XCrySDen—a new program for displaying crystalline structures and electron densities

Anton Kokalj

Department of Physical and Organic Chemistry, J. Stefan Institute, Ljubljana, Slovenia

XCrySDen is a molecular and crystalline-structure visualization program, but its main function is as a property analyzer program. It can run on most UNIX platforms, without any special hardware or software requirements. Special efforts were made to allow for appropriate display of 3D isosurfaces and 2D contours, which can be superimposed on crystalline structure and interactively rotated and manipulated. XCrySDen is also a graphical user interface for the CRYSTAL95/98 (Saunders, V. R., et al. CRYSTAL98—User's Manual. University of Torino, Turin, Italy, 1999) ab initio program and a visualization system for the WIEN97 (Blaha, P., et al. Comput. Phys. Commun. 1990, ***59,*** *399) ab initio program. In this article the program functions are presented with a short description of the algorithms. © 2000 by Elsevier Science Inc.*

Keywords: molecular graphics, crystalline graphics, isosurfaces, electronic density, modelling

INTRODUCTION

The tremendous progress in computer hardware has made it possible to use personal computers (PII generation and its clones) with a good graphics card as graphics stations, since such computers are more than capable of displaying several hundred atoms or an isosurface composed of several thousand triangles and perform real-time operations such as rotation and translation. There are many molecular and "protein" visualization programs, some of them are obtainable even free of charge (e.g., Rasmol,[1] Xmol,[2] Molden,[3] MolMol[4]) but there is still a lack of programs that are specialized for crystalline (periodic) structure visualization.

Since visualization of periodic structures requires some specific features, standard molecular visualization programs are not suited for that purpose. A rather common operation in periodic structure visualization is the switching between the primitive and conventional cell, or changing the number of cells displayed. But these simple operations cannot be achieved with "traditional" molecular visualization programs. A specialized periodic-structure visualization program must also possess additional specific features such as displaying the crystal cages (lattices) in direct and reciprocal space, and displaying the Wigner–Seitz cell (direct space) and the first Brillouin zone (reciprocal space). Beside measuring the distances, angles, and dihedrals, a measurement of crystal planes is almost mandatory.

In this article the XCrySDen program is presented and its functionality is explained. XCrySDen is a new program. The name stands for Crystalline Structures and Densities, and the X means it runs in the X-Window environment. It is available for most Unix platforms running X-Window.[5] The purpose of the program is to be a general solid-state visualization program, where visualization of physical properties is emphasized. A special effort was made to avoid any hardware and software requirements (the author successfully runs XCrySDen on an 80486/66MHz PC computer with only 12MB of memory and screen dimensions of 800 × 600). All libraries used by the program are either under general public license (GPL)[6] or freely available by other means.

PROGRAMMING CONSIDERATIONS

The programming tasks for XCrySDen can be divided into three groups, namely, numerical tasks, graphics, and a graphical user interface (GUI). For each group different programming languages and libraries are used. Because C is the most used programming language for writing various applications, and because the C support for various libraries is the strongest, the author chose C language and an OpenGL[7] application programming interface (API) for achieving high-quality graphics. Instead of commercial OpenGL, the freeware Mesa[8] emulation of OpenGL was used. For making the GUI there are several possibilities such as XForms,[9] Python,[10] GTk,[11] and Tcl/Tk[12] and its extensions ([Incr Tcl]/[Incr Tk]/[Incr Widgets], TclX, BLT, Tix). The author chose the latest, namely, Tcl/Tk, which is a Unix shell-like interpreter with capabilities

Color Plates for this article are on pages 215 and 216.

Corresponding author: Anton Kokalj, Department of Physical and Organic Chemistry, J. Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia

Journal of Molecular Graphics and Modelling 17, 176–179, 1999
© 2000 by Elsevier Science Inc.
655 Avenue of the Americas, New York, NY 10010

1093-3263/99/$–see front matter
PII S1093-3263(99)00028-5

of making toplevel windows, and comes with a huge collection of different widgets. On a LINUX[13] platform Tcl/Tk is one of the most used languages for making GUIs today. It is the adaptableness of Tcl/Tk that makes it so attractive. Beside being an interpreter it also has a large variety of C library functions support. It is easy to modify the interpreter by including application-specific custom commands. All that has to be done is to write a `Tcl_AppInit` function, which is called by the `Tk_Main` function (see Listing 1). This is how the link between Mesa and Tcl/Tk was done in XCrySDen, namely, by creating a custom interpreter with Mesa support.

The XCrySDen program consists of a custom Tcl/Tk interpreter, where all specific application commands are built in. Tcl/Tk scripts of XCrySDen use this custom interpreter and manage the GUI and all input–output (IO) protocols. In addition to that, some auxiliary one-purpose programs are written in Fortran and a few Awk[14] scripts are provided for needed text manipulations. The scheme of the XCrySDen programming structure is shown in Color Plate 1.

The graphical user interface of XCrySDen was especially developed to provide an easy-to-use and learn interface. Casual users should be able to exploit at least half of the whole functionality without devoting more than a few hours of effort to the task of learning the use of the program. A special toolbox panel is located on the left part of XCrySDen main window (see Color Plate 2) with a collection of command buttons. Since all the buttons cannot fit into the space devoted to the toolbox panel, the panel can be scrolled up and down with a scale widget in order to allow use of a large variety of command buttons, because pressing the buttons is usually easier and faster than following the logic of menus. The menus were deliberately constructed with just a few entries per menu, since overlarge menus usually confuse inexperienced users.

XCrySDen AS A MOLECULAR VISUALIZATION PROGRAM

XCrySDen can read a molecular structure from several standard file formats, such as XYZ,[15] PDB,[16] CHARMM[17] cdr, and display the molecule in several standard modes (lines, pointlines, sticks, ballsticks, spacefills, etc.).

Two levels of display modes are available, the so-called "lighting-off" and "lighting-on" modes. The usual approach for displaying objects is to build 3D models of the objects. For example, a model of a sphere is constructed by subdivision around the *z* axis into slices and along the *z* axis into stacks (similar to lines of longitude and latitude). Thus the model of a sphere is composed of many rectangular facets, where the edges of the facets are called vertices. Each vertex in such a model is composed of position coordinates and a normal vector, where the normal vector is used to calculate the shades, i.e., how bright some pixel is on the screen. Before objects are rendered, a hidden surface removal algorithm is employed via the *z* buffer, meaning that each new incoming screen pixel is tested against the *z* coordinate of an old pixel. If the *z* coordinate of the new pixel is lower than that of the old pixel *z* coordinate, the new pixel is discarded. This is called the depth buffer technique, and is time consuming if hardware acceleration is not present and may result in an enormous decrease in rendering performance. Since some display modes, such as wireframe, pointlines, and ballsticks, are simple, depth buffering can be omitted. Instead, a record of all objects consisting of a model of the molecule is built. To each object a vertex of *xyz* coordinates is assigned and the vertices are then sorted according to *z* coordinate. Objects are then rendered according to that sorting order, where objects with a lower *z* coordinate are rendered first. This is the so-called lighting-off mode, because to each object just one color is assigned and no shading is performed. For computers with no 3D graphics acceleration this mode is much faster than the

```
int
main(int argc, char *argv[])
{
  /* insert application-specific initialization here */

  Tk_Main(argc, argv, Tcl_AppInit);
  exit(0);
}

int
Tcl_AppInit(Tcl_Interp *interp) {
   /* Initialize Tcl & Tk packages */
   if (Tcl_Init(interp) == TCL_ERROR) return TCL_ERROR;
   if (Tk_Init(interp)  == TCL_ERROR) return TCL_ERROR;

   /* define application-specific commands here */
   for (i=0; i<number_of_custom_commands; i++)
      Tcl_CreateCommand(interp,
                        custom_command_name[i],
                        custom_command_proc[i],
                        (ClientData) Tk_MainWindow(interp),
                        (Tcl_CmdDeleteProc *) custom_command_deleteproc[i]);
   return TCL_OK;
}
```

Listing 1. Recipe for constructing a custom Tcl/Tk interpreter. The main function calls the `Tk_Main` *function, and the latter then calls the* `Tcl_AppInit` *function. In* `Tcl_AppInit` *function a* `Tcl/Tk` *interpreter is initialized (*`Tcl_Init, Tk_Init`*) and all application-specific commands are created (*`Tcl_CreateCommand`*). All necessary Mesa rendering is thus done through these custom commands.*

lighting-on mode, because all that is needed is to perform z orientation, and a sorting method such as heapsort[20] is an $n\log_2 n$ process, where n is the number of objects. It is easily possible to rotate and manipulate several hundred atoms on PC 80486 computers with the lighting-off mode.

XCrySDen can perform all standard operations, such as rotation, translation, and zooming. Measurement of distances, angles, dihedrals, and planes (crystals only) is possible as well. This can be achieved by clicking on two (distance), three (angle), or four (dihedral angle) atoms. While clicking the desired atoms some information such as atomic number and symbol, sequential number, and *xyz* coordinates, is displayed in a separate top-level window. It is possible to rotate the structure during the measurement of distances, angles, and other quantities and this feature can be quite useful for complex structures. Atomic labels and the *xyz* coordinate system can be turned on as well.

In addition, all "cosmetic" options are available. It is possible to change/set almost everything: the color and radius of atoms, chemical connectivity factor, thickness of bonds and lines, and size of points, balls, and spacefills. By default a bond between two atoms is drawn if their distance is shorter than 1.05 times the sum of their covalent radii. Spacefills and ballsticks can be displayed on the basis of covalent or van der Waals radii of atoms. It is also possible to set different OpenGL material and lighting parameters.

XCrySDen AS A CRYSTALLINE STRUCTURE VISUALIZATION PROGRAM

As already explained in the introduction, visualization of crystalline structures requires some specific features (switching between primitive and conventional cells, changing the number of cells).

Several types of unit cells are of interest, namely, primitive, conventional, or even Wigner–Seitz. Hexagonal and rhombohedral crystals can be displayed beside the parallelepiped shaped also in a hexagonal shaped cell (triple cell). Displaying the crystal cages is almost a necessity for crystalline structures and XCrySDen can visualize it as lines or sticks. If an algorithm for constructing a Wigner–Seitz cell is implemented, it can be used for construction of the first Brillouin zone as well, since the first Brillouin zone is nothing other than a Wigner–Seitz cell in reciprocal space. XCrySDen constructs the Wigner–Seitz cell in two steps. First, the coordinates of all facet points constructing the cell are calculated by an algorithm due to Finney.[18] The second step is to construct the convex polygons out of facet points. This step is usually referred to as "finding the convex hull," where Jarvis's march algorithm[19] is used.

When some special group of atoms should be emphasized in the crystalline structure, then it can be displayed as voronoi polyhedra, since this would give some information about the symmetry of the site. Again the algorithm for construction of Wigner–Seitz cell can be used.

XCrySDen AS A PROPERTY ANALYZER

A special emphasis of XCrySDen is the visualization of properties. One of the main goals of the program is to make the analysis of computed data (properties) much easier. The properties analyzer consists of two parts: a 3D visualizer and a 2D graphing system.

Plotting Properties with the 2D Graphing System

A special 2D XY and BAR graphing widget was implemented in XCrySDen. This widget is highly optimized for plotting some properties such as bandwidths, density of states, band structure, SCF convergence, and interaction potentials. For plotting the band structure a special widget displays the Brillouin zone and the path along the desired high symmetry points can be interactively selected by clicking the points with a mouse (Color Plate 3).

3D Property Visualizer

XCrySDen possesses several graphical procedures for selecting a portion of space or plane on which the user wants to compute some property. Two levels of such selections exist. For crystals (3D periodicity) the whole unit cell can be selected, whereas for slabs (2D periodicity) a third dimension must be specified. For polymers (1D periodicity) two nonperiodic dimensions must be specified. For molecules a bounding box with some margins can be specified. The second selection possibility is to interactively select the region following some sequence of tasks. Color Plate 4 shows feedback of XCrySDen to the user during selection of a region. The selected region is shown as a transparent parallelepiped.

Although plots like contour plots can easily be achieved, for example, by *Gnuplot*,[21] the advantages yielded by XCrySDen are as follows: (1) the region of space or plane for property plot can be selected interactively; (2) such a plot can be superimposed on the crystalline structure in order to enhance the readability of the plot; (3) crystalline structure can be transformed from primitive to conventional cell mode during the property display; (4) property display has a periodic attribute, meaning that it can be extended over several unit cells; (5) the whole picture can be interactively rotated, zoomed in real time, and displayed in the desired perspective; and (6) contours can be displayed as colorplanes or colorplanes with isolines.

Displaying the Properties on a Selected Plane

Some properties are particularly suited to be displayed as contours (isolines) or colorplanes or both together (see Color Plate 5). By *colorplane* is meant a plane where a property is not represented by isolines, but by the colors. There are several color schemes that can be used (black and white, rainbow, red–green–blue, cyan–magenta–yellow, and geographic [i.e., blue–green–yellow–brown–violet–white]).

Displaying the Isosurfaces

XCrySDen can display any regular 3D grid of points as an isosurface. The algorithm used for achieving the triangulation is the standard one, namely, *Marching Cubes*.[22] XCrySDen uses the algorithm by following the recipe of Bourke.[23] The initial step is the partitioning of a selected space region into smaller cubes. The algorithm then triangulates each individual cube. The *vertex recognition* algorithm is implemented as well, since it is needed in order to smoothly shade the triangulated isosurface. To perform the smooth shading of a triangulated isosurface, normals of vertices of touching triangles should be averaged to an average direction. A vertex recognition algorithm is thus needed for this averaging of the sharing vertex normals. The simplest vertex recognition algorithm would be to compare the *xyz* coordinates of vertices among themselves, but this would led into an n^2 process and would therefore require too much CPU time. Instead, vertex recognition can be traced during the triangulation process with the help of a lookup table. Let us define a sharing vertex as a vertex that is shared among several triangles that touch each other at least at this vertex. Sharing vertices can either occur inside the marching cube or among the nearest neighbor marching cubes. XCrySDen exploits this fact and performs vertex recognition and averaging of normals during the triangulation process.

Because the triangulation process is fast compared with grid calculation of, for example, "*ab initio*" electron density, it can be controlled interactively. Scanning through various isolevels is a relatively fast process for small structures on a moderate Pentium PC computer and can be performed interactively with XCrySDen.

Several display modes of isosurface are possible, namely, solid, wire, and dotted. It is also possible to set the isosurface as transparent. The isosurface can be smoothly or flatly shaded. XCrySDen also allows us to change/set various OpenGL parameters such as material properties of the isosurface or blending functions for achieving different types of transparency effects.

When a 3D grid of points is specified, then up to two different isosurfaces and three different property planes can be displayed simultaneously. This is more than enough for most cases, since more information on a single plot would result in complete unreadability of the plot.

ADDITIONAL INFORMATION

Additional up-to-date information about the XCrySDen package is available at the XCrySDen WWW home page, http://www-k3.ijs.si/kokalj/xc/XCrySDen.html.

ACKNOWLEDGMENTS

The author acknowledges helpful discussions with Mauro Causà, who gave the initial idea for this program. The author also thanks Milan Hodoščček and Piero Ugliengo for discussions about programming and related topics. The Slovenian Ministry of Science is acknowledged for financial support for the author's Ph.D. studies.

REFERENCES

1 Sayle, R. *Rasmol v2.5—A Molecular Visualisation Program*. Glaxo Research and Development Greenford, Middlesex, UK; Sayle, R.A., and Milner-White, S.C. RASMOL: Biomolecular graphics for all. *Trends Biochem. Sci*. 1995, **20,** 374–376

2 Minnesota Supercomputer Center. *Xmol User Guide*, version 1.3.1. Minnesota Supercomputer Center, Inc., Minneapolis, Minnesota, 1993

3 Schaftenaar, G. MOLDEN: A Portable Electron Density Program. *Quantum Chemical Program Exchange* QCPE619; MOLDEN: A pre and post-processing program of molecular and electronic structure. http://www.caos.kun.nl/~schaft/molden/molden.html

4 Koradi, R., Billeter, M., and Wüthrich, K. MOLMOL: A program for display and analysis of macromolecular structures. *J. Mol. Graphics* 1996, **14,** 51–55

5 Johnson, E.L., and Reichard, K. *X Window Applications Programming*, MIS, Inc., a subsidiary of Henry Holt and Company, New York, 1992

6 *GNU Library General Public License*, version 2. Free Software Foundation, Boston, 1991

7 Silicon Graphics, Inc. *The OpenGL Programming Guide*. Addison-Wesley, Reading, Massachusetts, 1993

8 Paul, B. *Mesa—The Free OpenGL Work-Alike Library*. http://www.mesa3d.org/. 1999

9 Zhao, T.C., and Overmars, M. *XForms—Forms Library: A Graphical User Interface Toolkit for X*. http://bragg.phys.uwm.edu/xforms. 1999

10 Ascher, D., and Lutz, M. *Learning Python*. O'Reilly & Associates, Beijing, 1999; *Python—interpreted, interactive, object-oriented, extensible programming language*. http://www.dstc.edu.au/python/. 1997

11 Amundson, S.T. *The GIMP Toolkit*. http://www.gtk.org/. 1999

12 Ousterhout, J.K. *Tcl and the Tk Toolkit*. Professional Computing Series. Addison-Wesley, Reading, Massachusetts, 1994; Welch, B.B. *Practical Programming in Tcl and TK*. Prentice Hall PTR, Englewood Cliffs, New Jersey, 1995; some information about Tcl/Tk can be browsed from http://www.tclconsortium.org/, http://www.elf.org/, and http://www.tcltk.com/

13 Linux is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world. Linux is an independent POSIX implementation and includes true multitasking, virtual memory, shared libraries, demand loading, proper memory management, TCP/IP networking, and other features consistent with Unix-type systems. Developed under the GNU General Public License,[6] the source code for Linux is freely available to everyone. The web site of Linux can be found at http://www.linux.org/

14 Awk was written by: Aho, V.A., Weinberger, P.J., and Kernighan, B.W. at AT&T Bell Laboratories, 1997; Robbins, A.D. *AWK Language Programming, A User's Guide for GNU AWK*, 1996

15 Network Computing Services, Inc. *XYZ Format*. http://www.msc.edu/msc/docs/xmol/XMol.html. 1995

16 Brookhaven Protein Data Bank. *PDB Format*. http://www.pdb.bnl.gov/. 1999

17 Brooks, B.R., Bruccoleri, R.E., Olafson, B.D., States, D.J., Swaminathan, S., and Karplus, M. *CHARMM: Chemistry at Harvard Macromolecular Mechanics*. *J. Comput. Chem*. 1983, **4,** 187

18 J.L. Finney. *J. Comput. Phys*. 1979, **32,** 137

19 Jarvis, R.A. On the identification of the convex hull of a finite set of points in the plane. *Inf. Proc. Lett*. 1973, **2,** 18–21; Cormen, T.H., et al. *Introduction to Algorithms*. MIT Press, Cambridge 1990

20 Knuth, D.E. Sorting and Searching. *The Art of Computer Programming.* Reading, MA: Addison-Wesley, 1973, vol. 3. Sedgewick, R., *Algorithms.* 2nd ed., Reading, MA: Addison-Wesley 1988, Chapter 11. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. *Numerical Recepies in Fortran 77: The Art of Scientific Computing.* 2nd ed., Cambridge University Press, 1992, Chapter 8.

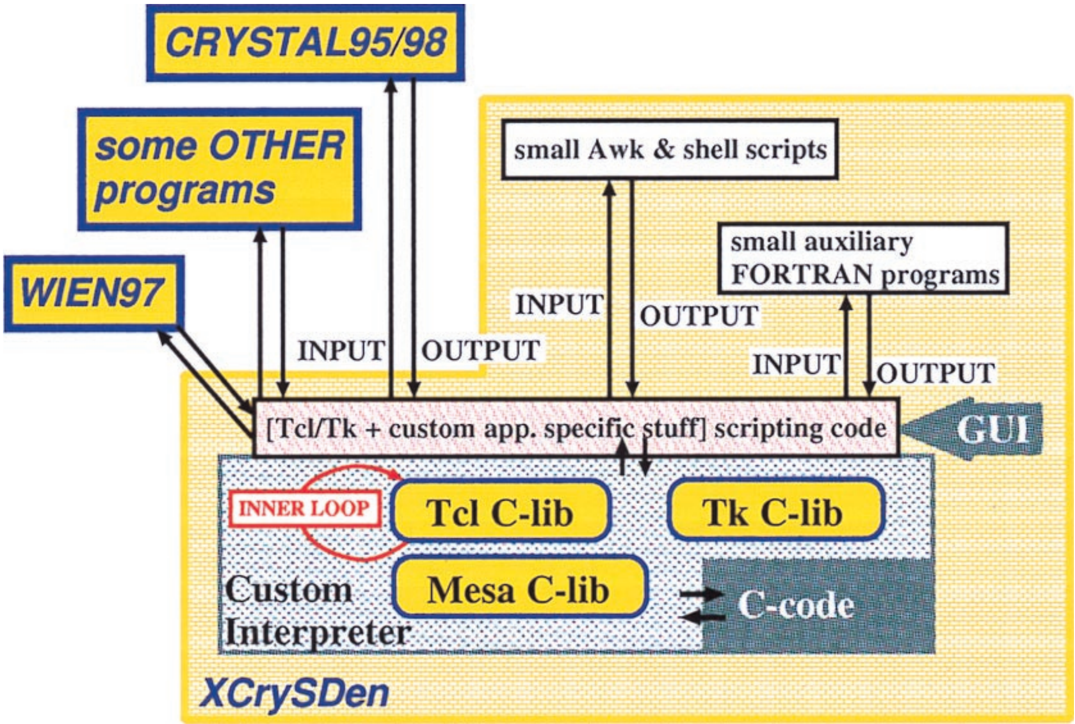
21 Williams, T., and Kelley, C. *Gnuplot–An Interactive Plotting Program*. http://www.cs.dartmouth.edu/gnuplot/gnuplot.html. 1995

22 Lorensen, W., and Cline, H. *Comput. Graphics* 1987, **21,** 163

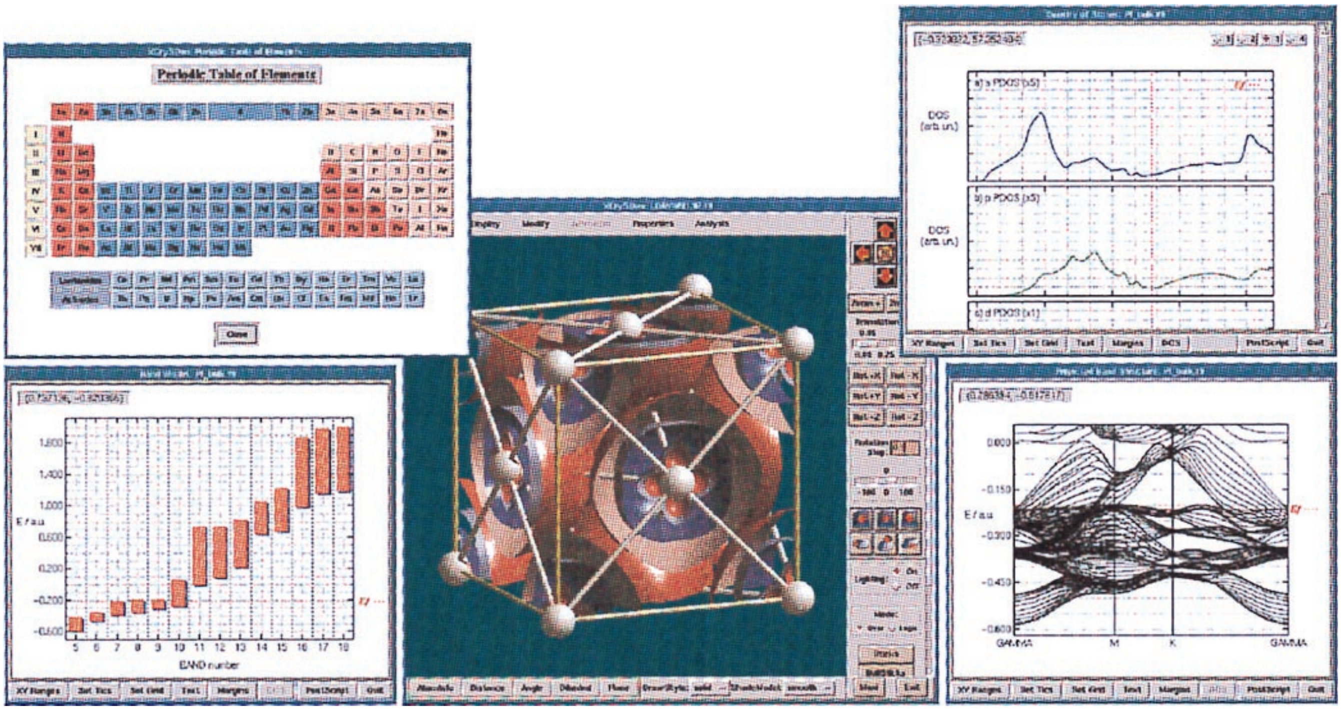
23 Bourke, P. *Modelling*. http://www.swin.edu.au/astronomy/pbourke/modelling/. 1999

Anton Kokalj

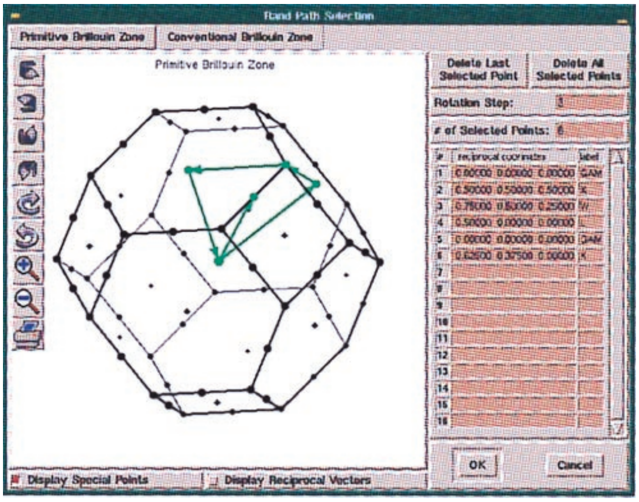
XCrySDen—a new program for displaying crystalline structures and electron densities



Color Plate 1. Structure of the XCrySDen program, shown schematically.



Color Plate 2. Snapshot of *Property Analyzer* of XCrySDen. Platinum bulk is analyzed. *Main window*: fcc cell of Pt with the difference electron density. *Top left window:* periodic table of elements. *Bottom left window*: bandwidths around the Fermi energy. *Top right window*: s- and p-projected density of states. *Bottom right window*: band structure projected to (111) plane.


Band Path Selection

Primitive Brillouin Zone | Conventional Brillouin Zone

Primitive Brillouin Zone

Delete Last Selected Point

Delete All Selected Points

Rotation Step: 3

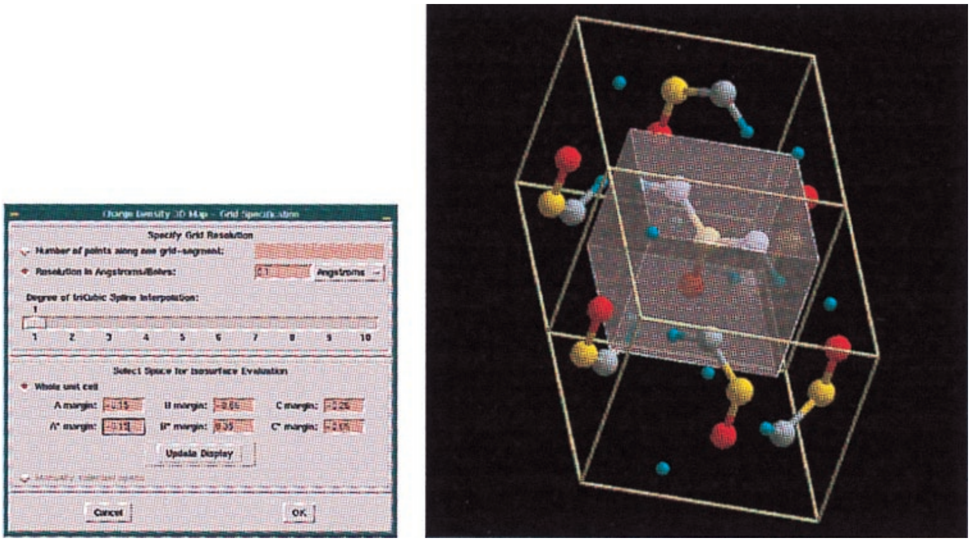
# of Selected Points: 6

#	reciprocal coordinates			label
1	0.00000	0.00000	0.00000	GAM
2	0.50000	0.50000	0.50000	K
3	0.75000	0.50000	0.25000	W
4	0.50000	0.00000	0.00000	L
5	0.00000	0.00000	0.00000	GAM
6	0.62500	0.37500	0.00000	K
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

☒ Display Special Points

□ Display Reciprocal Vectors

Color Plate 3. Snapshot of *k-path* selection.


Charge Density 3D Map – Grid Specification

Specify Grid Resolution

○ Number of points along one grid-segment:

● Resolution in Angstroms/Bohrs: 0.1 Angstroms

Degree of triCubic Spline Interpolation: 1

Select Space for Isosurface Evaluation

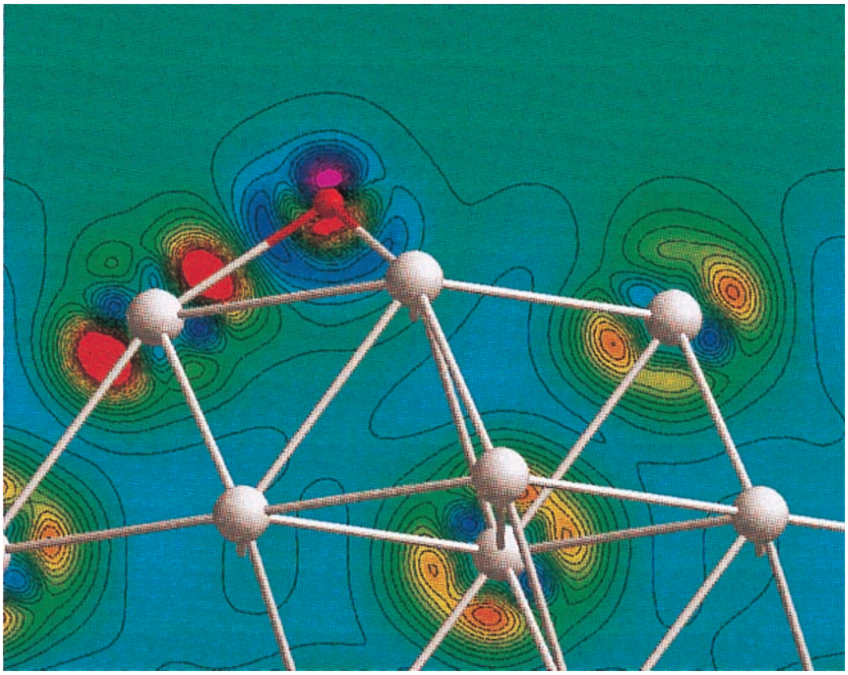
● Whole unit cell

A margin: -0.15 B margin: -0.05 C margin: -0.25

A* margin: -0.15 B* margin: 0.35 C* margin: -0.05

○ Manually selected space

Color Plate 4. Snapshot of region selection for property calculation on a molecular crystal of urea. The control window for selection of region and selected region are shown at left and right, respectively. The selected region is shown as a transparent box and comprises one whole urea molecule.



Color Plate 5. 2D difference electron density of an adsorbed oxygen on a Pt(111) surface in p(2 × 2) geometry shown as contours and rainbow colourplane. Red regions are electron poor, whereas blue and violet regions are electron rich.