# PEANUT: Computer graphics program to represent atomic displacement parameters

**W. Hummel, J. Hauser and H.-B. Bürgi**

*Laboratory of Chemical and Mineralogical Crystallography, University of Bern, Bern, Switzerland*

*PEANUT is an easy to use computer graphics program for the visualization and real-time manipulation of the atomic displacement parameters of small molecules. A flexible, dynamic data structure allows the user to compute complicated, nonspherical atomic surfaces and to handle the point group symmetry of the molecules automatically. Pictures with hidden-line removal may be plotted in publication quality on appropriate output devices.*

*Keywords: atomic displacement parameters, interactive computer graphics*

## INTRODUCTION

Atomic anisotropic displacement parameters (ADPs) account for atomic motion and disorder in crystals.[1] They are determined routinely together with atomic positional parameters during single-crystal structure analysis by diffraction methods. The publication of ADPs has become a rare event. If all goes well, ADPs are deposited as supplementary material; more often they are irretrievably lost. Even an ORTEP drawing[2] in which the ADPs are represented in terms of the well known ellipsoids (Figure 1a) is no longer a standard feature of communications reporting crystal structure data. The general neglect of ADPs is certainly one reason that they have had little impact on our understanding of atomic and molecular motion in crystals.[1] Another reason may be the lack of an appropriate computer graphics tool to visualize and manipulate in real time observed ADPs, ADPs calculated from models of motion or differences between the two. The program PEANUT, to be described in this communication, was developed with the aim of filling this gap. The use of program PEANUT for studying those aspects of molecular motion expressed in observed ADPs is described elsewhere.[3]

Address reprint requests to Dr. W. Hummel at his present address: Paul Scherrer Institute, CH-5232 Villigen PSI, Switzerland.

Received 21 February 1990; accepted 5 June 1990

## REPRESENTATION OF ADPs BY SURFACES

ADPs given in the form of atomic mean square displacement (MSD) tensors **U** may be visualized in several ways. One of them is to represent MSD $u^2$ in an arbitrary direction defined by the unit vector **n**

$$u^2(\mathbf{n}) = \mathbf{n}^T\mathbf{U}\mathbf{n} \quad (1)$$

From Equation 1 analytical formulas for two surfaces representing different aspects of ADPs may be derived.[4] For simplicity these formulas are referred to a Cartesian coordinate system whose axes $x_i$ point along the principal directions of **U** with principal values $u_i^2$.

The surface whose radius $|\mathbf{r}|$ in any direction equals the MSD in that direction is given by

$$(u_1^2 \cdot x_1^2 + u_2^2 \cdot x_2^2 + u_3^2 \cdot x_3^2)^2 = (x_1^2 + x_2^2 + x_3^2)^3 \quad (2)$$

Equation 2 describes a sixth-order surface known as an ovaloid (Figure 1b). The surface whose radius $|\mathbf{r}|$ equals the corresponding root mean square displacement (RMSD) is given by

$$(u_1^2 \cdot x_1^2 + u_2^2 \cdot x_2^2 + u_3^2 \cdot x_3^2) = (x_1^2 + x_2^2 + x_3^2)^2 \quad (3)$$

This is a fourth-order, or quartic, surface (Figure 1c).

The tensor **U** may also be interpreted as the variance–covariance matrix of an atomic Gaussian probability density distribution.[2] Surfaces of constant probability density take the form

$$x_1^2/u_1^2 + x_2^2/u_2^2 + x_3^2/u_3^2 = C^2 \quad (4)$$

Equation 4 defines the second-order, or ellipsoidal, surface often found in reports of crystal structure determinations (Figure 1a). If $C$ is unity, the probability of finding the atom inside the ellipsoid is 20% and the radii of the ellipsoidal surface along the principal axes $x_i$ (and only along them) are equal to the RMSDs in these directions (Figures 1a and 1c).

All three principal values $u_i^2$ have to be positive for Equation 4 to represent a closed surface. In contrast, MSD or RMSD surfaces are closed even if one or more of the principal values are negative. This is an important difference because it opens up the possibility of displaying *difference* MSDs or RMSDs as closed surfaces (Figures 1d and 2).

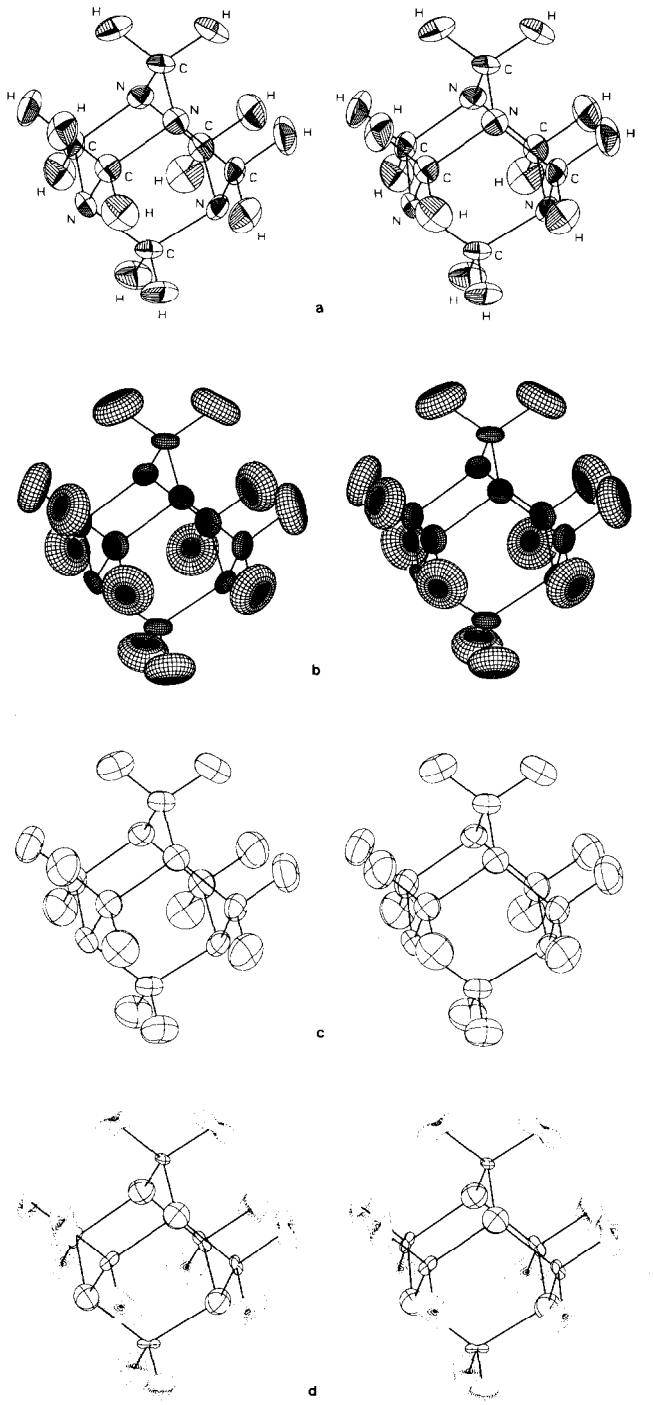

*Figure 1. ADP surfaces of hexamethylenetetramine.*[16] *The molecule has $\bar{4}3m$ ($T_d$) point symmetry. The site symmetry is 3m ($C_{3v}$) for N, mm ($C_{2v}$) for C and m ($C_s$) for H: (a) ADP(observed, neutron data), 20% probability ellipsoids shown as outlines with octant shading; (b) ADP(observed, neutron data), MSD surfaces shown as grid surfaces (scale = 4); (c) ADP(observed, neutron data), RMSD surfaces (scale = 1, which corresponds to axis lengths of 20% probability ellipsoids) shown as outlines; (d) ADP(observed, neutron data) − ADP(model, external and internal vibrations,*[16] *difference MSD surfaces (scale = 90), positive parts of the surfaces shown as outlines, negative parts shown as dot surfaces*

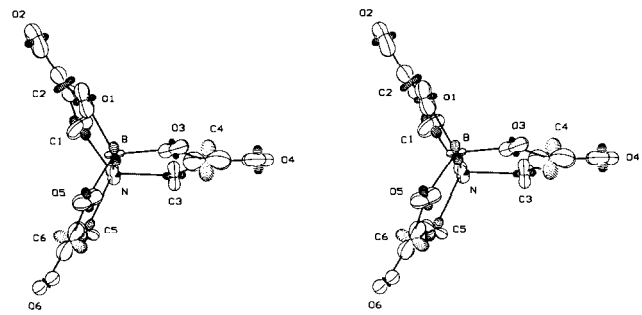

*Figure 2. RMSD difference surfaces with positive parts (outlines) and negative parts (dot surfaces) of comparable magnitudes; boron nitrilotriacetate,*[3] *scale = 12.5, ADP(spherical atom refinement) − ADP(multipole refinement)*

Initial experiences with PEANUT show[3] that the option to display difference surfaces, e.g., ADP(observed) − ADP(model), is very useful during the iterative process of constructing models of molecular motion from ADPs.

For actual computations the Cartesian coordinate system defined above is centered at the corresponding atomic positions. Points on the surfaces defined by Equations 2–4 are calculated using spherical coordinates $\varphi$, $\Theta$ and $r$. For example, the radius $r$ of the MSD surface (2) in the direction given by $\varphi$ and $\Theta$ is

$$r(\varphi, \Theta) = u_1^2 \cdot \sin^2\Theta \cdot \cos^2\varphi + u_2^2 \cdot \sin^2\Theta \cdot \sin^2\varphi + u_3^2 \cdot \cos^2\Theta \quad (5)$$

The coordinates of the corresponding surface point are simply:

$$\begin{aligned} x &= r(\varphi, \Theta) \cdot \sin\Theta \cdot \cos\varphi \\ y &= r(\varphi, \Theta) \cdot \sin\Theta \cdot \sin\varphi \\ z &= r(\varphi, \Theta) \cdot \cos\Theta \end{aligned} \quad (6)$$

Equations 6 also serve to calculate lines of constant altitude $\Theta$ or of constant latitude $\varphi$ in a straightforward way (Figure 1b). If the $u_i^2$'s have mixed signs, the eigenvalue with the unique sign is called $u_3^2$. If all of the principal values $u_i^2$ have the same sign, $u_3^2$ is defined such that $|u_1^2 - u_2^2|$ is the smallest difference between principal values.

## PRELIMINARY CONSIDERATIONS IN THE DESIGN OF PEANUT

A number of goals were formulated in the initial planning of PEANUT. Some of them are given below with the measures taken to achieve them.

It was assumed that the design and development of PEANUT would be an iterative process. To ensure as much flexibility as possible throughout this process, data structures and algorithms of PEANUT make extensive use of pointers, recursion and dynamic memory allocation. All code is written in C,[5] which is well suited for this purpose. As indicated by Equations 2–6 the calculations necessary

to obtain the ADP surfaces are time consuming. A fast graphics computer is therefore necessary to achieve real-time manipulations of the picture. PEANUT has been developed on a Silicon Graphics IRIS3130 workstation using the SGI graphics library and adapted to the IRIS-4D and Personal IRIS workstation. Systematic use of the symmetry of molecules and of graphical objects helps to minimize the amount of computations. Apart from considerations of flexibility and speed, emphasis was placed on user-friendly graphics tools. PEANUT is menu driven; the user may learn to handle the program by playing with a limited number of tools that are interactively selected from a pop-up menu using the mouse cursor. It is expected that an increasing number of crystal structure refinement programs will use the Standard Crystallographic File Structure-87 as one of their output options.[6] Input to PEANUT is therefore based on this structure. Finally, the results of using PEANUT should be available as pictures of publication quality. To this end routines handling hidden-line removal and direct access to a plotter or printer have been added to the package.

## COORDINATE SYSTEMS AND TRANSFORMATIONS

PEANUT is based on a hierarchy of coordinate systems (Figure 3). This allows one to carry out each computation in the coordinate system in which it is simplest. For each symmetry independent atom a local coordinate system is defined by the eigenvectors of **U** (object coordinates III, axes lengths 1 Å$^2$ for MSD surfaces and 1 Å for RMSD and probability ellipsoid surfaces). This brings into prominence the symmetry of the individual surfaces, which is mmm ($D_{2h}$) in all cases (Figure 1). Coordinates of surface points are calculated explicitly for one octant only; those in the other seven octants are obtained through a straightforward process of changing signs. The points representing a surface are transformed to fractional crystal coordinates (object coordinates II) using the transformation matrix "eigmat." The rotational part of the 4×4 transformation matrix is obtained from the matrix of eigenvectors of **U**,[8] expressed in object coordinates II and normalized to unit length; the

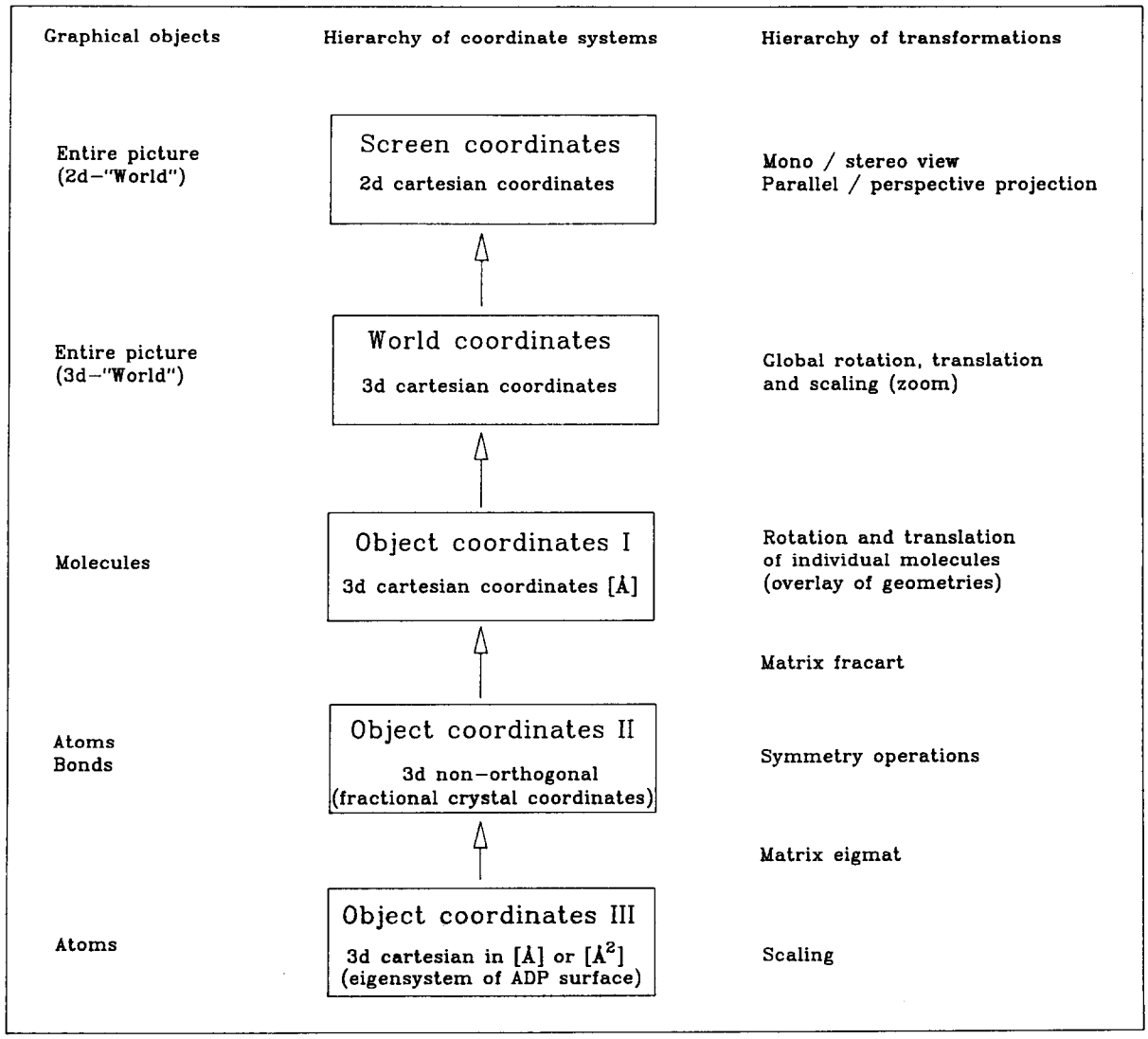


*Figure 3. PEANUT hierarchy of coordinate systems and transformations*

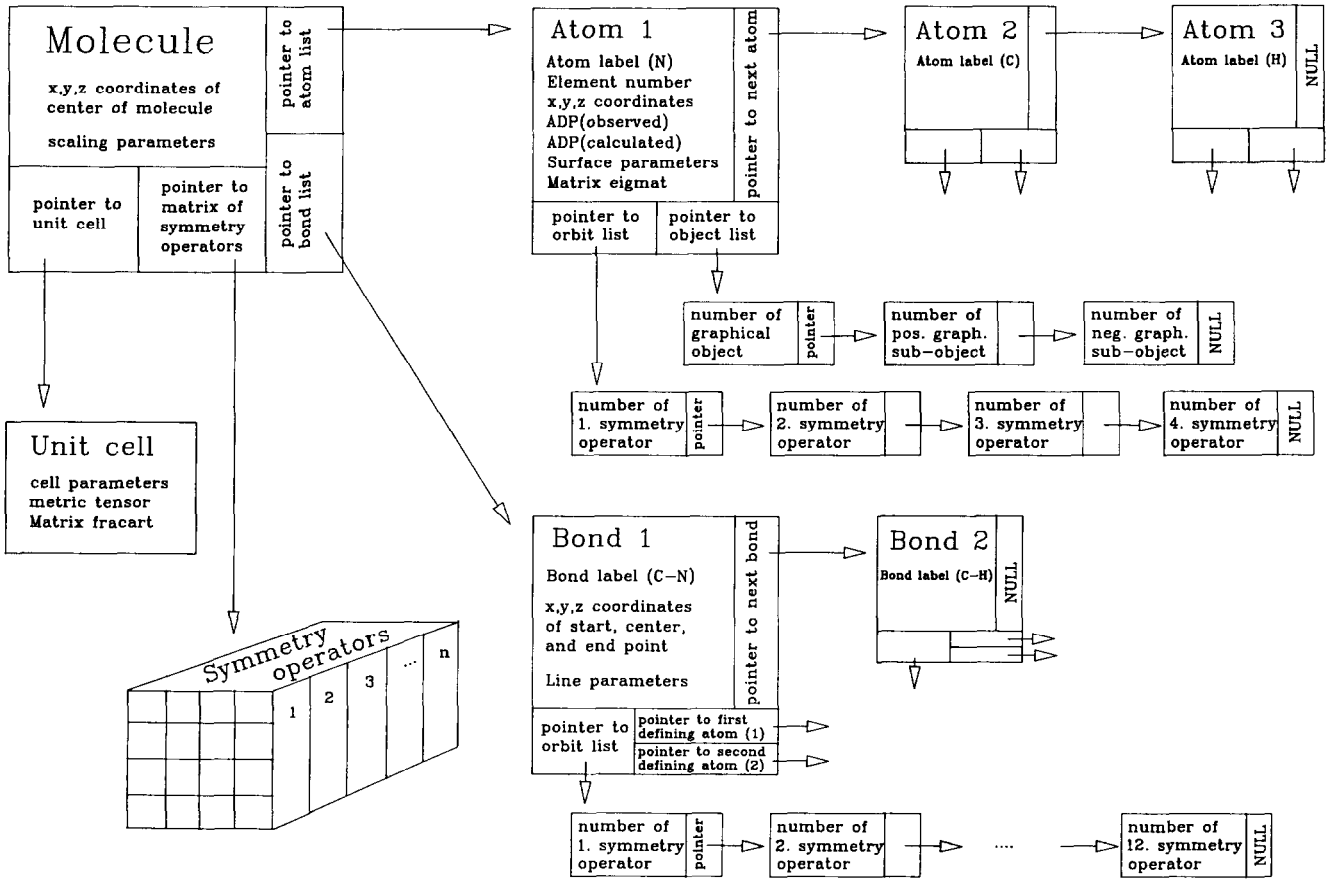


*Figure 4. PEANUT data structure. The example given corresponds to the data structure of hexamethylenetetramine (Figure 1d)*

translational part is obtained from the coordinates of the center of the atom. In fractional coordinates it is easy to perform crystallographic symmetry operations to produce all symmetry related atoms. The concept of crystallographic orbits[9] is implemented in this part of PEANUT to avoid duplications of atoms in special positions.

Bonds or, more generally, lines connecting two atoms are computed in object coordinates II. Again, as in the case of atoms, each symmetry independent line is transformed by the appropriate orbit of symmetry operations to create all symmetry related bonds without duplications.

All parts of a molecule, i.e., ADP surfaces and bonds, including all symmetry equivalent ones, are transformed from object coordinates II to a Cartesian coordinate system with axis lengths of 1 Å (object coordinates I) using the transformation matrix "fracart." The Cartesian $x$-axis (object coordinates I) is chosen to be parallel to the crystallographic $a$-axis (object coordinates II), the $y$-axis lies in the $a$-$b$ plane and the $z$-axis is perpendicular to it.[10] Object coordinates I are convenient for calculating global rotations and translations for the entire molecule. The design of PEANUT allows for the possibility of rotating and translating different molecules individually in object coordinates I, although this option is not implemented in the present version.

The transformations from object coordinates I to higher levels (Figure 3) affect all parts of a picture (the World). They include global rotation, translation, scaling and the projection necessary to generate a picture on the screen.

Once calculated a graphical object is transformed successively from level to level (i.e., from bottom to top in Figure 3) until the screen coordinates are reached. In practice a composite transformation matrix is built up from top to bottom using a stack of transformation matrices representing the different levels of coordinate systems. This matrix transforms the set of points of a given object directly from object coordinates III to screen coordinates.

## DATA STRUCTURE

The data structure of PEANUT has been designed with two goals in mind. First, it should be dynamic, i.e., free of any formal limits, such as maximum number of atoms and bonds or limits in the complexity of the ADP surfaces. Second, it should reflect the hierarchy of coordinate systems and transformations discussed above.[7]

Figure 4 summarizes the essential aspects of the data structure. It consists of a number of lists which are interconnected by pointers. The World (in world coordinates) of PEANUT consists of a linked list of molecules. Each molecule (in object coordinates I) is built up from linked lists of atoms and bonds (in object coordinates II). To each

atom a linked list of graphical objects is connected that defines an ADP representational surface (in object coordinates III). Each member of a linked list is built up in the same way and contains one or more of the following: some specific numerical information, a pointer to the next member of the list and a number of pointers to other lists containing further information.

The linked list of molecules (Figure 4, top left) has only one member at present with information characteristic for the entire molecule. It has pointers to the unit cell information, to an array of molecular symmetry operators, to the atom and to the bond list. The linked list of atoms (Figure 4, top right) has as members all symmetry independent atoms of the molecule. The entry for a member contains all information characterizing the atom as a whole, a pointer to the next atom and pointers to the crystallographic orbit and graphical object lists of this atom. The linked list of graphical objects (object list in Figure 4) is necessary to handle complex surfaces. In particular a difference surface, e.g., ADP(observed) − ADP(model), may consist of a positive part (graphical subobject 1) and a negative part (graphical subobject 2) with different colors or different graphical representations. The members of the orbit list contain a number that is interpreted as an index of the array of symmetry operators. With the help of this list all symmetry equivalent atoms in the molecule are generated exactly once. The symmetry operators of a molecule are stored in a three-dimensional array containing up to 48 4×4 point group symmetry operator matrices in object coordinates II (Figure 4). In a future version of PEANUT the orbit list may be extended to handle space group symmetry in addition to point group symmetry. The bond list is built in analogy to the atom list (Figure 4, bottom right). Additional lists analogous to the atom and bond lists may easily be introduced into the data structure, e.g., a triangulation list to represent coordination polyhedra or a chemical substituent list to enable changing the conformation of a molecule.

In principle the data structure of PEANUT may contain an unlimited number of molecules, each molecule consisting of an unlimited number of atoms and bonds; no limits are imposed on the number of graphical subobjects or the number of points and lines representing a surface. In practice limits are imposed by the amount of computer memory available and by the speed of the machine. Presumably both limits will continuously expand at a rate that will be determined primarily by developments in hardware technology. At present, ADPs of molecules with up to about 50 atoms may be handled reasonably in real time.

## THE TOOLBOX

To use PEANUT it is not necessary to know anything about the hierarchy of coordinate systems and transformations or to understand how the dynamic data structure is built up and processed. All the user will see is a pop-up menu that gives access to a toolbox containing options to manipulate the picture on the screen. The toolbox is a collection of operations on the data structure that may be grouped together according to the level at which they operate in the hierarchy of the data structure (Figure 5).

Operations on atoms include the selection of an ADP data set; available options are ADP(observed) and ADP(model) or, more generally, ADP(data set 1) and ADP(data set 2), and the difference between the two. Any of these ADP data sets may be displayed as a probability ellipsoid, a RMSD or MSD surface. Any of these surfaces may be shown as principal axes only, as outlines which are curves in the three planes defined by pairs of principal axes, as a grid surface built of lines of constant latitude and lines of constant altitude, or as a solid surface. The resolution of the surfaces, i.e., the number of straight lines that approximate the curves defining the surface representations may be varied. The ADP surfaces may also be scaled. In the case of ellipsoids the scale determines surfaces of a certain constant probability.

PEANUT includes an automatic bond searching routine. After reading new data, all intramolecular distances are checked and a bond is created if the distance between two atoms is smaller than 1.2 times the sum of their covalent radii.[11] This routine works well for most organic and organometallic molecules but cannot handle properly all inorganic molecules and structures. To get proper connectivities the user may add new bonds by selecting two atoms or delete unwanted lines by selecting them with the mouse cursor on screen.

| | | |
|---|---|---|
| ATOMS: | ADP data set: | observed<br>model<br>difference |
| | surface type: | ellipsoid<br>RMSD<br>MSD |
| | representation: | axes<br>outlines<br>grid<br>solid |
| | resolution | |
| | scale | |
| | labels: | on / off |
| BONDS: | add bond | |
| | delete bond | |
| PICTURE: | rotate: | x − y<br>z |
| | translate: | x − y |
| | zoom | |
| | mono | |
| | stereo: | normal<br>crossed |
| INPUT / OUTPUT: | read | |
| | hide | |
| | plot | |

*Figure 5. PEANUT toolbox*

Operations that affect the entire picture (the World of PEANUT) are global rotations and translations controlled by the mouse cursor. They work in real time even for complex ADP surfaces of small molecules at high resolution. In addition, the picture can be varied in size by zooming. Two stereo modes are included in PEANUT. The stereo option "normal" is for those used to stereo glasses, whereas the option "crossed" may be used by people who are able to see the left picture on screen with their right eye and vice versa without stereo glasses.

The read option inputs a data file written in SCFS-87 format.[6] It essentially contains unit cell data, positional parameters and ADPs for all symmetry independent atoms and symmetry operators. The program processes the selected file and creates a default picture of the molecule.

The picture may be plotted directly on paper using a plotter or a printer. At present, two drivers are available for PEANUT, one of them being selected during the installation of the program. One driver writes an ASCII file in HPGL format[12] for plotters; the other produces a Postscript file[13] for laser printers. To produce pictures of publication quality a hidden-line elimination algorithm was implemented in PEANUT and is discussed in the next section.

## HIDDEN-LINE REMOVAL

While it is sufficient to work with pictures of transparent objects on screen (hidden lines not removed), the same pictures plotted on paper may look rather confusing. Thus ADP surfaces are treated as opaque objects for plotting. The task of deciding which parts of the opaque objects should be shown and which parts should be omitted is known as the "hidden-line problem." Detailed discussions of the basic concepts of hidden-line removal algorithms are given in References 14 and 15.

The hidden-line algorithm implemented in PEANUT is an object-space algorithm.[14] In contrast to image-space algorithms,[14] where calculations are geared to the resolution of the output device (usually the screen), calculations in object space are performed to the precision available in the computer. A picture computed in object space is largely independent of magnification and of the resolution of the output device. Within program PEANUT a picture is composed of a number of graphical objects, each defined as a set of vertices and a set of connections between them. This allows one to build lines, areas and volumes. In principle, each (straight) line of a picture has to be tested against all areas of the picture to distinguish visible from invisible parts of the line. To keep the number of tests to a minimum, the algorithm uses the basic principles of sorting and object coherence.[14] PEANUT produces closed ADP surfaces only, which therefore may be treated as coherent graphical objects. The hidden-line algorithm constructs linked lists of ADP surfaces and of bonds sorted by depth ($z$), which are tested against each other only if they are overlapping in the $x$-$y$ plane. A further substantial reduction in the number of tests is achieved because the final plotting of ADP surfaces is usually done with high resolution to get smooth curves. Thus it is sufficient to test the vertices of a surface for visibility and to draw the (very small) straight lines connecting two vertices only if both defining vertices are "visible." Intersections of lines with areas are computed explicitly for bonds only.

A special problem arises in the calculation of $x$-$y$ boundary curves for ADP surfaces. The boundary curve is the edge of the shadow thrown by an (opaque) ADP surface if illuminated by parallel light along $z$. Because RMSD and MSD surfaces are fourth-order (Equation 3) and sixth-order (Equation 2) surfaces, respectively, no direct algebraic solution for the determination of their boundary curves (or boundary polygons) exists. Thus the boundary polygon vertices of an atom are computed numerically using a ternary tree search algorithm. First the atom is translated to the origin of the world coordinate system. A plane containing the $z$-axis is specified. In this plane three vectors at 45°, 90° and 135° from the $z$-axis are chosen, the surface points on these vectors are found and their distances from the $z$-axis are calculated. The angle $\alpha$ associated with the maximum distance is the starting point for the next step. Two vectors at $\alpha \pm 15°$ are specified, the surface points and their distances from the $z$-axis are calculated and the angle associated with the maximum distance is selected. The process is continued with successively smaller angles about $\alpha$: $\pm 5°$, $\pm 2°$ and $\pm 1°$. Only 11 surface points have to be calculated to determine the vertex of the boundary polygon in a given plane with an accuracy of $\pm 0.5°$. To locate the remaining vertices of the boundary polygon, the ternary tree search algorithm is repeated for other planes rotated about the $z$-axis. The number of vertices to be calculated, i.e., the number of planes to be searched, depends on the desired resolution. Finally the boundary polygon is translated back to the atomic position.

## ACKNOWLEDGEMENTS

We gratefully acknowledge financial support by the Schweizerischer Nationalfonds zur Förderung der wissenschaftlichen Forschung. We thank Mr. M. Knoblauch and Dr. M. Waldherr-Teschner, Dept. of Physical Chemistry, TH Darmstadt, Germany for the generous permission to use some of their modules from program MOLCAD.[17]

## REFERENCES

1 For a review see Dunitz, J.D., Schomaker, V. and Trueblood, K.N. Interpretation of atomic displacement parameters from diffraction studies of crystals. *J. Phys. Chem.* 1988, **92**, 856–67

2 Johnson, C.K. *ORTEP II: A FORTRAN thermal ellipsoid plot program for crystal structure illustrations* Oak Ridge Natl. Lab., report ORNL-5138 (1976) Oak Ridge, TN, USA

3 Hummel, W., Raselli, A. and Bürgi, H.B. Analysis of atomic displacement parameters and molecular motion in crystals., *Acta Crystallogr., Sect. B* 1990, **46**, in press

4 Nelmes, R.J. Representational surfaces for thermal motion. *Acta Crystallogr., Sect. A* 1969, **25**, 523–526

5 Kernighan, B.W. and Ritchie, D.M. *The C Programming Language* Prentice–Hall, Englewood Cliffs, (1978)

6 Brown, I.D. The standard crystallographic file structure. *Acta Crystallogr., Sect. A* 1983, **39**, 216–224

7 Foley, J.D. and VanDam, A. *Fundamentals of Inter-*

*active Computer Graphics* Addison–Wesley, Reading (1984)
8 Sands, D.E. *Vectors and Tensors in Crystallography* Addison–Wesley, Reading (1982) 72–76
9 Wondratschek, H. in *International Tables for Crystallography, Vol. A.* (Th. Hahn, Ed.) Reidel, Dordrecht (1983)
10 Sands, D.E. *op. cit.*, 58
11 *Table of Periodic Properties of the Elements.* Sargent-Welch Scientific Co., Skokie (1968)
12 *Interfacing and programming manual HP7475A graphics plotter.* Hewlett Packard, part no. 07475-90001 (1984)
13 *Postscript Language Reference Manual.* Addison–Wesley, Reading (1987)
14 Sutherland, I.E., Sproull, R.F. and Schumacker, R.A. A characterization of ten hidden-surface algorithms. *Computing Surveys* 1974, **6**, 1–55
15 Rogers, D.F. *Procedural Elements for Computer Graphics* McGraw-Hill, New York (1985)
16 Willis, B.T.M. and Howard, J.A.K. Do the ellipsoids of thermal vibration mean anything? Analysis of neutron diffraction measurements on hexamethylene-tetramine. *Acta Crystallogr., Sect. A* 1975, **31**, 514–22
17 Brickmann, J., Knoblauch, M. and Waldherr-Teschner, M. *MOLCAD IRIS-4D/xx Manual* Dept. of Phys. Chem., TH-Darmstadt (1988)