this time). You can try with the other proposed method, which is much faster but often fails.

Using the indirect method you can save some time by decreasing the accuracy of the surface determination. It is set as a default to 0.001 (see parameter \( \text{accur} \)). You could try to increase it up to 0.003 (but no more than 0.005). You lose somewhat in precision, but you certainly increase in speed. You could compare the results of these two computations on one of the atoms you have already integrated, \( N(\Omega), L(\Omega), \) etc. using:

1. first test : \( 64 \times 48 \times 120 \) \( \text{accur} =0.001 \)
2. second test : \( 64 \times 48 \times 120 \) \( \text{accur} =0.003 \)

Then you can decide if it is worth varying such a parameter and how much you can vary it.

Q: How do I check the accuracy of the integration?

A: Check the value of the integrated Lagrangian. For an "exact" integration it should vanish (for the divergence theorem). In practice:

1. it should be less than \( 5 \times 10^{-7} \) for H atoms, possibly around \( 1 \times 10^{-5} \). A value of \( 1 \times 10^{-4} \) could be perhaps acceptable, but not too precise.
2. for second row atoms (C,N,O, etc) it should not exceed \( 1 \times 10^{-3} \). Possibly \( 1 \times 10^{-4} \)
Q: You’ve mentioned that the computing time increases by a factor of $\phi \times \theta$ planes, but how does the \textit{nvi} parameter affect the elapsed time?

A: It will affect it, but in a very limited way, especially after the feeler ray step. Indeed the atoms reached during the feeler ray step are put at the top of the list of the \textit{nvi} reachable atoms. So that the DO loop in PATHEN and PATHEN2 (these DO’s run on the $3 \times nvi \times \text{star\_multiplicity}$ coordinates of the possible $\nabla \rho (r)$ attractors) are in most cases (>99%) terminated much before the end of the loop.

In practice you should not notice a CPU time increase with \textit{nvi} increase. Rather you could notice a decrease, if you have added an attractor that had to be enclosed. In this case the path oscillation is avoided and CPU time considerably saved.

Q. Sometimes I have problems with the integrated Lagrangian, which stays above $1 \times 10^{-3}$ despite the fact that I use $\text{accur}=0.001$ and $n\phi \times n\theta \times n\phi \times n\theta$ as large as $96 \times 64$. I remember that you have mentioned that decreasing the number of points might help, but when I reduce these numbers to $48 \times 32$ or $64 \times 48$ it still does not help. These problems usually occur with carbon and nitrogen atoms, never with oxygens or hydrogens. What do I do?

A: What about electroneutrality? Are you very far from it? The fact that one may get problems with carbon or nitrogen atoms and never with oxygen or hydrogen atoms seems to indicate, that the former have more complicated ZFSs than the latter (at least in the systems you are presently investigating). You could try to solve such a problem, by increasing the radius of the beta sphere for such atoms, thus reducing the size of the remaining part of the atomic basin. You could use for the beta sphere something like the distance of the closest bond critical point multiplied by 1.15 (the program then reduces this number by 20%). Furthermore, the increase (inside the code) of the number of theta and phi points in the inner beta sphere might help. Please contact us and we will send you instructions on how to do it...

Bibliography

Chapter 11
XD Utility programs

11.1 XDVIB1 - A Program to Calculate the Mean-Square Displacement Amplitudes from Harmonic Vibrational Frequencies and Normal Modes.

11.1.1 Overview

The XDVIB programs described in this chapter were developed to improve the significance of the ADPs obtained by LS refinement against X-ray data. This can be done by incorporating independent information based on spectroscopic and/or theoretical calculations into the refinement in terms of constraints or restraints.

A full description of nuclear motion in molecular crystals, within the mean-field approximation, is an \( M \)-parameter problem, where \( M \) is the number of elements of the symmetric mean square displacement amplitude matrix to be derived (\( M=3N(3N+1)/2 \), \( N \) being the number of nuclei in the molecule). This is a hopeless task, given the fact that even in an optimal case, only \( 6N \) ADPs are available from a diffraction experiment. Consequently, further approximations and independent data are needed. A feasible approximation is to neglect the coupling between relative motion of the nuclei (internal modes) and the overall motion of the molecule in the lattice (external modes, 3 translation and 3 rotation) yielding a reduction in the number of free parameters by \( 6n \), where \( n \) is the internal degree of freedom (\( n=3N-6 \) for nonlinear and \( 3N-5 \) for linear molecules). The knowledge of harmonic frequencies alone would further reduce the number of unknowns by \( N \). A complete knowledge of the internal vibrational modes, including the frequencies, \( (n(n+1)/2 \) elements of the corresponding MSDA matrix) leads to \( m=21 \) (6x7/2) parameters to be derived (\( m < 21 \) from molecules composed of \( N < 8 \) atoms). However, to estimate \( 6N \) ADPs (the diagonal, symmetric blocks of the total MSDA matrix) only \( 6N-m \) independent parameters, associated with internal modes are needed. But how to chose these remaining parameters?

When neglecting internal-external coupling, each ADP can be given as a sum of two terms:
\[
U_a = V_a + W_a
\]

where \( V_a \) and \( W_a \) are the MSDA tensors (ADP’s) of atom \( a \) corresponding to the internal and external modes, respectively. The latter term can be adequately accounted for in terms of the \( m \) parameters of the \( T,L \) and \( S \) tensors of the rigid-body model [1]:
\[
W_a = R_a^T LR_a + R_a^T S + S^T R_a + T
\]

where \( R_a \) is an antisymmetric tensor representing vector product
\[
R_a = \begin{pmatrix}
0 & -y_a z_a \\
y_a & 0 - x_a \\
- z_a x_a & 0
\end{pmatrix}
\]

with \( x_a, y_a \) and \( z_a \) being the Cartesian coordinates of nucleus \( a \) situated at position \( r_a \).
Since \( \mathbf{V} \) and \( \mathbf{W} \) are additive, they cannot be obtained independently via an LS refinement, but their different behaviour upon a special transformation can be used as extra information. It can be easily shown that the difference MSDA corresponding to \( \mathbf{W} \) (the difference of the components of \( \mathbf{W}^* \) and \( \mathbf{W} \) along the internuclear vector \( \mathbf{d}_{ab} = \mathbf{r}_b - \mathbf{r}_a \)) vanishes for all \( a-b \) links:

\[
\Delta(W)_{ab} = \mathbf{d}_{ab} \cdot (W_a - W_b) \mathbf{d}_{ab} = 0
\]

but does not necessarily for \( \mathbf{V} \). This property of the internal ADPs can be imposed as a restriction in the LS fit. Let us suppose we know \( \mathbf{V}_a \) for all atoms, start the refinement with these values and constrain the shifts in the ADPs according to the above requirement for a certain number of independent links:

\[
\Delta(\delta \mathbf{U})_{ab} = 0
\]

After the \( k \)-th cycle we obtain

\[
\mathbf{U}_{a}^{(k)} = \mathbf{V}_a + \delta \mathbf{U}_a^{(k)}, \quad \mathbf{U}_{b}^{(k)} = \mathbf{V}_b + \delta \mathbf{U}_b^{(k)} \quad \text{with} \quad \Delta(\delta \mathbf{U})_{ab}^{(k)} = \Delta(\mathbf{V})_{ab}
\]

Thus, the information on the intermolecular vibrations are preserved during the refinement. There are \( l = N(N-1)/2 \) links. Singular value decomposition of the matrix of constraints eliminates linear dependencies, yielding, in general, \( 6N - m \) independent constraints.

**XDVIB1** calculates MSDAs (the total \( 3N \times 3N \) \( \mathbf{V} \) matrix) from \( n \) frequencies and normal modes obtained by the *Gaussian* program package. Details of the vibrational analysis implemented in *Gaussian* are described in reference [2]. There is an important point to make concerning these calculations. Harmonic vibrational analysis based on theoretical force fields is valid only if the gradient of the electronic energy with respect of nuclear coordinates vanishes (equilibrium geometry) and the force-constant matrix (the Hessian of the energy or the second derivatives) is taken at the equilibrium configuration. Thus the molecular geometry used in the vibrational analysis must be optimized at the same level of theory, and utilizing the same basis set that the derivation of the frequencies is based on.

The *Gaussian* program works in terms of mass-weighted generalized coordinates to separate the internal (vibration) and external (translation and rotation) motion. The internal coordinates \( \mathbf{v} \) (\( n \)-vector) are generated to be orthogonal to the external ones, \( \mathbf{w} \) (\( 6 \)-vector related to the Eckart coordinates). The output displays the wavenumber (\( \nu \)), the reduced mass (\( \frac{\mu}{\hbar} \)) and the normalized Cartesian displacement vector for each normal mode (\( \mathbf{q} \)). The MSDA matrix is diagonal in terms of normal coordinates [3]

\[
< \mathbf{q} \mathbf{q}' > = \text{diag} (\delta_1, \delta_2, \delta_3, \cdots, \delta_n)
\]

where

\[
\delta_k = \frac{\hbar}{8\pi^2 c v_k} \coth \left( \frac{\hbar c v_k}{2kT} \right)
\]

while in Cartesian representation

\[
< \mathbf{v} \mathbf{v}' > = \mathbf{L} < \mathbf{q} \mathbf{q}' > L' , \quad \mathbf{q} = \mathbf{L} \mathbf{u}
\]

**11.1.2 Files used and created by XDVIB1**

Input: xd.mas, *Gaussian* frequency output
Output: xd_vib1.par, xd_vib1.sig xd_vib1.out
1.1.3 Input instructions for XDVIB1

11.1.3.1 SELECT

**SELECT temp 100 scale 1. (**nonlin linear**

**temp** temperature

The temperature (K) maintained during the data collection.

**scale** scalefactor

A number used to scale the calculated frequencies in wavenumbers [cm$^{-1}$], printed in the Gaussian output file, gaussian.out. Frequencies calculated at different level of theory and basis set are scaled by empirical factors to eliminate known systematic errors. Selected values taken from reference [2] are listed below. If experimental normal frequencies (corrected for anharmonicity) are available, one can scale the calculated ones directly to those observed.

<table>
<thead>
<tr>
<th>Method/basis</th>
<th>Scale factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>HF/3-21G</td>
<td>0.9085</td>
</tr>
<tr>
<td>MP2(Full)/6-31G(d)</td>
<td>0.9427</td>
</tr>
<tr>
<td>SVWN/6-31G(d)</td>
<td>0.9833</td>
</tr>
<tr>
<td>B3LYP/6-31G(d)</td>
<td>0.9613</td>
</tr>
<tr>
<td>HF/631-G(d)</td>
<td>0.8928</td>
</tr>
<tr>
<td>MP2(FC)/6-31G(d)</td>
<td>0.9434</td>
</tr>
<tr>
<td>BLYP/6-31G(d)</td>
<td>0.9940</td>
</tr>
</tbody>
</table>

11.1.3.2 MODES

**MODES (**all frqmin frqmin frqmax frqmax**

!MODES include 1 2 3 ...

!MODES exclude 1 2 3 ...

By these commands the vibrational modes used in the ADP calculation are selected. Only one type of **MODES** command is allowed.

**all**

The default option includes all normal modes.

**frqmin frqmin frqmax frqmax**

If the command is starred, wavenumbers in the range of $frqmin < \cdot < frqmax$ are included. The default upper cutoff value of 1500 cm$^{-1}$ limits the calculation to “soft” (large-amplitude) modes.

**include 1 2 3 ...**

The normal modes are listed in the gaussian.out file in order of increasing wavenumbers. The **include** command allows the user to select a set of normal modes according to their order number on the list. Those not listed in the command line will be ignored.

**exclude 1 2 3 ...**

The listed modes will be excluded. For the **MODES** command with **include** or **exclude** options more than one line can be entered (but not mixed!).
11.3.2.3 DATAFILE

DATAFILE *gaussian filename orient *standard

This command will be used to identify different files containing the calculated vibrational data.

*gaussian filename

The only type of data file allowed in the present version of XDVIB1 is a gaussian type output file. Its name is given as a character string filename.

orient *standard input

Atomic positional coordinates corresponding to the standard orientation of the molecule (the origin is placed at the center of molecular nuclear charge) are used. If the coordinates are read from the checkpoint file of a previous geometry optimization, they are listed as "Input Orientation" of the gaussian.out file of the frequency job, and in this case the input option should be starred.

11.2 XDVIB2 – A Program to Transform ADPs from Cartesian Systems to the Crystal Frame.

11.2.1 Overview

As mentioned above, the internal ADPs ($V$) calculated by XDVIB1 refer to the equilibrium molecular geometry. The conformation of the optimized molecule can considerably differ from that found in the crystal (experimental conformation). However, the comparison of $a_{ab}$ values corresponding to short links in structurally analogous molecules reveals only a moderate conformational dependence. For bonds formed by atoms of comparable nuclear mass, $a$ is invariant under rotation about the bond vector. 1-3 links show a similar transferability. This local symmetry of the ADPs is made use of when they are transformed from Cartesian (optimized molecule) to the crystal (experimental frame) coordinate system.

The procedure starts with generation of atomic local frames defined in the same way as for the spherical harmonics in the multipole model. XDVIB2 does that automatically, using the atomic connectivity of the input molecule or fragment. The calculated ADP’s are then transformed into the local frames

$$V_a = M_a V a M_a^T$$

where $M_a$ is an orthogonal 3x3 matrix whose column vectors are the components of the local basis vectors in the Cartesian system. The same transformation is applied to the experimental ADPs :

$$U_a = N_a O U_a O^{-1} N_a^T$$

where $N_a$ is based on the local connectivity of the experimental geometry transferred to a Cartesian system via the matrix $O$. In the last step the calculated ADP's are transferred to the crystal frame

$$V_a = P_a^{-1} U_a P_a$$

$P_a = N_a O$

The program prints out the full $\bullet(V)$ matrices in both representation (Cartesian and crystal). Their comparison can suggest a possible model for the segmentation that is applicable to the molecule during the LS refinement. If

$$\Delta_{ab}(\hat{V}) \equiv \Delta_{ab}(V)$$
for all links, the application of a full set of independent rigid-link constraints is feasible. Otherwise, the user should try to identify rigid groups and limit the constraints to intragroup links. The former approach corresponds to a rigid-body, while the latter to a segmented rigid-body model refinement.

11.2.2 Files used and created by XDVIB2

Input: xd.mas, xd_vib2.inp, xd.inp
Output: xd_vib2.res, xd_vib2.out

11.2.3 Input instructions for XDVIB2

There are no input instructions for XDVIB2. The program reads the xd_vib2.inp file containing the Cartesian atomic parameters (as calculated and output by XDIB1) for as many molecules or atomic groups as many is needed to build the unit cell. The molecules are separated by a GROUP command line. Let us suppose there are two independent molecules in the unit cell and we completed the two Gaussian calculations, each followed by an XDVIB1 run. The xd_vib1.par files obtained for the two calculations can than be merged to create the xd_vib2.inp files. Since the Cartesian-crystal transformation is done through the local systems, the parameters of the different groups can be given in different (but orthogonal) frames. This makes it easy to build a database of calculated ADPs.

Bibliography

2. J. W. Ochterski, Vibrational Analysis in Gaussian (1999) help@gaussian.com
11.3 XDCIF - A Program to Create an Archive CIF

XDCIF is a program to combine the CIFs produced by XD into an archive CIF called xd_archive.cif, which is suitable for submission to journals or deposition databases. Currently XD programs write the following CIFs: xd_lsm.cif, xd_fft.cif and xd_geo.cif. XDCIF will load these files (if found) and, with a view to future enhancements, will also look for the files xd_fou.cif and xd_pro.cif (these are not currently produced by XD). In addition, a file xd_dat.cif is also sought. This last file is not produced by the XD programs, but must be supplied by the user. It should contain those details of the study which are not accessible in any of the other CIFs. Such information could include unit cell dimensions and errors, crystal size, crystal colour, space group symbols, data reduction details etc. A suitable file could be that from a SHELX refinement or a WinGX archive CIF, but of course with all the details of the refinement and structural geometry removed.

XDCIF also needs to read a request file called xdcif.dat which must be placed in the directory pointed to by the environment variable XD_DATADIR (normally also the location of the XD databank files). If the environment variable XD_DATADIR is not set, the program will halt. This request file is user configurable and should include all the CIF data items which the user wishes to include in the final xd_archive.cif. Details of the syntax for these entries is given in the header to the default version of this file which is included in the XD release. Note that it is not important if a data item in this request list is not found in any CIF, but if a data item is not present in the request list, then it cannot be included in xd_archive.cif.

The program differs from other XD programs in being interactive. It will first ask the user if all the CIFs found in the working directory should be included in the output CIF. It then cycles through the items in the request list. If a data item is not found in any CIF, the user is given the opportunity of entering the data value manually. Alternatively, the user may respond "a" (for automatic mode), when the program will continue without further prompts.

11.3.1 Files used and created by XDCIF

Input: xd_*.cif (* = fft, lsm, geo, dat, fou, pro)
Output: xd_archive.cif
11.4 XDWTAN - A Program to Analyse the Weighting Scheme

XDWTAN analyses the structure factor file xd.fco written by XDLSM and provides a listing of discrepant reflections and tables of \( \omega(F_o-F_c)^2 \) and \( R \) values as functions of \( hkl \) indices, index parities, \( F_o \) and \( \sin \theta/\lambda \). It is based on the GX program WTANAL [1].

The reader should note that:

- the XDLSM refinement may be based on either \( F \) or \( F^2 \) (a user choice).
- regardless of the choice, the xd.fco file always contains \( F_o^2, F_c^2 \) and \( \sigma(F_o^2) \).
- the weighting analysis is always based on \( F_o, F_c \) and \( \sigma(F) \).

The calculated weights used in XDWTAN are based on these considerations, but may not necessarily exactly correspond with those used by XDLSM, due to approximations used in the program.

The problems of choosing the correct weights for the observations in the least squares procedures are well known and of course not restricted to multipole refinements. See reference [2] for a discussion of this topic. The weighting scheme used by XDLSM is the same as that utilized in SHELXL [3] when refinement is based on \( F^2 \), and the weight is based on this when refining against \( F \) (see Chapter 4). Ideally the weights should be chosen such that \( \langle (\Delta/\sigma)^2 \rangle \) is unity \((\Delta=Y_c-Y_o)\), but this can rarely be achieved. This condition corresponds to \( \langle (\omega \Delta^2) \rangle \sim 1 \), where \( \omega \) is the statistical weight equal to \( 1/\sigma(Y) \) and the so-called goodness of fit parameter (Gof or \( S \)) defined as

\[
\text{Gof (S)} = \left[ \Sigma \omega(F_o-F_c)^2 / (\text{nobs} - \text{npar}) \right]^{1/2}
\]

in XDWTAN provides a measure of this. The Gof value is generally greater than unity and this may be due either to an inadequacy of the model or to an underestimation of the observational errors. In XDWTAN, for the listing of discrepant reflections with \( \Delta/\sigma(F) \) greater than 3.0, the overall Gof is used as an effective scale factor for the weights. Ideally there should be little variation of \( \langle (\omega \Delta^2) \rangle \) with \( hkl \) indices, index parities, \( F_o \) and \( \sin \theta/\lambda \). The analysis is carried out firstly for all reflection data and secondly for only those reflections which were used in the least-squares refinement.

11.4.1 Files used and created by XDWTAN

Input: xd.fco, xd.mas
Output: xd_wta.out
11.5 AIM2TAB

The program AIM2TAB is interfaced to XD, but is distributed separately. Please send an email to Dr. Anatoliy Volkov (volkov@chem.buffalo.edu) to request the program.

AIM2TAB reads TOPOND9x, TOPXD or AIMPAC atomic integration files and calculates total molecular moments in original Cartesian coordinate system and Gaussian9x-like system using integrated atomic moments. AIM2TAB also prints out extensive information on other integrated properties, like atomic volumes, moments and integrated Lagrangians, etc. The program requires the file files.dat with names and locations of TOPOND/TOPXD/AIMPAC output files (1 line per filename).

There are several optional input files:

- **trans.dat** to transform the coordinates (and atomic dipoles) of integrated atoms using rotation matrices and translation vectors in crystal coordinate system (1 line per atom):
  
  \[ R_{11} \, R_{12} \, R_{13} \quad R_{21} \, R_{22} \, R_{23} \quad R_{31} \, R_{32} \, R_{33} \quad T_{1} \, T_{2} \, T_{3} \]

- **symm.dat** to generate symmetry-equivalents of integrated atoms - 1 line per each new symmetry-equivalent atom in the following format:
  
  \(<\text{atom label}> \quad R_{11} \, R_{12} \, R_{13} \quad R_{21} \, R_{22} \, R_{23} \quad R_{31} \, R_{32} \, R_{33} \quad T_{1} \, T_{2} \, T_{3}\)

  where \(<\text{atom label}>\) is the atomic label of the integrated "parent" atom, and \(R_{ij}\) and \(T_{i}\) are the rotations and translations in the crystal coordinate system, respectively.

- **new.dat** with additional information on atomic site occupations and number of electrons of the free atom (in the current version of AIM2TAB only atomic charges and volumes will be rescaled, NOT the higher moments). The format is (1 record per each atom):

  \(<\text{atom label}> \quad <\text{occupation}> \quad <n. \, \text{of electrons in free atom}>\)

11.5.1 Files used and created by AIM2TAB

Input:    TOPOND/TOPXD/AIMPAC outputfiles,  files.dat,  
           (trans.dat, symm.dat, new.dat)
Output:   aim2tab.out

11.6 LSDB

This is a program for automatic setup of the atomic local coordinate systems and chemical constraints starting from SHELX [3], PLATON [4] or XD structure files. The program is designed for interactive use and is self-explanatory. The atomic local coordinate systems can then be visualized in PLATON [4]. It is available upon request from Dr. Anatoliy Volkov (volkov@chem.buffalo.edu).

11.7 ADDGRID, SCALEGRID

These are two simple interactive routines to add and scale grid files produced by other XD sections. In particular, **ADDGRID** allows summing, substracting, multiplying, dividing,
scaling and applying exponents to a given number of grids. In this way, many properties derived (exactly or empirically) from the electron density and/or its derivatives can be visualized.

**Bibliography**

## Chapter 12

### Example input files

#### 11.1 Parameter file (XD.INP/XD.RES)

The new version of the parameter file has the following format:

```
!<<< X D PARAMETER FILE >>> $Revision: 4.04 (Feb 26 2003)$          23-Mar-03!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XDPARFILE VERSION  2
ICE8     MODEL   4 0 0 0
LIMITS nat  2000 ntx  31 lmx  4 nzz  30 nto  0 nsc  20 ntb  20 nov  2500
USAGE      2   0  4   2   0   1   0   1   2   8    19   0  10  1
-1.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
-0.000000  0.250000  0.193500
O(1)      3 2    3   1   2 0  1  1 4  1   0  0.000000  0.250000  0.107100
0.050000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
1.5653  0.0000  0.0000  0.0000 -0.0202 -0.0023  0.0000  0.0000  0.0213  0.0000
-0.0077  0.0000  0.0000 -0.0148  0.0000  0.0000 -0.0032  0.0000  0.0000  0.0000
-0.0045  0.0000  0.0000  0.0000  0.0015  0.0000
H(1)      1 2    3   2   1   0  2  2 2  1   0  0.000000  0.084300  0.193500  0.5000
0.050000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
0.4347  0.0000  0.0588  0.0435  0.0000 -0.0195  0.0000  0.0000  0.0129  0.0378
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
1  0.981203  1.038973  1.038973  1.038973  1.038973  1.038973  1.038973
2  1.128085  1.434702  1.434702  1.434702  1.434702  1.434702  1.434702
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.500000E+00
```

The old format style is still accepted and interpreted, though the user must be aware that some default definitions have been changed (see Section 2.5 for example) and therefore parameters refined with older version of XD (and their corresponding xd.mas file) may no longer be consistent.
Chapter 12 - Example Input Files

12.2 Master file

This is the new format of the master file. The old format is no longer readable by XD (due to the changes in SCAT table).
Chapter 12 - Example Input Files

!------------------------------------------------------------------------------
! TITLE ice8
! CELL 4.6560 4.6560 6.7750 90.0000 90.0000 90.0000
! WAVE 0.7107

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------
! BANK CR

!------------------------------------------------------------------------------
! SCAT CORE SPHV DEFV   1S   2S   3S   4S   2P   3P   4P   3D   4D   4F   5S   5P   6S   6P   5D   7S   6D   5F  DELF©   DELF©©  NSCTL
!------------------------------------------------------------------------------

!------------------------------------------------------------------------------
! ATOM     ATOM0    AX1 ATOM1    ATOM2   AX2 R/L TP TBL KAP LMX SITESYM  CHEMCON
!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------

!------------------------------------------------------------------------------
Chapter 12 - Example Input Files

OVTHP   0
SCALE   0
END KEY

END XDLSM

MODULE XDFFT
SELECT *fobs *fmod1 fmod2 snlmin 0. snlmax 2. sig 3. phase 0.
SELECT gridsize 0.2 scale 1. npeak 10 nhole 10 neutron gridf peakf
END XDFFT

MODULE XDFOUR
SELECT *fobs *fmod1 fmod2 print snlmin 0. snlmax 2.
GRID *3-points perp cryst
ATOM label ato(1) symm 1 trans 0 0 0 *mark on plot
ATOM label ato(2) symm 1 trans 0 0 0 *mark on plot
ATOM label ato(3) symm 1 trans 0 0 0 *mark on plot
LIMITS xmin -2.0 xmax 2.0 nx 50
LIMITS ymin -2.0 ymax 2.0 ny 50
LIMITS zmin 0.0 zmax 0.0 nz 1
END XDFOUR

MODULE XDPROP
MODEL iam *multipole
!APPLY symm 1 translations 0 0 0 ato(1) ato(2) ...
SELECT *local numdx check esd nocore
SELECT cpcut 0.0001 lmax 4 nstep 20 rcut 4.0
SELECT scale 0.05 dx 0.001 ds 0.005
SELECT radl 0.1 rad2 200. rad3 10. zone1 1 zone2 1
!GROUP ato(1) ato(2) ...
!DIPOLE *cmass center ucell
!QUADPOLE *cmass center ucell
!D-POP
PROPERTY *rho gradrho d2rho elpot core valence defden sigrho siglap esp
!QFIT grid 11 length 7.0 width 1.0
!CONSTRAIN ato(1) ato(2) ...
!STOKMOM defden lmin 0 lmax 4 *cmass center ucell debug
!STOKMOM minlim -3. -3. -3. maxlim 3. 3. 3. epsa 1.0d-4 epsr 1.0d-4
!STOKMOM atoms *all select ato(1) ato(2) ...
!POINT x y z
!LINE ato(1) ato(2) npts 50
!LINE points x1 y1 z1 x2 y2 z2 npts 50
!CUBE center x y z 30 0.1
!CUBE ato(1) ato(2) 20 0.1
!MAP ato(1) ato(2) npts 50 stepsize 0.1
!MAP bvec1 x1 y1 z1 bvec2 x2 y2 z2 cen x0 y0 z0 50 .1
!CPSEARCH bond ato(1) ato(2)
CPSEARCH bond rmin 1.2 rmax 1.6
!CPSEARCH ring ato(1) ato(2) ...
!CPSEARCH shell ato(1) rmin 0.3 rmax 0.5 nrad 5 nang 11 11 cutoff 16.0

137
Chapter 12 - Example Input Files

!CPSEARCH BUBBLE ato(1) rmin 0.3 rmax 0.5 curv -3 ncps 3
!CPSEARCH point x y z
!CPSEARCH start file.cps
!BPATH ato(1) ato(2) algorithm 2
END XDPROP

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
MODULE XDGEOM
SELECT rmin 0.8 rmax 1.8 tor *ato *bon *ang loc non
END XDGEOM

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
MODULE TOPXD
COMT just a comment for this run
DEBG symeqv deriv *check
CGEN alim -1. 2. blim -1. 2. clim -1. 2
MPAR rcut 4.0d0 dstep 5.d-3 au iam
!DGRD *use *gen fra gstep 0.2 0.2 0.2 read *ascii fort.69

!TRHO *seed all ail debug nstep 12 nnb 15 rmax 3.0 th 2.7
! fra 0. 0. 0.
! car 0. 0. 0.
!TRHO *cluster all ail debug nstep 11 nnb 10 rmax 3.0 th 2.7
!TRHO *pairs nr ail debug nstep 11 nnb 9 rmax 5. th 2.2 pc 0.3
!TRHO *points nr ail debug nstep 20 nnb 9 rmax 5 fra
! 0.0 0.0 0.0
! 0.5 0.5 0.5
!TRHO *line nr ail debug nstep 10 nnb 20 rmx 5. pc 0.3
! point fra 0. 0. 0. 0.5 0.5 0.5
! point car 0. 0. 0. 2. 2. 2.
!TRHO *grid nr ail debug nstep 10 nnb 9 rmax 5 ncons 0
! xmin 0. xmax 1. xstep 0.01
! ymin 0. ymax 1. ystep 0.01
! zmin 0. zmax 1. zstep 0.01
!TRHO *profile perstep 2.
! *atom ato(1) toneighbor 1 2 3
! *point fra 0. 0. 0. 1. 1. 1.
! *point car 0. 0. 0. 2. 2. 2.

!TLAP *auto ef CCCP ail debug nstep 15 nnb 10 rmax 3.0 ntheta 8 nphi 16
! atoms ato(1) ato(2) ... nmax 0 rstar 0.d0
! atoms ato(3) ato(4) ... nmax 2 rstar 0.d0
! nna x 1. y 1. z 1. nmax 4 rstar 3.2
! nna x 2. y 2. z 2. nmax 4 rstar 3.2
!TLAP *auto nr all debug nstep 20 nnb 12 rmax 3.0 ntheta 16 nphi 8
! atoms ato(1) ato(2) ... nmax 0 rstar 0.d0
!TLAP *points nr all debug nstep 23 nnb 11 rmax 4.0 nmax 14
! car 1. 1. 1.
! fra 0. 0. 0.
!TLAP *line nr all debug nstep 12 nnb 12 rmax 3.0 nmax 0
! atom ato(1) toneighbor 1 2 3
! points fra 0. 0. 0. 1. 1. 1.
Chapter 12 - Example Input Files

! points car 0. 0. 0.  2. 2. 2.
! ATBP Params PhInSph 48 ThInSp 32 *SavSurf
! ATBP AltGuess BigStep 0.5 Accur 1.D-3 MaxRInt 10. Rmax 10. Step0 0.02 A 0. B 0.
! ATBP Spheres ato(1) 0.2 ato(2) 0.2 ...
! ATBP *atoms ato(1) iZFS nvi 100 IRSur 0 *IRSav Rest Debug Phi 32 Th 24 Rad 120 Accur 1.D-3
! ATBP nna 0
! x 0. y 0. z 0. *integ sphere 0.2 iZFS nvi 5 irsur -1 irsav rest debug phi 8 th 4 rad 120 nbcp 0
!
! VZ3D *plot
! files rays.dat
! basins ato(1) ato(2) ...
! range *default xmi 0. ymi 0. zmi 0. xma 1. yma 1. zma 1.
! grid *default dx 0.05 dy 0.05 dz 0.05 rvec *default 0.20
! END VZ3D
!
! PL2D general
! point car 0. 0. 0.
! atom 1 0 -1 0
! atom 2 -1 0 -1
! plotdim xmin -2. xmax 2. xstep 0.5 ymin -2. ymax 2. ystep 0.5
! origin ishift 5. origin 0. 0. 0. vmod 0.5
! misc size a4 scale 0.4 name 'test2d' title 'TOPXD rulez!'
! PL2D *rhoo nstar 10 rmax 3.0 test cut 0.5 0.2
! PL2D *lapp nstar 10 rmax 3.0 test cut 0.5 0.2
! PL2D *lapm nstar 10 rmax 3.0 test cut 0.5 0.2
! PL2D *grho nstar 10 rmax 3.0 test cut 0.5 0.2
! PL2D *trajgrad nstar 10 rmax 5.0 test cut 0.5 0.2 toler 0.5 0.5 *plane npath 36 nextr 0
! PL2D *molgraph nstar 10 rmax 5.0 test cut 0.5 0.2 toler 0.5 0.5 *plane thr 1.6 *tr1 *tr2 *tr3
! PL2D *trajmolg nstar 10 rmax 5.0 test cut 0.5 0.2 toler 0.5 0.5 *plane thr 1.6 *tr1 *tr2 *tr3 npath 36 nextr 0
!
!
! PL3D general fra
! xmin 0.0  xmax 0.5  xstep 0.05
! ymin 0.0  ymax 0.5  ystep 0.05
! zmin 0.0  zmax 0.5  zstep 0.05
! name 'test3d'
! PL3D *plot rhoo lapp grho
! PL3DCRY *diff rhoo file1 file2
! PL3DCRY *diff lapp file3 file4
! PL3DCRY diff test2d rhoo
! PL3DCRY diff test2d trajmolg
! END TOPXD

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Chapter 13
XD Installation

13.1 What you need ...

First of all, you need the XD package itself. This is currently available as source code for
some operating systems, but it is tested and developed only for Linux.

You need a FORTRAN-77 compiler which knows about some common language extensions
like INCLUDE or DO ...ENDDO to compile most of the programs. The graphics program and
the master control program are written in C, so you need an ANSI C compiler, as well. Note
that some C compilers provided as part of the operating system are not ANSI compatible.

To be able to run the plotting program XDGRAPH you need the Tcl and Tk libraries and
optionally some OpenGL library. The program is tested with Tcl/Tk version 8.3.

Currently, the following parts of XD are available:

- **xd**  The complete package. These are just all files of the parts listed below packed
together. *If you have this, you don't need to retrieve any of the parts below.*
- **master**  The master control program (xd), and some utility programs (xdini, xdlspar),
the databank files and the makefiles used to install the complete package.
- **libxd**  The library used by all other programs.
- **xdsm**  The least-square program.
- **xdfour**  The Fourier program.
- **xdfft**  The Fast Fourier program
- **xdprop**  The property program.
- **xdgraph**  The graphics program.
- **xdgeom**  The program for analysing the geometries
- **topxd**  The program for a full topological analysis

13.2 Installing XD on VMS systems (no longer supported)

13.2.1 Tcl and Tk from source

There is no need to read this section, if Tcl and Tk are already installed on your system. If
you just have to install Tcl, but cannot make use of Tk, just ignore any references to that
package. You can get versions of Tcl and Tk modified to work under VMS by anonymous ftp from

```
mango.rsmas.miami.edu: /pub/VMS-tcl or
```

These are the original ports created by Angel Li. You can also find a slightly modified version
on ftp.fu-berlin.de. Basically, some unnecessary files are removed and command files to
compile and install the packages are provided. The tk-photo extension is removed as well. Here is how to install this distribution.

First of all, you have to pick a directory where to unpack and compile the source code.
Space requirements in blocks:

<table>
<thead>
<tr>
<th></th>
<th>zip archive</th>
<th>unpacked source</th>
<th>after compilation</th>
<th>installed</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcl</td>
<td>800</td>
<td>3000</td>
<td>4000</td>
<td>800</td>
</tr>
<tr>
<td>tk</td>
<td>1400</td>
<td>4500</td>
<td>8500</td>
<td>2000</td>
</tr>
</tbody>
</table>
Unpack both archives in the same directory. This will create sub-directories [.TCL73] and [.TK36]. Change into these directories and edit the file MAKE.COM. You have to change the five symbols at the top:

```
tcl_library | tk_library 
```

The directory, where the necessary Tcl scripts will be stored after installation. This has to be given in a unix-like notation. You can change the value at run-time by defining the logical TCL_LIBRARY or (TK_LIBRARY)

```
tcldir | tkdir
```

This is the same directory as tcl_library (tk_library), but in VMS notation. You have to specify this directory in both forms.

```
bindir
libdir
incdir
```

Where the executables will be stored.
Where the object libraries will be stored.
Where to put the include files.

Execute
$ make
And if everything is ok
$ make install
You might want to do
$ make test
in between.

**13.2.2 XD from sources**

Choose a directory where you want to install the sources, e.g. [SOURCES.XD]. Most parts of XD are installed in separate subdirectories that are created while unpacking the archives, but you have to create the top level directory which must hold all subdirectories.

Unpack the archive in the top level directory. Edit CONFIG.COM (see further notes in this file).

Rename fortran source code to from .f to .FOR by executing
$ @tovms
Compile and link the various programs
$ @make

Copy the necessary files into their destination directories
$ @make install

Finally, edit setup.com, which defines the foreign commands needed to start the programs. You won't be able to specify compound-id or model-id by starting the programs with a $ RUN command.

**13.3 Installing XD on Unix systems**

**13.3.1 Tcl and Tk from source**

There is no need to read this section, if Tcl and Tk are already installed on your system. If you just have to install Tcl, but can't make use of Tk, just ignore any references to that package.

Get Tcl and Tk from an ftp server next to you. Follow the instructions that come with those distributions.
Chapter 13 XD Installation

13.3.2 XD from sources

The UNIX source code of XD is available from the website
http://xd.chem.buffalo.edu

Choose a directory where you want to install the sources, e.g. /software/sources/xd, lets call it TOP. Most parts of XD are installed in separate subdirectories that are created while unpacking the archives, but you have to create the top level directory which must hold all subdirectories. To configure XD for your particular needs, copy $TOP/mk.config.dist to $TOP/mk.config and edit this file. The file contains further details what to do. The most important part consists of specifications where XD should finally be installed.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD_LIBDIR</td>
<td>Where to put the object libraries, for example:</td>
</tr>
<tr>
<td></td>
<td>/usr/local/lib.</td>
</tr>
<tr>
<td>XD_BINDIR</td>
<td>Where to put the executables, for example</td>
</tr>
<tr>
<td></td>
<td>/usr/local/bin.</td>
</tr>
<tr>
<td>XD_TCLDIR</td>
<td>Where to put the tcl/tk scripts coming with xdgraph. This should be a directory on its own, for example:</td>
</tr>
<tr>
<td></td>
<td>/usr/local/lib/xd/tcl.</td>
</tr>
<tr>
<td>XD_DATADIR</td>
<td>Where to put some datafiles (including the databanks). For example</td>
</tr>
<tr>
<td></td>
<td>/usr/local/lib/xd.</td>
</tr>
</tbody>
</table>

Finally, in directory $TOP you say

```
make arch
```

where arch is one of irix, aix, osf1, hpux, linux. If you want to compile for another architecture use unix and edit the appropriate files in $TOP/ports. Compilation will be done in architecture-dependent separate subdirectories of each part of the package. E.g. object files for the least squares program, compiled on a SGI Indigo running Irix will be created in $TOP/xdlsm/src/obj.irix.

After compilation you

```
make install
```

and all files will be copied to the directories given in mk.config.

Some parts of the XD code (for example XDCIF) additionally require that XD_DATADIR be defined as environment variable. It is also suggested to define XD_BINDIR as environment variable, in order to have access to all executables, without requiring a script file.

13.4 Installing XD for Windows

The MS Windows (95/98/ME/NT/2000/XP) version of XD is available from the website

http://xd.chem.buffalo.edu

It is supplied as a standard Windows installation program. This includes all executables (with all required run-time libraries) and all system files. It is strongly suggested to install into the default directory given by the installation program. In addition, the working directory containing the XD data files (xd.mas, xd.inp etc) should not contain an embedded blank in its name, or all programs may not work.

In order to function, the following environment variables need to be set

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD_DATADIR</td>
<td>(points to directory containing the data bank files)</td>
</tr>
<tr>
<td>XD_TCLDIR</td>
<td>(points to directory containing the XDGRAPH Tcl scripts)</td>
</tr>
<tr>
<td>TCL_LIBRARY</td>
<td>(points to directory containing the TCL 8.3 system scripts)</td>
</tr>
</tbody>
</table>
Example values for these variables would be

```
XD_DATADIR=<xddir>/lib/xd
XD_TCLDIR=<xddir>/lib/xdgraph
TCL_LIBRARY=<xddir>\bin\tcl8.3\n```

where `<xddir>` is the fullpath of the XD installation directory, e.g. c:/xd or c:\xd. **Note the use of Unix style forward slashes "/" rather than DOS backslashes "\" as delimiters for the directory names with XD_DATADIR & XD_TCLDIR.** These are a result of porting a Unix program to Windows and appear to be necessary. The standard DOS backslash should be used for the TCL_LIBRARY environment variable.