3-D Visualization of Cavities and Channels in Crystals

For more information contact **James L. Dye** , Dept. Of Chemistry
Michigan State University, East Lansing, MI 48824-1322
FAX: (517) 353-1793, Phone: (517) 355-9715 Ext.288
E-mail: dye@cemvax.cem.msu.edu

Program Development by

Tibor Nagy

1 Objectives

Many programs exist that permit one to visualize crystal structures by ball-and-stick models, ORTEP drawings, space-filling representations. Of importance to the understanding of the behavior of electrides is the geometry of the cavities and channels within the structure. Such information is also useful for zeolites, host-guest complexes. Although commercial programs permit construction of "solvent accessible surfaces", in which the sites that are accessible to spheres of a specified radius are displayed, there was, to our knowledge, no general way to display cavities and channels of arbitrary size, nor to easily determine the diameter of both cavities and channels with one program. The combination of Fortran programs (collectively referred to as the TiborChan programs) with a commercial isosurface 3D display program (AVS or Explorer) permits manipulation of 3D representations of the "empty spaces" in crystals and provides pictures of channels and cavities from any desired perspective.

2 Overview of the Method

To run the program, one needs the cartesian coordinates of the atoms in a large enough array to clearly show the cavities and channels. This involves the determination of the coordinates of the atoms in a number of unit cells. For triclinic space groups with one asymmetric unit per unit cell this might require 27 unit cells (3 x 3 x 3), while structures with z = 4 may require only 8 unit cells. The collection should be chosen to be as nearly cubic as possible. That is, one should avoid a collection of unit cells in which the overall dimensions are very different. Currently, we use the commercial program Insight-II to construct the coordinates. The output is in the Brookhaven Data Base (.bkv) format. Since one only needs to provide the space group and the internal coordinates of the atoms (based on the unit cell vectors) that constitute the asymmetric unit, input is easy. The data may be input from a crystal structure base or entered by hand. Presumably, the atom coordinates obtained from structure simulation programs would also be satisfactory.

The van der Waals radii of the atoms are available to the Fortran program in the file *atom.rad*, which is easily modified by the user. If the "standard" radii are satisfactory, the program *xeo* for "negative" displays of cavities and channels or the program *zeo* for the "positive" displays of atomic or molecular surfaces only needs the *.bkv* file for a given structure.

The "grid-point" programs described in detail below are rather "course-grained", typically 40 x 40 x 40 points so that the details of surfaces are not represented accurately. Thus, the output of *xeo* or *zeo*, when displayed as an isosurface, is not too useful, but is similar to the 1D equivalent of a coarse plot of points connected by straight line segments. To circumvent this difficulty, the program *smooth* is used, usually with two successive smoothing operations to provide a smoothed (but not completely accurate) picture of the surface.

The Fortran program assigns a number to each grid point from 0 to 255 depending on the distance of the point from the nearest van der Waals surface. The value 0 is assigned if the grid point falls within a van der Waals sphere. The value 255 corresponds to the maximum distance found in the crystal structure. Care must be taken to place the cubic collection of grid points *within* the collection of atoms (if possible) in order to eliminate edge effects. If a grid point lies far outside of the collection of atoms, it naturally gets a large number. While surface anomalies can be removed with the 3D visualization programs, the inclusion of "outside" points decreases the range of numbers available to cavities and channels. The cubic volume covered by the cubic grid is under user control when *xeo* or *zeo* are called. This is the *Expanding Factor*. For cubic space groups, a value of 0.9 is satisfactory. For a non-cubic array of positions (such as necessarily results from hexagonal unit cells) a reduction to 0.6 or even 0.4 may be required.

The Fortran program provides the collection of grid points and their numerical values in a file format *file.dat* appropriate for use by the isosurface option in AVS or Explorer on Silicon Graphics computers. The AVS and Explorer programs have a variety of display options, including the use of multiple colored lights to illuminate the surface from any desired position and control over the surface texture, camera position, depth perspective, etc. By producing two images rotated ±3 degree about y, stereographic images may also be produced. Cutting planes in front and behind the image may also be used to view only a segment of the display. This is very useful when edge effects are present. The user is referred to the AVS and Explorer instruction manual for the operations available within this collection of programs.

Although double smoothing can result in distance errors of up to 10% for the typical large collections of atoms required for electrides, the isosurface images at various levels can be used to determine cavity and channel sizes. Specifically the number associated with a given grid point is proportional to the distance from that point to the nearest van der Waals surface. The distance in Angstroms corresponding to the largest value (255) can be determined. Thus, the radius of the largest sphere that can be placed within a cavity, for example, can be determined from the value at which the cavity just disappears into the black background. Similarly the radius of the largest sphere that can pass through a channel is determined from the value at which the continuity of the channel is just destroyed. The program *xeo* also provides information on the fraction of void space in the structure. At any later time, the program *scale* can be used to find the distance equivalent to the value 255.

3 Detailed View of the Method

The problem: We have a given number of atoms at fixed position in the space:

$$\vec{c_i} = (x_i, y_i, z_i)$$

with fixed radius r_i , the radius of the i-th atom, with $i = 1, 2, ..., n$ where n is the number of atoms.
The question: Where are the empty regions in this structure and how can we construct a three-dimensional picture of these regions?
The solution: We place a grid (for simplicity a cubic grid) at positions:

$$\vec{g_i} = (u_j, v_j, w_j)$$

over the space occupied by the atoms, in which $j = 1, 2, ..., m$ and m is the number of grid points (usually 40x40x40 = 64,000 grid points are used). At every grid point we calculate the distance between the grid point and the closest atomic surface. (The atomic surface is defined by a sphere with the van der Waals radius of the atom in question.) The value assigned to each grid point is thus the distance:

$$d_j = min\{\sqrt{(x_i - u_j)^2 + (y_i - v_j)^2 + (z_i - w_j)^2} - r_i\},$$

$i = 1, 2, ..., n$ refers to the atoms and j is the label of the gridpoint. If the gridpoint is inside an atom this value is zero. As we can see, this procedure is nothing but placing the biggest possible sphere at every gridpoint without intersecting any atomic surface. After this calculation we get a "scalar field": every gridpoint in the space is assigned a real number between zero and the largest distance in the structure. We then transform this interval to the 0-255 integer interval. This permits the use of the commercial computer program - AVS or Explorer - to visualize this field by fitting an isosurface (three dimensional surface plot) to the gridpoints.

Implementation:

The following files are in the packet:

1.xeo.for
2.zeo.for
3.atom.rad
4.smooth.for
5.scale.for

1.Fortran code to calculate the "negative" channel picture.
2.Fortran code to calculate the "positive" atomic picture.

3.ASCII data file containing the radii of the different type of atoms.
4.Fortran code to smooth the pictures calculated by the first two programs.
5.Fortran code to read back the scaling information directly from a field-file without recalculation of the structure.

The whole system is given on this flow-chart:

Insight-II
↓
file.bkv
↓
xeo ; zeo ⟵ atom.rad
↓
file.dat
↓
smooth
↓
filesm.dat
↓
Visualization

The atomic positions are in files in Brookhaven data-format. The radii of the different type of atoms are in a file called "atom.rad". You can use arbitrary atomic radii by changing this file. We have entered the van der Waals radii given in the program Chem3D for most of the atoms that will be encountered. Others just have a question mark beside the element symbol. The ability to handle 99 different types is biult into the program.

The maximum number of atoms in the program is 19,999; the maximum size of the grid is 81 x 81 x 81. Usually we use 40 x 40 x 40 gridsize, because in this case the scalar-field-file is not too big (64 Kbyte) and the visualization is fast enough.

Because of this limited resolution the isosurfaces are slightly rough right after the calculations. The quality of the pictures can be improved by using smoothing. The smoothing is a simple three dimensional averaging with the following weight factors:

$$\frac{1}{64}\left[\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}\right]$$

The new smoothed value of a gridpoint depends on the old value of the point and the values of its 26 neighbors. The optimal number of the application of this smoothing is two according to our experience, but for different visualizations this number could be different. The user has the choice of the number of successive smoothings to use.

Installation:

The Fortran codes do not need any special compiler option. Two special functions are used in the codes: FGETC and FPUTC. These functions are available under every UNIX operating system. We managed to compile the programs without any changes in the source codes on three different types of machines (DEC-workstation, Ardent Titan, SGI Indigo) under different UNIX systems.

Program usage:

When the compilation is finished you can start to work. The *xeo* and *zeo* programs will ask for four parameters:

1. Input filename
2. AVS resolution
3. Expanding Factor
4. Output filename

1. The input filename is the name of your Brookhaven file (without the extension *.bkv*).
2. The resolution is an integer number between 1 and 81. The recommended value is 40. Remember that the calculation time is proportional to the number of gridpoints; that is, to the cube of this resolution.
3. The Expanding Factor is a real number. It is the ratio of the length of an edge of the cubic region to the *largest* dimension of the atomic structure. A factor less than 1.0 means magnifying, while greater than 1.0 means reducing. The recommended value is 0.4-0.6 to avoid edge effects, unless a cubic unit cell is involed, in which case, a value of 0.9 works fine.
4. The output filename to use is the name of the field-file (without the extension *.dat*).

When the program finishes the calculations, it gives the largest distance found in the structure. The unit of this number is the unit of the input coordinates and radii. In our case, for Brookhaven files this unit is Angstrom. This largest distance will be important information to have during the visualization.
The *smooth* program asks for the following parameters:

1. Input filename
2. number of smoothings
3. boundary condition
4. Output filename

1. The input filename is the name of the file to be smoothed (without extension *.dat*).
2. How many times would you like to smooth? Our recommendation is 2.
3. The grid points on the surface of the cube have not enough (26) neighbors. We have to give the boundary condition artificially by hand. Recommended value is zero.

4. The output filename is the name of the result file (without the extension *.dat*). Our naming convention for this file is *filesm2* if the number of smoothings is 2.

The *scale* program asks for only one parameter, the name of the field-file (without extension *.dat*). This program is useful when you forget the largest distance in a given structure. It just reads in the field-file and types this number to the screen. You can apply this *scale* program anytime to either the original field-files or to the smoothed ones.

Time consumption:

The calculation time is proportional to the number of atoms and to the number of gridpoints that is to the cube of the resolution.