XD-2015

A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental or Theoretical Structure Factors

Manual Version February 2015
Program Version 6.03 February 2015

XD Program authors

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

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
Middle Tennessee State University
avolkov@ mtsu.edu

T. Richter
Freie Universität Berlin

Table of Contents

HOW TO OBTAIN XD2015 V

ACKNOWLEDGEMENTS V

NOTATIONS VI

UPDATES VI

New features in version XD2015 .... *vi*
Corrections & improvements in versions after 4.9 .... *vii*
Corrections & improvements in version 4.9 .... *vii*

CHAPTER 1 1

INTRODUCTION 1
1.1 Synopsis for non-crystallographers *1*
1.2 Experimental electron densities *2*
1.3 Theoretical aspects of electron density determination *2*
1.4 Electron density - Structure factor models *3*
1.5 Conventional formalism *3*
1.6 High order refinement *3*
1.7 The aspherical-atom formalism *3*
1.8 Orbital vs. Multipole formalism *4*
1.9 Radial functions and scattering factors *5*
1.10 The temperature factor *5*
1.11 Deformation electron density *6*
1.12 Experimental requirements *7*
1.13 Determination of atomic and structural properties from charge distributions *7*

CHAPTER 2 11

THE XD SYSTEM FILES AND THE MASTER CONTROL PROGRAM 11
2.1 File Name Conventions *11*
2.2 The Master File xd.mas *13*
2.3 The parameter file xd.inp and xd.res *15*
2.4 The reflection file xd.hkl *15*
*2.5 The databank files xd.bnk_** *16*
2.6 The Fourier file xd.fou *17*
2.7 The design matrix xd.der *17*
2.8 The normal equation matrix xd.mat *18*
2.9 The variance-covariance matrix xd.cov *18*
2.10 Grid and path file format *18*
2.11 XD – The Master Control Program *18*

CHAPTER 3 23

XDINI – IMPORTING DATA INTO XD 23
3.1 Instructions for XDINI *23*

3.2 Examples *26*

CHAPTER 4 .... **27**

XDLSM - LEAST SQUARES PROGRAM FOR MULTIPOLE REFINEMENT 27
4.1 Overview *27*
4.2 Refinement strategy *33*
4.3 Dimensioning *33*
4.4 Variable names and order numbers *34*
4.5 Files used and created by XDLSM *34*
4.6 Input instructions for XDLSM *34*
4.7 Appendix *48*

CHAPTER 5 .... **63**

XDGEOM – GEOMETRY FUNCTIONS, ERRORS AND TABLES 63
5.1 Overview *63*
5.2 Files used and created by XDGEOM *64*
5.3 Input instructions for XDGEOM *64*
5.4 Example Master File segment for XDGEOM *65*

CHAPTER 6 .... **67**

XDPROP – ONE-ELECTRON PROPERTIES BASED ON THE MULTIPOLE REPRESENTATION OF THE ED 67
6.1 Overview *67*
6.2 General Instructions for XDPROP *68*
6.3 Instructions to Define the Cluster of Atoms Used *71*
6.4 Parameters for Topological Analysis *73*
6.5 Calculation of Properties *75*
6.6 Function Plots *82*
6.7 Topological Analysis *87*
6.8 Atomic Basin Integration *91*

CHAPTER 7 .... **95**

XDFOUR – A GENERAL 2-D AND 3-D FOURIER SYNTHESIS PROGRAM 95
7.1 Overview *95*
7.2 Files used and created by XDFOUR *96*
7.3 Input instructions for XDFOUR *96*

CHAPTER 8 .... **101**

XDFFT - A FAST FOURIER TRANSFORM PROGRAM 101
8.1 Overview *101*
8.2 Files used and created by XDFFT *101*
8.3 Input instructions for XDFFT *101*

CHAPTER 9 .... **103**

XDGRAPH - VISUALISING THE RESULTS 103
9.1 Overview *103*
9.2 The Command Line Interface *103*
9.3 The Graphical User Interface *104*
9.4 Running XDGRAPH *105*
9.5 Commands *105*

9.6 Common Options *113*
9.7 Common Subcommands *115*
9.8 Toolbox *115*
9.9 Display Driver *117*
9.10 Examples *117*
9.11 A few words about Tcl *118*

CHAPTER 10 .... **121**

TOPXD – FULL TOPOLOGICAL ANALYSIS 121
10.1 Overview *121*
10.2 TOPXD Instructions *123*
10.3 Topological Analysis of Electron Density *126*
10.4 Topological Analysis of the Laplacian of Electron Density *130*
10.5 Evaluation of atomic and/or NNA basin properties *133*
10.6 2-Dimensional (2D) Plots *136*
10.7 3-Dimensional (3D) plots *143*
10.8 3-Dimensional (3D) visualization of atomic basins *145*
10.9 Description of the atomic properties evaluated by TOPXD *147*
10.10 Frequently Asked Questions *148*

CHAPTER 11 .... **151**

XD UTILITY PROGRAMS 151
11.1 XDVIB1 - A Program to Calculate the Mean-Square Displacement Amplitudes from Harmonic Vibrational Frequencies and Normal Modes. *151*
11.2 XDVIB2 – A Program to Transform ADPs from Cartesian Systems to the Crystal Frame. *154*
11.3 XDPDF - A Program to Calculate the Probability Density Function of an Atom. *156*
11.4 XDCIF - A Program to Create an Archive CIF *160*
11.5 XDWTAN - A Program to Analyse the Weighting Scheme *160*
11.6 XDHKL - A Program to Analyse the Reflection Data *161*
11.7 AIM2TAB *162*
11.8 LSDB *162*
11.9 ADDGRID, SCALEGRID *163*
11.10 RAYS2POV *163*

CHAPTER 12 .... **165**

EXAMPLE INPUT FILES 165
11.1 Parameter file (XD.INP/XD.RES) *165*
12.2 Master file *165*

CHAPTER 13 .... **172**

XD2015 INSTALLATION 172
13.1 Installing XD2015 for Linux/Mac *172*
13.2 Installing XD2015 for Windows™ *173*
13.3 External programs for XD2015 .... *174*

How to obtain XD2015

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

For registered users only, the program distributions for Linux and MS Windows are available via the internet from the XD web-site :

Official XD web site : http://www.chem.gla.ac.uk/~louis/xd-home/

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 Glasgow
- Department of Chemistry, University of Wales, Cardiff
- Department of Chemistry, State University of New York at Buffalo
- Dipartimento di Chimica Strutturale e Stereochimica Inorganica, Università di Milano
- Istituto di Scienze e Tecnologie Molecolari CNR Milano

Finally, we would like to express our grateful thanks to S. Howard (Cardiff), Z. Su (Buffalo), N. K. Hansen (Nancy), F. Mamedov (Buffalo), Z. Kisiel (Warsaw) and M. Tafipolsky (TU, Munich) for contributions to the XD code.

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.

Updates

New features in version XD2015

These have been almost entirely implemented as options in the XDPROP program. See manual for further details.

New features since the release of version 4.10

These have been primarily implemented as options in the XDPROP program. These include:

- Calculation of intermolecular interactions (INTEREN/LATEN). Electrostatic interactions are calculated from the experimental multipole moments using the highly accurate EPMM method which eliminates the serious approximations of the non-overlapping charge density inherent in the Buckingham expression. Non-electrostatic interactions are evaluated through atom-atom potential curves which can be custom-selected. Lattice energy is also computed based on direct space summation. These features represent an updated and corrected version of XDINTER.
- New routines to compute the molecular electrostatic potential, field and field gradients, based on the separated integration of Slater-type density functions.

All the integrated properties available for $\rho(\mathbf{r})$ are available for the electrostatic potential as well.

- Gradient vector plots of $\rho(\mathbf{r})$ or the electrostatic potential (TRAJPLT). Vector plots of the electrostatic potential.
- Scattering factors and wave functions for atoms up to Z = 109 (including relativistic effects) are now available in a new databank (code VM).
- Direct interface to existing graphic programs (such as MOLEKEL, MOLDEN, GaussView03 and POV-Ray) and crystallographic programs (such as PLATON and ORTEP).
- Integration of atomic basins available for both the molecule in the crystal and the molecule extracted from the crystal (TOPINT).
- Local and basin-integrated Source Function at any point in space,
- The atomic probability density function (pdf) arising from thermal motion can be visualized, including anharmonic components as represented in the Gram-Charlier formalism (XDPDF).
- The experimental reflection data can be critically assessed for suitability for charge density analysis (XDHKL).
- An **INCLUDE** instruction is introduced, whereby external files may be inserted into the XD.MAS file at any point

The reference for this new version of XD2015 is :

XD2015 - a computer program for multipole refinement, topological analysis of charge densities and evaluation of intermolecular energies from experimental or theoretical structure factors. Volkov, A.; Macchi, P.; Farrugia, L. J.; Gatti, C.; Mallinson, P.; Richter, T.; Koritsanszky, T.; (2015).

Corrections & improvements in versions after 4.9

News of bug-fixes and modifications to XD are given at the XD web site

Corrections & improvements in version 4.9

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 3*d* and 4*s* 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 occured for n*s* 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 3*d* 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_1 and 6_5 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.

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 0.4×10^{-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

$$I_{tot} = \langle I \rangle_T \propto \langle | F(\mathbf{h},\mathbf{q}) |^2 \rangle_T \qquad \text{(Eq.1-1)}$$

where $F(\mathbf{h},\mathbf{q})$ is the Fourier transform of $\rho(\mathbf{r},\mathbf{q})$, the static ED at a given nuclear configuration $\mathbf{q}$, $\mathbf{h}$ is the Bragg vector with integral components h_1, h_2, h_3 relative to the

$$F(\mathbf{h},\mathbf{q}) = \int_V \rho(\mathbf{r},\mathbf{q}) \exp(2\pi\mathbf{hr}) d\mathbf{r} \qquad \text{(Eq. 1-2)}$$

reciprocal axes $\boldsymbol{a^*}$, $\boldsymbol{b^*}$, $\boldsymbol{c^*}$, V is the unit-cell volume and $\langle\ \rangle_T$ means thermal averaging over all vibrational states. By disregarding the diffuse scattering altogether

$$I_{tot} = I_{Bragg} = | \langle F(\mathbf{h},\mathbf{q}) \rangle_T |^2 \qquad \text{(Eq. 1-3)}$$

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). Based on this equation the ED in the crystal can be given by a Fourier summation

$$\rho(\mathbf{r}) = V^{-1} \sum_{\mathbf{h}} F_{\mathbf{h}} \exp(-2\pi\mathbf{hr}) \qquad \text{(Eq. 1-4)}$$

This direct evaluation of ρ 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

$$\langle \rho(\mathbf{r}) \rangle_T = \sum_k \rho_k (\mathbf{r} - \mathbf{q}_{ko} - \mathbf{u}_k) P(\mathbf{u}_k) d\mathbf{u}_k \qquad \text{(Eq. 1-5)}$$

where the summation runs over the density units ρ_k centered at $\mathbf{q}_{ko}$ and $P_k(\mathbf{u}_k)$ 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 $\langle \rho(\mathbf{r}) \rangle_T$

$$F(\mathbf{h}) = \sum_k f_k(\mathbf{h}) t_k(\mathbf{h}) \exp(2\pi i \mathbf{h}\mathbf{q}_{ko}) \qquad \text{(Eq. 1-6)}$$

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 ρ_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:

$$\rho(\mathbf{r}) = \rho_c(r) + P_v \rho_v(\kappa r) + \rho_d(\kappa' \mathbf{r}) \qquad \text{(Eq. 1-7)}$$

where ρ_c and ρ_v are the core and spherical valence densities (sphv), respectively and

$$\rho_d(\kappa'\mathbf{r}) = \sum_l R_l(\kappa' r) \sum_{m=-l}^{l} P_{lm} y_{lm}(\frac{\mathbf{r}}{r}) \qquad \text{(Eq. 1-8)}$$

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

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

$$= 1, l = 0 \qquad \text{(Eq. 1-10)}$$

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 (κ and κ', respectively) to account for the radial expansion or contraction of the valence shell. The corresponding scattering factor is

$$f(\mathbf{h}) = f_c(H) + P_v f_v\left(\frac{h}{\kappa}\right) + \sum_l \left\langle J\left(\frac{h}{\kappa'}\right)_l \right\rangle \sum_{m=-l}^{l} P_{lm} y_{lm}\left(\frac{\mathbf{h}}{h}\right) \qquad \text{(Eq. 1-11)}$$

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

$$\langle J_l \rangle = 4\pi i^l \int j_l(2\pi H r) R_l(r) r^2 dr \qquad \text{(Eq. 1-12)}$$

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

$$\rho = \sum_i n_i \, | \phi_i |^2 \qquad \text{(Eq. 1-13)}$$

where n_i is the orbital occupation number (1 or 2) of the ith atomic orbital,

$$\phi_i = \phi_{nlm} = R_{nl} y_{lm} \qquad \text{(Eq. 1-14)}$$

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

$$R_{nl} = \sum_j C_{nli} O_{lj} \qquad \text{(Eq. 1-15)}$$

the density unit ρ_{nlm} corresponding to ϕ_{nlm} is given by the following linear combination:

$$\rho_{nlm} = \left[\sum_{jk} D_{njk} O_{lj} O_{lk} \right] y_{lm} y_{lm} = R_{nl}^2 y_{lm} y_{lm}. \qquad \text{(Eq. 1-16)}$$

The spherical harmonics form a complete basis set, thus their product can be expanded over spherical harmonics:

$$y_{lm} y_{l'm'} = \sum_{LM} C_{Lll'Mmm'\mathrm{YLM}} \qquad \text{(Eq. 1-17)}$$

Clebsch-Gordon coefficients ($C_{Ll l' M m m'}$) 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:

$$O_l = [2n(l)!]^{-1/2} (2\zeta_l)^{(n(l)+1)/2} r^{n(l)} \exp(-\zeta_l r) \qquad \text{(Eq. 1-18)}$$

where ζ_l are energy optimized orbital exponents.

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

$$R_l(r) = \frac{a_l^{n(l)+3}}{(n(l)+2)!} r^{n(l)} \exp(-a_l r) \qquad \text{(Eq. 1-19)}$$

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

$l \setminus l'$	s	p	d
s	0	1	2
p		0 2	1 3
d			0 2 4

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 **u**, taken with respect to the equilibrium position ($\mathbf{u} = \mathbf{q}\text{-}\mathbf{q}_o$), is a normal distribution:

$$P_o(\mathbf{u}) = (2\pi)^{-3/2} (\det \mathbf{U})^{-1/2} \exp(-1/2\mathbf{u}'\mathbf{U}^{-1}\mathbf{u}) \qquad \text{(Eq. 1-20)}$$

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

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

$$t_o(\mathbf{h}) = \exp(-2\pi^2 \mathbf{h}'\mathbf{U}\mathbf{h}) \qquad \text{(Eq. 1-21)}$$

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:

$$P(\mathbf{u}) = (1 + \frac{1}{3!} C_{jkl} H_{jkl} + \frac{1}{4!} C_{jklm} H_{jklm} + \ldots) P_o \qquad \text{(Eq. 1-22)}$$

where $H_{jkl\cdots}$ are three dimensional Hermite polynomials being functions of **U** and **u**, while the coefficients $C_{jkl\cdots}$ 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:

$$T(H) = (1 - \frac{4}{3}\pi^3 i C_{jkl} h_j h_k h_l + \frac{2}{3}\pi^4 C_{jklm} h_j h_k h_l h_m + \ldots) T_o(H) \qquad \text{(Eq. 1-23)}$$

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].

$$\delta\rho(\mathbf{r}) = \rho_{mol}(\mathbf{r}) - \sum_k \rho_k(\mathbf{r} - \mathbf{r}_k) \qquad \text{(Eq. 1-24)}$$

To interpret the $\delta\rho(\mathbf{r})$ 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, ρ_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

$$\delta\rho(\mathbf{r}) = \sum_{\mathbf{h}} [F_o(\mathbf{h}) - F_c(\mathbf{h})] \exp(-2\pi i \mathbf{h}\mathbf{r}) \qquad \text{(Eq. 1-25)}$$

the series termination error is considerably decreased. The phases and the F_c 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_{ho}), (iii) full-data aspherical-atom refinement (X-X_{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 μ_K corresponds to the *K* absorption edge of any atom in the molecule and μ_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 $\rho(\mathbf{r})$, the features of which are summarized by the curvatures of $\rho(\mathbf{r})$ 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 $\rho(\mathbf{r})$ vanish. At such a point, denoted by position vector $\mathbf{r}_c$,

$$\nabla\rho(\mathbf{r}_c) = \hat{\mathbf{i}}\frac{\partial\rho}{\partial x} + \hat{\mathbf{j}}\frac{\partial\rho}{\partial y} + \hat{\mathbf{k}}\frac{\partial\rho}{\partial z} = 0$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$ 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 $\partial^2\rho/\partial x\partial y$ in the determination of the curvatures of ρ at a point in space. Their ordered 3×3 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 ρ are extremized. The principal axes and their corresponding curvatures at a critical point in ρ are obtained as the eigenvectors and corresponding eigenvalues (λ) of the Hessian matrix of $\rho(\mathbf{r})$. The *rank* ω of a critical point is the number of non-zero eigenvalues or curvatures of ρ at the critical point, while its *signature* σ is the algebraic sum of the signs of the curvatures at that point. The critical point is labelled by giving the pair of values (ω,σ). 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 ρ is a local maximum at $\mathbf{r}_c$.
(3,-1)	two curvatures are negative and ρ is a maximum at $\mathbf{r}_c$ in the plane defined by their corresponding axes. ρ is a minimum at $\mathbf{r}_c$ along the third axis, perpendicular to this plane.
(3,+1)	two curvatures are positive and ρ is a minimum at $\mathbf{r}_c$ in the plane defined by their corresponding axes. ρ is a maximum at $\mathbf{r}_c$ along the third axis, perpendicular to this plane.
(3,+3)	all curvatures are positive and ρ is a local minimum at $\mathbf{r}_c$.

The traditional association of nuclear positions with local maxima in ρ(**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 ∇ρ. The gradient vector points in the direction of the greatest increase in ρ, so these trajectories are perpendicular to the contour lines of ρ. They have the property of originating or terminating at critical points in ρ. The charge distribution is partitioned into disjoint regions by surfaces for which

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

where **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 ρ(**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, $\rho(\mathbf{r}_c)$, $\nabla^2\rho(\mathbf{r}_c)$ [35].

The value of ρ_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' ε) measures the degree of planarity or conjugation. More precisely, $\varepsilon = (\lambda_2/\lambda_1)-1$, where the λ's are the two eigenvalues of the Hessian corresponding to directions perpendicular to the bond.

Stationary points in ρ 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 ρ and has physical meaning as representing local concentrations, where $\nabla^2\rho(\mathbf{r}) < 0$, and depletions, where $\nabla^2\rho(\mathbf{r}) > 0$, 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 $\nabla^2\rho(\mathbf{r})$ are to be distinguished from local maxima and minima in the charge density itself. Although the topology of ρ yields a faithful mapping of the chemical concepts of atoms, bonds and structure, there is no indication of maxima in ρ 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 appears instead in the topology of the Laplacian of ρ, 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 $\nabla^2\rho(\mathbf{r})$, 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 [60-63].

Finally we note that the use of the bipolar model for characterising chemical bonds solely on the basis of the magnitude of $\rho(\mathbf{r}_c)$ and the sign and magnitude of $\nabla^2\rho(\mathbf{r}_c)$ [35], is a useful model for light atom compounds (*i.e.* elements from the first three periodic rows). It ceases to be so useful for compounds of heavier elements such as the transition metals, where the bond critical points involving these elements invariably lie in a region of positive $\nabla^2\rho(\mathbf{r}_c)$. The reader is directed to recent reviews [64,65], which discuss extentions to the bipolar model in considerable detail.

Bibliography

1. P. Coppens, *J. Chem. Ed.* **61**, 761-765 (1984).
2. K. Angermund, K.H. Claus, R. Goddard and C. Krüger, *Angew. Chem. Int. Ed.* **24**, 237-247 (1985).
3. P. Coppens, *J. Phys. Chem.* **93**, 7979-7984 (1989).
4. P. Coppens, *Ann. Rev. Phys. Chem.* **43**, 663-692 (1992).
5. (a) R.J. Weiss, *X-ray Determination of Electron Distributions*, North Holland Publishing Co.: Amstersdam, 1966. (b) B. Dawson, *Proc. Royal Soc.* **A298**, 255 (1967).
6. R.F. Stewart, *J. Chem. Phys.* **58**, 1668 (1973).
7. P. Coppens, T.N. Guru Row, P. Leung, E.D. Stevens, P.J. Becker & Y.W. Yang, *Acta Cryst.* **A35**, 63-72 (1979).
8. M. Souhassou, C. Lecomte, R. H. Blessing, A. Aubry, M. M. Rohmer, R. Wiest, M. Benard and M. Marraud, *Acta Cryst.* **B47**, 145 (1991).
9. S.T. Howard, M.B. Hursthouse, C.W. Lehmann, P.R. Mallinson and C.S. Frampton, *J. Chem. Phys.* **97**, 5616-5630 (1992).
10. P. Coppens and M.B. Hall, Eds. *'Electron Distributions and the Chemical Bond'*, Plenum Press 1982.
11. F. L. Hirshfeld, *Cryst. Rev.* **2**, 169-204 (1981).
12. P. Coppens and P.J. Becker, *International Tables for Crystallography*, Vol. C, 627-644, Kluwer Academic Publishers, 1992.
13. M. Born, *Zeitschr. für Physik* **38**, 803-811 (1926).
14. R.F.Stewart, *Isr.J. of Chem.* **16**, 137-143 (1977).
15. P.J. Becker, *Electron and Magnetization Densities in Molecules and Solids*, 173-211, Becker, P.J. Ed., Plenum Press: New York and London, 1980.
16. G.A.Jeffrey and D.W.J. Cruickshank, *Quart. Rev. Chem.Soc.* **7**, 335-376 (1953).
17. R.F. Stewart, *J.Chem.Phys.* **51**, 4569-4577 (1969).
18. R. Restori, *Acta Cryst.* **A46**, 150-151 (1990).
19. N.K. Hansen and P. Coppens, *Acta Cryst.* **A34**, 909-921 (1978).
20. E. Clementi and C. Roetti, *Atomic Data and Nuclear Data Tables* **14**, 177-478 (1974).
21. R.F. Stewart, *Israel J. Chem.* **16**, 124-131 (1977).
22. C.K. Johnson and H.A. Levy, *International Tables for X-ray Crystallography*, Vol. IV, 311-336, Birmingham: Kynoch Press, 1974.
23. M. Roux, S. Besnainou and R. Daudel, *J.Chim.Phys.* **53**, 218-223, 1956.
24. J.D. Dunitz, W.B. Schweizer and P. Seiler, *Helv. Chim. Acta* **66**, 123-133 (1983).

25. W.H.E. Schwarz, L. Mensching, P. Valtazanos and W. von Niessen, *Int. J. Quantum Chem.* **30**, 439-444 (1986).
26. K. Hermansson and S. Lunell, *Acta Cryst.* **B38**, 2563-2569 (1982).
27. J. Buschmann, T. Koritsanszky, R. Kuschel, P. Luger and K. Seppelt, *J.Am.Chem.Soc.* **113**, 233-238 (1991).
28. H. Takazawa, S. Ohba and Y. Saito, *Acta Cryst.* **B45**, 432-437 (1989).
29. P. Coppens, *Science* **158**, 1577-1579 (1967).
30. P. Coppens, *Isr. J. Chem.* **16**, 159-162 (1977).
31. R.H. Blessing, *Cryst. Rev.* **1**, 3-58 (1987).
32. P. Seiler, *Measurement of Accurate Bragg Intensities, in Accurate Molecular Structures*, Chapter 7, Domenicano, A., Hargittai, I., Eds., Oxford University Press, 1991.
33. P. Coppens, *J. Phys. Chem.* **93**, 7979 (1989).
34. P. Coppens, *Ann. Rev. Phys. Chem.* **43**, 663 (1992).
35. R.F.W. Bader, *Atoms in Molecules: A Quantum Theory*, Oxford University Press, 1990.
36. R.F.W. Bader, T.S. Slee, D. Cremer and E. Kraka, *J. Am. Chem. Soc.* **105** 5061 (1983).
37. S.T. Howard and T.M. Krygowski, *Can. J. Chem.* **75**, 1174-1181 (1995).
38. D. Cremer, E. Kraka, T.S. Slee, R.F.W. Bader, C.D.H. Lau, T.T. Nguyen-Dang and P.J. MacDougall, *J. Am. Chem. Soc.* **105**, 5069 (1983).
39. D. Cremer and E. Kraka, *J. Am. Chem. Soc.* **107**, 3800 (1985).
40. D. Cremer and E. Kraka, *J. Am. Chem. Soc.* **107**, 3811 (1985).
41. R.F.W. Bader and C. Chang, *J. Phys. Chem.* **93**, 2946 (1989).
42. R.F.W. Bader and C. Chang, *J. Phys. Chem.* **93**, 5095 (1989).
43. (a) S. Swaminathan, B.M. Craven and R.K. McMullan, *Acta Cryst.* **B40**, 300 (1984). (b) S. Swaminathan, B.M. Craven, M.A. Spackman and R.F. Stewart, *Acta Cryst.* **B40**, 398 (1984).
44. (a) J. Epstein, J.R. Ruble and B.M. Craven, *Acta Cryst.* **B38**, 140 (1982). (b) R.K. McMullan, J. Epstein, J.R. Ruble and B.M. Craven, *Acta Cryst.* **B35**, 688 (1979).
45. (a) R.K. McMullan, P. Benci and B.M. Craven, *Acta Cryst.* **B36**, 1424 (1980). (b) B.M. Craven and P. Benci, *Acta Cryst.* **B37**, 1584 (1981).
46. G.A. Jeffrey, J.R. Ruble, R.K. McMullan and J.A. Pople, *Proc. Roy. Soc.* **A414**, 47 (1987).
47. T. Slee, R.F.W. Bader*, J. Mol. Struct. THEOCHEM* **87**, 173 (1992).
48. P. Speers, K.E. Laidig, *J. Chem. Soc., Perkin Trans.* **2**, 799 (1994).
49. J.A. Platts, S.T. Howard and K. Woniak, *J. Org. Chem.* **59**, 4647 (1994).
50. S.T. Howard, J.A. Platts and R.W. Alder, *J. Org. Chem.* **60**, 6085-6090 (1995).
51. J.A. Platts and K.E. Laidig, *J. Phys. Chem.* **99**, 6487-6492 (1995).
52. S.T. Howard and J.A. Platts, *J. Phys. Chem.* **99**, 9027-9033, (1995).
53. Z. Shi and R.J. Boyd, *J. Phys. Chem.* **95**, 4698 (1991).
54. Y. Aray, J. Rodríguez, J. Murgich and F. Ruette, *J. Phys. Chem.* **97**, 8393 (1993).
55. Z. Shi and R. J. Boyd, *J. Phys. Chem.* **95**, 9614 (1993).
56. M. Alcamí, O. Mó, M. Yáñez and J.-L.M. Abboud, *J. Phys. Org. Chem.* **4**, 177 (1991).
57. R.J. Gillespie, and I. Hargittai, *The VSEPR model of Molecular Geometry*, Chapter 10, Allyn and Bacon: Boston, 1991.
58. M.T. Carroll and R.F.W. Bader, *Mol. Phys.* **65**, 695 (1988).
59. J.A. Platts, S.T. Howard and B.R.F. Bracke, *J. Am. Chem. Soc.* **118**, 2726-2733 (1996).
60. R.F. Stewart, in *The Application Of Charge Density Research to Chemistry and Drug Design*, Eds. G.A. Jeffrey and J.F. Piniella, NATO ASI Series B, Physics **250**, Plenum Press, 1991.
61. S.T. Howard, M.B. Hursthouse, C.W. Lehmann, P.R. Mallinson and C.S. Frampton, *J. Chem. Phys.* **97**, 5616 (1992).
62. S.T. Howard, J.P. Huke, P.R. Mallinson and C.S. Frampton, *Phys. Rev. B* **49**, 7124 (1994).
63. T. S. Koritsanszky and P. Coppens *Chem. Rev* **101** 1583-1627 (2001).
64. P. Macchi and A. Sironi *Coord Chem. Rev* **238-239**, 383-412 (2003).
65. C. Gatti *Z. Kristallogr* **220**, 399-457 (2005).

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*(_model-ID)(_property | program).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 *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 (*cf*. Section 2.11).

2.1.2 The Property Field

The *property* field refers to the property used for the calculations. It is mainly used for grid files and output from the property program.

`rho`	electron density
`defden`	deformation density
`gradrho`	gradient of the electron density
`d2rho`	Laplacian
`esp`	electrostatic potential
`fou`	Fourier map
`fft`	Fourier map
`core`	core density
`valence`	valence density
`nucpot`	nuclear potential
`sigrho`	error of the electron density
`siglap`	error of the Laplacian

2.1.3 The Program Field

The *program* field specifies the program which created the file. *program* can be ini, lsm, pro, geom, fft or fou.

2.1.4 The File Type

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

`mas`	the master file
`cyc`	cycle information
`hkl`	reflection data
`out`*‡	list-able output
`inp`*	see **res**
`res`*	all atomic and all refinable parameters
`fou`*	Fourier file (binary)
`cov`*	variance-covariance matrix (binary)
`der`*	matrix of structure factors derivatives (binary)
`mat`*	normal equation matrix (binary)
`grd`*†	property on a grid
`pth`*	bond path
`cps`*†	critical points
`ps`†	PostScript graphical output
`hpg`†	HPGL graphical output
`cgm`†	CGM (Computer Graphics Metafile) graphical output

2.1.5 Examples of File Names

`xd.mas`	(master file)
`xd_lsm.out`	(least squares results listing)
`xd.res`	(least squares parameter file on output)
`xd.fou`	(Fourier reflection file)
`xd.cov`	(variance-covariance matrix)
`xd_defden.grd`	(deformation density on a grid)

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 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 forwardslash (\), 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, **+inf** and **–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:

MODULE (*)*xdprogram*
where *xdprogram* is one of the program names (for example XDLSM, XDFOUR or XDPROP). A new line is read until the
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 **END** *TI* instruction.

Example:

```
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
```

2.2.1 General instructions

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

2.2.1.1 TITLE

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 `xd.hkl` and `xd.inp` files has to match with that in the master file, otherwise XDLSM terminates with an error message.

2.2.1.2 CELL

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 σ(*a*) σ(*b*) σ(*c*) σ(*alpha*) σ(*beta*) σ(*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 $I4_1$ (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 3×3 rotation matrix, for example:

```
SYMM 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 | VM]

BANK *databank type*
The databank type can be CR, BBB, SCM or VM (see Section 2.5 for a detailed description). In the absence of the BANK instruction, the databank CR is used as default.

2.2.1.7 INCLUDE <*filename*>

This instruction allows the user to include an external file containing further XD instructions. These instructions will be inserted at the point of the **INCLUDE** instruction, and **INCLUDE** instructions may be placed anywhere in the master file. An example of its use might be in XDPROP, when an extensive set of **CPSEARCH** and **BPATH** instructions are needed to calculation the topology. If all these instructions are placed in a separate file, it is only necessary to comment out a single line in the `xd.mas` file to remove this calculation. Include files may themselves contain **INCLUDE** instructions - there is no practical limit to the level of nesting. The allowed syntax is :

```
INCLUDE <filename>
INCLUDE "<filename>"
INCLUDE '<filename>' or even
INCLUDE<filename>
```

where <*filename*> is a character string, the name of the file with the XD instructions.

2.3 The parameter file xd.inp and 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. `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 `xd.inp` and `xd.res` may differ according to any changes made in `xd.mas`.

2.4 The reflection file xd.hkl

Type: ASCII, free-format, sequential
An input file containing the observations. It consists of as many records as observations are available. Unless anisotropic extinction is being refined, this file should normally contain only the unique reflections, with all space-group extinctions removed. If phase constraints

are to be utilised, the seventh data item must be the phase angle (in radians). See **Table 2-2** (end of this Chapter) for a detailed description.

2.5 The databank files 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:

$$b_{nl} = N_{nl} r^{(n-1)} \exp(-\zeta_n r) Y_{nl}$$

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 ζ are stored and used to calculate core and valence scattering factors according to a given 7electronic 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 `xd.mas` file with the **SCAT** table entries.

The databank file `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.

Four databanks are now available:

`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-ξ 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 $4s^1 3d^{n-1}$), 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),(i=1,6)),c.* Because the standard expansion of International Tables is up to i = 4, entries 9-12 are 0.0; the 13th entry is the constant term (which used to be the 9th 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-ξ functions are taken from Clementi and Raimondi [2] or Clementi and Roetti [1] (for atoms of the 5th 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-ξ functions as for databank BBB.

`xd.bnk_PBE-QZ4P-ZORA`: (flag **VM**)

STO atomic relativistic wavefunctions (Volkov & Macchi, unpublished work) obtained at PBE/QZ4P level of theory for neutral atoms H-Cf (*Z*=1-98) in the ground state configuration. The single-ξ exponents come from fitting of the density for each given orbital, and are thus slightly differ from the standard Clementi and Raimondi single-ξ functions.

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
 SZ      5.6727   1.6083   0.0000   0.0000   1.5679   0.0000   0.0000   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  0.000830  1.057490 2 -0.001760  1.524270 2
  0.005590  2.684350 2  0.003820  4.200960 2
 -0.208140  5.435990 1 -0.010710  9.482560 1  0.080990  1.057490 2  0.750450  1.524270 2
  0.335490  2.684350 2 -0.147650  4.200960 2
  0.282410  0.980730 2  0.546970  1.443610 2  0.231950  2.600510 2  0.010250  6.510030 2
```

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. An additional record at the end of the file reports the value of F(000), which is needed to ensure correct scaling for Fourier maps. Each record contains the following entries:

h k l fobs sig phase amod1 bmod1 amod2 bmod2

where

h, k, l	reciprocal lattice components of the scattering vector
fobs	observed structure factor (normally with anomalous dispersion removed)
sig	error of *fobs*
phase	phase angle 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*

The utility programs FOU2ASC and ASC2FOU convert this binary file into ASCII format and vice-versa. This allows users to transfer the file between different computer platforms and to get a readable output for other purposes.

2.7 The design matrix xd.der

Type: binary, sequential

nref, ibuf(4), 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),δy(i),i=1,nv)
where *B* and *δy* (real*8) are the coefficient matrix and vector of the system of least squares equations and *nv* is the number of variables refined:

$$B(i,j) = \sum_{h,k,l} w(h,k,l) * d\left|F(h,k,l)\right| / dp_i * d\left|F(h,k,l)\right| / dp_j$$

$$\delta y(i) = \sum_{h,k,l} w(h,k,l) * d\left|F(h,k,l)\right| / dp_i * (F_o - F_c)$$

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

xd (*options*) (*cid* (*mid*))

OPTIONS

–v
be more verbose (in XD itself, does *not* influence the output of individual modules)
–d
debug mode, show which commands would be executed, do not actually start them.
–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. See Chapters 9 & 10 for details about starting programs XDGRAPH and TOPXD respectively

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*
xdfft *cid*
xdgraph *options*
topxd <*output file*>

Table 2-1: The content¶ of the parameter file. The order of the parameters (U^{ij},C^{ijk},D^{ijkl},P_{lm},*extcn*) corresponds to the list given in Table 4-2

Record	Content	Description
1	*xdparfile version [1/2]*	version of the parameter file (1 is the older, 2 the most recent one)
2	*cid*	Compound identification
3	*nat, ntx, lmx, nzz, nto, nsc, ntb, nov*	software limits for some parameters in the current program version
4	(*kv(i),i*=1,14)	Dimensions of certain arrays in XDLSM, see table 4.1 in Chapter 4 for their meanings. 1: number of atoms (*na*), 2: number of displacement tensor components (*ntmx*), 3: maximum level of multipole expansion (*npolmax*), 4: number of kappa sets (*nz*), 5: not used (*nto*), 6: scale factors (*nq*), 7: extinction model (*next*), 8: number of constraints (*ncon*), 9: number of scattering factor tables (*ntbl*), 10: number of symmetry cards (*ns*), 11: number of variables (*nv*), 12: (*nqq*), 13: number of cycles (*nc*), 14: number of dummy atoms (*nad*)
5	*r1o, r2o, r1, r2, r1w, r2w, gof, sig*	Statistics of the fit
6	*a, b, c, d, e, f*	Parameters of the lsq weight
7…+*nad*	*dx(i), dy(i), dz(i)*	Dummy atom coordinates
do n=1,*na*		
+1	*atom,* *icor1, icor2, nax, nay1, nay2, jtf, itbl, isfz, lmax, isym, ichcon,*	Atom name (character*8) Code integers for defining the site coordinate system, The order of displacement tensor, scattering factor number, kappa set, max. level of spherical harmonics used, site symmetry code, chemical constraint,
+2	*x, y, z, amult* *U* for *jtf*(n)=1 or U^{ij} for *jtf*(n)≥2	Coordinates, multiplicity Isotropic or anisotropic *U*
+3,+4	C^{ijk} for *jtf*(n)≥3	3. order anharmonic tensor components
+5…+7	D^{ijkl} for *jtf*(n)=4	4. order anharmonic tensor components
+8…+10	(*Plm(j),j*=1,*npx*) *npx*=*lmax***lmax*+2**lmax*+2	Multipole populations
end do		
+1…+*nzz*	*ifz*(*i*),(*z*(*j*,*i*),*j*=0,*lmax*+2)	Scattering factor table to which the i'th kappa set refers, 6 kappa values
+1	(*extcn*(*i*),*i*=1,7)	Extinction parameters
+1	*out*	Overall thermal parameter
+1	(*sc*(*i*),*i*=1,*nsc*)	Scale factor

¶ The previous format is still accepted and interpreted by the code

Table 2-2: The content of the reflection file.

Record	Content	Description
1	*cid,* *fcod,* **NDAT** *ndat*	Compound ID **F** or **F^2** Number of entries for each observation (min.=6, max.=13)
do n=1,nref		
	h, k, l *iscgrp,* *obs,* *sigobs,* *tbar,* *u1,u2,u3,* *v1, v2, v3* *phase-angle*	Reflection indices Scale group number F or F^2 as given by *fcod* *Sigma*(F) or *sigma*(F^2) Absorption weighted path length (cm) 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 Alternatively, the seventh data item *tbar* may be replaced by the *phase-angle* (radians). This format initiates the phase-constrained refinement, and in this case NDAT must be set to -7
end do		

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

Record	Rec.-Id	Entries
1.	:ELEM	*z w dfpmo dfppmo dfpcu dfppcu sctl ira irc irn* *elem* Element symbol (H, Cu, Ti3+, …). It serves as the segment's identification *z* Atomic number *w* Atomic mass *dfpmo* Anomalous correction for Mo radiation ($\Delta f'$) *ddfpmo* Anomalous correction for Mo radiation ($\Delta f''$) *dfpcu* Anomalous correction for Cu radiation ($\Delta f'$) *ddfpcu* Anomalous correction for Cu radiation ($\Delta f''$) *sctl* Coherent neutron scattering length *ira* Atomic radii *irc* Covalent radii *irn* Van der Walls radii
2.	SPH	*a1 b1 a2 b2 a3 b3 a4 b4 a5 b5 a6 b6 c* Expansion coefficient for analytical approximation of the RHF spherical atomic scattering factors. For the hydrogen atom see Stewart *et al* [7]
3.	SZ	*(zet1(i),i=1,12)* *zet1* Single-ζ exponents used for the radial functions of the valence deformation density [2]
4.	STO	*((orb(i),ioc(i),nbf(i)),i=1,12)* *orb* Orbital type (1s,2s,3s,4s,2p,3p,4p,3d,4d,4f,5s,5p) *ioc* Occupation – negative number refers to valence electrons *nbf* Number of basis functions
5.		*(((bc(i,j),bx(i,j),nr(i,j)),j=1,nbf(i)),i=1,noc)* *bc* Coefficient of the basis function (C) *bx* Exponent of the basis function (ζ) *nr* n *noc* Number of occupied atomic orbitals (*ioc(i)*≠0)

Table 2-4: The content of the variance-covariance file.

nv, errwt (real*4)	Number of variables, square of GOF
((A(i,j),j=i,nv),i=1,nv) (real*4)	**A** = *inv*(**B**) * *errwt,* where **B** is the least squares matrix
(iatom(i),i=1,nv)	> 0 the sequence number of the atom to which parameter *i* relates = 0 *i* is not an atomic parameter < 0 point to Kappa set
(itype(i),i=1,nv)	The order number of parameter *i* as described in **Table 4.2**
(isfz(iatom(i)),i=1,nv) if *iatom(i)*>0	Kappa set

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

Record	Content	Description
1	*filetype version*	*filetype* can be any of **2DGRDFILE**, **3DGRDFILE** or **PATHFILE**. *version* gives the version number of the file format (currently 0).
2	*cid property*	Compound id and name of the mapped property
3	*title*	Title string
4	*nx ny (nz)*	Number of grid points
5	*ox oy oz*	Origin of the grid in 'world-coordinates'
6	*xdim ydim zdim*	Physical size of the grid (in Å)
7 +1…+*no*	*no* *name x y z (type)*	Number of objects *type* is either **ATOM** or **CP**
+1 +1…+*nc*	*nc* *object1 object2*	Number of connections List of bonds or other connecting lines to be drawn
The body for grid files		
	values	List of grid values, *x* varying fastest.
The body for path files		
+1	*ncurve*	Number of bond path curves
ncurve times		
+1 +1…+*npoints*	*object npoints type* *x y z*	*object* gives a starting point (usually a **CP** object). *type* is **BOND**.

Bibliography

1. E. Clementi and C. Roetti. *Atomic Data and Nuclear Data Tables* **14** 177-478 (1974).
2. E. Clementi and D. L. Raimondi, *J. Chem. Phys.* **38**, 2686-2689 (1963).
3. *International Tables for X-ray Crystallography*, Vol. IV, pp 103-144, Kynoch Press: Birmingham, 1974.
4. C. F. Bunge, J. A. Barrientos, A. V. Bunge *At. Data Nucl. Data Tables*, **53**, 113-162 (1993).
5. Z. Su and P. Coppens *Acta Cryst.*, ***A54***, 646 (1998).
6. P. Macchi and P. Coppens *Acta Cryst.*, ***A57***, 656 (2001).
7. R.F. Stewart, E.R. Davidson and W.T. Simpson, *J. Chem. Phys.* **42**, 3175 (1965).

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 (SHELXTL), CIF and XTAL 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** and **xtal** with the following input-file requirements:

prgname	file1	file2	file3
shelx	`shelx.ins`	-	`shelx.hkl`
cif	`xd.cif`	-	`xd.fcf`
xtal	`xtal.inp`	`xtal.stm`	`xtal.hkl`

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, SCM or VM (see section 2.5 for their meaning). If *bnkname* is not specified, `xd.mas` will be generated with the default BANK CR.

If cif format files are used as input, it is *essential* that the data_ block name is identical in both the parameter and reflection data files, *i.e.* both must have the same ***data_<name>*** at the start of the file (note that ***<name>*** is case insensitive).

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**, **BANKFILE** and **WAVE**, among which the first two always have to be given. The latter two segments have common instructions described below.

3.1.3 FILE

FILE *filename*

The data are 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:

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 by 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

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

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

dtyp	d_{ij}	A_{ij}
u	$2\pi^2 a_i a_j h_i h_j (2 - \delta_{ij})$	U_{ij}
b	$a_i a_j h_i h_j (2 - \delta_{ij})$	$B_{ij} = 8\pi^2 U^{ij}$
beta	$h_i h_j (2 - \delta_{ij})$	$b_{ij} = 2\pi^2 a_i a_j U^{ij}$

where

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

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

3.1.8 SCALE

SCALE *(sc(i),i=1,nsc) [1.0]*
nsc number of scale factors are read in. If omitted the data are supposed to be on absolute scale forming one scale group.

3.1.9 Default observation input

The following entries can be given for each observation:

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

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**] | **sinthl**)

The reflections are sorted either with respect to indices (*index1* varies first and *index2* last) or to the absolute value of the scattering vector (**sinthl**). 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 (LSEXP [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 $\mathbf{y}_c\{y_{c1},y_{c2},y_{c3},\ldots y_{cm}\}=\mathbf{y}_c(\mathbf{x})$, where $\mathbf{x}$ 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:

$$R^2 = (\mathbf{y}_o - \mathbf{y}_c)'W(\mathbf{y}_o - \mathbf{y}_c) = (\mathbf{y}_o - \mathbf{y}_c)'\mathbf{Q}'\mathbf{Q}(\mathbf{y}_o - \mathbf{y}_c) \qquad \text{(Eq. 4.1)}$$

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'Q** is the Cholesky decomposition of **W**. If $\mathbf{y}_c$ can be expanded about $\mathbf{x}_o$ in Taylor's series retaining only the first order terms, then

$$\mathbf{y}_c = \mathbf{y}_c(\mathbf{x}_o) + \mathbf{D}(\mathbf{x} - \mathbf{x}_o) \qquad \text{(Eq. 4.2)}$$

with $D_{ij}=dy_{ci}/\,d_{xj}$ being the design matrix. Eq. (4.1) becomes:

$$R^2 = [\Delta\mathbf{y} - \mathbf{Z}\Delta\mathbf{x}]'[\Delta\mathbf{y} - \mathbf{Z}\Delta\mathbf{x}] \qquad \text{(Eq. 4.3)}$$

where $\Delta\mathbf{y} = \mathbf{Q}(\mathbf{y}_o - \mathbf{y}_c(\mathbf{x}_o))$, $\mathbf{\Delta x}=\mathbf{x}\text{-}\mathbf{x}_o$ and $\mathbf{Z}=\mathbf{QD}$.

The n conditions

$$\left(\frac{dR^2}{dx_i}\right)_{x_i=x_{oi}} = 0 \text{ for } i = 1,2,3\ldots n \qquad \text{(Eq. 4.4)}$$

lead to the system of normal equations

$$\mathbf{Z}'\mathbf{Z}\Delta\mathbf{x} = \mathbf{Z}'\Delta\mathbf{y} \qquad \text{(Eq. 4.5}$$

whose solution vector is

$$\mathbf{x} = \mathbf{x}_o + \mathbf{B}^{-1}\mathbf{Z}'\Delta\mathbf{y} \qquad \text{(Eq. 4.6)}$$

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

$$\mathbf{Z} = \mathbf{UGV}' \qquad \text{(Eq. 4.7)}$$

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

$$\mathbf{Z}\Delta\mathbf{x} = \Delta\mathbf{y} \qquad \text{(Eq. 4.8)}$$

can be given as

$$\Delta\mathbf{x} = \mathbf{Z}^{-1}\Delta\mathbf{y} \qquad \text{(Eq. 4.9)}$$

where

$$\mathbf{Z}^{-1} = \mathbf{VG}^{-1}\mathbf{U}' \qquad \text{(Eq. 4.10)}$$

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

$$R = \left|\mathbf{Z}\Delta\mathbf{x} - \Delta\mathbf{y}\right| \qquad \text{(Eq. 4.11)}$$

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 $\mathbf{B}^{-1}$. 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:

$$\mathbf{B}_c = diag(\mathbf{B})^{-1/2}\mathbf{B}\mathrm{diag}(\mathbf{B})^{-1/2} \qquad \text{(Eq. 4.12)}$$

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 Householder reduction [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_{obs} - F_{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 $z^2_{A,B}$ 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

$$\Delta_{A,B} = z^2_{A,B} - z^2_{B,A} = 0$$

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 ($\langle u(\mathbf{n})\rangle^{1/2}=(\mathbf{n}'U\mathbf{n})$, where $\mathbf{n}$ 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:

$$\mathbf{C}(nc,nv)\Delta\mathbf{x}(nv) = \mathbf{a}(nc) \qquad \text{(Eq. 4.13)}$$

By decomposing the matrix **C**

$$\mathbf{C} = \mathbf{PSR}' = \mathbf{P}(nc,nv)\begin{pmatrix} \mathbf{S}(nr,nr) & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \mathbf{R}'_1(nr.nv) \\ \mathbf{R}'_2(nv-nr.nv) \end{pmatrix} \qquad \text{(Eq. 4.14)}$$

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

$$\Delta\bar{\mathbf{x}}_1(nr) = \mathbf{R}'_1\Delta\mathbf{x} \quad \Delta\bar{\mathbf{x}}_2(nv-nr) = \mathbf{R}'_2\Delta\mathbf{x} \qquad \text{(Eq. 4.15)}$$

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

$$\Delta\bar{\mathbf{x}}_1 = \mathbf{S}^{-1}\mathbf{P}'\mathbf{a} \qquad \text{(Eq. 4.16)}$$

This leads to a decomposition of the unconstrained variables

$$\Delta\mathbf{x} = \mathbf{R}\Delta\bar{\mathbf{x}} = \mathbf{R}_1\Delta\bar{\mathbf{x}}_1 + \mathbf{R}_2\Delta\bar{\mathbf{x}}_2 = \mathbf{R}_1\mathbf{S}^{-1}\mathbf{P}'\mathbf{a} + \mathbf{R}_2\Delta\bar{\mathbf{x}}_2 \qquad \text{(Eq. 4.17)}$$

The equations of observations 4.2 becomes

$$\Delta\mathbf{y} - \mathbf{D}\mathbf{R}_1\mathbf{S}^{-1}\mathbf{P}'\mathbf{a} = \mathbf{D}\mathbf{R}_2\Delta\bar{\mathbf{x}}_2 \qquad \text{(Eq. 4.18)}$$

and the system of normal equations is reduced to

$$\mathbf{B}_2\Delta\bar{\mathbf{x}}_2 = \mathbf{R}_2'\mathbf{Z}'\Delta\mathbf{y}_2 \qquad \text{(Eq. 4.19)}$$

where

$$\mathbf{B}_2 = \mathbf{R}_2'\mathbf{Z}'\mathbf{Z}\mathbf{R}_2 \ \textbf{and} \ \Delta\mathbf{y}_2 = \Delta\mathbf{y} - \mathbf{D}\mathbf{R}_1\mathbf{S}^{-1}\mathbf{P}'\mathbf{a} \qquad \text{(Eq. 4.20)}$$

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 4-5. 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 *Rl*'s are controlled by *n*(*l*) and α_l (see eq. 1.19), the latter being estimated from the Hartree-Fock-optimized single-ξ values. In case of quadrupolar atoms (which have only *ss, sp*, and *pp* type orbital products) the selection of α_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 κ' scaling of a. Even under this severe restriction κ' becomes highly correlated with the populations and convergence can be troublesome. In this respect, κ' 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. Incorporation of calculated ADP's for hydrogen atoms into charge density refinements has been recently reported by a number of authors, using differing methodologies [30-33]. See Chapter 11.1 concerning the program XDVIB for inclusion of calculated ADPS's.

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.

4.1.2.6 Restrictions on the reflection phases

There are well established problems [28,29] which arise when refining a multipole model for a non-centrosymmetric crystal structure. These arise because of the phase ambiguity and can result in poorly determinations of the odd-order multipole populations, which are invariant under certain crystal-class symmetry operations. In essence, some combinations of odd-order multipoles may make very small contributions to the structure factor amplitudes, but have significant contributions to the phases. In such cases, great care needs to be taken to ensure that physically meaningful parameters are obtained from the least squares procedure. As has been shown [28,29], the eigenvalue filtering method (used by default in XLDSM for singular or near-singular normal matrices) greatly alleviates these problems, though sometimes a 'slack constraint' on the scale factor (*e.g.* by including *F*(000) as an additional observation with unit weight) may also be necessary [29] to obtain accurate parameters.

An alternative solution is to use fixed phases, determined from, say, a fully periodic *ab initio* calculation. In XD2015, a phase-constrained refinement is implemented through a special format for the reflection file XD.HKL, see **Table 2-2**. The NDAT entry must be specified as -7, and the phase angle (in radians) must be supplied as the seventh data item for each reflection. No extra instructions in the master file are required. This phase-constrained strategy may be useful when refining against theoretical structure factors calculated on a non-centrosymmetric crystal structure.

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 (κ) can already be included. These variables, in contrast to those scaling the Slater exponents (κ'), 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 * U*eq*(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. These are the current limits for XDLSM. Future versions will have full dynamic memory allocation.

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 verbose** *verbose_level*
SELECT cycle *cycles* **dampk** *dampk* **cmin** *cmin* **cmax** *cmax* **eigcut** *r* **convcrit** *convcrit*

(*)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. If **model** is not starred, then it is assumed that *m3* = 1, *i.e.* that the reflection data contain anomalous dispersion effects. If theoretical data (or experimental data with anomalous dispersion removed) are being used, then ***model** *x x 0 x* must be used. An alternative approach in this case (if a global model control is not desired) is to set all the values of f' and f'' in the **SCAT** table (see **6.4.2.7** below) to zero.

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)

verbose *verbose_level* Setting *verbose_level* greater than one switches on extra printout to the log files.

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. This is singular value decomposition of the matrix of the constraints, also known as eigenvalue filtering.

convcrit *convcrit* **[0.0]** If the maximum shift/su in the current cycle of refinement is less than *convcrit*, the program immediately proceeds to the final cycle and finishes the requested XDLSM job.

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 Householder reduction and QL algorithms are used for the calculation of the eigenvalues and eigenvectors. These are considerably faster than the Jacobi method used in previous versions of XDLSM. If an eigenvalue fails the test based on the condition number (see **eigcut**), the corresponding eigenvector is eliminated and 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). It is important to note that the rejection criteria are *always* applied, starring these options merely allows the use to change the values of the default criteria. If more than one are starred, the "AND" logic is applied. The rejection criteria and default values are:

obs [0.0, 1.0e10] all observations for which *obsmin* > *obs* or *obs* > *obsmax*
sigobs [3.0, 1.0e10] observations for which *sigmin*sigobs* > *obs* or *obs* > *sigmax*sigobs*
sinthl [0.0, 2.0] lower and upper limit in $\sin\theta/\lambda$

IMPORTANT The **obs** and **sigobs** cutoffs are applied to the data in the XD.HKL reflection file, *after any conversion* implied by the **based_on** criterion (Section 4.6.1.1). This action (which is more logical) is different from that previously applied in XDLSM, where the cutoffs were applied *before* conversion.

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:

no h k l sinthl scgrp obs calc delta (del%|flag) extcn code

where

no	the order number of a reflection	
h k l	reciprocal-lattice components of the scattering vector	
sinthl	$\sin(\theta)/\lambda$	
scgrp	scale group number	
obs	F_o or F^2_o	
calc	F_c or F^2_c	
delta	$F_o - F_c$ or F^2_o or F^2_c	
flag	a flag based on $f = \lvert 100 * (obs - calc) / obs \rvert$ 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 $25 < f < 30$
del%	*f* is printed instead of a flag	
extcn	the extinction correction in percentage	
code	0	included in the refinement
	-1	rejected based on criterion **obs**
	-2	rejected based on criterion **sigobs**
	-4	rejected based on criterion **sinthl**
	-3	rejected based on criteria **obs** and **sigobs**
	-5	rejected based on criteria **obs** and **sinthl**
	-6	rejected based on criteria **sigobs** and **sinthl**
	-7	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.

(*)sinthl *snlmin snlmax* **[0 2]**

(*)obs *obsmin obsmax* **[0 10]**

(*)delta *dmin dmax* **[-50 50]**

(*)del% *min% max%* **[80 100]**

(*)extcn *extmin extmax* **[80 100]**

(*)abssc 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:

SCAT core sphv defv 1s 2s 3s 4s 2p 3p 4p 3d 4d 4f 5s 5p 6s 6p 5d 7s 6d 5f $\Delta f'$ $\Delta f''$ 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
$\Delta f'$	real part of anomalous dispersion correction
$\Delta f''$	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 assignements. To change the default assignement 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 ($\Delta f'$ and $\Delta f''$) 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**]	**rdsd**

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 $2\langle j_0\rangle$(1s1s) and $(2\langle j_0\rangle$(2s2s) + $2\langle j_0\rangle$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 sp^3 type carbon atom with the following input

```
   C    chfw chfw cszd   2 -1  0  0 -3
```

which assigns $(\langle j_0\rangle$(2s2s) + $3\langle j_0\rangle$(2p2p))/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 *ngrd* values (8 entries/lines) of the function taken at an equidistant grid of sinθ/λ with a step size of *grd*. *ngrd* 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 sinθ/λ. The parameters *ngrd* 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   1.03863   0.98506   0.93343
  0.88381   0.83628   0.79085   0.74754   0.70632   0.66717   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  -0.00502
 -0.00381  -0.00256  -0.00140  -0.00037   0.00048   0.00118   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, the radial functions of the valence deformation density are of single Slater-type (**cszd)**. The parameters of the radial functions ($n(l)$, $\xi(l)$) are obtained from the corresponding single-ξ 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-ξ orbitals (Clementi and Raimondi [15]) of the default configuration. The new version of the program now computes ξ's by weighting the orbitals by their occupation. For noble gases and closed shell ions, ξ 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 4th and 5th row (like Cu^+, Ga^{3+}, Sb^{5+}), which may have 28 or 46 electrons (single-ξ 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 ξ (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 $\langle J_l \rangle$ 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 $\langle J_l \rangle$ 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 $\langle J_l \rangle$ has to be specified for each *l*. Note that not all radial densities can be constructed from a given wavefunction. To satisfy Poisson's equation [27], $2(n-1)$ must be $\geq l$, so for example a 2s orbital cannot be used to construct octupole or hexadecapole radials parts. 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:

l [**cszd**]
l **chfw** *conf*
l **rdsd** *n(l) zeta(l)*
l **rdtb**
NOTE : The **chfw** option in XD means that the radial part (and only the radial part) of the specified orbital product will be used to construct a density function. The XD program allows such a radial function to be associated with any multipole, so that a **chfw** directive such as "1 chfw <3d3d>" is permissible, depite having limited physical meaning.

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
 0.00000   5.99918   4.95113   3.64245   2.42954   1.49816   0.87092   0.48586
 0.26409   0.14158   0.07557   0.04045   0.02182   0.01190   0.00658   0.00370
 0.00211   0.00122   0.00072   0.00043   0.00026   0.00016   0.00010   0.00006
 0.00004   0.00003   0.00002   0.00001   0.00001   0.00001   0.00000   0.00000
 0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000
```

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-ζ 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. **NOTE** Compounds containing elements with very large absorption cross-sections (B, Cd, In, Sm Eu, Gd, Dy), have an imaginary component of the scattering length, and cannot be dealt with in the current version of XDLSM.

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 ($\mathbf{v}_1$). This together with the second vector from *atom1* to *atom2* ($\mathbf{v}_2$) define the (*ax1*,*ax2*) plane. The third vector ($\mathbf{v}_3$) is taken perpendicular to this plane.

$\mathbf{v}_1 = (\mathbf{r}_0 - \mathbf{r})$ $\quad$ $\mathbf{v}_2 = (\mathbf{r}_2 - \mathbf{r}_1)$ $\quad$ $\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2$

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

$\mathbf{e}_{ax1} = \mathbf{v}_1/|\mathbf{v}_1|$ $\quad$ $\mathbf{e}_{ax2} = (\mathbf{v}_3 \times \mathbf{v}_1)/|(\mathbf{v}_3 \times \mathbf{v}_1)|$ $\quad$ $\mathbf{e}_{ax3} = \mathbf{v}_3/|\mathbf{v}_3|$

where $\mathbf{r}$, $\mathbf{r}_0$, $\mathbf{r}_1$ and $\mathbf{r}_2$ 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 3rd order Gram-Charlier expansion C^{ijk}
4	anharmonic 4th order Gram-Charlier expansion D^{ijkl}

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

xd.mas	xd.inp	
1	2	from anisotropic to isotropic
2	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 **SCAT** table.

4.6.3.5 kap - the Kappa Set

Defines the kappa set applied to the valence radial functions. *If a new set is introduced or the previous arrangement is redefined, the corresponding changes must also be made in the parameter file* `xd.inp` *and/or to the **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. *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

GROUP*n 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 **GROUP***n* command.

4.6.5 KEEP

The KEEP instruction simplifies the application of certain constraints.

KEEP kappa *set(1) ...*
For each set defined κ'_l, 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 κ', even if a single parameter is refined for all *l* values. The results of κ' refinement should *always* be critically examined and compared to those obtained with κ' = 1.0. For HF radial functions, the chance of obtaining convergence with reliable estimates of different κ'_l parameters is expected to be better than for single Slater orbitals. A separate κ'_l refinement is worth trying for transition metals.

KEEP charge [**group1**] **group*n*** ...

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**] **group*n*** ...
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.

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*

a [0.0] *b* [0.0] *c* [0.0] *d* [0.0] *e* [0.0] *f* [1/3]

For refinement on F^2, the weighting scheme as implemented in SHELXL is used:

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

where

$s_2 = \sigma(F_o^2)$
$p = f \times F_o^2 + (1\text{-}f) \times F_c^2$
$q = 1.0$ if $c = 0$
or $q = \exp[c \times (\sin\theta/\lambda)^2]$ if $c > 0$
or $q = 1.0 - \exp[c \times (\sin\theta/\lambda)^2]$ if $c < 0$

For refinement on F, the weight (w_1) is calculated as follows:

$$w_1 = [\; F_o \times \mathrm{sqrt}(w_2) \;+ \alpha \;]^2$$

where

$\alpha = 0.0$ if $\beta < 0.0$; $\alpha = \mathrm{sqrt}(\beta)$ if $\beta > 0.0$
$\beta = (F_o^2 \times w_2) + \mathrm{sqrt}(w_2)$ where w_2 is calculated as above.

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 [*i.e.* $w_2 = 1/\sigma^2(F_o^2)$ and $w_1 = 1/\sigma^2(F_o)$] will be applied. The default `XD.MAS` written by XDINI has $a = -2.0$, which sets this condition for refinement on either *F* or F^2. Note that the values of zero for the parameters *a* - *e* given above also imply statistical weights for refinement on F^2, but NOT for refinement on *F* !

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

The following instruction line for extinction correction appears in `xd.mas`:

EXTCN (*)iso (*)aniso (*)type_1 (*)type_2 (*)type_3 (*)distr_g (*)distr_1 (*)msc_0 (*)msc_1

For any extinction refinement, the absorption weighted path length (*tbar*) has to be stored in the reflection file `xd.hkl`. For an anisotropic extinction refinement, an additional six entries for each observation are required (see Table 2-2). If *tbar* is missing from `xd.hkl`, then for an isotropic extinction only, an identical value of 0.5 mm is assumed for all reflections (version 5.3 or later). This assumption will, in general, lead to incorrectly scaled extinction parameters, but this is not often of major concern.

By default the EXTCN command appears as a comment in `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):

extinction type:
- **type 1** (*type_1, default): mosaic spread, the g' coefficient is refined (variable EXT11);
- **type 2** (*type_2): particle size, the ρ coefficient is refined (variable EXT11);
- **type 3** (*type_3, *type_g is also accepted): generalized type, mosaic spread and particle size (g' and ρ) are simultaneously refined (variables EXT11 and EXT22, respectively);

mosaic spread distribution (active only for type_1 and 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):

extinction type:

- **type 1** (*type_1, default): mosaic spread, the **Z** tensor is refined (variables EXT11-EXT23) ($g(\mathbf{D})=(\mathbf{D}^t\mathbf{Z}\mathbf{D})^{1/2}$; **D** is a unit vector perpendicular to the diffraction plane);
- **type 2** (*type_2): particle size, the **W** tensor is refined (variables EXT11-EXT23) ($\rho(\mathbf{N})= \lambda(\mathbf{N}^t\mathbf{W}\mathbf{N})^{-1/2}$; **N** is a unit vector in the diffraction plane, perpendicular to the incident beam);
- **type 3** (*type_3, *type_g is also accepted): generalized type (mosaic spread and particle size), **Z** tensor and ρ simultaneously refined (variables EXT11-EXT23 for **Z**; variable RHOEX for ρ);

mosaic spread distribution (active only for type_1 and type_3):

- Gaussian (*distr_g, default): a Gaussian distribution is assumed;
- Lorentzian (*distr_l) : a Lorentzian distribution is assumed;

mosaic orientation (active only for type_1 and 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** ($g(\mathbf{D})=(\mathbf{D}^t\mathbf{Y}\mathbf{D})^{-1/2}$).

Given the expression for tensors **W**, **Y**, and **Z**, a switch from isotropic to anisotropic correction requires the following variables in `xd.inp`:

(1) g' to **Z**:
$EXT11 = EXT22 = EXT33 = (g'_{iso})^2$
EXT12 = EXT13 = EXT23 = 0.0
(2) g' to **Y**:
$EXT11 = EXT22 = EXT33 = 1/(g'_{iso})^2$
EXT12 = EXT13 = EXT23 = 0.0
(3) ρ to **W**:
$EXT11 = EXT22 = EXT33 = 1/(\rho_{iso})^2$
EXT12 = EXT13 = EXT23 = 0.0

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: the mosaic spread, η (η is proportional to 1/g', units in seconds) and the domain size *r* ($r = g'\lambda 10^{-4}$, units in centimeters) are given, as derived from the refined g' and/or ρ.
- for anisotropic extinction: The principal axes of mosaic spread η(**D**) (if *type_1 or *type_3) or domain size *r*(**N**) (*type_2) distribution are given; the corresponding 'equivalent' η and *r* scalars are computed (or the refined *r* is printed if *type_3 is applied).

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_{obs} and the phases. The latter quantities may have the effects of anomalous dispersion removed [34] as follows.

$$A^0_{obs} = \frac{F_{obs}A_{calc}}{F_{calc}} - (A_{calc} - A^0_{calc})$$

$$B^0_{obs} = \frac{F_{obs} B_{calc}}{F_{calc}} - (B_{calc} - B^0_{calc})$$

$$F^0_{obs} = \sqrt{(A^0_{obs})^2 + (B^0_{obs})^2}$$

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. The standard deviation of $F_{obs}{}^0$ is estimated as

$$\sigma\left(\left|F^0_{obs}\right|\right) = \sigma\left(\left|F_{obs}\right|\right) \frac{\sqrt{\left(A^0_{obs} A_{calc}\right)^2 + \left(B^0_{obs} B_{calc}\right)^2}}{F^0_{obs} F_{calc}}$$

The phases are based on the model applied in the refinement (see instruction **MODEL**).

Each of the structure factor models (**fmod1**, **fmod2**) is specified with four integers, in the same way as described above. The combination of **fobs**, **fmod1** and **fmod2** makes it possible to generate six different Fourier maps (see XDFOUR & XDFFT). 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 (*lmax*=4), the second one on a neutral spherical atom model (*lmax*=-1). Both are free of anomalous dispersion and extinction.

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

IMPORTANT In contrast to earlier versions of the program, it is the values of *m1.3* and *m1.4* for **fmod1** which determine the treatment of F_{obs}. Anomalous dispersion effects will only be removed from F_{obs} if *m1.3* is given as zero. Likewise, F_{obs} will only be corrected for extinction effects (if any) when *m1.4* is given as zero. Since it is normally desired that Fourier syntheses should represent the electron density, and be independent of the wavelength used for the experiment, the effects of anomalous dispersion must be removed from experimental F_{obs} [34,35]. The values of *m1.3* and *m1.4* must therefore be given as zero, as in the above example.

Another new feature in version 5.3 is the addition (at the end of the `xd.fou` file) of an entry for reflection *F*(000). The inclusion of *F*(000) in Fourier summations ensures a correct absolute scale for the electron density.

Only those reflections used in the refinement will be written to the XD.FOU file.

4.6.10 CON - General linear constraint

CON *a1 var1 a2 var2 a3 var3 ... = c*

The **CON** instruction defines a linear combination among a set of variables used as a constraint. A list of the coefficients (*a1,a2,a3,...*) and variable-symbols (*var1, var2, var3,...*) 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:

```
X/1, U12/12, C333/1, M1/2, H4+/11, KS/3, K2/2, SCALE/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.

Applications of great practical importance are the constraints due to crystallographic site symmetries. *Unlike in SHELXL97 and other modern refinement programs, these constraints are NOT treated automatically in XDLSM !* 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:

No.	atom	Wyckoff letter	x	y	z
1	K	a	0	0	1/4
2	F	h	x	x+1/2	0
3	H	d	0	1/2	0

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 in the use of Kubic harmonics. Table 4-4 shows the density normalized Kubic harmonics as linear combinations of density normalized spherical harmonics, and also indicates which Kubic harmonics are symmetric with respect to the cubic site symmetries (and hence are allowed). Suppose that atom 1 is at a site of *-43m* symmetry in a cubic space group. Table 4-4 shows that the only symmetry allowed Kubic harmonics (of relevance to the current version of XD) are $K_{0,1}$ $K_{3,1}$ and $K_{4,1}$. The first two simply correspond to M1,M2 and O2- (y_{00} and $y_{3,-2}$) while $K_{4,1}$ is a linear combination of the hexadecapoles H0 and H4+ (0.78245 H0 + 0.57939 H4+). The corresponding constraint is thus

```
CON  0.57939 H0/1 -0.78245  H4+/1 = 0
```

or more simply

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

Site symmetry restrictions on spherical harmonics are given in Table 4-5 [25]. Tables 4-6 to 4-9 (simplified versions of Tables 5.5A - 5.5D from reference 24) are reproduced with kind permission of the IUCr. These Tables provide the site symmetry restrictions on the U^{ij} anisotropic thermal motions tensors and on the Gram-Charlier anharmonic coefficients C^{ijk} and D^{ijkl}. The order of indices for C^{ijk} and D^{ijkl} in these Tables follows the order of symbolic names used by XD (see Table 4-2). Table 4-6 should be consulted first, to obtain the appropriate cross reference key for the crystallographic site symmetry in question.

IMPORTANT In the current version of the program, there is an idiosyncracy concerning the CON instruction. If more than two parameters are involved in a particular constraint, *e.g.* if U11=U22=U33, then all three constraints should be explicitly given. The linear dependency is noticed and removed, but the user should still use this procedure, otherwise the program may fail.

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^{ij}, C^{ijk}, D^{ijkl}, P_{lm}) have to be given for all of the atoms included in the structure model. These are followed by the "shared" (κ, κ'_l) 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,Cn,Dn	nth order displacement tensor components. There are 6, 10 and 15 components 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 (κ'_l, 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 devf (κ'_l, 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 the thermal motion 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) may either be fixed or free in the KEY table. On the other hand, parameters involved in any other constraint *must* be made variables (*i.e.* set to "1" in the KEY table). The program will halt if this is not the case. 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 ----------
```

4.7 Appendix

4.7.1 Treatment of the reflection data by XDLSM

Regardless of the format of Y_{obs} (*i.e* F_{obs} or F^2_{obs}) supplied in the reflection file `XD.HKL`, all the values F_{obs}, $\sigma(F_{obs})$, F^2_{obs} and $\sigma(F^2_{obs})$ are calculated. The standard interconversions, given below, are used. The refinement procedure minimises either $w(\Delta F)$ or $w(\Delta F^2)$, depending on the **based_on** record in XD.MAS, but R-indices for both F and F^2 are always reported. Since a zero or negative value for $\sigma(Y_{obs})$ leads to physically meaningless weights in a least-squares refinement, any record in XD.HKL which has a zero or negative $\sigma(Y_{obs})$ is *ignored* (*i.e.* it is immediately skipped and not counted). A warning message is issued on the console, and such reflections should be removed from XD.HKL

The expressions for deriving the quantities F_{obs}, $\sigma(F_{obs})$, F^2_{obs} and $\sigma(F^2_{obs})$ from each other are :

If F_{obs} and $\sigma(F_{obs})$ are supplied in XD.HKL then

$F^2_{obs} = (F_{obs})^2$ but F^2_{obs} *retains the sign* of F_{obs} (see note 1)
$\sigma(F^2_{obs}) = 2\times\sigma(F_{obs}) \times |F_{obs}|$ for the case where $|F_{obs}| > 0$
$\sigma(F^2_{obs}) = 2\times\sigma(F_{obs})$ for the case where $F_{obs} = 0$ (see note 2)

If F^2_{obs} and $\sigma(F^2_{obs})$ are supplied in XD.HKL then

$F_{obs} = \sqrt{|F^2_{obs}|}$ but F_{obs} *retains the sign* of F^2_{obs}

$\sigma(F_{obs}) = \dfrac{0.5\times\sigma(F^2_{obs})}{|F_{obs}|}$ for the case where $|F_{obs}| > 0$ (see note 2)

$\sigma(F_{obs}) = 0.5\times\sigma(F^2_{obs})$ for the case where $F_{obs} = 0$

These conventions ensure that $\sigma(F_{obs})$ and $\sigma(F^2_{obs})$ are always > 0, but F_{obs} or F^2_{obs} may be either negative, zero or positive.

4.7.2 Definitions of the Residual indices used in XDLSM

$$R(F) = \frac{\sum |F_{obs} - k_1 F_{calc}|}{\sum |F_{obs}|} \qquad R(F^2) = \frac{\sum |F^2_{obs} - k_2 F^2_{calc}|}{\sum |F^2_{obs}|}$$

$$wR(F) = \sqrt{\frac{\sum w(F_{obs} - k_1 F_{calc})^2}{\sum w |F^2_{obs}|}} \qquad wR(F^2) = \sqrt{\frac{\sum w(F^2_{obs} - k_2 F^2_{calc})^2}{\sum w(F^2_{obs})^2}}$$

The scale factors k_1/k_2 place Y_{obs} and Y_{calc} on the same scale. The moduli signs (| |) are only used here to indicate the *absolute value* of the appropriate quantity and *not* the commonly used alternative meaning of $|F|$ as the phaseless modulus of a structure factor. This distinction is important in summations where Y_{obs} may have a negative value (see note 3). The indices $R(F)$, $R(F^2)$, $wR(F)$ and $wR(F^2)$, reported in the file XD_LSM.OUT, are computed *only* for those reflections used in the refinement (*i.e.* those passing the rejection criteria on the SKIP directive). The indices $R_{all}(F)$ and $R_{all}(F^2)$ have the same definition as $R(F)$, $R(F^2)$, but also include those reflections rejected by the rejection criteria - this will mean *all* reflections in XD.HKL, except any ignored because of zero or negative $\sigma(Y_{obs})$.

4.7.2 Definitions of the Goodness of Fit indices used in XDLSM

The Goodness of Fit (*GOF*) indices are defined thus :

$$GOF = \sqrt{\frac{\sum (\Delta fs)^2}{(N_{ref} - N_{\text{var}})}} \qquad GOFw = \sqrt{\frac{\sum w(\Delta f)^2}{(N_{ref} - N_{\text{var}})}}$$

where N_{ref} is the number of observations and N_{var} the number of independent variables in the least-squares refinement. The definitions of Δf and Δfs depend of whether refinement was based on F or F^2, and unlike the R-indices, the GOF indices are only reported for the actual refinement mode.

For refinement on F

$\Delta f = (F_{obs} - k_1 F_{calc})$ and $\Delta fs = (F_{obs} - k_1 F_{calc})/\sigma(F_{obs})$

For refinement on F^2
$\Delta f = (F^2_{obs} - k_2F^2_{calc})$ and $\Delta fs = (F^2_{obs} - k_2F^2_{calc})/\sigma(F^2_{obs})$

GOFw uses the reflection weights *w* as defined on the WEIGHT directive in the XDLSM section of XD.MAS (see Section 4.6.6). *GOF* is the Goodness of Fit using statistical weights *i.e.* $1/\sigma^2(Y_{obs})$, so the two *GOF* indices will be identical if this weighting scheme is used (*i.e.* if the *a* parameter on the WEIGHT directive has a value less than -1.0). Note that the default values of zero given for parameters *a* - *e* in Section 4.6.6 will also imply statistical weighting if refinement is based on F^2, but NOT if refinement is based on *F* !

If the general form of the XD weighting scheme is used, the definitions of the weighted *R* indices given above have the unfortunate property of being dependent on the overall scale of the data being summed. The weighting scheme used for refinement on F^2 (which is the same scheme as used in SHELXL), in its most commonly used simple form, where only the *a* and *b* parameters are non-zero, is given by :

$$w = 1/(\sigma^2(F^2_{obs}) + a^2\mathrm{P}^2 + b\mathrm{P})$$

where $\mathrm{P} = (2F^2_{calc} + F^2_{obs})/3$ for the default value of parameter *f*.

If the observations used to accumulate the *R*-indices are on some arbitary scale compared with the absolute scale, then F^2_{obs}, F^2_{calc} and $\sigma(F^2_{obs})$ are all effectively multiplied by a constant *s*, so we are dealing with sF^2_{obs}, sF^2_{calc} and $s\sigma(F^2_{obs})$. The expression for $wR(F^2)$ becomes

$$wR(F^2) = \sqrt{\frac{\sum ws^2(F^2_{obs} - k_2F^2_{calc})^2}{\sum ws^2(F^2_{obs})^2}}$$

If unit weights are used, then clearly the w*R* indices are independent of *s*. The same pertains if statistical weights are used. In this case $w = 1/s^2\sigma^2(F^2_{obs})$ and the constant s^2 is eliminated. *However* if the general form of the weighting scheme given above is used, the weight is now given by

$$w = 1/(s^2\sigma^2(F^2_{obs}) + s^2a^2\mathrm{P}^2 + sb\mathrm{P})$$

and a dependence on *s* remains. Since the current version of XDLSM accumulates the summations on the *observational* scale, while the structure factors in XD.FCO are on an *absolute scale*, it is impossible to correctly compute the weighted *R*-indices if a general form of the weighting function is used. It would be more logical to accumulate the indices in XDLSM on an absolute scale.

4.7.3 NOTES

Some minor changes in the above definitions in XD for version 6.02 may result in small changes to reported *R* values, compared with earlier versions.

1. Previously F^2_{obs} was simply taken as $(F_{obs})^2$, but this has been changed to distinguish between positive and negative F_{obs}. Since the majority of data reduction programs will never report negative F_{obs}, this change is unlikely to have any consequences.

2. For this case $\sigma(F^2_{obs})$ or $\sigma(F_{obs})$ were previously either undefined, or arbitarily set to the value of the input $\sigma(Y_{obs})$. For refinements which use any type of sigma cut-off, this change will not have any consequences.

3. Previously, the denominator for $R(F^2)$ was simply $\Sigma\ F^2_{obs}$, while that for $R(F)$ was $\Sigma|F_{obs}|$. For consistency, both summations now use the *absolute* value of the summand, as indicated above. In cases where data sets contain negative F^2_{obs}, this change may means small differences in $R_{all}(F^2)$ and possibly in $R(F^2)$, compared with previous versions.

Table 4-1 : Dimensioning of XDLSM

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

Related to these the following parameters are also in use:

Name	Value	Description
npop	*lmx * lmx + 2 * lmx + 2*	maximum number of multipole populations
nap	*3 + ntx + npop*	maximum number of atomic parameters
npp	*nap * nat + (lmx + 2) * nzz + nsc + 8*	total number of parameters

Table 4-2 : Variable names and order numbers

Parameter	Symbolic name	Order number
Fractional Coordinates	X, Y, Z	1-3
Displacement Tensor Components		
2nd order U^{ij}	U11, U22, U33, U12, U13, U23	4-9
3rd order C^{ijk}	C111, C222, C333, C112, C122, C113, C133, C223, C233, C123	10-19
4th order D^{ijkl}	D1111, D2222, D3333, D1112, D1222, D1113, D1333, D2223, D2333, D1122, D1133, D2233, D1123, D1223, D1233	19-34
Multipole Populations		
Monopoles	M1, M2	35-36
Dipoles	D1+, D1-, D0	37-39
Quadrupoles	Q0, Q1+, Q1-, Q2+, Q2-	40-44
Octupoles	O0, O1+, O1-, O2+, O2-, O3+, O3-	45-51
Hexadecapoles	H0, H1+, H1-, H2+, H2-, H3+, H3-, H4+, H4-	52-60
Radial Screening		
$\kappa, \kappa'(l)$	KS, K0, K1, K2, K3, K4	61-66
Isotropic and Anisotropic Extinction	EX11,EX22,EX33,EX12,EX13, EX23, RHOEX (for isotropic case ISOEX ≡ EX11)	67-73
Overall U	OVTHP	74
Scale Factor	SCALE	75

IMPORTANT NOTE : Following normal conventions, the magnitudes of the Gram-Charlier coefficients C^{ijk} and D^{ijkl}, which are reported in XD_LSM.OUT and XD_LSM.CIF, are multiplied by 10^3 and 10^4 respectively.

Table 4-3 : The model limits

m1 static scattering models	
-4	neutron
-3	core
-2	conventional, spherical-atom promolecule model with RHF scattering factors taken from the International Tables
-1	neutral, spherical-atom model with HF scattering factors generated from Slater-type wavefunctions [14]
lmax	aspherical-atom model: frozen-core, spherical valence, multipolar deformation density up to *lmax* in the expansion over spherical harmonics [1]
m2 thermal motion models	
-1	overall-isotropic-harmonic
0	static
1	isotropic – harmonic
2	anisotropic – harmonic
tmax	anharmonic model: Gram-Charlier expansion up to 4th order [24]
m3 anomalous dispersion	
0	excluded
1	included
m4 extinction	
0	excluded
1	included

Table 4-4 : Density normalised Kubic harmonics K_{lj}

(a) as linear combinations of spherical harmonics

	mp			
Even *l* *l* *j*	0+	2+	4+	6+
0 1	1.0			
4 1	0.78245		0.57939	
6 1	0.37790		-0.91682	
6 2		0.83848		-0.5
Odd *l* *l* *j*	2-	4-	6-	8-
3 1	1.0			
7 1	0.73145		0.63290	

(b) site symmetry

l *j*	23 *T*	m-3 T_h	432 O	-43m T_d	m-3m O_h
0 1	yes	yes	yes	yes	yes
3 1	yes	no	no	yes	no
4 1	yes	yes	yes	yes	yes
6 1	yes	yes	yes	yes	yes
6 2	yes	yes	no	no	no
7 1	yes	no	no	yes	no

Table 4-5 : Index Picking Rules of Site-Symmetric Spherical Harmonics [25]

Symmetry	Choice of coordinate axes	Indices of symmetric y_{lmp} (λ, μ are integers)
1	any	all (*l*,m,±)
$\bar{1}$	any	(2λ,m, ±)
2	2 || z	(*l*,2μ, ±)
m	m ⊥ z	(*l*,*l*-2μ, ±)
2/m	2 || z, m ⊥ z	(2λ,2μ, ±)
222	2 || z, 2 || y, (2 || x)	(2λ,2μ,+), (2λ+1,2μ,–)
mm2	2 || z, m ⊥ y, (m ⊥ x)	(*l*,2μ,+)
mmm	m ⊥ z, m ⊥ y, m ⊥ x	(2λ,2μ, +)
4	4 || z	(*l*,4μ,±)
$\bar{4}$	$\bar{4}$ || z	(2λ,4μ, ±), (2λ+1,4μ+2,±)
4/m	4 || z, m ⊥ z	(2λ,4μ, ±)
422	4 || z, 2 || y, (2 || x)	(2λ,4μ,+), (2λ+1,4μ,–)
4mm	4 || z, m ⊥ y, (m ⊥ x)	(*l*,4μ,+)
$\bar{4}$2m	$\bar{4}$ || z, 2 || x, $(m \hat{=} xy \rightarrow yx)$	(2λ,4μ,+),(2λ+1,4μ+2,–)
	m ⊥ y, $(2 \hat{=} xyz \rightarrow yx\bar{z})$	(2λ,4μ,+),(2λ+1,4μ+2,+)
4/mmm	4 || z, m ⊥ z, (m ⊥ x), $(m \hat{=} xy \rightarrow yx)$	(2λ,4μ,+)
3	3 || z	(*l*,3μ,±)
$\bar{3}$	$\bar{3}$ || z	(2λ,3μ,±)
32	3 || z, 2 || y,	(2λ,3μ,+),(2λ+1,3μ,–)
	2 || x	(3μ+2*j*,3μ,+),(3μ+2*j*+1,3μ,–)
3m	3 || z, m ⊥ y	(*l*,3μ,+)
	m ⊥ x	(*l*,6μ,+),(*l*,6μ+3,–)
$\bar{3}$m	$\bar{3}$ || z, m ⊥ y	(2λ,3μ,+)
	m ⊥ x	(2λ,6μ,+),(2λ,6μ+3,–)
6	6 || z	(*l*,6μ,±)
$\bar{6}$	$\bar{6}$ || z = (3 || z, m ⊥ z)	(2λ,6μ,±),(2λ+1,6μ+3,±)
6/m	6 || z, m ⊥ y 2 || y, (2 || x)	(2λ,6μ,±)
622	6 || z, 2 || y, (2 || x)	(2λ,6μ,+),(2λ+1,6μ,–)
6mm	6 || z, m ⊥ y, (m ⊥ x)	(*l*,6μ,+)
$\bar{6}$m2	$\bar{6}$ || z, m ⊥ y, (2 || x)	(2λ,6μ,+),(2λ+1,6μ+3,+)
	m ⊥ x, (2 || y)	(2λ,6μ,+),(2λ+1,6μ+3,–)
6/mmm	6 || z, m ⊥ z, m ⊥ y, (m ⊥ x)	(2λ,6μ,+)

Table 4-6 : Site Symmetry Table Giving the Key to Tables 4-7 to 4-9 (Hex indicates hexagonal axes).

Point Symmetry at Special Position					Position	Cross reference key		
Symmetry Axes		Point Group Generators			x,y,z	4-7	4-8	4-9
mšm		4[0,0,1]	3[1,1,1]	˜	0,0,0	B1	C0	D1
›3m		›[0,0,1]	3[1,1,1]		0,0,0	B1	C1	D1
432		4[0,0,1]	3[1,1,1]		0,0,0	B1	C0	D1
mš		3[1,1,1]	2[0,0,1]	˜	0,0,0	B1	C0	D1
23		3[1,1,1]	2[0,0,1]		0,0,0	B1	C1	D1
6/mmm	Hex	6[0,0,1]	2[1,0,0]	˜	0,0,0	B9	C0	D2
œm2	Hex	œ[0,0,1]	2[1,0,0]		0,0,0	B9	C9	D2
œm2	Hex	œ[0,0,1]	2[1,2,0]		0,0,0	B9	C10	D2
6mm	Hex	6[0,0,1]	™[1,0,0]		0,0,z	B9	C19	D2
622	Hex	6[0,0,1]	2[1,0,0]		0,0,0	B9	C0	D2
6/m	Hex	6[0,0,1]	˜		0,0,0	B9	C0	D2
œ	Hex	œ[0,0,1]			0,0,0	B9	C20	D2
6	Hex	6[0,0,1]			0,0,z	B9	C19	D2
4/mmm		4[0,0,1]	2[1,0,0]	˜	0,0,0	B2	C0	D3
4/mmm		4[0,1,0]	2[0,0,1]	˜	0,0,0	B3	C0	D4
4/mmm		4[1,0,0]	2[0,1,0]	˜	0,0,0	B4	C0	D5
›2m		›[0,0,1]	2[1,0,0]		0,0,0	B2	C1	D3
›2m		›[0,0,1]	2[1,1,0]		0,0,0	B2	C2	D3
›2m		›[0,1,0]	2[0,0,1]		0,0,0	B3	C1	D4
›2m		›[0,1,0]	2[1,0,1]		0,0,0	B3	C3	D4
›2m		›[1,0,0]	2[0,1,0]		0,0,0	B4	C1	D5
›2m		›[1,0,0]	2[0,1,1]		0,0,0	B4	C4	D5
4mm		4[0,0,1]	™[1,0,0]		0,0,z	B2	C13	D3
4mm		4[0,1,0]	™[0,0,1]		0,y,0	B3	C14	D4
4mm		4[1,0,0]	™[0,1,0]		x,0,0	B4	C15	D5
422		4[0,0,1]	2[1,0,0]		0,0,0	B2	C0	D3
422		4[0,1,0]	2[0,0,1]		0,0,0	B3	C0	D4
422		4[1,0,0]	2[0,1,0]		0,0,0	B4	C0	D5
4/m		4[0,0,1]	˜		0,0,0	B2	C0	D12
4/m		4[0,1,0]	˜		0,0,0	B3	C0	D13
4/m		4[1,0,0]	˜		0,0,0	B4	C0	D14
›		›[0,0,1]			0,0,0	B2	C16	D12
›		›[0,1,0]			0,0,0	B3	C17	D13
›		›[1,0,0]			0,0,0	B4	C18	D14
4		4[0,0,1]			0,0,z	B2	C13	D12
4		4[0,1,0]			0,y,0	B3	C14	D13
4		4[1,0,0]			x,0,0	B4	C15	D14
šm		3[1,1,1]	2[1,˜,0]	˜	0,0,0	B5	C0	D6
šm		3[1,1,˜]	2[1,˜,0]	˜	0,0,0	B6	C0	D7
šm		3[1,˜,1]	2[1,1,0]	˜	0,0,0	B7	C0	D8
šm		3[˜,1,1]	2[1,1,0]	˜	0,0,0	B8	C0	D9
šm	Hex	3[0,0,1]	2[1,0,0]	˜	0,0,0	B9	C0	D10
šm	Hex	3[0,0,1]	2[1,2,0]	˜	0,0,0	B9	C0	D11
3m		3[1,1,1]	™[1,˜,0]		x,x,x	B5	C33	D6
3m		3[1,1,˜]	™[1,˜,0]		x,x,Đ	B6	C34	D7
3m		3[1,˜,1]	™[1,1,0]		x,Đ,x	B7	C35	D8
3m		3[˜,1,1]	™[1,1,0]		Đ,x,x	B8	C36	D9
3m	Hex	3[0,0,1]	™[1,0,0]		0,0,z	B9	C37	D10
3m	Hex	3[0,0,1]	™[1,2,0]		0,0,z	B9	C38	D11
32		3[1,1,1]	2[1,˜,0]		0,0,0	B5	C5	D6

Table 4-6 : (cont)

Point Symmetry at Special Position					Position	Cross reference key		
Symmetry Axes		Point Group Generators			x,y,z	**4-7**	**4-8**	**4-9**
32		3[1,1,˜]	2[1,˜,0]		0,0,0	B6	C6	D7
32		3[1,˜,1]	2[1,1,0]		0,0,0	B7	C7	D8
32		3[˜,1,1]	2[1,1,0]		0,0,0	B8	C8	D9
32	Hex	3[0,0,1]	2[1,0,0]		0,0,0	B9	C9	D10
32	Hex	3[0,0,1]	2[1,2,0]		0,0,0	B9	C10	D11
š		š[1,1,1]			0,0,0	B5	C0	D15
š		š[1,1,˜]			0,0,0	B6	C0	D16
š		š[1,˜,1]			0,0,0	B7	C0	D17
š		š[˜,1,1]			0,0,0	B8	C0	D18
š	Hex	š[0,0,1]			0,0,0	B9	C0	D19
3		3[1,1,1]			x,x,x	B5	C54	D15
3		3[1,1,˜]			x,x,Đ	B6	C55	D16
3		3[1,˜,1]			x,Đ,x	B7	C56	D17
3		3[˜,1,1]			Đ,x,x	B8	C57	D18
3	Hex	3[0,0,1]			0,0,z	B9	C58	D19
mmm		2[0,0,1]	2[1,0,0]	˜	0,0,0	B10	C0	D20
mmm		2[0,0,1]	2[1,1,0]	˜	0,0,0	B11	C0	D21
mmm		2[0,1,0]	2[1,0,1]	˜	0,0,0	B12	C0	D22
mmm		2[1,0,0]	2[0,1,1]	˜	0,0,0	B13	C0	D23
mmm	Hex	2[0,0,1]	2[1,0,0]	˜	0,0,0	B14	C0	D24
mmm	Hex	2[0,0,1]	2[1,1,0]	˜	0,0,0	B11	C0	D21
mmm	Hex	2[0,0,1]	2[0,1,0]	˜	0,0,0	B15	C0	D25
mm		2[0,0,1]	™[1,0,0]		0,0,z	B10	C21	D20
mm		2[0,0,1]	™[1,1,0]		0,0,z	B11	C22	D21
mm		2[0,1,0]	™[0,0,1]		0,y,0	B10	C23	D20
mm		2[0,1,0]	™[1,0,1]		0,y,0	B12	C24	D22
mm		2[1,0,0]	™[0,0,1]		x,0,0	B10	C25	D20
mm		2[1,0,0]	™[0,1,1]		x,0,0	B13	C26	D23
mm		2[1,1,0]	™[0,0,1]		x,x,0	B11	C27	D21
mm		2[1,˜,0]	™[0,0,1]		x,Đ,0	B11	C28	D21
mm		2[1,0,1]	™[0,1,0]		x,0,x	B12	C29	D22
mm		2[1,0,˜]	™[0,1,0]		x,0,Đ	B12	C30	D22
mm		2[0,1,1]	™[1,0,0]		0,y,y	B13	C31	D23
mm		2[0,1,˜]	™[1,0,0]		0,y,‹	B13	C32	D23
mm	Hex	2[0,0,1]	™[1,0,0]		0,0,z	B14	C40	D24
mm	Hex	2[0,0,1]	™[1,1,0]		0,0,z	B11	C22	D21
mm	Hex	2[0,0,1]	™[0,1,0]		0,0,z	B15	C39	D25
mm	Hex	2[1,0,0]	™[0,0,1]		x,0,0	B14	C41	D24
mm	Hex	2[2,1,0]	™[0,0,1]		2x,x,0	B15	C42	D25
mm	Hex	2[1,1,0]	™[0,0,1]		x,x,0	B11	C27	D21
mm	Hex	2[1,2,0]	™[0,0,1]		x,2x,0	B14	C43	D24
mm	Hex	2[0,1,0]	™[0,0,1]		0,y,0	B15	C44	D25
mm	Hex	2[1,˜,0]	™[0,0,1]		x,Đ,0	B11	C28	D21
222		2[0,0,1]	2[1,0,0]		0,0,0	B10	C1	D20
222		2[0,0,1]	2[1,1,0]		0,0,0	B11	C2	D21
222		2[0,1,0]	2[1,0,1]		0,0,0	B12	C3	D22
222		2[1,0,0]	2[0,1,1]		0,0,0	B13	C4	D23
222	Hex	2[0,0,1]	2[1,0,0]		0,0,0	B14	C11	D24
222	Hex	2[0,0,1]	2[1,1,0]		0,0,0	B11	C2	D21
222	Hex	2[0,0,1]	2[0,1,0]		0,0,0	B15	C12	D25
2/m		2[0,0,1]	˜		0,0,0	B16	C0	D26

Table 4-6 : (cont)

Point Symmetry at Special Position		Position	Cross reference key		
Symmetry Axes	Point Group Generators	x,y,z	**4-7**	**4-8**	**4-9**
2/m	2[0,1,0] ˜	0,0,0	B17	C0	D27
2/m	2[1,0,0] ˜	0,0,0	B18	C0	D28
2/m	2[1,1,0] ˜	0,0,0	B19	C0	D29
2/m	2[1,˜,0] ˜	0,0,0	B20	C0	D30
2/m	2[1,0,1] ˜	0,0,0	B21	C0	D31
2/m	2[1,0,˜] ˜	0,0,0	B22	C0	D32
2/m	2[0,1,1] ˜	0,0,0	B23	C0	D33
2/m	2[0,1,˜] ˜	0,0,0	B24	C0	D34
2/m Hex	2[0,0,1] ˜	0,0,0	B16	C0	D26
2/m Hex	2[1,0,0] ˜	0,0,0	B25	C0	D35
2/m Hex	2[2,1,0] ˜	0,0,0	B26	C0	D36
2/m Hex	2[1,1,0] ˜	0,0,0	B19	C0	D29
2/m Hex	2[1,2,0] ˜	0,0,0	B27	C0	D37
2/m Hex	2[0,1,0] ˜	0,0,0	B28	C0	D38
2/m Hex	2[1,˜,0] ˜	0,0,0	B20	C0	D30
m	™[0,1,0]	x,0,z	B17	C64	D27
m	™[1,0,0]	0,y,z	B18	C65	D28
m	™[1,1,0]	x,Đ,z	B19	C66	D29
m	™[1,˜,0]	x,x,z	B20	C67	D30
m	™[1,0,1]	x,y,Đ	B21	C68	D31
m	™[1,0,˜]	x,y,x	B22	C69	D32
m	™[0,1,1]	x,y,‹	B23	C70	D33
m	™[0,1,˜]	x,y,y	B24	C71	D34
m Hex	™[0,0,1]	x,y,0	B16	C63	D26
m Hex	™[1,0,0]	x,2x,z	B25	C72	D35
m Hex	™[2,1,0]	0,y,z	B26	C73	D36
m Hex	™[1,1,0]	x,Đ,z	B19	C66	D29
m Hex	™[1,2,0]	x,0,z	B27	C74	D37
m Hex	™[0,1,0]	2x,x,z	B28	C75	D38
m Hex	™[1,˜,0]	x,x,z	B20	C67	D30
2	2[0,0,1]	0,0,z	B16	C45	D26
2	2[0,1,0]	0,y,0	B17	C46	D27
2	2[1,0,0]	x,0,0	B18	C47	D28
2	2[1,1,0]	x,x,0	B19	C48	D29
2	2[1,˜ ,0]	x,Đ,0	B20	C49	D30
2	2[1,0,1]	x,0,x	B21	C50	D31
2	2[1,0,˜]	x,0,Đ	B22	C51	D32
2	2[0,1,1]	0,y,y	B23	C52	D33
2	2[0,1,˜]	0,y,‹	B24	C53	D34
2 Hex	2[0,0,1]	0,0,z	B16	C45	D26
2 Hex	2[1,0,0]	x,0,0	B25	C59	D35
2 Hex	2[2,1,0]	2x,x,0	B26	C60	D36
2 Hex	2[1,1,0]	x,x,0	B19	C48	D29
2 Hex	2[1,2,0]	x,2x,0	B27	C61	D37
2 Hex	2[0,1,0]	0,y,0	B28	C62	D38
2 Hex	2[1,˜ ,0]	x,Đ,0	B20	C49	D30
˜	˜	0,0,0	B29	C0	D39
˜ Hex	˜	0,0,0	B29	C0	D39
1	1	x,y,z	B29	C76	D39
1 Hex	1	x,y,z	B29	C76	D39

Table 4-7 : Site symmetry restrictions on coefficients of the U^{ij} tensor.

Cross reference from Table 4-6	Number of indep. variables	Symbols and coeffcient indices					
		A (1) (1)	B (2) (2)	C (3) (3)	D (1) (2)	E (1) (3)	F (2) (3)
B1	1	A	A	A	0	0	0
B2	2	A	A	C	0	0	0
B3	2	A	B	A	0	0	0
B4	2	A	B	B	0	0	0
B5	2	A	A	A	D	D	D
B6	2	A	A	A	D	-D	-D
B7	2	A	A	A	D	-D	D
B8	2	A	A	A	D	D	-D
B9	2	A	A	C	A/2	0	0
B10	3	A	B	C	0	0	0
B11	3	A	A	C	D	0	0
B12	3	A	B	A	0	E	0
B13	3	A	B	B	0	0	F
B14	3	A	B	C	B/2	0	0
B15	3	A	B	C	A/2	0	0
B16	4	A	B	C	D	0	0
B17	4	A	B	C	0	E	0
B18	4	A	B	C	0	0	F
B19	4	A	A	C	D	E	-E
B20	4	A	A	C	D	E	E
B21	4	A	B	A	D	E	-D
B22	4	A	B	A	D	E	D
B23	4	A	B	B	D	-D	F
B24	4	A	B	B	D	D	F
B25	4	A	B	C	B/2	F/2	F
B26	4	A	B	C	A/2	0	F
B27	4	A	B	C	B/2	E	0
B28	4	A	B	C	A/2	E	E/2
B29	6	A	B	C	D	E	F

Table 4-8 : Site symmetry restrictions on third-order Gram-Charlier coefficients C^{ijk}.

Cross reference from Table 4-6	Number of indep. variables	Symbols and coefficient indices									
		A	B	C	D	E	F	G	H	I	J
		(1)	(2)	(3)	(1)	(1)	(1)	(1)	(2)	(2)	(1)
		(1)	(2)	(3)	(1)	(2)	(1)	(3)	(2)	(3)	(2)
		(1)	(2)	(3)	(2)	(2)	(3)	(3)	(3)	(3)	(3)
C0	0	0	0	0	0	0	0	0	0	0	0
C1	1	0	0	0	0	0	0	0	0	0	J
C2	1	0	0	0	0	0	F	0	-F	0	0
C3	1	0	0	0	D	0	0	0	0	-D	0
C4	1	0	0	0	0	E	0	-E	0	0	0
C5	1	0	0	0	D	-D	-D	D	D	-D	0
C6	1	0	0	0	D	-D	D	D	-D	-D	0
C7	1	0	0	0	D	D	D	-D	-D	-D	0
C8	1	0	0	0	D	D	-D	-D	D	-D	0
C9	1	0	0	0	D	D	0	0	0	0	0
C10	1	A	-A	0	A/2	-A/2	0	0	0	0	0
C11	1	0	0	0	0	0	F	0	0	0	F
C12	1	0	0	0	0	0	0	0	H	0	H
C13	2	0	0	C	0	0	F	0	F	0	0
C14	2	0	B	0	D	0	0	0	0	D	0
C15	2	A	0	0	0	E	0	E	0	0	0
C16	2	0	0	0	0	0	F	0	-F	0	J
C17	2	0	0	0	D	0	0	0	0	-D	J
C18	2	0	0	0	0	E	0	-E	0	0	J
C19	2	0	0	C	0	0	F	0	F	0	F/2
C20	2	A	-A	0	D	D-A	0	0	0	0	0
C21	3	0	0	C	0	0	F	0	H	0	0
C22	3	0	0	C	0	0	F	0	F	0	J
C23	3	0	B	0	D	0	0	0	0	I	0
C24	3	0	B	0	D	0	0	0	0	D	J
C25	3	A	0	0	0	E	0	G	0	0	0
C26	3	A	0	0	0	E	0	E	0	0	J
C27	3	A	A	0	D	D	0	G	0	G	0
C28	3	A	-A	0	D	-D	0	G	0	-G	0
C29	3	A	0	A	0	E	F	F	E	0	0
C30	3	A	0	-A	0	E	F	-F	-E	0	0
C31	3	0	B	B	D	0	D	0	H	H	0
C32	3	0	B	-B	D	0	-D	0	H	-H	0
C33	3	A	A	A	D	D	D	D	D	D	J
C34	3	A	A	-A	D	D	-D	D	-D	D	J
C35	3	A	-A	A	D	-D	-D	-D	-D	D	J
C36	3	A	-A	-A	D	-D	D	-D	D	D	J
C37	3	A	-A	C	A/2	-A/2	F	0	F	0	F/2
C38	3	0	0	C	D	D	F	0	F	0	F/2
C39	3	0	0	C	0	0	F	0	H	0	F/2
C40	3	0	0	C	0	0	F	0	H	0	H/2
C41	3	A	0	0	D	D	0	G	0	0	0
C42	3	A	B	0	A/2	A/6+2B/3	0	G	0	G/2	0
C43	3	A	B	0	B/6+2A/3	B/2	0	I/2	0	I	0
C44	3	0	B	0	D	D	0	0	0	I	0
C45	4	0	0	C	0	0	F	0	H	0	J
C46	4	0	B	0	D	0	0	0	0	I	J
C47	4	A	0	0	0	E	0	G	0	0	J
C48	4	A	A	0	D	D	F	G	-F	G	0
C49	4	A	-A	0	D	-D	F	G	-F	-G	0

Table 4-8 : (cont)

Cross reference from Table 4-6	Number of indep. variables	Symbols and coeffIcient indices									
		A	B	C	D	E	F	G	H	I	J
		(1)	(2)	(3)	(1)	(1)	(1)	(1)	(2)	(2)	(1)
		(1)	(2)	(3)	(1)	(2)	(1)	(3)	(2)	(3)	(2)
		(1)	(2)	(3)	(2)	(2)	(3)	(3)	(3)	(3)	(3)
C50	4	A	0	A	D	E	F	F	E	-D	0
C51	4	A	0	-A	D	E	F	-F	-E	-D	0
C52	4	0	B	B	D	E	D	-E	H	H	0
C53	4	0	B	-B	D	E	-D	-E	H	-H	0
C54	4	A	A	A	D	E	E	D	D	E	J
C55	4	A	A	-A	D	E	-E	D	-D	E	J
C56	4	A	-A	A	D	E	E	-D	-D	-E	J
C57	4	A	-A	-A	D	E	-E	-D	D	-E	J
C58	4	A	-A	C	D	D-A	F	0	F	0	F/2
C59	4	A	0	0	D	D	F	G	0	0	F
C60	4	A	B	0	A/2	A/6+2B/3	0	G	H	G/2	H
C61	4	A	B	0	B/6+2A/3	B/2	F	I/2	0	I	F
C62	4	0	B	0	D	D	0	0	H	I	H
C63	6	A	B	0	D	E	0	G	0	I	0
C64	6	A	0	C	0	E	F	G	H	0	0
C65	6	0	B	C	D	0	F	0	H	I	0
C66	6	A	-A	C	D	-D	F	G	F	-G	J
C67	6	A	A	C	D	D	F	G	F	G	J
C68	6	A	B	-A	D	E	F	-F	-E	D	J
C69	6	A	B	A	D	E	F	F	E	D	J
C70	6	A	B	-B	D	E	-D	E	H	-H	J
C71	6	A	B	B	D	E	D	E	H	H	J
C72	6	A	B	C	B/6+2A/3	B/2	F	I/2	H	I	H/2
C73	6	0	B	C	D	D	F	0	H	I	F/2
C74	6	A	0	C	D	D	F	G	H	0	H/2
C75	6	A	B	C	A/2	A/6+2B/3	F	G	H	G/2	F/2
C76	10	A	B	C	D	E	F	G	H	I	J

Table 4-9 : Site symmetry restrictions on fourth-order Gram-Charlier coefficients D^{ijkl}.

Cross reference from Table 4-6	Number of indep. variables	Symbols and coefficient indices														
		A (1) (1) (1) (1)	B (2) (2) (2) (2)	C (3) (3) (3) (3)	D (1) (1) (1) (2)	E (1) (2) (2) (2)	F (1) (1) (1) (3)	G (1) (3) (3) (3)	H (2) (2) (2) (3)	I (2) (3) (3) (3)	J (1) (1) (2) (2)	K (1) (1) (3) (3)	L (2) (2) (3) (3)	M (1) (1) (2) (3)	N (1) (2) (2) (3)	P (1) (2) (3) (3)
D1	2	A	A	A	0	0	0	0	0	0	J	J	J	0	0	0
D2	3	A	A	C	A/2	A/2	0	0	0	0	A/2	K	K	0	0	K/2
D3	3	A	A	C	0	0	0	0	0	0	J	K	K	0	0	0
D4	4	A	B	A	0	0	0	0	0	0	J	K	J	0	0	0
D5	4	A	B	B	0	0	0	0	0	0	J	J	L	0	0	0
D6	4	A	A	A	D	D	D	D	D	D	J	J	J	M	M	M
D7	4	A	A	A	D	D	-D	-D	-D	-D	J	J	J	M	M	-M
D8	4	A	A	A	D	D	-D	-D	D	D	J	J	J	M	-M	M
D9	4	A	A	A	D	D	D	D	-D	-D	J	J	J	M	-M	-M
D10	4	A	A	C	A/2	A/2	F	0	-F	0	A/2	K	K	F/2	-F/2	K/2
D11	4	A	A	C	A/2	A/2	0	0	0	0	A/2	K	K	M	M	K/2
D12	5	A	A	C	D	-D	0	0	0	0	J	K	K	0	0	0
D13	5	A	B	A	0	0	F	-F	0	0	J	K	J	0	0	0
D14	5	A	B	B	0	0	0	0	H	-H	J	J	L	0	0	0
D15	5	A	A	A	D	E	E	D	D	E	J	J	J	M	M	M
D16	5	A	A	A	D	E	-E	-D	-D	-E	J	J	J	M	M	-M
D17	5	A	A	A	D	E	-E	-D	D	E	J	J	J	M	-M	M
D18	5	A	A	A	D	E	E	D	-D	-E	J	J	J	M	-M	-M
D19	5	A	A	C	A/2	A/2	F	0	-F	0	A/2	K	K	M	M-F	K/2
D20	6	A	B	C	0	0	0	0	0	0	J	K	L	0	0	0
D21	6	A	A	C	D	D	0	0	0	0	J	K	K	0	0	P
D22	6	A	B	A	0	0	F	F	0	0	J	K	J	0	N	0
D23	6	A	B	B	0	0	0	0	H	H	J	J	L	M	0	0
D24	6	A	B	C	D	B/2	0	0	0	0	B/6+2D/3	K	L	0	0	L/2
D25	6	A	B	C	A/2	E	0	0	0	0	A/6+2E/3	K	L	0	0	K/2
D26	9	A	B	C	D	E	0	0	0	0	J	K	L	0	0	P
D27	9	A	B	C	0	0	F	G	0	0	J	K	L	0	N	0
D28	9	A	B	C	0	0	0	0	H	I	J	K	L	M	0	0
D29	9	A	A	C	D	D	F	G	-F	-G	J	K	K	M	-M	P
D30	9	A	A	C	D	D	F	G	F	G	J	K	K	M	M	P
D31	9	A	B	A	D	E	F	F	-E	-D	J	K	J	M	N	-M
D32	9	A	B	A	D	E	F	F	E	D	J	K	J	M	N	M
D33	9	A	B	B	D	E	-D	-E	H	H	J	J	L	M	N	-N
D34	9	A	B	B	D	E	D	E	H	H	J	J	L	M	N	N
D35	9	A	B	C	D	B/2	F	I/2	H	I	B/6+2D/3	K	L	H/6+2F/3	H/2	L/2
D36	9	A	B	C	A/2	E	0	0	H	I	A/6+2E/3	K	L	M	M	K/2
D37	9	A	B	C	D	B/2	F	G	0	0	B/6+2D/3	K	L	M	M	L/2
D38	9	A	B	C	A/2	E	F	G	H	G/2	A/6+2E/3	K	L	F/2	F/6+2H/3	K/2
D39	15	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P

Bibliography

1. N.K. Hansen and P. Coppens, *Acta Cryst.* **A34**, 909-921 (1978).
2. (a) F.L. Hirshfeld, *Acta Cryst.* **B27**, 769-781 (1971). (b) F.L. Hirshfeld, *Isr. J. Chem.* **16**, 198-201 (1977).
3. E. Prince and P.T. Boggs, *International Tables for X-ray Crystallography*, Vol. C, pp 594-603, Kluwer Academic Publishers, 1992.
4. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes*, Cambridge University Press, 1992.
5. P.R. Mallinson, T. Koritsanszky, E. Elkaim, N. Li and P. Coppens, *Acta Cryst.* **A44**, 336-342 (1988).
6. V. Schomaker and K.N. Trueblood, *Acta Cryst.* **B24**, 63-77 (1968).
7. F.L. Hirshfeld, *Acta Cryst.* **A32**, 239-244 (1976).
8. W. Hummel, J. Hauser and H.-B. Bürgi, *J.Mol.Graphics* **8.**, 214-220 (1990).
9. W. Hummel, A. Raselli and H.-B. Bürgi, *Acta Cryst.* **B46**, 683-692 (1990).
10. K. Ruedenberg, *J. Chem. Phys.* **19**, 1433-1434 (1951).
11. J. Epstein, J. Bentley and R.F. Stewart, *J.Chem. Phys.* **66**, 5564-5567 (1977).
12. N. Li, P. Coppens and J. Landrum, *Inorg. Chem.* **27**, 482-488 (1988).
13. R.F. Stewart, E.R. Davidson and W.T. Simpson, *J. Chem. Phys.* **42**, 3175 (1965).
14. E. Clementi and C. Roetti. *Atomic Data and Nuclear Data Tables* **14** 177-478 (1974).
15. E. Clementi and D. L. Raimondi, *J. Chem. Phys.* **38**, 2686-2689 (1963)
16. *International Tables for X-ray Crystallography*, Vol. IV, pp 103-144, Kynoch Press: Birmingham, 1974.
17. Z. Su and P. Coppens *Acta Crysta.* **A54**, 646- (1998).
18. P. Macchi and P. Coppens *Acta Cryst.* **A57**, 656- (2001).
19. P.J. Becker and P. Coppens, P. *Acta Cryst.* **A30**, 129-147 (1974).
20. P.J. Becker and P. Coppens, *Acta Cryst.* **A30**, 148-153 (1974).
21. P.J. Becker and P. Coppens, *Acta Cryst.* **A31**, 417-425 (1975).
22. P.Coppens, and W. C. Hamilton, *Acta Cryst.* ***A26***, 71-83 (1970).
23. F. R. Thorney and R. J. Nelmes, *Acta Cryst.*, **A30**, 748-757 (1974).
24. C. K. Johnson and H. A. Levy *International Tables for X-ray Crystallography*, Vol. IV, pp 324-336, Kynoch Press: Birmingham, 1974.
25. K. Kurki-Suonio, *Israel J. Chem.* **16**, 115-123 (1977).
26. R. H. Blessing, *Acta Cryst.* **B51**, 816-823 (1995).
27. R.F. Stewart, *Israel J. Chem.* **16**, 124-131 (1977).
28. A. El Haouzi, N. K. Hansen, C. Le Henaff and J. Protas *Acta Cryst.*, **A52**, 291-301 (1996)
29. M. A. Spackman and P. G. Byrom *Acta Cryst.*, **B53**, 553-564 (1997).
30. A. Ø. Madsen, H. S. Osholm. R. F. Stewart, C. Flensburg and S. Larsen *Acta Cryst.*, **A60**, 550-561 (2004).
31. A. Ø. Madsen *J. Appl. Cryst.* **39**, 757-758 (2006).
32. P. Roversi and R. Destro *Chem. Phys. Lett* **386**, 472-478 (2004).
33. A. E. Whitten and M. A. Spackman *Acta Cryst.*, **B62**, 875-888 (2006) and refs therein.
34. J. A. Ibers and W. C. Hamilton *Acta Cryst.*, **17**, 781-782 (1964).
35 (a) A. L. Patterson *Acta Cryst.*, **16**, 1255-1256 (1963). (b) A. C. Larson in Crystallographic Computing Techniques (ed A Ahmed), Munkgaard, (1976), pp 396-398.

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 $c_i = \sigma(p_i)$. 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 **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:

$$C_{ij} = \mathrm{cov}(p_i, p_j) = c_i c_j r_{ij}$$

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 *rmin* 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 conjuction 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

1. W.R. Busing, K.O. Martin and H. Levy, *ORFFE – A Fortran Crystallographic Function and Error Program*, Oak Ridge National Laboratory (1964), ORNL-TM-306.
2. R.H. Stanford and J. Waser *Acta Cryst.* **A28**, 213-215, (1972)
3. J.S. Rollett, in *Computing Methods in Crystallography*, Ed. Rollett, J.S., Pergamon Press, 1965.
4. S. R. Hall, F. H. Allen and I. D. Brown *Acta Cryst.* **A47**, 655-685 (1991).
5. P. R. Mallinson and I. D. Brown, *International Tables for X-ray Crystallography*, Vol. G, Sect 3.5, Kluwer Academic Publishers, to be published.

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). 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 recent books and reviews [3].

The version of XDPROP in XD2015 is substantially upgraded from the previous release (4.10), and is under continuous development. New features include the calculation of lattice energies and interaction energies (**LATEN** & **INTEREN**), gradient trajectory plots of ρ and the esp (**TRAJPLT**), vector plots of the electric field (**VECPLOT**), atom basin integration of properties (**TOPINT**) and calculation of the local and integrated source function (**SOURCE**). Experienced users of XD are advised to read this Chapter carefully, as a number of options have been significantly modified.

6.1.1 Abbreviations

This chapter uses the following notation and abbreviations: ρ (total electron density); CP (Critical Point); $\Delta\rho$ (Deformation Density); IAM (Independent Atom Model); $V(\mathbf{r})$ (electrostatic potential); $v(r)$ (nuclear potential); BP (Bond Path - a line of maximum charge density connecting two nuclei); PD charges (electrostatic Potential-Derived charges); 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 ζ_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_<prop>.cps` critical points with eigenvectors of Hessian
	`xd_<prop>.grd` 2D or 3D XD grid files from MAP/CUBE
	`xd_<prop>.cube` Gaussian-style grid files from MAP/CUBE
	`xd.pth` bond path file for visualisation with XDGRAPH
	`<prop>_traj.ps` PostScript graphic file from TRAJPLT
	`<prop>_traj.cgm` CGM graphic file from TRAJPLT
	`xd_<prop>.dat` vector map for visualisation with PLOTMTV

`xd_bubble.spf` critical points in Laplacian for PLATON
`min16.inp` atomic moments in spherical tensor notation for MIN16
`orient.inp` atomic moments in spherical tensor notation for ORIENT

6.1.4 Coordinate systems

Although XDLSM uses local coordinate systems on the pseudoatoms, XDPROP works almost exclusively in a single, "global" system of Cartesian coordinates $\mathbf{r}$ generated from the fractional co-ordinates $\mathbf{r}_f$ in `xd.res` by the transformation

$$\mathbf{r} = \mathbf{M}\,\mathbf{r}_f \qquad \text{(Eq. 6.1)}$$

where **M** is the matrix [4]

$$\begin{bmatrix} & b\cos\gamma & c\cos\beta \\ 0 & b\sin\gamma & c(\cos\alpha-\cos\beta\cos\gamma)/\sin\gamma \\ 0 & 0 & \frac{c}{\sin\gamma}\sqrt{\left[\sin^2\gamma-\left(\cos^2\alpha+\cos^2\beta+\cos^2\gamma-2\cos\alpha\cos\beta\cos\gamma\right)\right]} \end{bmatrix}$$

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 ρ and $-\nabla^2\rho$ has some severe limitations at the moment, namely: (i) only contributions from the multipole populations are currently taken into account (*i.e.* not κ's or coordinates) (ii) contributions to the error from symmetry-generated atoms are not taken into account.
(b) The potential-derived charge fitting subroutine **QFIT** assumes that the fragment is molecular (neutral), so it will give nonsensical results for ions.
(c) It is only possible to compute errors on the dipole moment, ρ or $-\nabla^2\rho$ if Slater-type radial functions have been used for all multipoles.

6.2 General Instructions for XDPROP

The Master file entry for XDPROP should begin with MODULE *XDPROP and be terminated by the END XDPROP line. In general, the order of directives is not important, with a few exceptions. The ordering given here follows that written into the default `xd.mas` by XDINI. These general instructions apply to all sub options in XDPROP

6.2.1 SIZE

SIZE napl *napl*
In XD2015, many of the large arrays in XDPROP are dynamically allocated, according to the number of atoms actually present in the asymmetric unit. However, it is algorithmically inconvenient to dynamically allocate the arrays which use the parameter *napl* (the maximum number of symmetry equivalent positions *for any single atom*, generated by the APPLY/GROUP/OMIT instructions). A relatively modest value (*napl*=8) is used for these arrays by default. The **SIZE** directive may be used to increase this default value, if it is too small for the problem under investigation. *If used, the SIZE directive must be the first one in the XDPROP section.*

6.2.2 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 κ parameters are set to unity.

6.2.3 PROPERTY

PROPERTY **prop*

XDPROP will analyse the property which is flagged with an asterisk (*) on the **PROPERTY** card (there should be only one such a card). If no property is flagged, then the default property (rho) is chosen. The following properties are currently available:

rho	Total density (ρ)
gradrho	magnitude of gradient vector of ρ ($\nabla\rho$)
d2rho	Laplacian of ρ ($\nabla^2\rho$)
nucpot	Nuclear Potential ($v(\mathbf{r})$)
core	Core density
valence	Valence density
defden	Deformation density ($\Delta\rho$)
sigrho	Error in ρ
siglap	Error in $\nabla^2\rho$
esp	Electrostatic Potential $V(\mathbf{r})$
ef	Electric field vector
efg	Electric field gradient tensor
nci	Reduced density gradient

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 $\rho - \rho_{IAM}$.

The electrostatic potential $V(\mathbf{r})$ is defined as

$$V(\mathbf{r}) = \sum_j \frac{Z_j}{\left|\mathbf{r} - \mathbf{R}_j\right|} - \int \frac{\rho(\mathbf{r})}{\left|\mathbf{r} - \mathbf{r}'\right|} d\mathbf{r}' \qquad \text{(Eq. 6-2)}$$

where $\mathbf{R}_j$ 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 [33,34]

$$V(\mathbf{r}) = \sum_j \frac{Z_j}{\left|\mathbf{r} - \mathbf{R}_j\right|} + \sum_j \left[\frac{q_j}{r_j} + \frac{\mathbf{p_j} \cdot \mathbf{e_j}}{\left|\mathbf{r_j}\right|^3} + \frac{1}{2} \sum_k \sum_l Q_{jkl} \frac{r_{ji} r_{jk}}{\left|\mathbf{r_j}\right|^5} + \ldots \right] \qquad \text{(Eq. 6-3)}$$

where $q, \mathbf{p}, Q$... represent pseudoatom moments. A number of experimental charge density analyses using this type of multipolar representation of the potential have been reported [35-38]. In previous versions of XDPROP, $V(\mathbf{r})$ was computed using central contributions according to Su and Coppens [6], and peripheral contributions according the ADF method [7-9]. In the current version of XDPROP, the central contributions are now obtained by the method recently described by Volkov *et al* [56]. Thus the potential may be computed at points inside the charge distribution, permitting the calculation of V, the electric field vector **ef,** and the electric field gradient **efg** at the nuclei, and exploration of the topology of $V(\mathbf{r})$ [39].

NOTE: Since the **efg** is not scalar or vector quantitity, this property can only be calculated at a single point (*i.e.* properties in a line or 2D or 3D maps *cannot* be calculated).

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

$$V(\mathbf{r}) = \sum_j Z_j / \left| \mathbf{r} - \mathbf{R}_j \right| \qquad \text{(Eq. 6-4)}$$

sigrho and **siglap** are computed with equation 6.2: *currently only the contributions from population parameters and κ'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 !

nci option calculates the reduced density gradient for displaying Non-Covalent Interactions (nci) as proposed by Johnson *et al* [58]. If the **CUBE** option is selected with this property, then five 3D grid files are computed automatically, the reduced density gradient RDG (in `xd_nci.grd/cube`), the signed electron density (in `xd_rhos.grd/cube`), the kinetic energy density *G*(**r**) according to the Abramov approximation [59] (in `xd_ked.grd/cube`), the potential energy density *V*(**r**) from the virial theorem in local form [46] (in `xd_ped.grd/cube`) and the total energy density *H*(**r**) summed from the previous energies (in `xd_ted.grd/cube`). All these grid files are computed in a.u. The computation is compute-intensive and the time taken will be considerably longer than for 3D grid files of other properties! An isosurface of low RDG value, mapped and colour coded with the signed electron density provides an indication of the existence and nature of non-covalent interactions [58]. Saleh *et al* [60] have proposed similar mappings using the kinetic energy density *G*(**r**), the potential energy density *V*(**r**) and total energy density *H*(**r**). The Abramov approximation for *G*(**r**) is only valid in regions of low electron density, so is a reasonable approximation for this purpose.

Two other properties **elf** and **oep** may also be calculated, but are somewhat deprecated and not included in the standard `xd.mas` file. The electron localisation function **elf** is computed using the approximation of Kirzhnits [40] for the kinetic energy density, as recently applied by Tsirelson [41]. The one-electron-potential function **oep** [43-45] is defined as:

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

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. Moreover, the approximate **elf** has been shown by Tsirelson *et al.* [41] and Jayatilaka *et al.* [42] to introduce undesirable artefacts.

6.2.4 SELECT (*)flag

This first type of SELECT command involves flags, namely **numdx**, **esd**, and **au**, which require no further qualifiers and **verbose**.

(*)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.

(*)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, ρ (or $\Delta\rho$) and $\nabla^2\rho$. In

least squares refinement, the variance in some property A, derived from the N_{Vars} refined parameters $\{P_{ij}\}$, may be estimated from the variance-covariance matrix ε [5]:

$$\sigma^2(A) = S^2 \sum_{i,j}^{N\,\mathrm{vars}} \left(\frac{\partial A}{\partial P_i}\right)\varepsilon_{ij}\left(\frac{\partial A}{\partial P_j}\right) \qquad \text{(Eq. 6-5)}$$

where S is the goodness-of-fit.

(*)au
If selected, properties are output in atomic units. This DOES NOT affect the input parameters in xd.mas and xd.inp which are generally in Angstrom units. **Note :** not all program options accept this directive.

verbose *verbose_level*
Setting *verbose_level* greater than one switches on extra printout to the screen and to the log files. The old flags ***check** and ***debug** will also still work.

(*)local
This keyword is only mentioned because it was used in old versions of the program. It is no longer present in the default `xd.mas` file written by the current version of XDINI, and is ignored in the current version of XDPROP. The program action is always to use a set of multipole populations transformed into a *global coordinate system* for property computation. It is no longer possible to run calculations using the local coordinate systems.

6.3 Instructions to Define the Cluster of Atoms Used

6.3.1 APPLY

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

6.3.2 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.

6.3.3 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 substracted from the total density in creating the deformation density. The normal option (the default) is to substract 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)
```

6.3.4 OMIT

This instruction allows the user to exclude atoms from further calculations. Either an explicit or implicit list of atom serial numbers may be given. For example, the three **OMIT** instructions shown below all have the same result of excluding atoms 3 to 8.

```
OMIT  3 4 5 6 7 8
OMIT  3 -8
OMIT  3 -5 6 -8
```

6.3.5 CRYSTAL

CRYSTAL alim *xmin xmax* **blim** *ymin ymax* **clim** *zmin zmax*
This directive adds in all atoms contained in the volume defined by the unit cell vectors **alim**, **blim** and **clim**. *xmin xmax* are the 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 $-1. < \boldsymbol{x} < 2.$ will be generated. *ymin ymax zmin zmax* are similarly defined along the unit cell axes ***b*** and ***c*** respectively and have the same defaults.

6.4 Parameters for Topological Analysis

6.4.1 SELECT

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 sub-options correspond to system variables, and the numerical value of the variable must follow the keyword.

cpcut *dxcut* **[1.0e-6]**
This is used in CP searching. A CP is considered to have been located at some point $\mathbf{r}_c$ if $|\nabla f(\mathbf{r}_c)| < dxcut$. The default value is 1.0×10^{-6} Å, which is generally fine for ρ, but may be quite inappropriate for searching other functions.

lmax *lmax* **[4]**
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_{max} on individual pseudoatoms using these master file instructions, it could of course be achieved by manually editing the `xd.res` file.

nstep *nstep* **[20]**

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.

rcut *rcut* **[4.0]**

Local properties $f(\mathbf{r})$ are computed as a sum over pseudoatom contributions: $f(\mathbf{r})=\Sigma_j f_j(\mathbf{r})$. For many properties, notably ρ, $\nabla^2\rho$ and $\Delta\rho$, it is an excellent approximation to ignore contributions of pseudoatoms which lie further than a distance rcut 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 $V(\mathbf{r})$ or $v(\mathbf{r})$ (iii) when some pseudoatoms have rather small values of κ and/or κ'_l. *Users are therefore urged to check that the property being computed is converged with respect to this parameter.*

When the properties esp, ef or efg are computed, the parameter **rcut** has a rather different action. At the distance specified by **rcut**, the calculation switches from the exact to an approximate formula, and for this purpose, the value of **rcut** should be increased. For lighter elements a value of 6.0 Å has given good results, and an even larger value may be appropriate for heavier elements such as the transition metals.

scale *scale* **[0.05]**

CP searching involves consecutive steps δs 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 *scale*, with a default of 0.05 Å. This value works well for searching ρ, but other (most likely smaller) values should be considered in searching more rapidly-changing functions such as $\nabla^2\rho$

dx *xstep* **[0.001]**

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

ds *pstep* **[0.005]**

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

The following options apply to the calculation of the electrostatic potential (**esp**) only:

rEcrit *rEcrit* **[0.00001]**

rEcrit is the critical distance from the nucleus at which the calculation of the electronic potential switches from the "central" method of Volkov *et al* [56] to the "peripheral" ADF method [7-9].

rNcrit *rNcrit* **[0.00001]**
rNcrit is the critical distance to a nucleus, below which the nuclear potential is omitted entirely.

6.4.2 QUADINT

QUADINT iqt *iqt* **Nrad** *Nrad* **Nang** *Nang* **(*)Becke (*)Stock**

This instruction controls the quadrature integration procedures for the stockholder moments (**STOCKMOM**), the interaction energies (**INTEREN**), the lattice energy (**LATEN**) and the atomic basin integrations (**TOPINT**) calculations.

iqt *iqt* **[2]**

iqt defines the type of radial quadrature used. If *iqt* = 1, the Gauss-Legendre quadrature with Becke's mapping [10] is used. If *iqt* = 2, the Gauss-Chebyshev quadrature [11] with the Treutler-Ahlrichs M4 mapping [12] is used. If *iqt* = 3, the Euler-MacLaurin quadrature [13] with the Mura-Knowles mapping [14] is used.

Nrad *Nrad* **[50]**

Nrad is the number of radial points used.

Nang *Nang* **[194]**

Nang is the number of angular points used. The Lebedev-Laikov quadrature [15] is used for the angular integration. Almost all of the Lebedev quadratures are available, allowed values of *Nang* range from 6 to 5810. The user may enter any value for *Nang* - the program automatically assigns the closest allowed Lebedev quadrature to the input value.

Weighting functions of the Becke form [10] (***Becke**) or the stockholder form [16] (***Stock**) can be used to partition the 3-D space. These can be both switched off, for example for pseudoatom-pseudoatom integration.

6.4.3 ODESOLVE

ODESOLVE *rk (*)bs eps *eps* **stepi** *stepi*

This instruction controls parameters for the ODE (Ordinary Differential Equation) solver used in the integration procedures for the **TRAJPLT** (Section 6.6.6) and **TOPINT** (Section 6.8) options. Two algorithms are available, the fifth-order Runge-Kutta (***rk**) and the Bulirsch-Stoer (***bs**). Both methods use monitoring of the local truncation error to ensure accuracy and automatic adjustment of the stepsize.

eps *eps* **[1.0d-6]**
The required accuracy of integration

stepi *stepi* **[0.01]**
The initial stepsize in Angstroms

6.4.4 ORIGIN

ORIGIN *mass (*)charge (*)geom (*)ucell

The instruction defines the origin used in the calculation of molecular moments. This will affect ALL instructions which calculate molecular moments, *i.e.* **LATEN**, **INTEREN**, **STOCKMOM, MULTMOM** and **TOPINT**. One of four frames may be specified : **mass** uses the center of mass as the origin, **geom** the geometric center, **ucell** the origin of the unit cell and **charge** the center of charge distribution (*i.e Z*). The origin so defined remains effective until another **ORIGIN** instruction is given.

6.4.5 EXPORT

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

This instruction, in conjuction with the directives **MULTMOM, STOCKMOM** or **TOPINT**, causes XDPROP to write out atomic moments in spherical tensor notation [21] in the input formats for the programs ORIENT [22] (when **orient** is flagged) and/or MIN16 [23] (when **min16** is flagged). *lmax* defines the maximum level of atomic moments to be written. *nmol* 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. The **EXPORT** instruction MUST precede any **MULTMOM, STOCKMOM** or **TOPINT** directives.

6.5 Calculation of Properties

6.5.1 MULTMOM

The traceless Cartesian molecular moments up to the hexadecapole level are calculated from the multipole populations. This option replaces the previous **DIPOLE**, **QUADPOLE** and **MOLMOM** directives, which are now aliases for **MULTMOM**. The molecular moments can be calculated in one of four frames dependent on the **ORIGIN** directive (see Section 6.4.4).

The user is referred to the excellent review by Spackman on computing molecular multipole moments from X-ray diffraction data [17]. The computation of the molecular dipole moment requires some definition of a "molecule" in the crystal [18-20] 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

$$\mathbf{p} = \sum_{j=1}^{n} \mathbf{p}_i \qquad \text{(Eq. 6-6)}$$

where

$$\mathbf{p}_j = Z_j \mathbf{R}_j - \int \mathbf{r} \rho_j (\mathbf{r}_j) d\mathbf{r} \qquad \text{(Eq. 6-7)}$$

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

$$\mathbf{p}_j = Z_j\mathbf{R}_j - \int \rho_j(\mathbf{r}_j)(\mathbf{r}_j + \mathbf{R}_j)d\mathbf{r} = q_j\mathbf{R}_j \qquad \text{(Eq. 6-8)}$$

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

$$R_l(r) = \frac{\zeta_l^{n+3}}{(n+2)!} r^{n_l} e^{-\zeta_l r} \qquad \text{(Eq. 6-9)}$$

then the remaining integrations may be performed analytically, to give

$$\begin{aligned}\mathbf{p}_j &= q_j\mathbf{R}_j - P_{1,+1}\int d_{1,+1}R_l x d\mathbf{r} \qquad \text{(Eq. 6-10)}\\ &\quad - P_{1,-1}\int d_{1,-1}R_1 y d\mathbf{r} - P_{1,0}\int d_{1,0}R_1 z d\mathbf{r}\\ &= q_j\mathbf{R}_j - \frac{4\mathbf{P}'_j(n_l+3)!}{3\kappa''_j\zeta_j(n_l+2)!}\end{aligned}$$

where $\mathbf{P}'_j$ is the vector consisting of the dipole populations for the j^{th} atom. In general, each atom has its own local coordinate system so $\mathbf{P}_j$ must include the effect of a rotation $\mathbf{A}$ to a common Cartesian frame: $\mathbf{P}'_j=\mathbf{A}_j\mathbf{P}_j$.

The dipole moment **p** is independent of the choice of origin (for a neutral moiety), but the dipole moment variance $\sigma(\mathbf{p})$ would normally be origin-dependent. However, the application of the neutrality constraint in XDLSM has an effect equivalent to minimizing the dipole variance with respect to the origin. Therefore molecular or fragment dipole moment variances computed with XDPROP should be origin-independent, *provided that the neutrality constraint was applied appropriately during refinement.*

6.5.2 D-POP

D-POP use *Pv *P00

Occupancies of the *d* orbitals are derived, (with the assumption of low *sp* to *d* overlap) according to the approach suggested by Holladay, Leung and Coppens [32]. The calculated *d* orbital occupancies are given in the output file `xd_pro.out`. Either or both of the populations in Pv and P00 may be used in the calculation. If the ***esd** flag in the **SELECT** heading is set, estimated errors are also tabulated. For example:

```
MULTIPOLE POPULATIONS
   P00
  2.376
   P20   P21+   P21-   P22+   P22-
  0.094  0.000  0.000  0.000  0.000
   P40   P41+   P41-   P42+   P42-   P43+   P43-   P44+   P44-
  0.213  0.000  0.000  0.000  0.000  0.001  0.011  0.000  0.000
ORBITAL POPULATIONS
   z2    =  0.87065
   xz    =  0.32579
   yz    =  0.32579
 x2-y2   =  0.42701
   xy    =  0.42701
 z2/xz   =  0.00000
 z2/yz   =  0.00000
z2/x2-y2 =  0.00000
 z2/xy   =  0.00000
 xz/yz   =  0.00000
```

```
xz/x2-y2 =  0.00126
 xz/xy   =  0.02387
yz/x2-y2 =  0.02387
 yz/xy   = -0.00126
x2-y2/xy =  0.00000
tot d-pop =  2.37625
```

6.5.3 STOCKMOM

Quadrature integration of the electron density, partitioned according to the Stockholder method [16], and calculation of the corresponding unabridged moments is performed. The relevant instructions in the `xd.mas` file are:

QUADINT iqt *iqt* **Nrad** *Nrad* **Nang** *Nang* **(*)Becke (*)Stock**
STOCKMOM atoms (*)all *unique (*)select *ato(1) ato(2) ...*

For the **QUADINT** instruction, see Section 6.4.2

atoms (*)all *unique (*)select *ato(1) ato(2) ...*
If **select** is flagged, only the specified atoms *ato(1) ato(2)* ... will be integrated, if **all** is flagged all atoms in the atom list will be integrated, or if **unique** is flagged then only the atoms in the asymmetric unit will be integrated (default).

6.5.4 INTEREN

Interaction energies between two molecular fragments or atoms are calculated by numerical (quadrature) integration and/or by the Buckingham approximation using either atomic or total molecular multipolar moments, supplemented with various atom-atom potentials for calculation of the exchange-repulsion, dispersion and induction energies. The total interaction energy may be expressed [21] as the sum of electrostatic, exchange-repulsion, dispersion and induction terms.

$$E_{\text{int}} = E_{\text{es}} + E_{\text{ex-rep}} + E_{\text{disp}} + E_{\text{ind}}$$

The electrostatic term E_{es} is usually the major contribution to E_{int}. The exchange-repulsion terms arises from the antisymmetrisation requirements of the Pauli principle, the dispersion term from the induced charge distributions on different fragments, and the induction term from the interaction of the unperturbed charge density on one fragment with the induced charge distribution another (and vice-versa). In XDPROP, the latter two terms are obtained using empirically derived atom-atom potentials. The induction term is not normally required for calculations involving experimental data, since the induction effects of the crystal lattice should already be included in the experimentally determined pseudoatom parameters. While several potentials are available, the user should note that none have been proven reliable for all classes of compounds in all types of molecular environment. In the current version of XDPROP, it is assumed that the atoms comprising the two interacting fragments are entered consecutively in the `xd.mas` file. The following instructions are relevant:

QUADINT iqt *iqt* **Nrad** *Nrad* **Nang** *Nang* **(*)Becke (*)Stock**
ATATPOT *exrep (*)spack *wilcox (*)lj
ATATPOT *dispr (*)spack *wilcox (*)lj
HBONDS Hatom1 Xatom1 Distance1, (Hatom2 Xatom2 Distance2 .....)
HPOLAR Hatom1, Hatom2 ......
INTEREN frag 1 *nat1* **-***nat2* **(*)neutral**
INTEREN frag 2 *nat3* **-***nat4* **(*)neutral**
INTEREN (*)EP (*)aMM (*)mMM (*)EPMM rCrit1 *rCrit1* **rCrit2** *rCrit2*

The **ATATPOT** instructions are required if calculation of the exchange-repulsion, dispersion and induction terms is to be included. The **HBONDS/HPOLAR** instructions might also be required in the presence of H-bonds.

For parameters controlling the quadrature integration **QUADINT**, see Section 6.4.2 The instructions **ATATPOT/HBONDS/HPOLAR** are common to both the **INTEREN** and **LATEN** calculations and are described first.

IMPORTANT : Atom-atom parameters are currently available only for a limited number of atomic types. If the structure contains other elements, only the electrostatic term of the interaction energy can sensibly be computed. Contributions to the exchange-repulsion, dispersion and induction terms from any atomic type with unavailable parameters are set to zero !

The ATATPOT/HBONDS/HPOLAR instructions MUST precede any INTEREN or LATEN instructions in the `xd.mas` file.

ATATPOT *exrep (*)spack *wilcox (*)lj

This instruction specifies the type of atom-atom potentials to be used for the *exchange-repulsion* term of the total energy. The instruction is only executed if the **exrep** parameter is selected (*i.e.* starred).

spack	Spackman's potentials for exchange-repulsion and dispersion [24]. Available for elements H, C-F, Al-Cl, Ga-Br.
wilcox	Williams and Cox potentials for exchange-repulsion and dispersion [25]. Available for elements H, C, N, O
lj	Pairwise Lennard-Jones 12-6 (and 12-10 for H-bonds) potentials [26] Available for elements H, C, N, O

ATATPOT *dispr (*)spack *wilcox (*)lj

This instruction specifies the type of atom-atom potentials to be used for the *dispersion* term of the total energy. The allowed options are the same as for the exchange-repulsion instruction. The instruction is only executed if the **dispr** parameter is selected (*i.e.* starred).

IMPORTANT : Since Spackman's potentials for exchange-repulsion contain a spherical-density/spherical-density term, these potentials SHOULD NOT be used in conjuction with the EPMM method. The latter already includes this term, and using both will result in an overestimation of exchange-repulsion. The Williams and Cox potentials for exchange-repulsion should be used in preference, as the Lennard-Jones potentials are very approximate, and acceptable only for very weak interactions.

Spackman's potentials for exchange-repulsion COULD be used with the **aMM** (atomic multipole moments) method, but this procedure is not very good for short contacts, although it might be acceptable for less tightly bound crystals. For the dispersion term either Spackman's or Williams and Cox potentials may be used with any method (they give generally similar results).

The same considerations also apply to the **LATEN** instruction.

HBONDS Hatom1 Xatom1 Distance1, ...

This instruction specifies which hydrogen atoms are involved in H-bonds, and is used to set special potentials for those atoms. Xatom1 is the acceptor atom for set #1 of hydrogen bonds and Distance1 is the distance from the H atom to the acceptor atom. Up to 200 sets of H-bonds may be specified.

HPOLAR Hatom1, Hatom2 ......

This instruction specifies which hydrogen atoms involved in H-bonds are to be considered as polar. Different potentials are used for such atoms.

INTEREN frag *n n1 –n2* *neutral

This instruction defines the atomic fragment as a consecutive run between atoms *n1* to *n2*. The fragment number *n* may be either 1 or 2 - the program will halt if more than two fragments are defined. If the **neutral** directive is selected, the fragment will be made electroneutral.

INTEREN (*)EP (*)aMM (*)mMM *EPMM rCrit1 *rCrit1* **rCrit2** *rCrit2*

EP/aMM/mMM/EPMM control the type of calculation used to compute the potential. Only **one** method should be selected. The EPMM hybrid method is a combination of the three methods and is the preferred approach. It is described in references [27,28], and has been shown [29] to provide accurate electrostatic interaction energies.

***EP**	The Exact Potential [30] is computed. **NOTE :** very compute intensive !!
***aMM**	The Buckingham approximation [31] using **atomic** moments is used.
***mMM**	The Buckingham approximation [32] using **molecular** moments is used.
***EPMM**	The EP/MM hybrid method [27,28] is used.

rCrit1 *rCrit1* **[4.0]**

The critical distance (Angstroms) at which the **EPMM** calculation switches between **EP** and **aMM**

rCrit2 *rCrit2* **[15.0]**

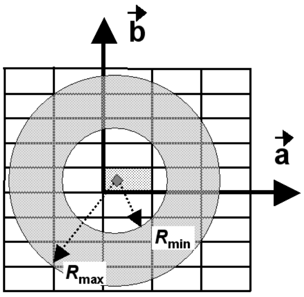
The critical distance at which the **EPMM** calculation switches to **mMM**.

If *verbose_level* (Section 6.2.4) is greater than 1, then additional information is printed to Fortran units 48 and 49.

IMPORTANT: When using EP method, one should always check the accuracy of integration by comparing numerical values for two integrals: integral of rho(A)*elpot(B) and integral of rho(B)*elpot(A) where A and B are the pair of pseudoatoms between which Ees is evaluated. For more information, see [28b], equation 3 and discussion therein. This can be done by setting 'verbose 3'. In this mode, XDPROP will create two files: fort.48 and fort.49 In file fort.49, the values of the two integrals mentioned above are compared. For each pair of pseudoatoms integrated, check the differences between the two integrals listed in the fort.49 file. If it is too large even for one pair, simply increase the accuracy of integration by giving larger values for Nrad and Nang in QUADINT, for example, going from 50x194 to something like 70x194, or 70x302, 100x302 etc, until you get an acceptable agreement between the two integrals for ALL pseudoatom pairs.

6.5.5 LATEN

The electrostatic crystal binding energy is calculated using the same methodology as described above for **INTEREN**. All intermolecular interactions are calculated directly within a spherical shell centred about the reference molecule in the asymmetric unit. The radii of the internal and external "surfaces" of the shell are defined by *Rmin* and *Rmax*.



The following instructions are relevant in the `xd.mas` file:

QUADINT iqt *iqt* **Nrad** *Nrad* **Nang** *Nang* **(*)Becke (*)Stock**
ATATPOT *exrep (*)spack *wilcox (*)lj
ATATPOT *dispr (*)spack *wilcox (*)lj
HBONDS Hatom1 Xatom1 Distance1 (Hatom2 Xatom2 Distance2 .....)
HPOLAR Hatom1, Hatom2 ......
LATEN frag1 *nat1* **-***nat2* ***neutral**
LATEN radii *Rmin Rmax*
LATEN (*)EP *aMM (*)mMM (*)EPMM rCrit1 *rCrit1* **rCrit2** *rCrit2*
For **QUADINT**, see Section 6.4.2. For **ATATPOT/HBONDS/HPOLAR** instructions see Section 6.5.4.

LATEN frag1 *nat1* **-***nat2* ***neutral**
Defines the fragment to be calculated, from *nat1* to *nat2*. If the **neutral** directive is selected, the fragment will be made electroneutral.

LATEN radii *rad1 rad2*
radii *rad1 rad2* **[0.0, 100.0]**
These provide a definition of the shells, in terms of radii in Angstroms.

LATEN (*)EP *aMM (*)mMM (*)EPMM rCrit1 ***rCrit1*** **rCrit2** ***rCrit2***

EP/aMM/mMM/EPMM are directives controlling the type of calculation undertaken (see corresponding **INTEREN** instruction for details). Only **one** should be selected.

rCrit1 *rCrit1* **[4.0]**
The critical distance (Angstroms) at which the **EPMM** calculation switches between **EP** and **aMM**

rCrit2 *rCrit2* **[15.0]**
The critical distance at which the **EPMM** calculation switches to **mMM**.

If *verbose_level* is greater than 1 (Section 6.2.4) , then additional information is printed to Fortran units 48 and 49.

6.5.6 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 [48,49]. A cubic grid of (*npts*)^3 points is centered at the centroid of coordinates of the active molecule. This grid is of size *length* Å. $V(\mathbf{r})$ is computed over this grid, and then a subset of these points in a shell of thickness *width* 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

$$\chi^2 = \sum_i^m \omega_i \left[V_i^o - \sum_j^{n-1} q_j r_{ij}^{-1} + \left(\sum_j^{n-1} q_j - Z \right) r_{in}^{-1} \right]^2 \qquad \text{(Eq. 6-11)}$$

(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 w_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

$$\sigma = \left[\frac{1}{m} \sum_i^m \left(V_i^o - V_i^{\mathrm{model}} \right)^2 \right]^{1/2} \qquad \text{(Eq. 6-12)}$$

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:

```
CONSTRAIN   1 2 3 4 4 5 5 5
```

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 [50]. 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 [51]. This gives charges which have some internal chemical significance for the molecule, rather than optimally predicting its intermolecular interactions.

6.5.7 NUCPROP (*)all (*)select ato1, ato1 -ato3...

This command invokes the calculation of the esp, ef and efg at the nuclear positions. The exact properties calculated depend on the specification of the PROPERTY directive (see **6.2.3**). If the property selected is esp, then only the esp is calculated. If the property selected is ef, then the esp and ef are calculated, while if the property selected is efg, then all three are calculated. By default (*i.e.* if any other property is selected) then all three are calculated.

6.5.8 EPFG EP aMM mMM *EPMM rCrit1 *rCrit1* rCrit2 *rCrit2*

This command controls the calculation of the esp, ef and efg. It is needed because their evaluations via the exact formula of Volkov *et al* [56] are very slow if a lot of atoms are present in the list, *i.e.* if the **LATEN** or **CRYSTAL** options are used.

EP	- the exact formula of Volkov *et al* [56]
aMM	- using atomic multipoles
mMM	- using molecular mutipoles (this approximation is only available for **LATEN** - in the calulation of the esp, ef and efg for plots and properties at the nucleus, only the **aMM** approximation is available)
EPMM	- hybrid method (**EP/aMM/mMM** in **LATEN**, **EP/aMM** everywhere else)
	rCrit1 and **rCrit2** are for **EPMM** only
rCrit1	- interatomic distance for switching from **EP** to **aMM**, default 6 Å
rCrit2	- interatomic distance for switching from **EP/aMM** to **mMM** (LATEN only), default 15 Å

6.6 Function Plots

The following 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 these instructions may be present. However, depending on the computer operating system, the various 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.

6.6.1 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`.

6.6.2 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.

6.6.3 VECTOR PLOTS

If a 2D or 3D *vector map* of the vector property electric field (ef) is required, an extra directive must be given *before* the MAP or CUBE directives. In this case, as well as the standard format XD grid file (`xd_<prop>.grd`) containing the *magnitude* of the vector (which can be visualised using XDGRAPH), an MTVDAT file called `xd_<prop>.dat` is also produced. MTVDAT files may be visualised using the program **PLOTMTV** (for Linux versions see http://rpmfind.net). The program is invoked with the command line :

plotmtv -colorps -plotall xd_<prop>.dat

VECPLOT (*)labels (*)zcut *zcut* **omitrad** *omitrad* **(*)vcut** *vcut* **vscale** *vscale* **xytol** *xytol* **(*)dp** *dipole_scale*

If ***labels** is selected, atomic labels are included in the MTVDAT file

(*)zcut *zcut* **[1.2]**
The range in Å along the z-axis of the plot to include atoms.

omitrad *omitrad* **[0.2]**
Omit grid points within *omitrad* around each nucleus

(*)vcut *vcut* **[0.8]**
Do not plot vectors with magnitude > *vcut.* Units depend on the property.

vscale *vscale* **[1.0]**
Scale factor for vectors in the plot

xytol *xytol* **[0.4]**
Tolerance in the xy-direction of the 2-D plot to include atoms.

(*)dipole *dipole_scale*
Starring this option will plot a projection of the dipole moment vector. The number *dipole_scale* is a scale factor, different from the scaling of vectors *vscale.*

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

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

MAP bvect1 *x1 y1 z1* **bvect2** *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** × **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 limited only by the system memory available.

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

CUBE center *x0 y0 z0* **npts** *npts* **step** *step*

CUBE *atom1 atom2* **npts** *npts* **step** *step*
This generates an *npts* × *npts* × *npts* cube of the active property, with a grid spacing of *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:

```
CUBE CENTER 0.3 9.82 1.5 NPTS 21 STEP 0.3
CUBE C(1) O(3) NPTS 21 STEP 0.3
```

A map file is produced with the filename structure `xd_mid_prop.grd`. The maximum value of *npts* is limited only by the system memory available. A new feature of XD-2015 is that, in addition to the standard format XD grid files, the MAP and CUBE instructions now also generate CUBE files in Gaussian format. These may be used as input for programs like MOLEKEL v4.3 [54], so that one property, such as the electrostatic potential, may be mapped onto an isosurface of another property such as the density, as shown in Figure 6-1.

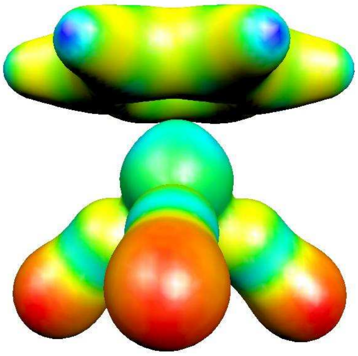


Fig 6-1 Illustrative plot of isosurface mapping. The electrostatic potential is mapped onto an isosurface of the density (0.1 a.u.) in Cr(benzene)(CO)$_3$. Red indicates the smallest (most negative) value of the potential and blue the largest (most positive). This plot was produced using the program MOLEKEL v4.3 [54] from the Gaussian format CUBE files `xd_rho.cube` & `xd_esp.cube` output by XDPROP.

CUBE hirshf *<xd_iam_mol.grd> <xd_iam_cry.grd>*

This is a special option to create a cube file with Hirshfeld weights in XD format, which can then be used to draw properties mapped onto the so called Hirshfeld surface [55]. Two cube files in XD format must be prepared previously, using the *IAM option in XDPROP to create promolecule/procrystal densities. These cube file must have identical origins and dimensions. The first cube file *xd_iam_mol.grd* is a rho grid file of the isolated molecule, and the second cube file *xd_iam_cry.grd* is a rho grid file of the molecule in a crystal, i.e. completely surrounded by other molecules. This is best achieved using the **CRYSTAL** option (6.3.5) to expand the atom list. Cube calculations of this type are very time consuming, so some care should be taken to use the minimum expansion that is adequate. A new cube file *xd_hirsh.grd* is created, which contains the ratio of the molecular:crystal density at all points. The Hirshfeld surface is the isosurface where this value is 0.5. This cube file may then be used as the iso-grid file with programs such as MolIso/MoleCoolQt to map properties in a third cube file (Figure 6-2). This property cube file must again have the same origin and dimensions as the previous cube files.

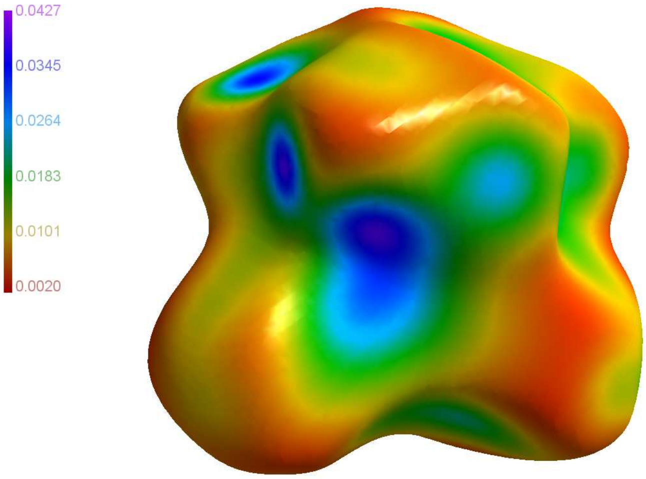


Fig 6-2 Illustrative plot of isosurface mapping. The electrostatic potential is mapped onto the molecular Hirshfeld surface of Fe(TMM)(CO)$_3$, using MoleCoolQt.

6.6.6 TRAJPLT

Plots of the gradient field trajectories of either the total electron density ρ or the electrostatic potential (esp) may be obtained. The plot is written to a PostScript file `<prop>_traj.ps`, which may be visualised with the GhostView program (available on Linux & Windows platforms). In addition, a Computer Graphics Metafile `<prop>_traj.cgm` is produced, which may be manually edited and converted to a PostScript file using the utility program RALCGM, which is available from the official XD website as a Linux RPM or Windows distribution.

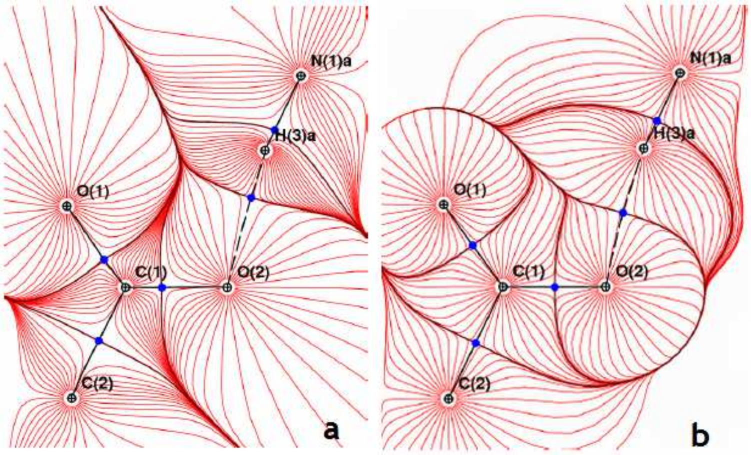


Fig 6-3 Illustrative TRAJPLT gradient-field trajectory plots of (a) the density and (b) the electrostatic potential of an alanine dimer in the plane of the carboxylate group.

The relevant instructions in the `xd.mas` file are :

ODESOLVE *rk (*)bs eps *eps* **stepi** *stepi*
See Section 6.4.3 for more details. The default value of eps may not be suitable for TRAJPLT plots. Smaller values of eps give smoother gradients vector lines, but increase the time taken. Values between 1.0×10^{-6} and 1.0×10^{-8} should give suitable plots for publication purposes. The fifth-order Runge-Kutta algorithm (***rk**) generally gives better results.

TRAJPLT origin atom *ato1*
ato1 is an atom name which defines the origin of the plot.

TRAJPLT origin point *<x,y,z>*
Alternatively the origin may be given with a point , using global Cartesian coordinates (Å).

TRAJPLT xaxis atom *ato2* **xdim1** *xdim1* **xdim2** *xdim2*
ato2 is an atom name which defines the x-axis of the plot, along the direction atom1-atom2. *xdim1* and *xdim2* are the minimum and maximum displacements of the plot from the origin

TRAJPLT yaxis atom *ato3* **ydim1** *ydim1* **ydim2** *ydim2*
The x-y plane of the plot is defined by *ato3*. The y-axis lies in this plane at 90 degrees to the x-axis. Alternatively, both the xaxis and yaxis definitions may also be given using **point** instead of **atom** as for the origin (see above).

TRAJPLT mark (*)atoms (*)labels (*)bonds (*)cps (*)basins (*)hbonds
This directive allows the user to select which of the graphical objects appear in the plot. Most are self evident - the ***hbonds** directive results in long bonds (such as hydrogen bonds) being drawn as dashed lines, otherwise they will be drawn as a solid line, as 'normal' bonds.

TRAJPLT params Circle *rad1* **ATrad** *rad2* **CPrad** *rad3* **CPgrid** *step* **CPlim** *cplim*
This directive allows the user to modify the default values of the drawn radius of the circle around each atom from where the trajectories originate (*rad1*), the drawn radii of the atoms (*rad2*) and critical points (*rad3*). *step* is the grid size in Å for the CP search in the plot plane. (3,-1) critical points are drawn as blue circles, (3,+1) critical points as green circles and (3,-3) critical points as magenta circles. If the function value at a found critical point is lower than *cplim*, then that CP is ignored

TRAJPLT (*)plot (*)plane npath *npath* **(*)zcut** *zcut* **(*)xytol** *xytol* **(*)all select** *ato4*
A plotfile is only produced if ***plot** is given. If the plot plane is not a crystallographic mirror plane (as will normally be the case), then the gradient trajectories will, in general, leave the plot plane at some points. To force the plotted trajectories onto the plot plane, use the ***plane** option. On the other hand, if plotting in a crystallographic mirror plane, then it is probably better not to use this option. *npath* is the number of downhill gradient trajectories - select an integer factor of 360. If ***zcut** is selected, only those atoms within ± *zcut* from the plot plane are included. If ***xytol** is selected, only those atoms within ± *xytol* from the plot plane edges are included. Either all atoms (***all**) are processed or only those selected atoms given **(*select**).

6.6.7 SOURCE

Bader and Gatti [55] have shown that it is possible to view the electron density at any point **r** within a molecule to be determined by a source operating at all other points **r′**. The local source at position vector **r** from **r'**, $LS(\mathbf{r},\mathbf{r}')$, is given by

$$LS(\mathbf{r},\mathbf{r}') = -(4\pi\,|\mathbf{r}-\mathbf{r}'|\,)^{-1}\,\nabla^2\rho(\mathbf{r} \tag{Eq 6-13}$$

The *local source* is calculated at a point, along a line, or in a 2D or 3D grid (for computation of the *integrated source* function see Section 6.8). The only properties for which the source may be calculated are the density rho and the electrostatic potential esp. If another function is selected, the program halts. It is normal practise to use bond critical points as the reference points, since these are minimally biased, but any arbitary point may be chosen.

SOURCE refpoint <x,y,z>
This instruction defines a reference point **r** in global Cartesian coordinates (Å) for the source calculation. Up to 100 reference points may be defined.

SOURCE point <x,y,z>
The local source is calculated at the point **r'** *<x,y,z>*, given in global Cartesian coordinates (Å), for each of the reference points **r.**
SOURCE LINE *atom1, atom2* **npts** *npts*
The local source will be calculated at npts along the line defined by *atom1* and *atom2*, for each of the reference points **r** .

SOURCE MAP atoms *atom1, atom2, atom3* **npts** *npts* **stepsize** *step*
SOURCE CUBE atoms *atom1, atom2, atom3* **npts** *npts* **stepsize** *step*

2D or 3D gridfile of the local source will be produced (`xd_source.grd` & `xd_source.cube`), with the definition of the plane given in the same way as the MAP instruction (see Section 6.1.16). Only one gridfile is produced with the **SOURCE MAP** or **SOURCE CUBE** options. If more than one source **SOURCE refpoint** has been specified, then the source is calculated for the *last* refpoint given.

6.7 Topological Analysis

6.7.1 CPSEARCH

CPSEARCH bond *atom1 atom2* (**frac** *frac*) (**scan** *delta*)
CPSEARCH bond rmin *rmin* **rmax** *rmax*
CPSEARCH ring *atom1 atom2 …*
CPSEARCH point *x y z*
CPSEARCH line vec1 *x1,y1,z1* **vec2** *x2,y2,z2* **npts** *npts*
CPSEARCH shell *atom* **rmin** *rmin* **rmax** *rmax* **nrad** *nrad* **nang** *theta phi* **cutoff** *cutoff*
CPSEARCH start *filename*
CPSEARCH bubble *atom* **rmin** *rmin* **rmax** *rmax* **curv** *icurv* **ncps** *npoints*

Topological analysis of the density is concerned with the scalar fields ρ 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 ρ [46]. 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 × 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 $\mathbf{r}_c$ is found from iterative application of

$$\mathbf{r}_c^{(n+1)} = \mathbf{r}_c^{(n)} - H^{-1} \cdot \nabla f \qquad \text{(Eq. 6-13)}$$

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 six 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.

```
CPSEARCH  BOND C(1) O(1)
```

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

```
CPSEARCH  BOND C(1) O(1)  FRAC  0.7
```

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

```
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 $\{x_o\pm\delta y_o,z_o\}$, $\{x_o,y_o\pm\delta,z_o\}$ and $\{x_o,y_o\pm\delta,z_o\pm\delta\}$, where delta is the step (in Å) which appears after scan.

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

```
CPSEARCH  BOND  RMIN  1.15  RMAX  1.6
```

(ii) **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:

```
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) **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:

```
CPSEARCH  POINT  1.2  -4.3  9.3
```

(iv) **line** starts a search along the line from the point defined by **vec1** to point defined by **vec2**. Both points are defined in the global Cartesian system. A total of *npts* points along this line will be used as starting points for the CP search, including the two defining points. So the form of the command is:

```
CPSEARCH  LINE  VEC1 1.2 -4.3 9.3  VEC2 2.2 -5.3 10.3 NPTS 15
```

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

```
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 *nrad*=3), over an 11 × 11 angular grid of points in θ and ϕ (polar coordinates, referred to the global cartesian system). The minimum number of points in both θ and ϕ (**nang**) is two, and the minimum number of radial shells (*nrad*) is one. The algorithm operates in the following manner. Denoting the property being searched by $f(\mathbf{r})$, $|\nabla f(\mathbf{r})|$ 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 $|\nabla f(\mathbf{r})| < 20.0$, 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 *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.

(vi) **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 ρ. An example of the command would be:

```
CPSEARCH  START  XD_FORMAMIDE.CPS
```

where the last string is a filename.

(vii) **bubble** allows a complete search of critical points around a given atom, particularly useful when searching critical points of the Laplacian. **NOTE :** unlike all other CPSEARCH options, **bubble** searches the function $L(\mathbf{r}) \equiv -\nabla^2\rho$ when **property** is set to d2rho (*i.e.* critical points will have inverted signature compared with all other search options):

```
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 [47].

Note: in this version of XDPROP, only the unique set of critical points is stored (previous versions allowed multiple finds).

6.7.2 BPATH (Bond Path analysis)

BPATH *atom1 atom2* **algrithm (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)  algrithm 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 [53] ($\rho(r)$; $\nabla^2\rho(r)$ and ε) is performed and reported in the file xd_profile.dat.

```
BPATH ato(1) ato(2) algrithm 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
     .......      ......      ......          ......
```

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 more points along the profile are required, then decrease the value of the *pstep* parameter from its default value of 0.005 (see **SELECT ds** above).

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.).

6.7.3 MOLGR (Automatic Molecular Graph Analysis)

MOLGR *auto level *nlevel* **rmin** *rmin* **rmax** *rmax* **algrithm (2|6) (profile)**

This directive provides for an automatic search for the full molecular graph. It uses the CPSEARCH and BPATH routines described above, though their output has been simplified when they are run under the **MOLGR** directive.
Note that this option uses the atoms in the current atom list (including those added/removed through **APPLY/GROUP/OMIT/CRYSTAL** commands). So, if a molecule sits on a symmetry element, it will be necessary to explicitly include the required **APPLY/etc** commands.

If **auto** is flagged (*i.e.* starred, the default)**, MOLGR** instigates a fully automatic procedure to determine the atomic graph, otherwise the procedure relies on immediately previous **CPSEARCH** instructions (*e.g.* **CPSEARCH start**) which have located all the critical points.

The **level** setting *nlevel* determines how comprehensive the automatic critical point search is. There are currently three allowed *nlevel* settings 1, 2 or 3. For *nlevel* = 1, the automatic procedure firstly uses the mid-points of all pairs of atoms which fulfill the **rmin/rmax** conditions as starting points for the Newton-Raphson search. This will generally find all the bond critical points, and if **rmax** is set sufficiently high, a number of ring critical points may also be obtained. If the resultant set of critical points satisfies the Poincaré-Hopf relationship, *i.e.* nat-nbcp+nrcp-nccp = 1, the procedure then immediately moves onto the tracing of the bond paths. For this, the **algrithm (2|6) (profile)** directives are used exactly as for the **BPATH** command given above. If on the other hand the Poincaré-Hopf relationship is NOT satisfied, then a second cp search is instigated, which uses all triads of atoms satisfying the **rmin/rmax** conditions to find any missing ring and/or cage critical points. As soon as the Poincaré-Hopf relationship is satisfied, this procedure exits and moves onto the bond path tracing.

For *nlevel* = 2, the procedure is similar, except that (i) the **rmin/rmax** conditions are not used, *.i.e.* ALL pairs/triads are used for the Newton-Raphson search, and (ii) the Poincaré-Hopf relationship is checked but never used to exit the routine, *.i.e.* both sets of cp searches are carried to completion. This is naturally a more time-consuming process, but is usually worthwhile, in order to be certain that the full graph has been obtained. It is important to be aware that the Poincaré-Hopf relationship is a *necessary* but not *sufficient* condition for the completeness of the molecular graph, so the fact the relationship is satisfied is no guarantee that the molecular graph is actually complete.

The *nlevel* = 3 setting is experimental, and uses a comprehensive grid search for critical points. It is very compute-intensive, untested, and is only intended for desperate situations !

6.8 Atomic Basin Integration

Atomic properties are integrated in the atomic basins. Most of the properties integrated by **TOPXD** (see Section 10.9) are also provided with **TOPINT**, and in addition, the integrated source function may be computed. **TOPINT** works either in the *crystal* environment, or in the *isolated molecule (gas-phase)* environment (similar to many quantum calculations), depending on whether the **CRYSTAL** directive is active. This is in contrast to the **TOPXD** program (Section 10), which designed to work fairly exclusively in the *crystal* environment.

The relevant instructions in the `xd.mas` file are :

EXPORT ....
ORIGIN *mass (*)charge (*)geom (*)ucell
QUADINT iqt 2 Nrad *Nrad* **Nang** *Nang* **(*)Becke (*)Stock**
ODESOLVE (*)rk (*)bs eps *eps* stepi *stepi*
CRYSTAL alim *xmin xmax* **blim** *ymin ymax* **clim** *zmin zmax*
SOURCE refpoint <x,y,z>

TOPINT step *step* **accur** *accur* **rmax** *rmax*
TOPINT spheres *ato(1) r1* ***ato(2) r2 ....***
TOPINT atoms (*)all *unique (*)select *ato(1) ato(2) ...*

The only parameters used from the **QUADINT** instruction are **Nrad** and **Nang**, the number of radial and angular quadrature points. The value of **Nrad** can be set large, as it does not greatly affect the CPU time. Larger values of **Nang** should result in better integration, but are more time consuming. Gauss-Legendre quadrature is used for the radial integration, and Lebedev quadrature [15] for the angular integration. Any value may be set for **Nang**, the program automatically assigns the closest allowed Lebedev quadrature. Allowed values of **Nang** range from 6 - 5810.

For the parameters associated with the **ODESOLVE** directive, see section 6.4.3. If the Runge-Kutta (***rk**) algorithm gives step underflow errors, then switch to the Bulirsch-Stoer (***bs**) method.

If the **CRYSTAL** directive is absent, the integration is carried out over the cluster of atoms as defined normally in XDPROP (see Section 6.3).

SOURCE refpoint <x,y,z>
Reference point **r** for source calculation, <x,y,z> in global Cartesian coordinates

The presence of **SOURCE refpoint** directives activates the calculation of the integrated form of the source function for each atomic basin specified in the **TOPINT atoms** directive described below, at each of the given reference point(s) **r**. Up to 100 reference points may be specified. The reference points would normally be bond critical points (as the most unbiased reference points) but any point may be specified.

IMPORTANT: The only properties for which the source may be calculated are the density rho and the electrostatic potential esp. If another function is selected, the program ignores any **SOURCE refpoint** instructions. All the **SOURCE refpoint** instruction(s) MUST precede any **TOPINT** instructions in the `xd.mas` file.

The local source (*LS*) contribution at position vector **r** from **r'** is given above by Eq 6-13. By integrating over the regions of space bounded by the zero-flux surfaces, the density may be equated to a sum of atomic contributions $S(\mathbf{r},\Omega)$.

$$\rho(\mathbf{r}) = \sum_{\Omega} \int_{\Omega} LS(\mathbf{r},\mathbf{r}')d\mathbf{r}' \equiv \sum_{\Omega} S(\mathbf{r},\Omega) \qquad \text{(Eq 6-14)}$$

The integrated form of the source function (SF) $S(\mathbf{r},\Omega)$ provides a measure of the relative importance each atom has in determining the density at a specific point [55]. This property has great power in illustrating the non-local properties of the electron density.

Note that Eq 6-14 is only useful for decomposing ρ(**r**) into contributions from the integrated source function of atomic basins in the case of a *closed system* where all zero flux surfaces terminate at infinity (*e.g* in an isolated molecule). For a molecule in a crystal, the density at point **r** within atomic basin Ω, is alternatively given [55] by Eq (6-15)

$$\rho(\mathbf{r}) = -(1/4\pi)\left\{ \int_{\Omega} \frac{\nabla^2\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' + \oint_{S\Omega} dS(\mathbf{r}_s)\cdot\varepsilon(\mathbf{r}-\mathbf{r}_s) \right\} \qquad \text{(Eq 6-15)}$$

The first integration term represents the basin average of the potential at **r** exerted by the Laplacian of the density, while the second integration term represents the flux through the surface boundary of Ω of the electric field density, **ε(r - r$_s$)** at **r** due to the density, and corresponds to the source contributions from the basins of all other atoms.

IMPORTANT: The current version of **TOPINT** does not calculate this surface integral, so the source function for a *crystal* environment cannot be properly calculated.

6.8.1 TOPINT Instructions

TOPINT step *step* **accur** *accur* **rmax** *rmax* **nmax** *nmax*

step *step* **[0.01]**
The step distance along the integration ray, in Angstroms.

accur *accur* **[0.001]**
The required accuracy in the final surface determination in Angstroms. Surface determination is done via the bisection method. In order to use step-by-step method, set **step** = **accur**. In principle, the smaller the values of **step** and **accur** the better, but this becomes much more time consuming. It does not make sense to set **step** smaller than **accur**.

rmax *rmax* **[8.0]**
The maximum allowed length of the integration ray, in Angstroms.

nmax *nmax* **[10]**
The maximum allowed number of atoms in the cluster around the integrated atom. The default value is probably suitable for most organic compounds, but will probably need to be increased for inorganic and organometallic compounds.

TOPINT spheres *ato(1) r1 ato(2) r2* **....**
spheres
The β-sphere radii for individual atoms may be given explicitly. These would normally be slighty smaller than the distance to the closest bond critical point. If no **spheres** instructions are given, then the β-sphere radii for all atoms are calculated automatically, and are printed in the output file `xd_pro.out`. These radii can then be used in subsequent **TOPINT** jobs.

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

Specifies the atoms to be integrated. ***all** indicates that all atoms in list are to be integrated, ***unique** specifies just the asymmetric unit, while ***select** is followed by a list of specific atoms to integrate.

No errors for the integrated properties are obtained, but the accuracy of the integration may be gauged by the value of the integrated Lagrangian, (property L) which should ideally have a value of zero. The smaller the value the better, and it should ideally be below 1.0×10^{-4} au. Increasing the value of **Nang** in the **QUADINT** instruction will generally lead to a better integration, at the expense of increased CPU time, but the response is by no means linear and increasing **Nang** may even cause the Lagrangian to increase ! see Fig 6-3.

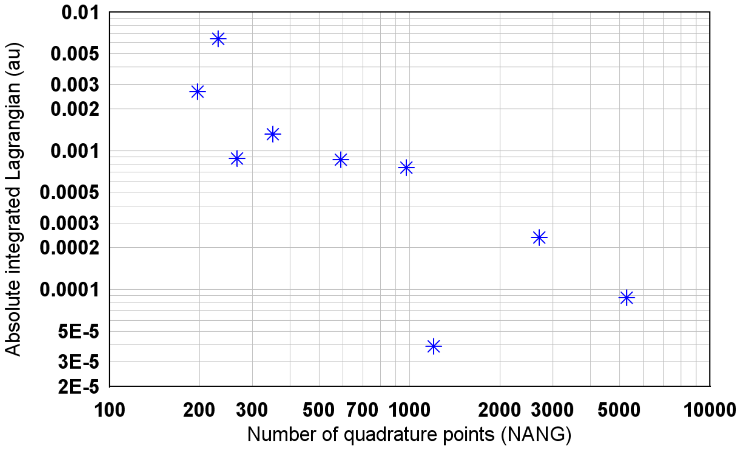


Fig 6-4 Typical plot of the integrated Lagrangian function versus number of angular quadrature points (Nang) for integration of the oxygen atom in water dimer (log-log scaling) using TOPINT.

Bibliography

1. R.F. Stewart and M.A. Spackman, *VALRAY users manual*, Department of Chemistry, Carnegie-Mellon University, Pittsburgh, PA, U.S.A.
2. N. K. Hansen and P. Coppens, *Acta Cryst.* **A34**, 909-921 (1978).
3. (a) P. Coppens *X-ray charge densities and chemical bonding*, Oxford University Press, 1997. (b) V. G. Tsirelson and R. P. Ozerov *Electron density and bonding in crystals*, Institute of Physics Publishing, Bristol, 1996. (c) T. Koritsanszky and P. Coppens, *Chem. Rev.* **101**, 1583 (2001). (d) P. Macchi and A. Sironi, *Coord Chem. Rev.* **238-239**, 383 (2003). (e) C. Gatti, *Z Kristallogr.* **220**, 399 (2005).
4. E. Prince, in *Mathematical Techniques In Crystallography And Materials Science*, Springer-Verlag: New York, 1982.
5. D.W.J. Cruickshank and A.P. Robertson, *Acta Cryst.* **6**, 698 (1953).
6. Z.W. Su and P. Coppens, *Acta Cryst.* **A48**, 188 (1992).
7. G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, T. Ziegler, *J. Comp. Chem.* **22**, 931 (2001).
8. BAND - A Fortran program for Band Structure Calculations. B te Velde (1990)
9. Modern Density Functional Theory, N. C. Handy .
10. A. D. Becke, *J. Chem. Phys.* **88**, 1053 (1988).
11. J. M. Perez-Jorda, A. San-Fabian, F. Moscardo, *Comp. Phys. Comm.* **70**, 271 (1992).
12. O. Treutler, R. Ahlrichs, *J. Chem. Phys* **102**, 346 (1995).
13. M. Abramowitz, I. E. Stegun "Handbook of Mathematical Functions" (1970).
14. M. E. Mura, P. J. Knowles, *J. Chem. Phys.* **104**, 9848, 1996) ; P. M. W. Gill, S.-H. Chien, *J. Comp. Chem.* **24**, 732 (2003).
15. V. I. Lebedev, D. N. Laikov, *Doklady Math.* **59**, 477 (1999).
16. F. L. Hirshfeld, *Theor. Chim. Acta* **44**, 129 (1977).
17. M.A. Spackman, *Chem. Rev.* **92**, 1769 (1992).
18. P. Coppens and N.K. Hansen, *Israel J. Chem.* **16**, 163 (1977).
19. P. Coppens, G. Moss and N.K. Hansen, in *Computing In Crystallography*, Eds. R. Diamond, S. Ramaseshan, K. Venkatesan, Indian Academy of Sciences, Bangalore, 1980.
20. F.L. Hirshfeld, *J. Mol. Struct.* **130**, 125 (1985).
21. A. J. Stone, The Theory of Intermolecular Forces; International Series of Monographs on Chemistry 32; Oxford University Press; Oxford, 1996.
22. A. J. Stone, Program ORIENT, http://www-stone.ch.cam.ac.uk/programs.html
23. Z. .Kisiel, PROSPE - Programs for ROtational SPEctroscopy, http://info.ifpan.edu.pl/~kisiel/prospe.htm.
24. M. A. Spackman *J. Chem. Phys.* **85**, 6579 (1986).
25. D. E. Williams, S. R. Cox *Acta Cryst* **B40**, 404 (1984).
26. J. E. Lennard-Jones, *Proc. Phys Soc* **43**, 461 (1931).

27. A. Volkov, C. Nygren, M. Messerschmidt, X. Li, P. Coppens, in preparation (2006).
28. (a) A. Volkov, P. Coppens *J. Comp. Chem.* **25**, 921 (2004). (b) A. Volkov, Koritsanszky and P. Coppens, *Chem. Phys. Lett.* **391**, 170 (2004)
29 A. Volkov, X. Li, T. Koritsanszky, P. Coppens *J. Phys. Chem. A* **108**, 4283 (2004).
30. (a) M. A. Spackman *J. Chem. Phys.* **85**, 6579 (1986), (b) *ibid,* 6587 (c) M. A. Spackman *J. Chem. Phys.* **91**, 3179 (1987),
31. A. D. Buckingham *Intermolecular Interactions : from Diatomics to Biopolymers* ed. B. Pullman, pp 1.67, New York, Wiley, Chichester; A. D. Buckingham *Adv. Chem. Phys* **12**, 107 (1967).
32. A. Holladay, P. Leung and P. Coppens, P. *Acta Cryst* **A39**, 377-387 (1883).
33. K.E. Laidig, *J. Phys. Chem.* **97**, 12760 (1993).
34. J.D. Jackson, *Classical Electrodynamics*, J. Wiley & Sons: New York, p98, 1963.
35. G. Moss and D. Feil, *Acta Cryst.* **A37**, 414 (1981).
36. R. Destro, R. Bianchi and G. Morosi, *J. Phys. Chem.* **93**, 4447 (1989).
37. W.T. Klooster, S. Swaminathan, R. Nanni and B.M. Craven, *Acta Cryst.* **B48**, 217 (1992).
38. N. Ghermani, N. Bouhmaida, C. Lecomte, A. Papet and A. Marsura, *J. Phys. Chem.* **98**, 6287 (1994).
39. R.K. Pathak and S.R. Gadre, *J. Chem. Phys.* **93**, 1770 (1990).
40. A. Kirzhnits*, Sov. Phys JETP*, **5**, 65 (1957).
41. V. Tsirelson and A. Stash, *Chem. Phys. Lett.,* **351**, 142 (2002).
42. D. Jayatilaka, D. Grimwood *Acta Cryst.* **A60**, 111 (2004).
43. E. N. Lassettre, *J. Chem. Phys.,* **83**, 1709, (1985).
44. G. Hunter, *Int.. J. Quantum Chem.*, **29**, 197 (1986).
45. M. Kohout, *Int. J. Quantum Chem.,* **83**, 324 (2001).
46. R. F. W. Bader, *Atoms in Molecules: A Quantum Theory*, Oxford University Press, 1990.
47. PLATON, A Multipurpose Crystallographic Tool, Utrecht University, Utrecht, The Netherlands, A. L. Spek, 2003.
48. D.E. Williams and S.R. Cox, *J. Comput. Chem.* **2**, 304 (1981).
49. D.E. Williams, *J. Am. Chem. Soc.* **105**, 4143 (1983).
50. C.M. Breneman and K.B. Wiberg, *J. Comp. Chem.* **11**, 361 (1990).
51. Z.W. Su, *J. Comp. Chem.* **14**, 1036 (1993).
52. R.F.W. Bader and H. Essén, *J. Chem. Phys.* **80**, 1943 (1984).
53. M. Tafipolsky, W. Scherer,, K. Ofele, G. Artus, B. Pedersen, W. Herrmann, G. S. McGrady *J. Am. Chem. Soc.* **124**, 5865 (2002).
54. MOLEKEL is a free graphics program available from http://www.cscs.ch/molekel/
55. (a) R. W. F. Bader and C Gatti *Chem. Phys Lett* **287**, 233 (1998). C. Gatti, F. Cargnoni, L. Bertini *J. Comput Chem* **24**, 422 (2003)
56. A. Volkov, H. F. King, P. Coppens and L. J. Farrugia *Acta Cryst* **A62**, 400-408 (2006).
57. M. A. Spackman and D. Jayatilaka *CrystEngComm* **11**, 19-32 (2009).
58. E. R. Johnson, S. Keinan, P. Mori-Sánchez, J. Contreras-García, A. J. Cohen and W. Yang *J. Am. Chem. Soc.* **132**, 6498 (2010).
59. Yu. A. Abramov *Acta Cryst.* **A53** 264 (1997).
60. G. Saleh, L. Lo Presti, C. Gatti and D. Ceresoli *J. Appl. Cryst.* **46**, 1513 (2013).

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:

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

$F_{sph}(\mathbf{h})$ 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:

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

(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*:

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

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

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

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

where $A(\mathbf{h})+iB(\mathbf{h})=F(\mathbf{h})$. For the *total density map* the Fourier coefficients are

$$A(\mathbf{h}) = \left|F_o(\mathbf{h})\right|\cos\phi_{mul}$$

$$B(\mathbf{h}) = \left|F_o(\mathbf{h})\right|\sin\phi_{mul}$$

For the *experimental deformation map*

$$A(\mathbf{h}) = \left|F_o(\mathbf{h})\right|\cos\phi_{mul} - \left|F_{sph}(\mathbf{h})\right|\cos\phi_{sph}$$

$$B(\mathbf{h}) = \left|F_o(\mathbf{h})\right|\sin\phi_{mul} - \left|F_{sph}(\mathbf{h})\right|\sin\phi_{sph}$$

For the *dynamic model map*

$$A(\mathbf{h}) = \left|F_{mul}(\mathbf{h})\right|\cos\phi_{mul} - \left|F_{sph}(\mathbf{h})\right|\cos\phi_{sph}$$

$$B(\mathbf{h}) = \left|F_{mul}(\mathbf{h})\right|\sin\phi_{mul} - \left|F_{sph}(\mathbf{h})\right|\sin\phi_{sph}$$

For the *residual map*

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

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. The value of F(000) is taken from `xd.fou`, and is included in the Fourier summations, ensuring the electron density is on a correct scale.

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 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 (see **FOUR** instruction for XDLSM.) 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-$F_{multipole}$. In this case the F_c set labelled fmod1 has been defined in the least squares program as $F_{multipole}$.

```
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_o 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.
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** preceeded 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 **LIMI**ts, **TRAN**slate and **PERM**ute 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 sytem, 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*

α, β, γ are the Eulerian rotation angles: first a rotation of α degrees about the *z*-axis, then β degrees about the new *y*-axis, and finally γ 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.

```
ROTATE EULERIAN  45 -54.5 90
ROTATE   Z  -90
ROTA     Y  54.5
ROTATE   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:

```
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 × 41 points. There are no arbitary limits to the grid size which may be calculated - the only limits are those imposed by the system memory. However please note that the graphical programs may not be able to display extremely large grid files.

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 ***NlogN*** 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. The value of F(000) is taken from `xd.fou`, and is included in the Fourier summations, ensuring the electron density is on a correct scale.

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 and treatment 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 $F/\sigma(F)$ > *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 *grdsize* **scale** *scale* **npeak** *peaks* **nhole** *holes* **(*)neutron (*)gridf (*)peakf**

gridsize *grdsize*

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`. The peak searching algorithm uses a 19-point interpolation, giving a more accurate location of maxima and minima.

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. All atoms in the unit cell are included in the header, though not necessarily as complete molecules.

peakf

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

Bibliography

1. L. F. Ten Eyck. *Acta Cryst.* **A29**, 183-191 (1973).
2. P. R. Mallinson, K. W. Muir, *J. Appl. Cryst.* **18**, 51-53 (1985).

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.

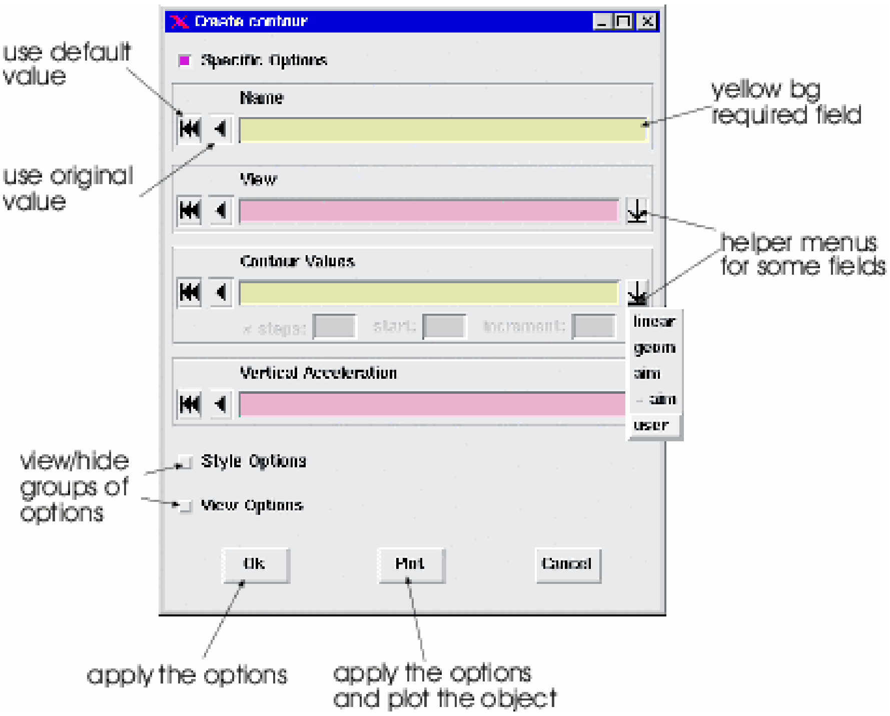


Figure 9-1 : A sample dialog box from the graphical user interface

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*
-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 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. **NOTE** *This option is only available with the OpenGL driver, i.e NOT with the Windows version of XD2015.*

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.

-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. **NOTE** *This option is only available with the OpenGL driver, i.e NOT with the Windows version of XD2015.*

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** arc 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 *cutoffValwe* | *{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 *{x y z}*
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*
clear*
info*
plot*
remove*

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 **xydiag** and **relief**). The way molecules are represented depends on the type of view. E.g. in a **contour** view it is a simple line drawing whereas in a **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 **-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:
contour
height
path
relief
surface
xydiag

SYNOPSIS
view *name* **-type** *type ?options?*

OPTIONS

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

SUBCOMMANDS
configure* *options*
Change options for an existing view. See previous section for a list of possible options.
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 **ddr**
objects
Returns a list of objects connected to this view.
rotate *options*
Rotate the view. **Not yet implemented!!!**
-x *angle*
-y *angle*
-z *angle*
translate {x y z}
Translate the view. **Not yet implemented!!!**

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 Angstrom to cm, a value of one meaning that 1cm in the plot corresponds to 1Å 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 **-vrange** 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 *pi p2 p3*
An easy way to give origin and orientation. $\vec{p}_1$ 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_x p_y p_y}
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
{**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 n'th 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 PPM file (Portable Pixel Map 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*
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 .02 .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, because it requires the (Unix specific) GLX library.*

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
```

9.10.2 colramp.tcl

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.

```
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 the word is terminated by the matching close brace ("}"). Braces nest within the word: for each additional open brace there must be an additional close brace (however, if an open brace or close brace within the word is quoted with a backslash then it is not counted in locating the matching close brace). No substitutions are performed on the characters between the braces except for backslash-newline substitutions described below, nor do semi-colons, newlines, close brackets, or white space receive any special interpretation. The word will consist of exactly the characters between the outer braces, not including the braces themselves.

- If a word contains an open bracket ("[") 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.

\a	Audible alert (bell) (0x7).
\b	Backspace (0x8).
\f	Form feed (0xc).
\n	Newline (0xa).
\r	Carriage-return (0xd).
\t	Tab (0x9).
\v	Vertical tab (0xb).
\<newline>*whiteSpace*	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.
****	Backslash ("\").
ooo	The digits *ooo* (one, two, or three of them)]give the octal value of the character.
x*hh*	The hexadecimal digits *hh* give the hexadecimal value of the character. Any number of digits may be present.

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.
[chars]	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 *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 runtime libraries TCL83.DLL and TK83.DLL, as well as all the necessary Tcl/tk system files are supplied in the release, so there are no external dependencies. Unfortunately, the ***OpenGL*** driver is *not available* for this version of XDGRAPH, so several functions available in the Linux/Unix versions (like iso-surface plots) do not work in the *Windows™* version. An alternative program for viewing isosurfaces is *MolIso*, written by Christian B. Hübschle (Freie Universität Berlin), and freely available for academic users. See Section 13.3 for download details.

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 ρ 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 and 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]:

$$f'_x = \frac{-f_{x+2h} + 8f_{x+h} - 8f_{x-h} + f_{x-2h}}{12h} ,$$

$$f''_x = \frac{-f_{x+2h} + 16f_{x+h} - 30f_x + 16f_{x-h} + f_{x-2h}}{12h^2} ,$$

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:

$$\frac{d^3\rho}{dx^2dy} = \frac{d}{dx}\left[\frac{d^2\rho}{dxdy}\right] = \frac{d}{dy}\left[\frac{d^2\rho}{dx^2}\right] = \frac{d^2}{dx^2}\left[\frac{d\rho}{dy}\right] ,$$

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 1×10^{-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 1×10^{-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 $d^2\rho/dx^2$, 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:

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

Acronym	**Description**
QTAM	Quantum Theory of Atoms in Molecules (R.F.W. Bader)
$\rho(\mathbf{r})$	Electron density
$\nabla^2\rho(\mathbf{r})$	Laplacian ($\nabla\bullet\nabla$) of $\rho(\mathbf{r})$
$\boldsymbol{H}$ ($\rho(\mathbf{r})$)	The Hessian (second derivatives) matrix of $\rho(\mathbf{r})$
λ_k	Eigenvalues ($\lambda_1\leq\lambda_2\leq\lambda_3$) of $\boldsymbol{H}(\rho(\mathbf{r}))$
CP(s)	Critical Point(s) : a point **r** where a given scalar f has $\nabla f(\mathbf{r})=0$
(m,n) CP	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 $\boldsymbol{H}$ ($\rho(\mathbf{r})$) at the CP.
BCP	Bond Critical Point : a (3,-1) CP in the $\rho(\mathbf{r})$ scalar field
ZFS	Zero-Flux-Surface : $\nabla\rho(\mathbf{r})\bullet\mathbf{n}(\mathbf{r}) = 0 \quad \forall\ \mathbf{r} \in$ *surface*
Atomic Basin	The space traversed by all the uphill $\nabla\rho(\mathbf{r})$ 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.
NNA	Non-Nuclear Attractor: a (3,-3) CP of ρ at a position other than nuclei
NEA	Non-Equivalent (unique) Atom
AIL	Atomic Interaction Line : a curved path joining two nuclei along which ρ is a maximum with respect to any lateral displacement
"Crystal Graph"	The network of AIL's for a given crystal geometry. It is the crystal correspondent of the QTAM molecular graph for an isolated molecule
AGL	Atomic Graph Line : a curved path joining two $-\nabla^2\rho(\mathbf{r})$ (3,-3) CPs along which $-\nabla^2\rho(\mathbf{r})$ is a maximum with respect to any lateral displacement
VSCC	Valence Shell Charge Concentration : the *atomic* valence shell region where $\nabla^2\rho(\mathbf{r})$ is negative
IR	Integration Ray : in atomic properties evaluation)
EF	Eigenvector Following method (CP search)

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:

keyword	meaning
TRHO	topological analysis of $\rho(\mathbf{r})$
TLAP	topological analysis of $\nabla^2\rho(\mathbf{r})$
ATBP	atomic basin properties
PL2D	2D plots
P2DCRY	Visualization of 2D plots
PL3D	3D plots
VZ3D	3D visualization of atomic basins

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. *ymin ymax zmin zmax* are similarly defined along the unit cell axes ***b*** and ***c*** respectively and have the same defaults.

10.2.1.4 MPAR

MPAR rcut *rcut* **dstep** *dstep* **(*)au (*)iam (*)esp**

rcut
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.

dstep
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 ($l \geq 1$) are set to zero; monopole populations are set to free atom values and κ parameters are set to unity. Otherwise the multipole model specified in the `xd.mas` file is used.

(*)esp
If starred, a topological analysis of the electrostatic potential will be undertaken, instead of the density.

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 `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 `xd.mas`.

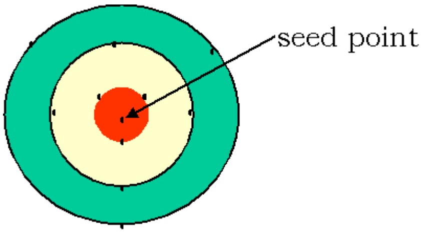
(*)ascii

If starred, the gridfile specified by the *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 *nstep* **nnb** *nnb* **rmax** *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 *nnb*
The value of *nnb* determines the maximum number of "coordination shells" of atoms to be included in the cluster generated around the "seed point". *nnb* also defines the number of neighbours in the nearest neighbour analysis around each unique critical point (of any kind) found. In general, *nnb* should be set to be as large as is required to generate all neighbours around a "seed point"

rmax *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 *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 *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** *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 *th*
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** *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 : $\mathbf{r}'_{\text{start}} = \mathbf{r}_A + pc * (\mathbf{r}_B - \mathbf{r}_A)$; $\mathbf{r}''_{\text{start}} = \mathbf{r}_A + (1. - pc) * (\mathbf{r}_B - \mathbf{r}_A)$,
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** *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... i(n)*
Critical point search 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 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 ρ(r), ∇²ρ(r) and λ₃ along the line joining two nuclei or two general points

Profiles of $\rho(\mathbf{r})$, $\nabla^2\rho(\mathbf{r})$ and λ_3 are written to Fortran units 95, 96, 97, respectively. The units of $\rho(\mathbf{r})$, $\nabla^2\rho(\mathbf{r})$ and λ_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= $0.01\times R_{A\text{-}B}$ (or $R_{a\text{-}b}$)
if n=100, s= $1\times R_{A\text{-}B}$ (or $R_{a\text{-}b}$)

<*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 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 below

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 four-character string, either ***cccp / s1cp / s2cp / cdcp*** for (3,-3), (3,-1), (3,+1) or (3,+3) critical points respectively.

(*)ail
If this keyword is starred, atomic graph line (AGL) lengths and termini are evaluated numerically for each unique (3,-1) critical point. **This is a compute-intensive option !** The AGL is the union of the unique pair of $\nabla(\nabla^2\rho)$ trajectories that originate at the (3,-1) $-\nabla^2\rho$ 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

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 θ and φ, respectively

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)

(*)atoms *label1…label(n)* **nmax** *nmax* **rstar** *rstar*
The critical point search will only be performed if the keyword **atoms** is starred.

label1..label(n)
Labels of unique atoms for which the critical point search will be performed.

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.

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).

NNA specifications given in the following format. Insert one record for each NNA :

(*)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.

x y z - Cartesian coordinates of the current NNA (Å)

nmax , rstar

As above for **(*)atoms**

10.4.2 Critical point search started from a given set of points

TLAP (*)points *meth* **(*)ail (*)debug nstep** *nstep* **nnb** *nnb* **rmax** *rmax* **nmax** *nmax*
(*)car x y z (Several of either of these records may be given)
(*)fra x y z

(*)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 10.4.1. The usage of keywords **(*)ail**, **(*)debug**, **nstep**, **nnb**, **rmax, nmax** is also 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.

(*)car *x y z*
(*)fra *x y z*

These commands must each start on a new line! Starting point coordinates (*x y z*) in either Cartesian (Å) (**car**) or fractional (**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 *meth* **(*)ail (*)debug nstep** *nstep* **nnb** *nnb* **rmax** *rmax* **nmax** *nmax*
<*line specifications*> (Several of these line may be given)

(*)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 10.4.1. It is recommended to use the Newton-Raphson search (**nr**) unless a specific type of critical point is being sought to the exclusion of all others.

The usage of keywords **(*)ail**, **(*)debug**, **nstep**, **nnb**, **rmax, nmax** is exactly as described in Section 10.4.1

<*line specifications*> can be given in one of the following formats:

(*)atom *label* **toneighbor *i1... i(n)***
Critical point search 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 search will only be performed if the keyword **atom** is starred.

(*)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 either 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

10.5 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
  ATBP params ...
- Alternative method for the Zero Flux Surface (ZFS) search
  ATBP altguess .....
- 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 decribed 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 advange 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** *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 :

$$step = step0 \cdot \exp(a|\cos \omega|^{b})$$

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 ρ(**r**) bond critical points which lie on the ∇ρ(**r**) ZFS, enclosing the atomic basin of the unique atom with label *label*. The default of 0.2 Å is safe for almost all atoms, but is not computationally efficient.

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).

ATPB (*)atoms *label* [**izfs** | **zfs**] **nvi** *nvi* **irsur** *irsur* **(*)irsav** **(*)rest** **(*)debug** **phi** *nphi* **th** *nth* **rad** *rad* **accur** *accur* { **nbcp** *nbcp* [**car** | **fra**] }

{**. . . .** *if nbcp > 0 insert nbcp records with BCP x y z coordinates* **. . . . .** }

(*)atoms
If starred, this requests the integration of unique atoms specified with *label* instruction, otherwise no integration will be performed.

label
Specifies which atoms will be integrated if the **atoms** keyword is starred. There are several possible format specifications for *label.*

1. Labels of unique atoms, for example:
   ATBP *atoms O(1) C(2) N(10) H(2A) ....

2. Keyword **all** for all unique atoms to be integrated, for example:
   ATBP *atoms all ....

3. Atomic symbols – all atoms with the same atomic symbols will be integrated, for example :
   ATBP *atoms O H ....

izfs
Indirect determination of the ZFS [12]. This is the **recommended** and more accurate method for determination of ZFSs, but is computationally rather demanding.

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

NOTE: The ZFS method is still experimental and has not been thoroughly tested. It may well fail !

nvi *nvi*
nvi is the number of "coordination shells" of neighbours of the current unique atom(s) which have to be considered as possible attractors of the $\nabla\rho(\mathbf{r})$ paths launched from points along the integration rays.

irsur *irsur*
irsur = 0 – normal run
irsur = 1 – the lengths of the integration rays obtained in a previous run are read from Fortran unit 97 (file `fort.97`) and used as an initial guess for the IR lengths
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 `fort.97`).

(*)irsav
When starred the lengths of the integration rays are saved in Fortran unit 98 (file `fort.98`). **NOTE:** The use of *irsur* ≠ 0 requires that ***irsav*** was set in a previous ATBP run. The ZFS thus saved on Fortran unit 98 (file `fort.98`) may be used in a following run (*irsur*≠0), by copying it on Fortran unit 97 (file `fort.97`). Use *irsur* = -1 to run the integration step separate from the ZFS determination step. Put *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.

(*)rest
When starred, the run is restarted from (partial) surface data stored in Fortran unit 96 (file `fort.96`) from a previous aborted run

(*)debug
Activates the debug printing during the ZFS determination and integration

phi *nphi* **th** *nth*
Angular integration parameters used for OUTSIDE the β-sphere: number of φ (*nphi*) and θ (*nth*) grid points (see also FAQ section)

rad *rad*
rad is the number of radial integration points inside β-sphere

accur *accur*
Parameter *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 *accur* value reduces the computational time at the expense of accuracy (see also FAQ section).

{ **nbcp** *nbcp* [**car** | **atom**] }
These optional parameters should only be used for a two-step procedure in ZFS determination (see keyword **zfs** above). *nbcp* is number of (3,-1) critical points associated with the atomic interaction lines (AIL) [see **TRHO** section] linking the current unique atom to other atoms and/or NNAs. The keyword **car|atom** determines the format for specification of (3,-1) critical points to be read in the following *nbcp* records (if *nbcp* > 0).

1. Keyword **car** specifies that the Cartesian coordinates of a (3,-1) critical point should be given:
{ *x y z* }

2. Keyword **atom** specifies the atom linked to the current unique atom
{ *inum itx ity itz* }

inum is the sequence number of a linked atom in the TOPXD printing of all atoms in the unit cell, while *itx ity* and *itz* specify the indices (direct cell) of the cell where the linked atom *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*
nna is the number of NNAs in the structure. The default is zero, *i.e.* no NNAs.

If *nna > 0* insert *nna* 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* **irsur** *irsur* **(*)irsav (*)rest (*)debug phi** *nphi* **th** *nth* **rad** *rad* **accur** *accur* } { **nbcp** *ncp* [**car** | **atom**] }

{ **. . . .** *if nbcp > 0 insert here nbcp records with BCP x y z coordinates* **. . . . .** }

There should be as many lines with NNA specifications as the number of NNA's 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 $\rho(\mathbf{r})$ bond critical points which lie on the $\nabla\rho(\mathbf{r})$ 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 *xmin* **xmax** *xmax* **xstep** *xstep* **ymin** *ymin* **ymax** *ymax* **ystep** *ystep*
origin shift *ishft* **origin** *xo yo zo* **vmod** *vmod*
misc size *ax* **scale** *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 $\nabla\rho(\mathbf{r})$ 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 *xmin* **xmax** *xmax* **xstep** *xstep* **ymin** *ymin* **ymax** *ymax* **ystep** *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 *ishft* **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 no longer lie in the XY plot plane.
=1 – the origin of the plot is translated to a point lying on the ABC plane (must specify the *xo yo* and *yo* coordinates, see below)
= 2 – the origin of the 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.

= 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)
= 4 – the origin of the plot is at atom/point A
= 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 then 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

Scalar function (or $\nabla\rho(\mathbf{r})$ plot) type	Filename	
Electron density	$\rho(\mathbf{r})$	SURFRHOO//*name*
Laplacian of $\rho(\mathbf{r})$	$\nabla^2\rho(\mathbf{r})$	SURFLAPP//*name*
Negative of the Laplacian $\rho(\mathbf{r})$	$-\nabla^2\rho(\mathbf{r})$	SURFLAPM//*name*
Magnitude of the gradient of $\rho(\mathbf{r})$	$\mid \nabla\rho(\mathbf{r}) \mid$	SURFGRHO//*name*
$\nabla\rho(\mathbf{r})$ trajectories only		TRAJGRAD//*name*
Molecular/crystal graph (and atomic basin boundaries)		MOLGRAPH//*name*
$\nabla\rho(\mathbf{r})$ trajectories and molecular/crystal graph		TRAJMOLG//*name*

10.6.2 2D plot specific instructions

These instructions must follow the general 2D plot instructions.

PL2D (*)*func* **nstar** *nstar* **rmax** *rmax* **(*)test (*)cut** *cutr 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 ($\rho(\mathbf{r})$)
***lapp**	Laplacian of the electron density ($\nabla^2\rho(\mathbf{r})$)
***lapm**	negative Laplacian of the electron density ($-\nabla^2\rho(\mathbf{r})$)
***grho**	magnitude of the gradient of electron density ($\lvert\nabla\rho(\mathbf{r})\rvert$)
***trajgrad**	$\nabla\rho(\mathbf{r})$ trajectories only
***molgraph**	molecular/crystal graph (and atomic basin boundaries)
***trajmolg**	$\nabla\rho(\mathbf{r})$ trajectories and molecular/crystal graph

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.

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 $\lvert\nabla\rho(\mathbf{r})\rvert$ 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 ±*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.

10.6.2.1 *func* = trajgrad

Add the following parameters to the **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-plot coordinate is less than *tol2* (Å).

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-oordinate is less than *tol2* (Å).

If keyword **plane** is starred, the starting point of each ∇ρ(**r**) trajectory segment (which are typically 10^{-4} to 10^{-2} Å long) is forced on the XY plane, while the default is the normal tracing of ∇ρ(**r**) trajectories (with projection regulated by *tol2*).
The number of downhill ∇ρ(**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 ∇ρ(**r**) trajectories. A bond critical point is seen as a (2,-2) attractor whenever its associated ZFS lies in the plot plane. If *nextr* is greater than zero, add *nextr* lines with Cartesian (Å) *x y z* coordinates of attractors of ∇ρ(**r**) trajectories (one set of *x y z* coordinates per line)

10.6.2.2 *func* = molgraph

Add the following parameters to the **PL2D plot molgraph** instruction:

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 *tol1* (Å) and its absolute z-plot coordinate is less than *tol2* (Å). See note in Section 10.6.2.1 regarding "correct" tracing of molecular/crystal graphs.

If the keyword **plane** is starred, the starting point of each ∇ρ(**r**) trajectory segment is forced on the XY plane, while the default is the normal tracing of ∇ρ(**r**) trajectories (with projection regulated by *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 *thr* (Å).
The ∇ρ(**r**) trajectories that originate at bond critical points and have as the initial direction the *K*-th eigenvector of Hessian of ρ(**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 λ_K ($\lambda_1 \leq \lambda_2 \leq \lambda_3$).

10.6.2.3 *func* = trajmolg

Add the following parameters to the **PL2D plot trajmolg** instruction:

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 *tol1* (Å) and its absolute z-plot coordinate is less than *tol2* (Å). See note in Section 10.6.2.1 regarding "correct" tracing of molecular/crystal graphs.

The keywords **(*)plane thr** *thr* **(*)tr1 (*)tr2 (*)tr3** are as described in Section 10.6.2.2

The number of downhill ∇ρ(**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 bond critical point, to be considered as origins of downhill ∇ρ(**r**) trajectories. A bond critical point is seen as a (2,-2) attractor whenever its associated ZFS lies in the plot plane.
If *nextr* is greater than zero, add *nextr* lines with Cartesian (Å) *x y z* coordinates of attractors of ∇ρ(**r**) trajectories (one set of *x y z* coordinates per line)

10.6.3 Output of TOPXD 2D plot data into HPGL & PostScript format

P2DCRY (*)diff *var1 var2 var3*

(*)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** ($\nabla^2\rho$), **lapm** ($-\nabla^2\rho$), **grho** ($|\nabla\rho|$), **trajgrad** ($\nabla\rho(\mathbf{r})$ trajectories), **molgraph** (molecular ghraph), **trajmolg** (molecular graph and $\nabla\rho(\mathbf{r})$ trajectories), **rhoomolg** (ρ and molecular graph), **lappmolg** ($\nabla^2\rho$ and molecular graph), **lapmmolg** ($-\nabla^2\rho$ and molecular graph).

10.6.4 Visualisation of 2D HPGL files 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:

http://www.gnu.org/software/hp2xx/hp2xx.html

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 shown below:

Pen number	Corresponding object in the drawing
1	contours (and $\nabla\rho(\mathbf{r})$ trajectories)
2	dash lines
3	bond critical point position
4	bond path
5	nuclear positions
6	plot info and border
7	atomic basin boundaries
8	not used

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**)

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**

The image was created with the following **hp2xx** instruction:

```
hp2xx PL2Dform  -d 150  -p 218282  -c 276411  -f form.pcx  -m pcx  -r 270
```

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.

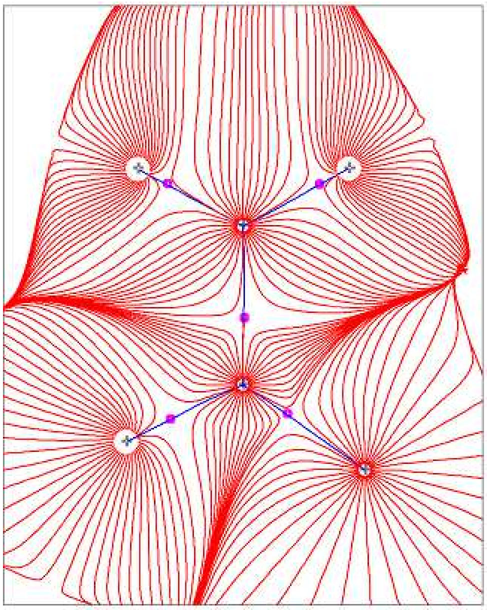


Figure 10-1 ρ(r) trajectories and molecular graph of formamide molecule in the crystal created from TOPXD/P2DCRY2000 data with hp2xx. The ∇ρ(r) trajectories are shown with red lines, nuclear positions are marked with + (black), BCPs are marked with ○ (magenta), bond paths are shown with blue lines.

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 the plot is defined in a 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**]
xmin *xmin* **xmax** *xmax* **xstep** *xstep*
ymin *ymin* **ymax** *ymax* **ystep** *ystep*
zmin *zmin* **zmax** *zmax* **zstep** *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.

xmin xmax xstep
Minimum and maximum values and grid interval along the crystal ***X***-axis (**fra**) (in fractional units) or cartesian ***X***-axis (**car**) (in Ångstroms).

ymin ymax ystepb zmin zmax zstep
These are similarly defined along the ***Y***-axis and ***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).

Scalar function type	**File names**		
	SciAn SY format	**SciAn STF format**	**XDGRAPH format**
Electron density ρ(**r**)	**3DRHOO**//*name*	**3DRHOO**//*name*//**.stf**	**3DRHOO**//*name*//**.grd**
Laplacian $\nabla^2\rho(\mathbf{r})$	**3DLAPP**//*name*	**3DLAPP**//*name*//**.stf**	**3DLAPP**//*name*//**.grd**
Negative of the Laplacian $-\nabla^2\rho(\mathbf{r})$	**3DLAPM**//*name*	**3DLAPM**//*name*//**.stf**	**3DLAPM**//*name*//**.grd**
Magnitude of the gradient $\nabla\rho(\mathbf{r})$	**3DGRHO**//*name*	**3DGRHO**//*name*//**.stf**	**3DGRHO**//*name*//**.grd**

10.7.2 3D plot specific instructions

These instructions must follow the general 3D plot instructions.

PL3D (*)plot *func(1)… func(n)*

NOTE There is no limit for **PL3D plot** instructions and number of *func* instructions in each line, except the 256 character string limitation in the latter case.

(*)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 ($\rho(\mathbf{r})$)
lapp – laplacian of the electron density ($\nabla^2\rho(\mathbf{r})$)
lapm – negative laplacian of the electron density ($-\nabla^2\rho(\mathbf{r})$)
grho – magnitude of the gradient of electron density ($|\nabla\rho(\mathbf{r})|$)

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 **ATBP Params** 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, and the utility program **rays2pov** provides an interface to the ray-tracing program POV-Ray [16].

VZ3D (*)plot
files *file(1) file(2) … file(n)*
basins *label(1) label(2) … label(n)*
range (*)default xmi *xmi* **xma** *xma* **ymi** *ymi* **yma** *yma* **zmi** *zmi* **zma** *zma*
grid (*)default dx *dx* **dy** *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.

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.

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.

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.

grid (*)default dx *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 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).

The file `rays.dat` may also be visualised using the utilities program **rays2pov**, which produces an input file for the ray tracing program POV-Ray (http://www.povray.org). This porgram is available for a number of platforms, including Linux and Windows. See figure 10-3 for an example plot, and see Section 11.8 for full details on using **rays2pov.**

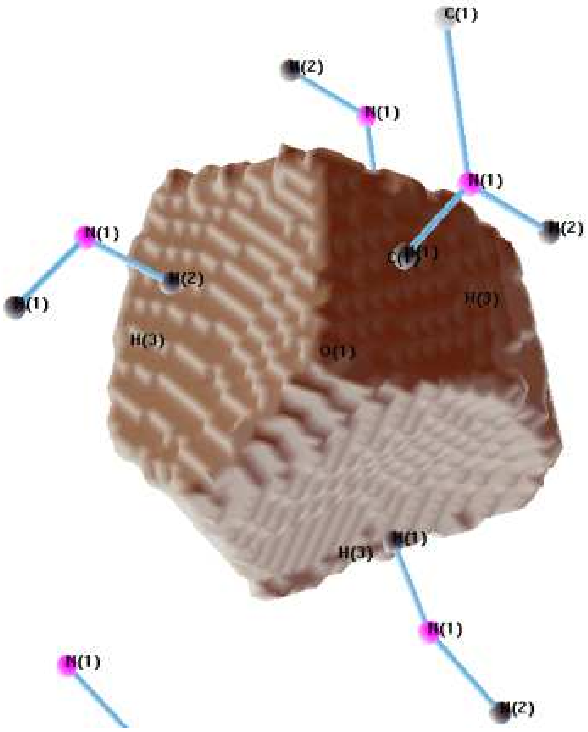


Figure 10-2. Atomic basin of O(1) atom in methyl carbamate

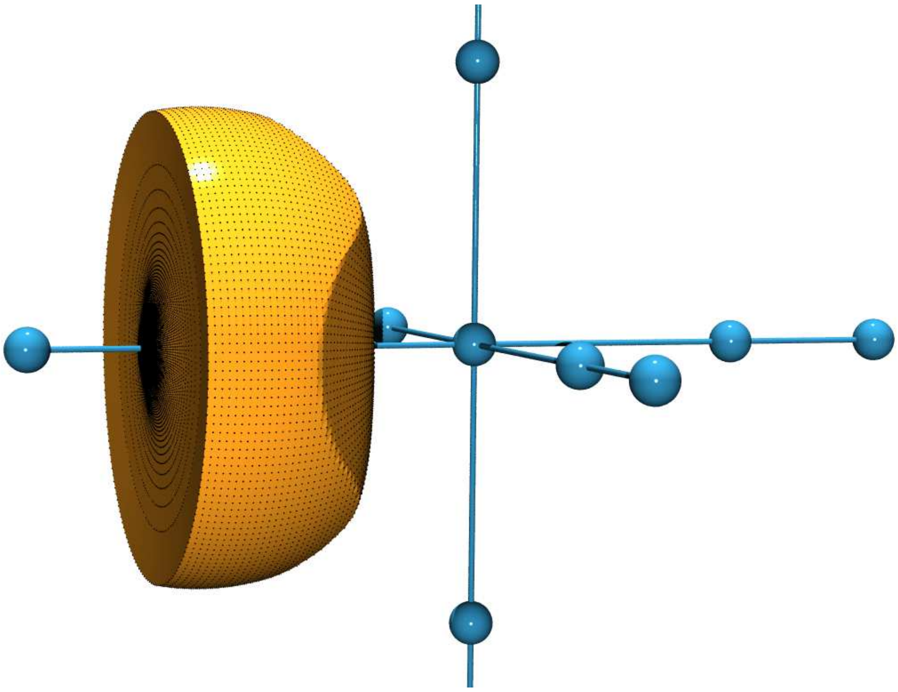


Figure 10-3. Atomic basin of C atom in Cr(CO)$_6$ using the utilities program rays2pov.

10.9 Description of the atomic properties evaluated by TOPXD

Populations		
Atomic population	N	$N(\Omega)=\int_\Omega \rho(\mathbf{r})\ d\tau$
Net charge	Q	$q(\Omega)=Z(\Omega)\ -N(\Omega)$
Energies		
Atomic Lagrangian (the error in L is a measure of the accuracy of the numerical integration)	L	$L(\Omega)=\ -1/4\int_\Omega \nabla^2\rho d\tau$
Atomic value of nuclear-electron potential energy with its own nucleus; $\mathbf{R}_\Omega$ is the position vector of Ω in the system frame	VNEO	$VNEO(\Omega)=-\int_\Omega (Z_\Omega/r_\Omega)\rho(\mathbf{r})d\tau$ $\mathbf{r}_\Omega=\mathbf{r}-\mathbf{R}_\Omega$; $r_\Omega=\lvert \mathbf{r}_\Omega \rvert$
Atomic Forces		
Atomic force components Force on nucleus of atom Ω by the electron density of atom Ω	FAXA FAYA FAZA	$FAXA(\Omega)=\int_\Omega (Z_\Omega\ /r^3_\Omega)x_\Omega\rho(\mathbf{r})d\tau$ $FAYA(\Omega)=\int_\Omega (Z_\Omega\ /r^3_\Omega)y_\Omega\rho(\mathbf{r})d\tau$ $FAZA(\Omega)=\int_\Omega (Z_\Omega\ /r^3_\Omega)z_\Omega\rho(\mathbf{r})d\tau$ $x_\Omega,y_\Omega,z_\Omega$, xyz comp. of $\mathbf{r}_\Omega$
'Radial' Atomic expectation values		
Atomic expectation value of the operator r^n_Ω	R(-1) R(+1) R(+2) R(+3) R(+4)	$R^N(\Omega)=\int_\Omega r^n_\Omega\ \rho(\mathbf{r})d\tau$
Atomic expectation values of r^n_Ω averaged over $r_\Omega \bullet \nabla\rho(\mathbf{r})$. It reflects the distortion of the$\nabla\rho(\mathbf{r})$ field of the charge density that is caused by the formation of chemical bond. For n=0 and for a free atom (or a perfectly spherical atom in a molecule or in a crystal) is equal to -3N(Ω)	GR(-1) GR(0) GR(1) GR(2)	$GR^N(\Omega)=\int_\Omega r^n\mathbf{r}_\Omega\ \bullet\boldsymbol{\nabla}\rho(\mathbf{r})d\tau$.
Atomic volumes and related populations		
Volume of the region of the atomic basin where $\rho(\mathbf{r})$ is greater or equal 0.001 au	V001	$V001(\Omega)=\int_\Omega d\tau_{0.001}$ where $d\tau_{0.001}$ are the infinitesimal volume elements where $\rho(\mathbf{r})$ exceeds or is equal to 0.001 au
Electron population in the V001 region	N001	$N001(\Omega)=\int_\Omega \rho(\mathbf{r})d\tau_{0.001}$
Ratio of electron populations in V001 and in the atomic basin	R001	R001=N001/N
Volume of the region of the atomic basin where $\rho(\mathbf{r})$ is greater or equal 0.002 au	V002	$V002(\Omega)=\int_\Omega d\tau_{0.00}$
Electron population in the V002 region	N002	$N002(\Omega)=\int_\Omega \rho(\mathbf{r})d\tau_{0.002}$
Ratio of electron populations in V002 and in the atomic basin	R002	R002=N002/N
Total atomic volume	VTOT	$VTOT(\Omega)=\int_\Omega d\tau$

Atomic unabridged moments		
Atomic dipole components	DX DY DZ	$DX(\Omega)= -\int_\Omega \rho(\mathbf{r}) x_\Omega d\tau$ $DY(\Omega)= -\int_\Omega \rho(\mathbf{r}) y_\Omega d\tau$ $DZ(\Omega)=-\int_\Omega \rho(\mathbf{r}) z_\Omega d\tau$ $x_\Omega, y_\Omega, z_\Omega$, xyz comp. of $\mathbf{r}_\Omega$
Atomic dipole magnitude	DM	$DM(\Omega)=\lvert \mathbf{DM}(\Omega) \rvert$ where $\mathbf{DM}(\Omega)$ is the atomic dipole vector
Components of atomic displacement vector	DCX DCY DCZ	DCX= - DX/N(Ω) DCY= - DY/N(Ω) DCZ= - DZ/N(Ω)
Coordinates of the centroid of negative charge	CX CY CZ	$CX= DCX+X_\Omega$ $CY= DCY+Y_\Omega$ $CZ= DCZ+Z_\Omega$ $X_\Omega, Y_\Omega, Z_\Omega$, xyz components of $\mathbf{R}_\Omega$
Atomic 2nd moment tensor components	QXX QXY QXZ QYY QYZ QZZ	$QXX(\Omega)= -\int_\Omega x_\Omega^2 \rho(\mathbf{r})\, d\tau$ $QXY(\Omega)= -\int_\Omega x_\Omega y_\Omega \rho(\mathbf{r})\, d\tau$ $QXZ(\Omega)= -\int_\Omega x_\Omega z_\Omega \rho(\mathbf{r})\, d\tau$ $QYY(\Omega)= -\int_\Omega y_\Omega^2 \rho(\mathbf{r})\, d\tau$ $QYZ(\Omega)= -\int_\Omega y_\Omega z_\Omega \rho(\mathbf{r})\, d\tau$ $QZZ(\Omega)= -\int_\Omega z_\Omega^2 \rho(\mathbf{r})\, d\tau$
Eigenvalues and eigenvectors of the atomic 2nd moment tensor		
Atomic 3rd moment tensor components		
Atomic 4th moment tensor components		
Atomic Shannon information entropy		
Atomic information (missing information function). This is the integral of ρ′lnρ′ where ρ′ is the un-normalized electron density (IUN) or the normalized electron density to unity over the atomic basin (INO) [15]	IUN INO	$I(\Omega)= - \int_\Omega \rho' \ln\rho' d\tau$

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 ρ(**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 $\nabla^2\rho(\mathbf{r})$ 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:

64×48×120 (φ×θ×radial) for non-H atoms
32×24×96 (φ×θ×radial) for hydrogen atoms (if not involved in H-bond)
48×32×96 (φ×θ×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 $n\varphi \times n\theta$. 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 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 ***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(Ω), L (Ω), etc. using:

1. first test : 64×48×120 ***accur*** =0.001
2. second test : 64×48×120 ***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 5×10^{-5} for H atoms, possibly around 1×10^{-5}. A value of 1×10^{-4} could be perhaps acceptable, but not too precise.
2. for second row atoms (C,N,O, etc) it should not exceed 1×10^{-3}. Possibly 1×10^{-4}

Q: You've mentioned that the computing time increases by a factor of φ×θ planes, but how does the *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 ***nvi*** reachable atoms. So that the DO loop in PATHEN and PATHEN2 (these DO's run on the 3 × ***nvi*** ×

star_multiplicity coordinates of the possible $\nabla\rho(\mathbf{r})$ 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 ***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 1×10^{-3} despite the fact that I use *accur*=0.001 and $n\varphi\times n\theta$ as large as 96×64. I remember that you have mentioned that decreasing the number of points might help, but when I reduce these numbers to 48×32 or 64×48 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

1. A. Volkov, C. Gatti, Yu Abramov and P. Coppens, *Acta Cryst.* **A56**, 252 (2002).
2. R. F. W. Bader *Atoms in Molecules: A Quantum Theory;* Claredon Press: Oxford, 1990.
3. C. Gatti, *TOPOND98 User's Manual*; CNR-CSRSRC, Milano, Italy, 1999.
4. V. R. Saunders, R. Dovesi, C. Roetti, M. Causà, N.M. Harrison, R. Orlando, and C. M. Zicovich-Wilson, *CRYSTAL98 User's Manual;* University of Torino, Italy, 1998.
5. J. Baker, *J. Comput. Chem.* **7**, 385 (1986).
6. P. L. A. Popelier, *Chem. Phys. Lett.* **228**, 160, (1994).
7. Y. Aray, J. Rodríguez and R. López-Boada, *J. Phys. Chem. A*, **101**, 2178. (1997).
8. C. F. Gerald and P. O. Wheatley, *Applied numerical analysis*. Addison-Wesley Pub. Co., 1989.
9. J. Cioslowski and A. Nanayakkara, *Chem. Phys. Lett.* **219**,151. (1994).
10. P. L. A. Popelier *Chem. Phys. Comm.* **108**, 180 (1998).
11. C. Katan, P. Rabiller, C. Lecomte, M. Guezo and M. J.Souhassou, *Appl. Cryst.* **36**, 65, (2003).
12. T. A. Keith, *Ph.D. thesis*, McMaster University, Ontario, Canada, 1993.
13. F. W. Biegler-König, R. F. W. Bader, and T. H. Tang, *J. Comput. Chem.* **13**, 317 (1982).
14. E. Pepke and J. Lyons, *SciAn.* Supercomputer Computations Research Institute, Florida State University, Tallahassee, Florida.
15. M. Hô et al. *J. Chem. Phys.*, **108,** 5469, (1998)
16. http://www.povray.org/

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 anisotropic displacement parameters (adp's) 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 $6N$ adp's 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 $m=21$ (6×7/2) parameters to be derived ($m < 21$ from molecules composed of $N < 8$ atoms]. However, to estimate $6N$ adp's (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:

$$\mathbf{U_a} = \mathbf{V}_a + \mathbf{W_a}$$

where $\mathbf{V}_a$ and $\mathbf{W}_a$ 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]:

$$\mathbf{W_a} = \mathbf{R'_a L R_a} + \mathbf{R'_a S} + \mathbf{S'R_a} + \mathbf{T}$$

where $\mathbf{R_a}$ is an antisymmetric tensor representing the vector product

$$\mathbf{R}_a = \begin{pmatrix} 0 - y_a z_a \\ y_a 0 - x_a \\ - z_a x_a 0 \end{pmatrix}$$

with x_a, y_a and z_a being the Cartesian coordinates of nucleus **a** situated at position $\mathbf{r}_a$.
Since $\mathbf{V}_a$ and $\mathbf{W}_a$ 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 **W** (the difference of the components of $\mathbf{W_a}$ and $\mathbf{W_b}$ along the internuclear vector $\mathbf{d_{ab}} = \mathbf{r_b} - \mathbf{r_a}$) vanishes for all **a-b** links:

$$\Delta(\mathbf{W})_{\mathbf{ab}} = \mathbf{d}'_{\mathbf{ab}}(\mathbf{W}_{\mathbf{a}} - \mathbf{W}_{\mathbf{b}})\mathbf{d}_{\mathbf{ab}} = \mathbf{0}$$

but does not necessarily for **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:

$$\Delta(\delta\mathbf{U})_{\mathbf{ab}} = 0$$

After the *k-th* cycle we obtain

$$\mathbf{U}_{\mathbf{a}}^{(k)} = \mathbf{V}_{\mathbf{a}} + \delta\mathbf{U}_{\mathbf{a}}^{(k)} \quad , \qquad \mathbf{U}_{\mathbf{b}}^{(k)} = \mathbf{V}_{\mathbf{b}} + \delta\mathbf{U}_{\mathbf{b}}^{(k)} \quad \text{with} \quad \Delta(\mathbf{U}^{(\mathbf{k})})_{\mathbf{ab}} = \Delta(\mathbf{V})_{\mathbf{ab}}$$

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 *3Nx3N* **V** matrix) from *n* frequencies and normal modes obtained by the *Gaussian98/03* program packages. Details of the vibrational analysis implemented in *Gaussian98/03* are described in reference [2]. Users should note that it is necessary to run the *Gaussian98/03* frequency analysis with the IOP(7/33=1) option to obtain the normal coordinates in a form readable by **XDVIB1**. It is also imperative that the "opt" option is not used in the Gaussian job. There is an important point to make concerning these calculations. Harmonic vibrational analyses based on theoretical force fields are 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 *Gaussian98/03* programs work in terms of mass-weighted generalized coordinates to separate the internal (vibration) and external (translation and rotation) motion. The internal coordinates **v** (*n*-vector) are generated to be orthogonal to the external ones, **w** (6-vector related to the Eckart coordinates). The output displays the wavenumber (υ_k), the reduced mass (μ_k) and the normalized Cartesian displacement vector for each normal mode ($\mathbf{q}_k$).

Important The six external modes calculated by Gaussian are often far from zero, even if the opt=VeryTight option is used. This seems to result in normal modes which are not orthogonal to each other. As a consequence, the MSDA matrix calculated by XDVIB may have some 'contamination' from external (artefact) modes (especially if default opt is used). This problem is under investigation.

The MSDA matrix is diagonal in terms of normal coordinates [3]

$$<\mathbf{qq}'> = \mathrm{diag}(\delta_1, \delta_2, \delta_3, \ldots \delta_n)$$

where

$$\delta_k = \frac{h}{8\pi^2 c\upsilon_k}\coth\left(\frac{hc\upsilon_k}{2kT}\right)$$

while in Cartesian representation

$$<\mathbf{vv}'> = \mathbf{V} = \mathbf{L} <\mathbf{qq}'> \mathbf{L}' \quad , \qquad \mathbf{q} = \mathbf{Lu}$$

11.1.2 Files used and created by XDVIB1

Input: `xd.mas, `*`Gaussian98/03`*` output file`
Output: `xd_vib1.par, xd_vib1_sig.dat xd_vib1.out`

11.1.3 Input instructions for XDVIB1

11.1.3.1 SELECT

SELECT temp 100.0 **scale** 1.0 **(*)nlinear 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.

Method/basis	Scale factor	Method/basis	Scale factor
HF/3-21G	0.9085	HF/631-G(d)	0.8928
MP2(Full)/6-31G(d)	0.9427	MP2(FC)/6-31G(d)	0.9434
SVWN/6-31G(d)	0.9833	BLYP/6-31G(d)	0.9940
B3LYP/6-31G(d)	0.9613		

(*)nlinear linear

The molecule must be described either as a nonlinear or a linear molecule.

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 $frqmin < \nu < frqmax$ 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*98/03 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" in 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 adp's ($\mathbf{V}_a$) 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 Δ_{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, Δ is invariant under rotation about the bond vector. 1-3 links show a similar transferability. This local symmetry of the adp's 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 than transformed into the local frames

$$\overline{\mathbf{V}}_{\mathbf{a}} = \mathbf{M}_{\mathbf{a}}\mathbf{V}_{\mathbf{a}}\mathbf{M}_{\mathbf{a}}^{'}$$

where $\mathbf{M}_a$ 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 adp's :

$$\overline{\mathbf{U}}_\mathbf{a} = \mathbf{N}_\mathbf{a}\mathbf{O}\hat{\mathbf{U}}_\mathbf{a}\mathbf{O}^{-1}\mathbf{N}_\mathbf{a}^{'}$$

where $\mathbf{N}_a$ 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

$$\hat{\mathbf{V}}_\mathbf{a} = \mathbf{P}_\mathbf{a}^{-1}\overline{\mathbf{V}}_\mathbf{a}\mathbf{P}_\mathbf{a} \qquad \mathbf{P}_\mathbf{a} = \mathbf{N}_\mathbf{a}\mathbf{O}$$

The program prints out the full $\Delta(\mathbf{V})$ matrices in both representations (Cartesian and crystal). Their comparison can suggest a possible model for the segmentation that is applicable to the molecule during the LS refinement.

If

$$\Delta_{\mathbf{ab}}(\hat{\mathbf{V}}) \cong \Delta_{\mathbf{ab}}(\mathbf{V})$$

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 intra-group links. The former approach corresponds to a rigid-body, while the latter to a segmented rigid-body model refinement.

For a recent publication describing the use of the **XDVIB** programs see reference 4.

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 file containing the Cartesian atomic parameters (`xd_vib1.par` as calculated and output by **XDVIB1** and renamed as `xd_vib2.inp`) 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 adp's.

Bibliography

1. V. Schomaker and K. N. Trueblood, K.N. *Acta Cryst.* **B54**, 507 (1968)
2. J. W. Ochterski, Vibrational Analysis in *Gaussian* (October, 1999) PDF file available from : http://gaussian.com/g_whitepap/vib/vib.pdf
3. S. J. Cyvin, *Molecular Vibration and Mean Square Amplitudes*, Amsterdam: Elsevier, 1968.
4. A. E. Whitten and M. A. Spackman *Acta Cryst.* **B62**, 875-888 (2006)

11.3 XDPDF - A Program to Calculate the Probability Density Function of an Atom.

The program XDPDF calculates the vibrational probability density function (pdf) of a single atom using the least-squares estimated thermal tensors up to the fourth order Gram-Charlier expansion [1-3]. The pdf is obtained either as a 3-D grid in sections parallel to the real cell axes (or the U^{ij} basis vectors), or as a 2-D grid in a specified plane. XDPDF also provides an estimation [2] of the data resolution required for satisfactory refinement of anharmonic coefficients.

As outlined in the Introduction, in the harmonic approximation, the vibrational pdf of the nuclear displacement vector **u**, taken with respect to the equilibrium position ($\mathbf{u} = \mathbf{q}\text{-}\mathbf{q}_o$), is a normal trivariate Gaussian distribution:

$$P_o(\mathbf{u}) = (2\pi)^{-3/2}(\det \mathbf{U})^{-1/2}\exp[-(\mathbf{u}^T\mathbf{U}^{-1}\mathbf{u})/2] \qquad [1]$$

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

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

$$t_o(\mathbf{h}) = \exp(-2\pi^2\mathbf{h}^T\mathbf{U}\mathbf{h}) \qquad [2]$$

In XD, the anharmonic model used is the Gram-Charlier expansion [1], in which the anharmonic pdf is approximated in terms of zero and higher derivatives of the normal distribution:

$$P(\mathbf{u}) = (1 + \frac{1}{3!}C_{jkl}H_{jkl} + \frac{1}{4!}D_{jklm}H_{jklm} + \ldots)P_o \qquad [3]$$

where H_{jkl}, H_{jklm} are three dimensional Hermite polynomials being functions of **U** and **u**, while the coefficients C_{jkl}, D_{jklm} 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:

$$T(H) = (1 - \frac{4}{3}\pi^3 iC_{jkl}h_jh_kh_l + \frac{2}{3}\pi^4 D_{jklm}h_jh_kh_lh_m + \ldots)T_o(H) \qquad [4]$$

It is always important to ascertain whether the thermal parameters obtained by the least-squares refinement represent physically realistic atomic motion. For harmonic anisotropic motion this is straight-forward - if any of the eigenvalues of the U_{ij} tensor are negative (*i.e.* the tensor is non-positive definite), this implies an imaginary displacement. When anharmonic motion is involved however, it is less obvious, since the Gram-Charlier expansion coefficients C^{jkl} and D^{jklm} have (in general) no direct physical meaning [3]. The best way is to examine carefully the derived pdf function. A 3D map encompassing the full pdf should be examined (***cell** option). Negative regions of the pdf are physically meaningless - however, due to the limited accuracy of the refined coefficients, the pdf may have some negative features far from the center, with small absolute values. If negative regions are found close to the center of the pdf, especially if they are surrounded by positive areas, it is likely that the refined anharmonic coefficients are meaningless. The program reports the ranges in the pdf for each section, and also writes an XD grid file. To allow for better inspection of any negative regions, the pdf may be visualised in XDGRAPH using an iso-surface plot or a relief or contour plot (see below).

It should always be remembered that there is a close mathematical resemblance between the formalisms used to describe non-Gaussian pdf's and aspherical electron densities, see for example Mallinson *et al* [4]. There are likely to be high correlations between the Gram-Charlier coefficients and the multipole populations for the same atom in the least-squares refinement, and great care should be taken to ensure that a true minimum is reached. Kuhs [2] has provided an estimation of the minimum required resolution in the data Q_n for a satisfactory least-squares refinement for each order *n* of the anharmonic term. XDPDF prints a table of these Q_n values for all anisotropic atoms in the parameter file. The program expects the parameter file to contain the refined C^{jkl} and D^{ijklm} Gram-Charlier coefficients from recent versions (4.12 or later) of XDLSM. If a parameter file from an earlier version of XD is supplied, the program assumes that the anharmonic parameters are of the form U^{jkl} and U^{jklm}, and makes the necessary conversion. Note that the reported C^{jkl} and D^{ijklm} coefficients in XD are multiplied by 10^3 and 10^4 respectively, following the standard convention.

11.3.1 Files used and created by XDPDF

Input:	`xd.res, xd.mas`
Optional input	`xd.inp` (if `xd.res` does not exist)
Output :	`xd_pdf.out, xd_pdf.grd`

11.3.2 Input instructions for XDPDF

Input instructions for XDPDF are given in the master file `xd.mas` and are placed between the MODULE *XDPDF and END XDPDF lines. The program writes two files, the list file `xd_pdf.out` and the grid file `xd_pdf.grd`. The grid files may be visualised using either XDGRAPH or external programs such as Moliso [5] or MolecoolQT [6] (see Figure 11-1).

11.3.2.1 SELECT

SELECT atom *atom(1)* **scale** 1.0 **(*)angstrom**

atom *atom(1)*
The label of the atom for which the pdf is to be calculated. If this label does not correspond to a non-isotropic atom in the parameter file, the pdf calculation is halted.

scale *scalefactor*
The pdf is multiplied by this scalefactor.

(*)angstrom
The pdf is normally calculated using the unitless coefficients β^{ij}, C^{jkl} and D^{jklm}, in sections parallel to the real axial directions. The displacement vector **u** is in unitless cell fractions along this basis and the resultant pdf is therefore unitless (effectively per unit cell). If ***angstrom** is selected, then the pdf in computed in units of Å^{-3}. The coefficients U^{ij}, U^{jkl}, U^{jklm} are used in expressions 1 and 2 above, as these have units of Å^2, Å^3 and Å^4 respectively. In this case the displacement vector **u** is in units of Å along the basis vector ($a^*\mathbf{a}$, $b^*\mathbf{b}$, $c^*\mathbf{c}$) – see ref 3. Only for orthorhombic, tetragonal and cubic unit cells will this basis form an orthogonal set. For these unit cells only, the unitless pdf can be converted to units of Å^{-3}, by dividing by the cell volume in Å^{-3}. For all other cells, this is not the case and the view of the grid file `xd_pdf.grd` in XDGRAPH will be distorted, since the latter program expects an orthogonal isometric grid.

11.3.2.2 ADDPDF

ADDPDF *atom(1), atom(2), atom(3) …….*

As well as the central atom selected above, the pdf's of the atoms *atom(1)…* etc will also be included in the calculation of the grid file. Isotropic atoms are not allowed in this list. Their contributions are not reported in the output file `xd_pdf.out`, which just lists the pdf for the central atom.

11.3.2.3 QMORD

QMORD (*)second (*)third (*)fourth

This selects which order of quasi-moments are to be included in the calculation of the pdf. If only ***second** is selected, then the Gaussian anisotropic (harmonic) pdf is calculated, while selecting ***third** or ***fourth** includes also the third and fourth order Gram-Charlier anharmonic components. Any combination of quasi-moments is acceptable - if none are selected, the harmonic pdf is calculated. The previous keyword **CUMORD** is also acceptable.

11.3.2.3 GRID

GRID (*)3-points *cryst

If ***3-points** is selected, then a single section of the pdf is calculated through a plane described by three points. The first point is *atom(1),* the origin of the plot plane. The other two points are given by two subsequent **ATOM** instructions, which specify respectively the x-direction (*atom(1)* → *atom(2)*) and the x-y plane (*atom(1)* → *atom(3)*)

There are 2 types of format for **ATOM** instructions

4. **ATOM label** *atom* (**symm** *symop*) (**trans** *tx ty tz*)
   atom must be identical to an atom label given in the parameter file. **NOTE :** ***symm*** *and* ***trans*** *are included for future enhancement and are not operative in the current version of XDPDF.*

5. **ATOM xyz** *x y z*
   the position of a point in crystallographic fractional coordinates

If ***cryst** is selected (the default), then the pdf is calculated in sections parallel to the real cell axes **a**, **b**, **c** (unitless pdf) or to the basis vector (a^***a**, b^***b**, c^***c**) (pdf in units of Å^{-3}), controlled by the **LIMITS** instructions.

11.3.2.4 LIMITS

This command defines the limits of the calculated grid file. For the ***cryst** option, a 3-D grid is obtained when all three axes have more than one grid point.

LIMITS (*keyword value*) ...

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

xmin	[-0.8]	**xmax**	[0.8]	**nx**	[50]
ymin	[-0.8]	**ymax**	[0.8]	**ny**	[50]
zmin	[-0.8]	**zmax**	[0.8]	**nz**	[50]

If the ***cryst** option is used, *?min* and *?max* are the minimum and maximum displacements in Å from the equilibrium position of the atom along the respective basis. *n?* is the number of grid points in total in this direction.

If the ***3-points** option is used, the values of *?min* and *?max* are the minimum and maximum fractional displacements (in arbitary units) from the equilibrium position of the atom along x- and y-directions in the plot plane. The values for **zmin, zmax** and **nz** are ignored. It is recommended to use equal fractions and numbers of grid points along the axes. The atom indicated in the **SELECT atom** instruction is placed in the center of the plot.

Fig 11-1 Illustrative iso-surface plots of the density modification from the 3rd and 4th order anharmonic components to a pdf, showing the positive (cyan) and negative (magenta) regions. The graphical programs used were (a) MCE [6] and (b) XDGRAPH.

Conclusion

In conclusion, to establish whether a set of refined Gram-Charlier coefficients represent a physically reasonable model.

(a) the total pdf should be calculated, using all the refined Gram-Charlier coefficients. The total pdf is best visualized as an isosurface plot, which includes both positive and negative isosurfaces.

(b) ideally the pdf should be positive everywhere. Negative parts of the pdf are physically non-meaningful. In most cases the changes due to anharmonicity are only a few percent of the total probability density, and the gaussian distribution from the harmonic part is not visibly affected.

(c) due to the limited accuracy of the refined coefficients, there may be small volumes of negative pdf, distant from the nucleus.

(d) large volumes of very negative pdf, especially near the nucleus should be regarded with the greatest suspicion. Very probably the computed anharmonic coefficients are physically meaningless. It should be realized however that the pdf close to the nucleus is very sensitive to experimental error and to the limitations of the anharmonic expansion, particularly for heavier elements. These cases need to be considered on an individual basis.

Bibliography

1. C.K. Johnson and H.A. Levy, *International Tables for X-ray Crystallography,* Vol. IV, 311-336, Birmingham: Kynoch Press, 1974.
2. W. F. Kuhs *Acta Cryst.* **A48**, 80-98 (1992).
3. K. N. Trueblood, H.-B. Bürgi, H. Burzlaff, C. M. Gramaccioli, H. H. Schultz, U. Shmueli, S. C. Abrahams *Acta Cryst.* **A52**, 770-781 (1996).
4. P. R. Mallinson, T. Koritsansky, E. Elkaim, N. Li, P. Coppens *Acta Cryst* **A44**, 336-342 (1988).
5. C. Hübschle, P. Luger, *J. Appl. Cryst*, **38**, 901 (2006).
6. C. Hübschle, B. Dittrich, *J. Appl. Cryst*, **44**, 238 (2011).

11.4 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 will also look for the files `xd_fou.cif` and `xd_pro.cif`. These latter files are not currently produced by XD, but are included for future enhancements.

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 may be edited by the user, and it 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 :** 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.

11.4.1 Files used and created by XDCIF

Input:	`xd_*.cif (* = fft, lsm, geo, dat,` *`fou`*`,` *`pro`*`)`
Output :	`xd_archive.cif`

11.5 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 $<w(F_o-F_c)^2>$ 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 $\sigma(F_o)$.

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 Section 4.6.6). Ideally the weights should be chosen such that $<(\Delta/\sigma)^2>$ is unity ($\Delta = Y_o - Y_c$), but this can rarely be achieved. This condition corresponds to $<(w\Delta^2)> \sim 1$, where w is the statistical weight equal to $1/\sigma^2(Y)$ and the so-called goodness of fit parameter (Gof or S) defined as

$$\text{Gof } (S) = [\Sigma\, w(F_o - F_c)^2 \,/\, (\text{nobs} - \text{npar})]^{1/2}$$

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 $<(w\Delta^2)>$ with *hkl* indices, index parities, F_o and $\sin\theta/\lambda$. The analysis is carried out twice: firstly using all reflection data and secondly using only those reflections which were included in the least-squares refinement.

11.5.1 Files used and created by XDWTAN

Input: `xd.fco, xd.mas`
Output : `xd_wta.out`

Bibliography

1. P. R. Mallinson and K. W. Muir, *J. Appl. Cryst.* **18**, 51 (1985).
2. J. D. Dunitz, *X-Ray Analysis and the Structures of Organic Molecules*, VCH Weinheim, pp 213-222 (1995).

11.6 XDHKL - A Program to Analyse the Reflection Data

XDHKL analyses the reflection file `xd.hkl` (by default) or SHELX format files (if the file name is given on the command line). The file format is automatically determined by the program, which

- examines the lattice centering absences and suggests an appropriate lattice type
- prints a summary of the reflection intensity statistics
- checks for systematic absences according to the space group implied in `xd.mas` and reports whether any are found, and
- writes out a merged reflection file, containing just the unique data and with all systematic absences removed. There are no arbitrary limits on the number of reflections handled, other than the system memory on the computer.

Note that the merging process involves a simple averaging, and so is *quite inappropriate* for experimental data, where a much more sophisticated program such as SORTAV [1] should be used. The program is, however, useful for merging synthetic reflection data, *e.g.* as obtained from quantum chemical programs such as CRYSTAL [2], where there are no statistical issues involved. If the file `xd.hkl` contains phase angles (format NDAT -7, see Table 2-2), these are transferred to the merged reflection file.

11.6.1 Files used and created by XDHKL

Input: `xd.hkl, xd.mas`
Output : `xd_hkl.out, xd.hkl.merged`

Bibliography

1. R. H. Blessing, *J. Appl. Cryst.* **22**, 396 (1989).
2. http://www.crystal.unito.it/

11.7 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): *R11 R12 R13 R21 R22 R23 R31 R32 R33 T1 T2 T3*
`symm.dat`	to generate symmetry-equivalents of integrated atoms - 1 line per each new symmetry-equivalent atom in the following format: *<atom label> R11 R12 R13 R21 R22 R23 R31 R32 R33 T1 T2 T3* where *<atom label>* is the atomic label of the integrated "parent" atom, and *Rij* and *Ti* 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): *<atom label> <occupation> <n. of electrons in free atom>*

11.7.1 Files used and created by AIM2TAB

Input: `TOPOND/TOPXD/AIMPAC outputfiles, files.dat, (trans.dat, symm.dat, new.dat)`
Output : `aim2tab.out`

11.8 LSDB

This is a program [1] for automatic setup of the atomic local coordinate systems and chemical constraints starting from SHELX [2], PLATON [3] or XD structure files. It links into the UBDB (the Buffalo Aspherical Atom Database). The program is designed for interactive use and is self-explanatory. The atomic local coordinate systems can then be visualized in PLATON. Current versions of the program for Linux and Windows and the UBDB are freely available for academic users from

http://harker.chem.buffalo.edu/group/databank.html

Bibliography

1. (a) A. Volkov, X. Li, T. Koritsanszky, P. Coppens *J. Phys. Chem* (2004) *108*, 4283-4300 (b) P. M. Dominiak, A. Volkov, X. Li,. M. Messerschmidt, P. Coppens *J. Chem. Theo. Comp.* (2006) in press.

2. G. M. Sheldrick, SHELX97. *Programs for Crystal Structure Analysis (Release 97-2).* University of Göttingen, Germany (1997)
3. A. L. Spek *PLATON, A Multipurpose Crystallographic Tool* Urecht University, Utrecht, The Netherlands (2003)

11.9 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.

11.10 RAYS2POV

This program provides another way of visualising the inter-atomic surface file `rays.dat`, which is obtained when the ***SaveSurf** option is used in TOPXD (see Section 10.8). It writes an output file for the well known ray-tracing program POV-Ray, obtainable for both Windows™ and Unix/Linux platforms.

It is invoked by the command line : **rays2pov** <*input_file*> <*output_file*>

The command line parameters <*input_file*> <*output_file*> are optional, available if the user wishes to change the defaults. The default input file is `rays.dat` and the default output POV-Ray scene description file is `xd_ias.pov`. The program is controlled by the instruction file `rays2pov.ini`, which the program looks for first in the local working directory, and then in the directory pointed to by the environment variable XD_DATADIR. If this file is not found, the program defaults will be used. A default version of `rays2pov.ini,` as supplied with the installation, is shown below.

```
# colors - only colors given in colors.inc are acceptable (case sensitive !)

SurfaceColor=Copper
AtomColor=SteelBlue
BondColor=SteelBlue
PointColor=Black
CP_Color=Red
TextColor=Black

# textures

SurfaceTexture=Metal
AtomTexture=Shiny
BondTexture=Shiny
PointTexture=Shiny
CP_Texture=Shiny

# numerical parameters - all in Angstroms except rCutoff in a.u.

rCutOff=3.0
BondTol=0.2
TextOffset=0.1
MinimumBond=0.8
MaximumBond=3.0
TextWidth=0.1
TextScale=0.1
AtomRadius=0.15
BondRadius=0.02
PointRadius=0.005
CP_Radius=0.1
```

```
X_Rotation=0.0
Y_Rotation=0.0
Z_Rotation=0.0
X_Translation=0.0
Y_Translation=0.0
Z_Translation=0.0
ZoomFactor=1.0
LabelAtoms=no
RunPovray=yes
ShowIntersections=yes
```

The meaning of the parameters are :

SurfaceColor	Colour of IAS - any legal POV-Ray colour
AtomColor	Colour of atoms - any legal POV-Ray colour
BondColor	Colour of bonds - any legal POV-Ray colour
PointColor	Colour of points where rays intersect the IAS - any legal POV-Ray colour
CP_Color	Colour of bond critical points - any legal POV-Ray colour
TextColor	Colour of labels - any legal POV-Ray colour
SurfaceTexture	Any legal POV-Ray texture
AtomTexture	Any legal POV-Ray texture
BondTexture	Any legal POV-Ray texture
PointTexture	Any legal POV-Ray texture
CP_Texture	Any legal POV-Ray texture
rCutOff	Cutoff for IAS - in atomic units !
BondTol	bond is drawn if distance between A and B is < (Ra +Rb + bontol)
TextOffset	offset form atom position
MinimumBond	minimum distance for bond to be drawn
MaximumBond	maximum distance for bond to be drawn
TextWidth	
TextScale	
AtomRadius	
BondRadius	
PointRadius	
CP_Radius	
X_Rotation	
Y_Rotation	
Z_Rotation	
X_Translation	
Y_Translation	
Z_Translation	
ZoomFactor	
LabelAtoms	whether to label the atoms - yes/no
RunPovray	whether to run POV-Ray after writing output file - yes/no
ShowIntersections	whether to show the points where the rays intersect the IAS - yes/no

The output file `xd_ias.pov` can be very large and take a significant time for POV-Ray to parse. The structure is intended for easy manual editing. For instance, if labelling is chosen it will probably be necessary to manually edit the individual positions of the labels.
NOTE: The option TOPINT in XDPROP also write out a rays.dat file, but these files cannot be processed by the stand-alone version of RAYS2POV.

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 >>> $Revision: 5.42 (Sep 18 2007)$          18-Sep-07!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
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
  0.004206  0.004527  0.004206  0.004527  0.003877  0.003499  0.020006 0.656E-01
 -1.000000  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 0.2500
  0.050000  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  0.0000 -0.0032  0.0000  0.0000
 -0.0043  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
  0.4347  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
  1  0.981203  1.038973  1.038973  1.038973  1.038973  1.038973
  2  1.128085  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.

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).

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! <<< X D MASTER FILE >>> $Revision: 5.00a (Aug 07 2006)$            10-Aug-08!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
TITLE XD
CELL      8.0247   10.3068    3.7492    90.000    90.000    90.000
WAVE     0.71073
CELLSD    0.0001    0.0001    0.0001     0.000     0.000     0.000
LATT  A P
SYMM -X, -Y, Z
SYMM 0.5-X, 0.5+Y, -Z
SYMM 0.5+X, 0.5-Y, -Z
BANK CR
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    MODULE *XDLSM
SELECT  model 0 2 1 0 based_on F test verbose 1
SELECT  cycle -1 dampk 1. cmin 0.6 cmax 1. eigcut 1.d-09 convcrit 0.0
SAVE  deriv lsqmat *cormat
SOLVE  *inv diag *cond
!------------------------------------------------------------------------------
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
O    CHFW CHFW CSZD    2  -2   0   0  -4   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0106  0.0060  0.580
N    CHFW CHFW CSZD    2  -2   0   0  -3   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0061  0.0033  0.936
C    CHFW CHFW CSZD    2  -2   0   0  -2   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0033  0.0016  0.665
H    CHFW CHFW CSZD   -1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  0.0000  0.0000 -0.374
END SCAT
!------------------------------------------------------------------------------
ATOM     ATOM0    AX1 ATOM1    ATOM2   AX2 R/L TP TBL KAP LMX SITESYM  CHEMCON
O(1)     C(1)      Z  O(1)     H(2)     Y   R   2  1   1   4  NO
O(2)     C(1)      Z  O(2)     O(1)     Y   R   2  1   1   4  NO
O(3)     H(5)      Z  O(3)     O(1)     Y   R   2  1   1   4  NO
N(1)     H(4)      Z  N(1)     H(2)     Y   R   2  2   2   4  NO
C(1)     O(1)      Z  C(1)     O(2)     Y   R   3  3   3   4  NO
H(1)     N(1)      Z  H(1)     H(4)     Y   R   4  4   4   4  NO
H(2)     N(1)      Z  H(2)     H(4)     Y   R   4  4   4   4  NO
H(3)     N(1)      Z  H(3)     H(2)     Y   R   4  4   4   4  NO
H(4)     N(1)      Z  H(4)     H(1)     Y   R   4  4   4   4  NO
H(5)     O(3)      Z  H(5)     O(1)     Y   R   4  4   4   4  NO
DUM0      0.0000  0.0000  0.0000
END ATOM
!------------------------------------------------------------------------------
!GROUP2   atom1 atom2 ...
KEEP     kappa  1  2  3  4
KEEP     charge group1
!KEEP     rigid group1
!RESET    bond C(1) H(1) 1.09 ...
WEIGHT   -2.0 .0 .0 .0 .0 0.3333
SKIP     obs 0. 1.d10 *sigobs 3. 1.d06 sinthl 0. 2.
PRINT    sinthl .0 2. obs 0. 15. delta 0. 10. *del% 80 100 extcn 80. 100. *abssc
!EXTCN    *iso aniso *type_1 type_2 type_3 distr_g *distr_l msc_0  msc_1
!DMSDA    1.1  1.8
```

```
!FOUR     fmod1 4 2 0 0  fmod2 -1 2 0 0
!CON      num1 par1/iat1 num2 par2/iat2 ... = num0
!-------------------------------------------------------------------------------
KEY     XYZ --U2-- ----U3---- ------U4------- M- -D- --Q-- ---O--- ----H----
O(1)    111 111111 0000000000 000000000000000 00 000 00000 0000000 000000000
O(2)    111 111111 0000000000 000000000000000 00 000 00000 0000000 000000000
O(3)    111 111111 0000000000 000000000000000 00 000 00000 0000000 000000000
N(1)    111 111111 0000000000 000000000000000 00 000 00000 0000000 000000000
C(1)    111 111111 0000000000 000000000000000 00 000 00000 0000000 000000000
H(1)    111 100000 0000000000 000000000000000 00 000 00000 0000000 000000000
H(2)    111 100000 0000000000 000000000000000 00 000 00000 0000000 000000000
H(3)    111 100000 0000000000 000000000000000 00 000 00000 0000000 000000000
H(4)    111 100000 0000000000 000000000000000 00 000 00000 0000000 000000000
H(5)    111 100000 0000000000 000000000000000 00 000 00000 0000000 000000000
KAPPA   000000
KAPPA   000000
KAPPA   000000
KAPPA   000000
EXTCN   0000000
OVTHP   0
SCALE   1
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
SIZE     napl 200
MODEL    iam *multipole
! *au will print results in a.u. but input is always given in Ang
SELECT   numdx  esd  au  verbose 1
PROPERTY *rho gradrho d2rho nucpot core valence defden esp ef efg
! Define the cluster of atoms to be included:
!CRYSTAL alim [amin] [amax] blim [bmin] [bmax] clim [cmin] [cmax]
!APPLY    symm 1 translations 0 0 0 ato(1) ato(2) ...
```

```
!GROUP    ato(1) ato(2) ... (atom label)
!OMIT     1 2 ... 4 -8 (number in atom list)
!
! Parameters for topological analysis:
SELECT   cpcut 1.0d-6 lmax 4  nstep 20  rcut 4.0
SELECT   scale 0.05  dx 0.001  ds 0.005
SELECT   rEcrit 0.00001  rNcrit 0.00001
!
! Parameters for the integration routine:
ODESOLVE *rk bs eps 1.D-06 stepi 0.01
QUADINT  iqt 2  Nrad 50  Nang 194  Becke  *Stock
!
! Origin of the coordinate system for molecular moments
ORIGIN  *mass charge geom ucell
!
! Export atomic moments to Orient/MIN16
!EXPORT  *orient *min16 lmax [4] nmol [k] natmol [i=1..k]
!
!------End of General section-----
!
! Atomic/Molecular moments from pseudoatoms :
MULTMOM
!
! Calculate d-orbital populations for transition metals
!D-POP
!
! Atomic/Molecular from STOCKHOLDER partitioning:
!STOCKMOM atoms all *unique select ato(1) ato(2) ...
!
! Evaluation of intermolecular interaction energy and lattice energy:
!
!ATATPOT  *EXREP spack  *wilcox lj
!ATATPOT  *DISPR spack  *wilcox lj
!HBONDS   atoH(1) ato(1) rHBOND(1) atoH(2) ato(2) rHBOND(2) ...
!HPOLAR   atoH(1) atoH(2) ....
!
!INTEREN  frag 1  nat1 -nat2 *neutral
!INTEREN  frag 2  nat3 -nat4 *neutral
!INTEREN  EP aMM  mMM *EPMM  rCrit1 4. rCrit2 10. debug
!
!LATEN    frag1 1 -10  *neutral
!LATEN    radii 0.0 100.0
!LATEN    EP  *aMM  mMM  EPMM  rCrit1 4.0 rCrit2 10. lapf debug
!
NUCPROP (*)all (*)selected [n1] [n2] [-n3] ...
!
! Atomic charges fitting electrostatic potential:
!QFIT     grid 11 length 7.0 width 1.0 constrain false
!CONSTRAIN ato(1) ato(2) ...
!
```

```
! Function plots:
!POINT    x y z
!LINE     ato(1) ato(2)  npts 50
!LINE     points x1 y1 z1 x2 y2 z2 npts 50
!VECPLOT  *labels *zcut 1.2 omitrad 0.2 *vcut 0.8  vscale 1. xytol 0.4 *dipole 1.
!MAP      atoms ato(1) ato(2) ato(3) npts 50 stepsize 0.1
!MAP      bvect1 x1 y1 z1 bvect2 x2 y2 z2 cen x0 y0 z0 npts 50 stepsize 0.1
!CUBE     centre x y z   npts 30  stepsize 0.1
!CUBE     ato(1) ato(2)  npts 20  stepsize 0.1
!
! Gradient field plots (only for rho and esp functions):
!TRAJPLT  origin atom ato(1)
!TRAJPLT  xaxis  atom ato(2) Xdim1 -3.0 Xdim2 3.0
!TRAJPLT  yaxis  point x y z Ydim1 -3.0 Ydim2 3.0
!TRAJPLT  mark *atoms *labels *bonds *cps *basins *hbonds
!TRAJPLT  params Circle 0.1 ATrad 0.05 CPrad 0.08 CPgrid 0.3 CPlim 1.0d-4
!TRAJPLT  *plot *plane npath 36 *zcut 0.3 *xytol 0.5 *all select ato(1)
!
! Local Source Function (only for rho and esp functions):
!SOURCE   refpoint x y z
!SOURCE   point x y z
!SOURCE   line  ato(1) ato(2)  npts 50
!SOURCE   map atoms ato(1) ato(2) ato(3) npts 100 stepsize 0.1
!SOURCE   cube ato(1) ato(2) ato(3) npts 20 stepsize 0.1
!
! Topological analysis:
!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
!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) algrithm 2
!
! Atomic basin integration:
!TOPINT step 0.01 accur 0.001 rmax 8.0
!TOPINT spheres ato(1) r1 ato(2) r2 .....
!TOPINT atoms all *unique select ato(1) ato(2) ...
!
   END XDPROP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   MODULE XDGEOM
SELECT   rmin 0.8 rmax 1.8  tor *ato *bon *ang loc non
   END    XDGEOM
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   MODULE XDVIB
SELECT   temp 100. scale 1. *nlinear linear
MODES    *all frqmin 0. frqmax 1500.
!MODES    include 1 2 3 ...
```

```
!MODES    exclude 1 2 3 ...
DATAFILE *gaussian g98.out orient *standard input
   END  XDVIB
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   MODULE TOPXD
SIZE napl 2000
COMT just a comment for this run
DEBG symeqv deriv check
TRSH eigenvec 1.0d-10 nr 1.D-08
CGEN alim -0.5 1.5 blim -0.5 1.5  clim -0.5 1.5
MPAR  rcut 4.0d0  dstep 5.d-3  au iam esp
!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 ail 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 ail debug nstep 23 nnb 11 rmax 4.0 nmax 14
!  car  1. 1. 1.
!  fra  0. 0. 0.
!TLAP *line nr ail 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.
!  points car 0. 0. 0.  2. 2. 2.
!ATBP Params PhInSph 48 ThInSp 32  *SavSurf
!ATBP Spheres ato(1) 0.2 ato(2) 0.2 ...
!ATBP AltGuess BigStep 0.5 Accur 0.002 MaxRInt 10.00 Rmax 10.0 Step0 0.025 A 0.0 B 0.0
```

```
!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 ncp 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
!
!P2DCRY *diff rhoo file1 file2
!P2DCRY *diff lapm file3 file4
!P2DCRY  diff test2d rhoo
!P2DCRY  diff test2d trajmolg
   END  TOPXD
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   MODULE XDPDF
SELECT   atom atom(1) scale 1.0 orth
CUMORD   second  third  fourth
GRID     3-points *cryst
!ATOM     label atom(2) symm 1 trans 0 0 0
!ATOM     xyz  0. 0. 0.
LIMITS   xmin -0.8  xmax 0.8  nx 51
LIMITS   ymin -0.8  ymax 0.8  ny 51
LIMITS   zmin -0.8  zmax 0.8  nz 51
   END  XDPDF
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Chapter 13

XD2015 Installation

The XD2015 package is supplied as precompiled executables, for a variety of modern platforms. Once a user has registered for the package, he/she will be issued with the password(s) required to download the package from the XD website.

http://xd.chem.buffalo.edu/download.html

13.1 Installing XD2015 for Linux/Mac

Statically linked executables of the XD2015 package for Linux are provided for 32 and 64-bit Linux operating systems. They should run under any (relevant) flavour of Linux. The plotting program XDGRAPH is quite platform dependent, so that precompiled executables may not work properly. For this reason the source code and makefile are also supplied. To compile XDGRAPH you need an ANSI C compiler. Note that some C compilers provided as part of the operating system are *not* ANSI compatible. In addition, the ***Tcl/Tk*** libraries (available from http://www.tcl.tk) and optionally the ***OpenGL*** libraries (or substitutes such as the ***MESA*** libraries usually supplied with Linux installations) are required. The program is tested with ***Tcl/Tk*** version 8.3. The file `mk.config` will need to be edited to suit your local installation of Linux.

The program is provided in a tar GZipped archive. This should be copied to a suitable directory (probably root privileges are required) and unzipped,

```
gunzip xd2015.tar.gz
tar -xvf xd2015.tar
```

The resultant directory structure is something like :

xd2015	containing the Makefile and config files for XDGRAPH
xd2015/ports	config files for various compilers for XDGRAPH
xd2015/bin	containing all the executables
xd2015/xdgraph/src	containing the source code for XDGRAPH
xd2015/lib/xd	containing system files and databanks
xd2015/lib/xdgraph	containing system files for XDGRAPH

The environment variable XD_DATADIR points to the full pathname of xd2015/lib/xd, and the executables directory xd2015/bin need to be added to the PATH environment variable of each user. The environment variable XD_TCLDIR points to directory containing the XDGRAPH Tcl scripts,

For Mac users, dynamically linked OS-X 10 executables for the PowerPC and Intel processors are provided. These executables are linked to Mac system libraries (which should be present in all OS-X 10 environments), and also to the **gfortran** libraries. These latter libraries (and libraries for XDGRAPH) are provided in the download, and are placed in the same directory as the program executables. The environment variable DYLD_LIBRARY_PATH is essential, and points to this directory. So, for example, if you have installed XD2015 into /usr/xd2015_macm, then

```
export DYLD_LIBRARY_PATH=/usr/xd2015_macm/bin:$DYLD_LIBRARY_PATH
```

13.2 Installing XD2015 for *Windows™*

XD2015 is available for *MS Windows* (XP/Vista/7/8/10 – 32- or 64-bit versions). The installation is self-contained, and includes all executables, all required run-time libraries and all system files, with no dependencies on third-part libraries. 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.

Two versions are provided :

The **console version** works like a DOS program, driven by the command line, in a virtually identical manner to the Linux/Mac versions of the program. It is necessary to open a DOS/Command prompt window, change to the working directory using "cd" and issue text commands.

The **GUI version** is controlled by the *WinXD* GUI (which has its own separate manual). The XD system programs (XDLSM, XDPROP *etc*) have exactly the same functionality as their console versions, but the text output is directed to the text window of the GUI. Programs may be executed, and the parameter and master files edited, using standard *Windows™* drop-down menus and dialog boxes. In addition to the standard functionality of XD, the *WinXD* GUI has a number of extra useful graphical features.

13.2.1 Environment variables

In order for both *Windows™* versions to function, the following environment variables need to be set

XD_DATADIR	(points to directory containing the data bank files)
XD_TCLDIR	(points to directory containing the XDGRAPH Tcl scripts)
TCL_LIBRARY	(points to directory containing the TCL 8.3 system scripts)

Example values for these variables would be

XD_DATADIR=*<xddir>/lib/xd*
XD_TCLDIR=*<xddir>/lib/xdgraph*
TCL_LIBRARY=*<xddir>\bin\tcl8.3*

where *<xddir>* is the fullpath of the XD installation directory, *e.g.* "c:\xd2015".

The GUI version requires an additional environment variable PGFONT, to point to the fullpathname of the file grfont.dat, e.g.

PGFONT=*<xddir>\lib\xd\grfont.dat*

NOTE : the use of Unix style forward slashes "/" rather than DOS backslashes "\" as delimiters for the directory names with XD_DATADIR & XD_TCLDIR is essential. This is a result of porting a Unix program to Windows and is necessary for XDGRAPH to work correctly. The standard DOS backslash "\" should be used for the TCL_LIBRARY and PGFONT environment variables.

Finally, the console version requires that the directory *<xddir>/bin* is in the PATH, so that *Windows™* has access to all the executables.

13.3 External programs for XD2015

XD2015 writes a number of files (primarily graphical files) which require external, publically available programs to view. To access all the new features of XD2015, it is recommended that these programs are also installed. They are all available either as *Windows*™ or Linux executables.

MoleCoolQT - a program for visualising iso-surfaces and mapped iso-surfaces

http://www.molecoolqt.de/

An excellent program written by Christian B. Hübschle is now freely available for academic users. This produces iso-surface plots directly from the XD grid files or Gaussian format cube files output by XDPROP, and alleviates the deficiencies of the *Windows™* version of XDGRAPH.

Molekel version 4.3 - a program for visualising iso-surfaces and mapped iso-surfaces

http://molekel.software.informer.com/5.4/

This program reads the Gaussian style cube files written by XDPROP. It is important to use version 4.3, rather than more recent versions (5.1 or later), since the latter do not (as yet) allow plotting of mapped isosurfaces.

Ghostview - a program for visualising PostScript files

http://www.cs.wisc.edu/~ghost/

This program reads the PostScript files written by XDPROP.

POV-Ray - a ray-tracing program for high quality graphical rendering

http://www.povray.org/

This program reads the scene description files written by the WinXD GUI and the utility program rays2pov.

Plotmtv - a graphics program for drawing vector plots

http://rpmfind.net/linux/rpm2html/search.php?query=plotmtv

This program reads the MTVDAT files written by XDPROP and plots vector plots (as well as many other types actually). Linux rpm distributions and source code from this site.