

# Chapter 10

## TOPXD – Full Topological Analysis

### 10.1 Overview

TOPXD [1] has been developed in order to fully incorporate the Quantum Theory of Atoms in Molecules [2] (QTAM) into routine X-ray charge density studies. The program TOPOND98 [3], originally written for the CRYSTAL98 package [4] has been adapted for the experimental charge-density package XD. While the evaluation of several charge density properties at the critical points is already included in the XDPROP program, the TOPXD program provides several additional features. The main ones are its capability to define atomic basin boundaries and to integrate density functions within the basins, thus producing an extensive set of atomic properties, including net charges, dipole and higher electrostatic moments.

TOPXD allows the user to undertake :

- fully-automated chain-like searching for critical points in the  $\rho$  and  $\nabla^2\rho$  scalar fields, using either conventional Newton-Raphson techniques or the eigenvector following method [5,6]
- grid searching of critical points in the asymmetric unit
- evaluation of atomic properties
- finely tuned algorithms for the evaluation of atomic interaction lines 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  $h=5\times 10^{-3}$ , the expected error in the numerical derivatives is only  $O(h^4)=6.25\times 10^{-10}$ . Numerical examples show the actual error to be less than  $1\times 10^{-9}$  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  $1\times 10^{-9}$ . 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 Ångströms. 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/Ångströms**. 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 ( $\nabla\cdot\nabla$ ) of $\rho(\mathbf{r})$
$\mathbf{H}(\rho(\mathbf{r}))$	The Hessian (second derivatives) matrix of $\rho(\mathbf{r})$
$\lambda_k$	Eigenvalues ( $\lambda_1\leq\lambda_2\leq\lambda_3$ ) of $\mathbf{H}(\rho(\mathbf{r}))$
CP(s)	Critical Point(s) : a point $\mathbf{r}$ 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 $\mathbf{H}(\rho(\mathbf{r}))$ at the CP.
BCP	Bond Critical Point : a (3,-1) CP in the $\rho(\mathbf{r})$ scalar field
ZFS	Zero-Flux-Surface : $\nabla\rho(\mathbf{r})\cdot\mathbf{n}(\mathbf{r}) = 0 \quad \forall \mathbf{r} \in \text{surface}$
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 $\rho$ at a position other than nuclei
NEA	Non-Equivalent (unique) Atom
AIL	Atomic Interaction Line : a curved path joining two nuclei along which $\rho$ 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 <i>atomic</i> 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  $\kappa$  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*.

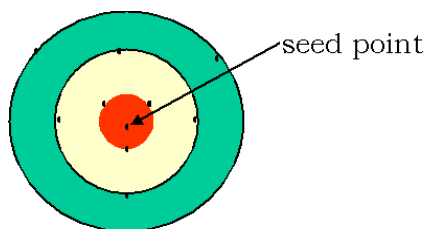
##### **(\*)ascii**

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 **nbp/bcp/rcp/ccp** for (3,-3), (3,-1), (3,+1) or (3,+3) critical points respectively. Use the **nbp** 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 $\rho(\mathbf{r})$ , $\nabla^2\rho(\mathbf{r})$ and $\lambda_3$ along the line joining two nuclei or two general points

Profiles of  $\rho(\mathbf{r})$ ,  $\nabla^2\rho(\mathbf{r})$  and  $\lambda_3$  are written to Fortran units 95, 96, 97, respectively. The units of  $\rho(\mathbf{r})$ ,  $\nabla^2\rho(\mathbf{r})$  and  $\lambda_3$  are determined by the keyword **(\*)au** (see Section 10.1).

#### **TRHO (\*)profile perstep n**

*<profile specification>* (Several of these may be given)

#### **(\*)profile**

If starred, the search is performed. Otherwise no search is undertaken.

#### **perstep n**

Determines the percentage step **s** along **A-B** or **a-b**

if  $n=1$  ,    $s = 0.01 \times R_{A-B}$  ( or  $R_{a-b}$  )

if  $n=100$ ,    $s = 1 \times R_{A-B}$  ( or  $R_{a-b}$  )

*<profile specification> On a new line ! It can be given in one of the following formats:*

#### **(\*)atom label toneighbor i1... i(n)**

Profile along the line(s) joining the unique atom **A** with label *label* and its *i1...i(n)* neighbour(s) (atom **B**), where *i* is the "NEW" number in the "Clusters around each of the unique atom" printing at the beginning of the TOPXD output. The profiling will only be performed if keyword **atom** is starred. Repeat this line **m** times for the **m** unique atoms to be considered.

#### **(\*)point [ car | fra ] x1 y1 z1 x2 y2 z2**

Profile along the line joining two points **a** and **b** with coordinates (*x1 y1 z1*) and (*x2 y2 z2*), respectively. The coordinates are in Cartesian (Å) (**car**) or fractional (**fra**) units. The profiling will only be performed if the keyword **point** is starred. Repeat this line **n** times for **n** point pairs.

## 10.4 Topological Analysis of the Laplacian of Electron Density

### 10.4.1 Auto critical point search within the concentration (or depletion) shells of unique atoms and/or non-nuclear attractors

Usually the search is performed for critical points in the valence shell charge concentration (VSCC) of each selected unique atom. Nevertheless, a suitable choice for the sphere radius (*rstar* parameter, see below) allows for a critical point search in (any of) the depletion shells of the selected unique atom.

#### **TLAP (\*)auto meth (\*)ail (\*)debug nstep nstep nnb nnb rmax rmax ntheta ntheta nphi nphi**

*<atom 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  $\theta$  and  $\phi$ , 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** *Phi* *InSph* *phi* *ThInSph* *theta* **\*SavSurf**

The *phi* and *theta* parameters define the angular integration parameters which are used INSIDE the  $\beta$ -sphere, *i.e.* the number of  $\phi$  (*phi*) and  $\theta$  (*theta*) grid points. **SavSurf**, when starred, enables TOPXD to write out the lengths and coordinates of integrations rays to file *rays.dat* for visualization in **VZ3D** (see section 10.6.1).

### 10.5.2 Alternative method for Zero Flux Surface (ZFS) search

Activating this set of (optional) instructions enables an alternative method for the ZFS search, based on algorithms described in references [10-11]. In addition, a second-order Runge-Kutta method is used when tracing the gradient paths instead of a Predictor-Corrector method. The advantage of this method is that it can significantly improve the speed of the ZFS search, but can result in a less accurate ZFS if incorrect parameters are specified.

**ATBP altguess** **bigstep** *bigstep* **accur** *accur* **maxrint** *maxrint* **rmax** *rmax* **step0** *step0* **A** *a* **B** *b*

*bigstep* (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  $\omega$  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  $\rho(\mathbf{r})$  bond critical points which lie on the  $\nabla\rho(\mathbf{r})$  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*  $\neq$  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* $\neq$ 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  $\beta$ -sphere: number of  $\varphi$  (*nphi*) and  $\theta$  (*nth*) grid points (see also FAQ section)

### **rad rad**

*rad* is the number of radial integration points inside  $\beta$ -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 nbcp [ car | atom ] }
```

{ . . . . if *nbc*p > 0 insert here *nbc*p 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 xmax xstep ymin ymax ystep**

**origin shift ishft origin xo yo zo vmod vmod**

**misc size ax scale name 'name' title 'title'**

The format is exactly as laid out above, i.e. each command starts a new line. See Chapter 12 for an example.

*point/atom A/B/C specification*

Three atoms or points (A,B,C) must be given to define the plot plane. The specification format is different for atoms and points:

1. Atoms can be specified using the following format

**atom inum itx ity itz**

*inum* is the serial number of atom in TOPXD printout of all atoms in unit cell

*itx ity itz* – translations applied to fractional coordinates of atom with number *inum* along X, Y and Z-crystal axis, respectively

2. General points can be specified using the following format:

**point [ car | fra ] x y z**

where *x y z* are fractional (**fra**) or Cartesian (Å) (**car**) coordinates

**NOTE:** If the evaluation of a molecular/crystal graph or, generally, of  $\nabla\rho(\mathbf{r})$  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 xmax xstep ymin ymax ystep**

The plot plane is **XY**. *xmin* and *xmax* define the minimum and maximum values along the plot X-axis, respectively, while *xstep* defines the grid interval along X. *ymin ymax* and *ystep* have the same meaning but for Y-axis of the plot. All these parameters must be given in Ångstroms.

**origin shift 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* 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 than 1 are also allowed.

**misc size ax scale scale name name title title**

*ax* = **a4** – plot size is A4

= **a3** – plot size is A3

*scale* – Plot scale (Å/cm)

*name*

Suffix name enclosed in single quotes " " (maximum 24 characters, no blanks and no special symbols like ‘,’,’^ etc) for the files which contain the values of each computed function *func* (see below) and the common information for the XYZ plot (system geometry, Euler rotation angles from crystal to plot frame, plot size, etc.). These files form the input for the **P2DCRY** section. Full names of the files are listed in **Table 10-1** below ("/" means character concatenation). Common information for the XY plot is saved in file: P2DCRYIN//*name*. The files prefixed with "SURF" may also be read from the SURFER program (a PC-DOS program for representing the scalar function in the plot plane as a 2D surface in 3D space, something that it is often referred to as a 3D graph).

*title* – Plot title enclosed in single quotes " " (maximum 80 symbols, blanks are allowed, no special symbols like ‘,’,’^ ).

**Table 10-1**

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

### 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.

<b>*rho</b>	electron density ( $\rho(\mathbf{r})$ )
<b>*lapp</b>	Laplacian of the electron density ( $\nabla^2\rho(\mathbf{r})$ )
<b>*lapm</b>	negative Laplacian of the electron density ( $-\nabla^2\rho(\mathbf{r})$ )
<b>*grho</b>	magnitude of the gradient of electron density ( $ \nabla\rho(\mathbf{r}) $ )
<b>*trajgrad</b>	$\nabla\rho(\mathbf{r})$ trajectories only
<b>*molgraph</b>	molecular/crystal graph (and atomic basin boundaries)
<b>*trajmolg</b>	$\nabla\rho(\mathbf{r})$ 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  $|\nabla\rho(\mathbf{r})|$  are cut at the value of *cutr* and/or values of  $\nabla^2\rho(\mathbf{r})$  and  $-\nabla^2\rho(\mathbf{r})$  are cut at  $\pm cutl$ , according to their sign (*cutr cutl* must be given electrons and Angstroms). Cutting of the scalar function is generally required, especially in the case of the Laplacian, for representing the function in the plot plane as a 3D graph.

{+function-dependent parameters in **Plot 2D** specific instructions }

For certain plotted functions, some extra instructions may be necessary. These extra instructions are outlined in Sections 10.6.2.1 - 10.6.2.3.

**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*-coordinate is less than *tol2* (Å).

If keyword **plane** is starred, the starting point of each  $\nabla\rho(\mathbf{r})$  trajectory segment (which are typically  $10^{-4}$  to  $10^{-2}$  Å long) is forced on the XY plane, while the default is the normal tracing of  $\nabla\rho(\mathbf{r})$  trajectories (with projection regulated by *tol2*).

The number of downhill  $\nabla\rho(\mathbf{r})$  trajectories is defined by variable *npath*. The recommended value for *npath* is 36.

Variable *nexttr* defines the number of other attractors, like non-nuclear (NNAs) attractors or the 2D attractors associated with a BCP, to be considered as origins of downhill  $\nabla\rho(\mathbf{r})$  trajectories. A bond critical point is seen as a (2,-2) attractor whenever its associated ZFS lies in the plot plane. If *nexttr* is greater than zero, add *nexttr* lines with Cartesian (Å) *x y z* coordinates of attractors of  $\nabla\rho(\mathbf{r})$  trajectories (one set of *x y z* coordinates per line)

### 10.6.2.2 *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  $\nabla\rho(\mathbf{r})$  trajectory segment is forced on the XY plane, while the default is the normal tracing of  $\nabla\rho(\mathbf{r})$  trajectories (with projection regulated by *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  $\nabla\rho(\mathbf{r})$  trajectories that originate at bond critical points and have as the initial direction the *K*-th eigenvector of Hessian of  $\rho(\mathbf{r})$  at BCPs, are traced out ( **\*trK** ) or skipped ( **trK** ). The *K*-th eigenvector is associated with the *K*-th eigenvalue of the Hessian  $\lambda_K$  (  $\lambda_1 \leq \lambda_2 \leq \lambda_3$  ).

### 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* **nexttr** *nexttr*

{ if *nexttr* is greater than zero, add *nexttr* 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  $\nabla\rho(\mathbf{r})$  trajectories is defined by variable *npath* (the recommended value for *npath* is 36).

Variable *nexttr* defines the number of other attractors, like non-nuclear (NNAs) attractors or the 2D attractors associated with a bond critical point, to be considered as origins of downhill  $\nabla\rho(\mathbf{r})$  trajectories. A bond critical point is seen as a (2,-2) attractor whenever its associated ZFS lies in the plot plane.

If *nexttr* is greater than zero, add *nexttr* lines with Cartesian (Å) *x y z* coordinates of attractors of  $\nabla\rho(\mathbf{r})$  trajectories (one set of *x y z* coordinates per line)

### 10.6.3 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** ( $\rho$ ), **lapp** ( $\nabla^2\rho$ ), **lapm** ( $-\nabla^2\rho$ ), **grho** ( $|\nabla\rho|$ ), **trajgrad** ( $\nabla\rho(\mathbf{r})$  trajectories), **molgraph** (molecular graph), **trajmolg** (molecular graph and  $\nabla\rho(\mathbf{r})$  trajectories), **rhoomolg** ( $\rho$  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:

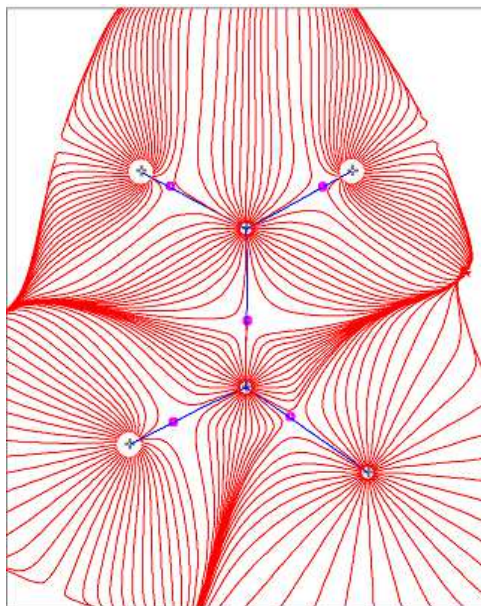
```
hp2xx PL2Dform -d 150 -p 218282 -c 276411 -f form.pcx -m pcx -r 270
```

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**  $\rho(r)$  trajectories and molecular graph of formamide molecule in the crystal created from TOPXD/P2DCRY2000 data with hp2xx. The  $V\rho(r)$  trajectories are shown with red lines, nuclear positions are marked with + (black), BCPs are marked with o (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 Ångströms).

*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 $\rho(\mathbf{r})$	<b>3DRHOO</b> // <i>name</i>	<b>3DRHOO</b> // <i>name</i> // .stf	<b>3DRHOO</b> // <i>name</i> // .grd
Laplacian $\nabla^2\rho(\mathbf{r})$	<b>3DLAPP</b> // <i>name</i>	<b>3DLAPP</b> // <i>name</i> // .stf	<b>3DLAPP</b> // <i>name</i> // .grd
Negative of the Laplacian $-\nabla^2\rho(\mathbf{r})$	<b>3DLAPM</b> // <i>name</i>	<b>3DLAPM</b> // <i>name</i> // .stf	<b>3DLAPM</b> // <i>name</i> // .grd
Magnitude of the gradient $\nabla\rho(\mathbf{r})$	<b>3DGRHO</b> // <i>name</i>	<b>3DGRHO</b> // <i>name</i> // .stf	<b>3DGRHO</b> // <i>name</i> // .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**

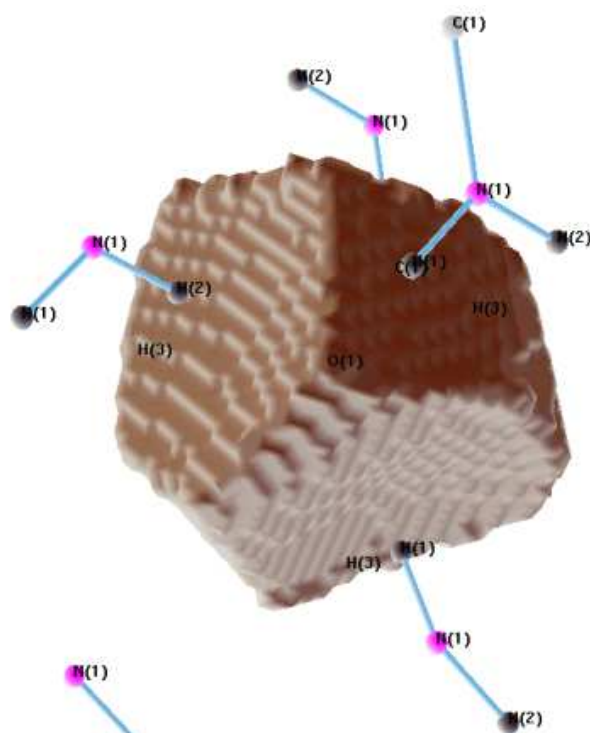
Definition of limits of the Cartesian grid (in Å) generated for visualization in XDGRAPH. If keyword **default** is starred, these limits will be automatically determined based on the data read from all files specified in **files** directive.

**grid (\*)default dx dx dy dy dz dz rvec (\*)default rvec**

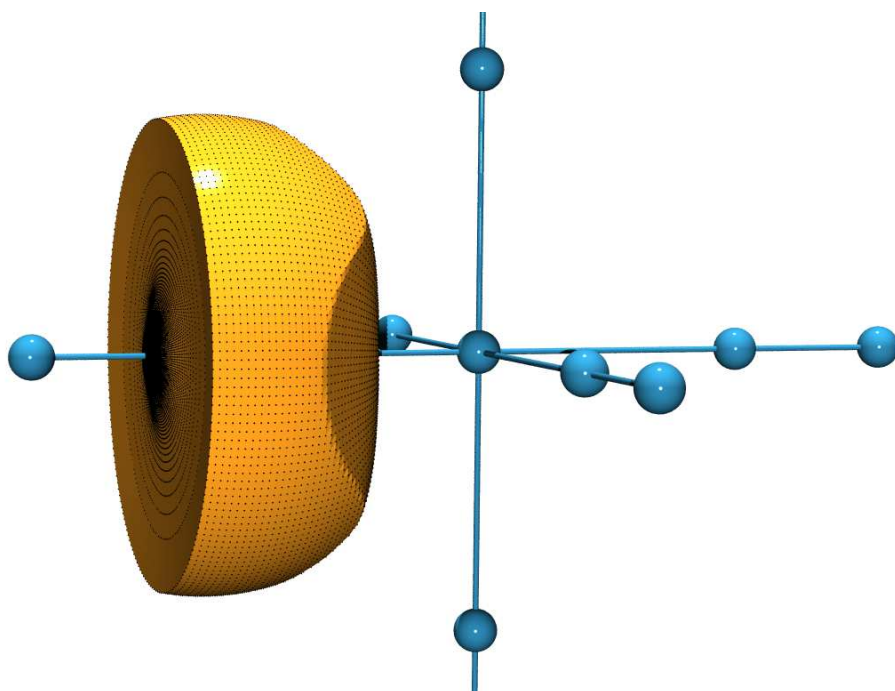
Definition of the grid spacings *dx* *dy* and *dz* of the Cartesian grid (in Å). If keyword **default** is starred, the default value of 0.1 Å will be used for *dx* *dy* and *dz*. The parameter *rvec* specifies the radius of the sphere (in Å) centered on each grid point. If the boundary of the atomic 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 program 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)<sub>6</sub> 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 $\Omega$ 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} =  \mathbf{r}_{\Omega} $
Atomic Forces		
Atomic force components Force on nucleus of atom $\Omega$ by the electron density of atom $\Omega$	FAXA FAYA FAZA	$FAXA(\Omega) = \int_{\Omega} (Z_{\Omega} / r_{\Omega}^3) x_{\Omega} \rho(\mathbf{r}) d\tau$ $FAYA(\Omega) = \int_{\Omega} (Z_{\Omega} / r_{\Omega}^3) y_{\Omega} \rho(\mathbf{r}) d\tau$ $FAZA(\Omega) = \int_{\Omega} (Z_{\Omega} / r_{\Omega}^3) z_{\Omega} \rho(\mathbf{r}) d\tau$ $x_{\Omega}, y_{\Omega}, z_{\Omega}$ , xyz comp. of $\mathbf{r}_{\Omega}$
'Radial' Atomic expectation values		
Atomic expectation value of the operator $r_{\Omega}^n$	R(-1) R(+1) R(+2) R(+3) R(+4)	$R^N(\Omega) = \int_{\Omega} r_{\Omega}^n \rho(\mathbf{r}) d\tau$
Atomic expectation values of $r_{\Omega}^n$ 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 $-3N(\Omega)$	GR(-1) GR( 0) GR( 1) GR( 2)	$GR^N(\Omega) = \int_{\Omega} r_{\Omega}^n \mathbf{r}_{\Omega} \bullet \nabla \rho(\mathbf{r}) d\tau.$
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	$V002(\Omega) = \int_{\Omega} d\tau_{0.002}$
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) =  \mathbf{DM}(\Omega) $ where $\mathbf{DM}(\Omega)$ is the atomic dipole vector
Components of atomic displacement vector	DCX DCY DCZ	$DCX = -DX/N(\Omega)$ $DCY = -DY/N(\Omega)$ $DCZ = -DZ/N(\Omega)$
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 2 <sup>nd</sup> 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 2 <sup>nd</sup> moment tensor		
Atomic 3 <sup>rd</sup> moment tensor components		
Atomic 4 <sup>th</sup> moment tensor components		
Atomic Shannon information entropy		
Atomic information (missing information function). This is the integral of $\rho' \ln \rho'$ where $\rho'$ 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  $\rho(\mathbf{r})$  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 ( $\phi\times\theta\times$ radial) for non-H atoms

32×24×96 ( $\phi\times\theta\times$ radial) for hydrogen atoms (if not involved in H-bond)

48×32×96 ( $\phi\times\theta\times$ radial) for hydrogen atoms (if involved in H-bond)

Note that the number of angular points and of radial points refers to the integration within beta sphere and outside beta sphere, respectively.

The computational time is roughly proportional to  $n\phi \times n\theta$ . The number of radial points is very important for the precision, but hardly affects the total integration time, as it is operative only in the integration step and NOT in the ZFS determination (which takes about 95% of the total time).

**Q: Integration of an atomic basin takes a very long time. What options do we have to speed up the calculation ?**

**A:** Unfortunately the integration step is very 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( $\Omega$ ), L ( $\Omega$ ), 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  $5\times 10^{-5}$  for H atoms, possibly around  $1\times 10^{-5}$ . A value of  $1\times 10^{-4}$  could be perhaps acceptable, but not too precise.
2. for second row atoms (C,N,O, etc) it should not exceed  $1\times 10^{-3}$ . Possibly  $1\times 10^{-4}$

**Q: You've mentioned that the computing time increases by a factor of  $\phi\times\theta$  planes, but how does the *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  $3 \times nvi \times$

*star\_multiplicity* coordinates of the possible  $\nabla p(\mathbf{r})$  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  $1 \times 10^{-3}$  despite the fact that I use *accur*=0.001 and *n $\phi$*  $\times$ *n $\theta$*  as large as 96 $\times$ 64. I remember that you have mentioned that decreasing the number of points might help, but when I reduce these numbers to 48 $\times$ 32 or 64 $\times$ 48 it still does not help. These problems usually occur with carbon and nitrogen atoms, never with oxygens or hydrogens. What do I do?**

**A:** What about electroneutrality ? Are you very far from it ? The fact that one may get problems with carbon or nitrogen atoms and never with oxygen or hydrogen atoms seems to indicate that the former have more complicated ZFSs than the latter (at least in the systems you are presently investigating). You could try to solve such a problem by increasing the radius of the beta sphere for such atoms, thus reducing the size of the remaining part of the atomic basin. You could use for the beta sphere something like the distance of the closest bond critical point multiplied by 1.15 (the program then reduces this number by 20%). Furthermore, the increase (inside the code) of the number of theta and phi points in the inner beta sphere might help. Please contact us and we will send you instructions on how to do it...

## Bibliography

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*; Clarendon Press: Oxford, 1990.
3. C. Gatti, *TOPOND98 User's Manual*; CNR-CSR SRC, 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/>