ELSEVIER

Computational Materials Science 28 (2003) 155–168

COMPUTATIONAL MATERIALS SCIENCE

www.elsevier.com/locate/commatsci

Computer graphics and graphical user interfaces as tools in simulations of matter at the atomic scale

Anton Kokalj [*]

Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia
SISSA–Scuola Internazionale Superiore di Studi Avanzati, and INFM DEMOCRITOS National Simulation Center, I-34014 Trieste, Italy

Abstract

The role of computer graphics in different aspects of simulating matter on the atomic scale is discussed. The computer graphics is useful in specifying and examining chemical structures, since it is nowadays possible to study—with density functional theory—complex systems containing up to a few hundreds in-equivalent atoms. Furthermore, computer graphics is also an indispensable tool in analysing computed data and facilitates interpretation of results. In this context `XCrySDen` (http://www.xcrysden.org/) is presented, a crystalline- and molecular-structure visualisation program, which aims at display of isosurfaces and contours, which can be superimposed on crystalline structures and interactively rotated and manipulated. Another aspect of computer utilisation in simulations that takes advantage of the computer's graphics capabilities, is that it provides intuitive graphical user interfaces for the simulation setup. It is demonstrated how such interfaces are easily built using the developed `GUIB` software (http://www-k3.ijs.si/kokalj/guib/).

1. Introduction

Graphical representations of molecular structures are an important adjunct to simulations of matter at the atomic scale and have exerted a very important influence in computer-aided modelling. Historically, molecules have been most commonly visualised with computer graphics using "ball-and-stick" and "space-filling" representations of the Dreiding and Corey–Pauling–Koltun (CPK) mechanical models [1].

In this paper the role and applicability of computer graphics in simulations of matter at the atomic scale are discussed. Attention will be focused on the graphical needs associated with simulations performed with the so-called *first principle* or ab initio electronic structure calculations, and the usefulness of computer graphics will be demonstrated through a particular example of such simulations. These needs are, however, by no means restricted solely to this level of simulation, but many of them are more generally applicable.

The tremendous growth in computer power and accessibility has made ab initio modelling an ever-increasing area of interest in chemistry, physics and materials science. The simulation software currently available for the quantum-mechanical

[*] Tel.: +386-1-477-35-23; fax: +386-1-477-38-11.
E-mail address: tone.kokalj@ijs.si
URL: http://www-k3.ijs.si/kokalj

0927-0256/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0927-0256(03)00104-6

study of properties of molecular and crystalline systems encourages its widespread use also in a rapidly growing community of non-specialised users, such as experimentalists. In this respect the graphical user interfaces, which are aimed at a more intuitive simulations setup and easier analysis of the computed data, become valuable tools.

This paper is organised as follows. In the next section various aspects of the usefulness of computer graphics and graphical user interfaces are discussed in the present context. Two software applications are then presented. As an example of an application that exploits computer graphics for visualising chemical structures and other physico-chemical properties, the `XCrySDen` program is presented in Section 3 and its functioning is described. In the context of providing intuitive graphical user interfaces (GUI), the newly implemented `GUIB` software is presented in Section 4 along with a simple example of GUI construction.

2. Role of computer graphics and graphical user interfaces

The graphical presentation of chemical structures is of fundamental importance. Nevertheless the applicability of computer graphics is far broader, because it can be used in a variety of aspects of simulations. To demonstrate this point, let us consider a typical ab initio simulation of matter at the atomic scale. Such a simulation consists of three steps (see Fig. 1):

(i) the first step is the selection of the model and method. The level of approximation is selected, because the Schrödinger equation is not analytically solvable (except for a few trivial cases). Thus an approximate Hamiltonian of the system is chosen. Basis functions for the expansion of wave-functions and other parameters also have to be specified. At this stage the specification of the chemical structure of the model (the term chemical structure stands for either a molecule or atoms within the unit cell of a crystal) is an essential input. This can be rather tedious, since it is possible nowadays to model, with density functional theory, complex systems containing up to a few hundred in-equivalent atoms. In such cases, computer graphics can be efficiently exploited and becomes a valuable auxiliary tool in building the chemical structure, by means of interactive graphical procedures, as well as for examining the chemical structure itself, because the human eye is very sensitive to any kind of irregularities, this being especially true in the case of crystalline structures;

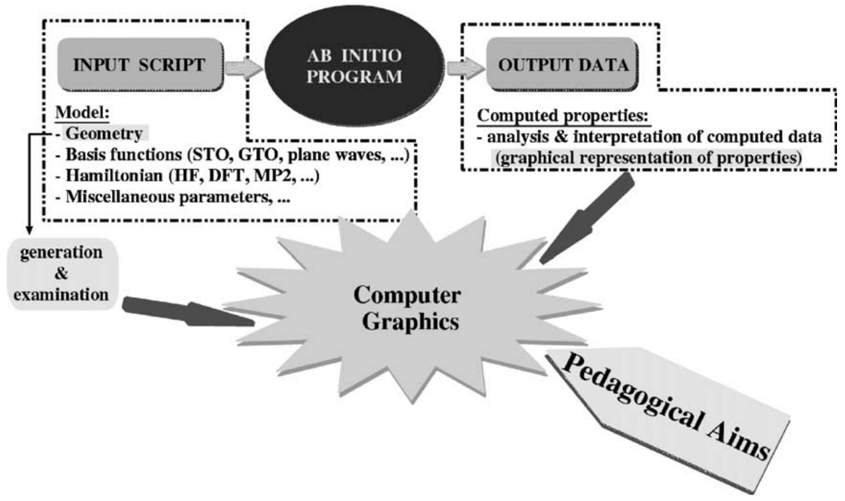

INPUT SCRIPT
AB INITIO PROGRAM
OUTPUT DATA
Model:
- Geometry
- Basis functions (STO, GTO, plane waves, ...)
- Hamiltonian (HF, DFT, MP2, ...)
- Miscellaneous parameters, ...
Computed properties:
- analysis & interpretation of computed data
(graphical representation of properties)
generation & examination
Computer Graphics
Pedagogical Aims

Fig. 1. A simplified scheme of ab initio atomistic simulation and the concomitant use of computer graphics.

(ii) the second step is the calculation itself, for example, a total energy calculation or an optimisation of the geometry of a given chemical structure. Here also, computer graphics can be used, for example, to examine the evolving chemical structure during geometry optimisation or the molecular dynamics run;

(iii) the last step is the analysis and interpretation of the calculated properties. Here, computer graphics is indispensable for use in a variety of cases, starting from simple two-dimensional (2D) data plotting (i.e. *XY* plots) of, for example, the total-energy vs. unit-cell-volume relationship, to more complex isosurfaces of the computed properties such as molecular orbitals, electron densities, and Fermi surfaces.

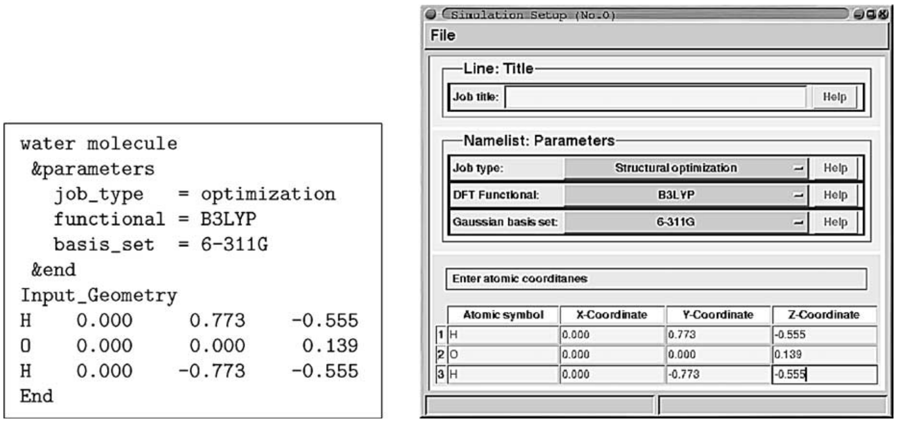
Here we focus on the computer graphics aspects devoted to modelling crystalline (i.e. periodic) structures. In particular, the visualisation needs of the condensed-matter simulation community will be addressed. Because the visualisation of periodic structures requires specific features, standard molecular visualisation programs are not well suited for this purpose. A rather common operation is to switch between the primitive and conventional cell settings, or to change the number of cells displayed. But these simple operations cannot be achieved with "traditional" molecular visualisation programs. A specialised periodic structure visualisation program should also possess additional specific features such as displaying the crystal lattices in direct and reciprocal space, displaying the Wigner–Seitz cell (direct space) and the first Brillouin zone (reciprocal space). Another important example for the condensed-matter simulation community is the visualisation of Fermi surfaces although, unfortunately, appropriate visualisation tools are not widely accessible (free or low-cost available software is considered here). On the basis of the growing need for such visualisation tools, the `XCrySDen` project [2,3] was initiated several years ago. Programming started in 1996 and the first implementation of the program was made available in 1999. To begin with, the program was aimed to be a visualisation system for the `CRYSTAL` program [4]—an ab initio electronic structure program for the periodic systems. Later, `XCrySDen` was extended to support the visualisation of several other electronic structure programs such as `WIEN` [5], `PWscf` [6], and `FHI98MD` [7]. It can also be used independently of the above programs.

A somewhat related field to computer graphics, in the sense that it takes advantage of the computer's graphics capabilities, includes the whole branch of graphical user interfaces, which make programs easier and more intuitive to use. A powerful, though simple, example is the use of GUIs for creating input files for numerical simulation programs. The typical input of such a program consists of several records, with well defined sequential order and format. Hence, a user must either memorise the input syntax or browse the manual every time an input is created. In contrast, GUI offers an easier approach, not requiring the input-syntax to be memorised, but offering instead a description of each entity arranged in a suitable set of widgets. A comparison of these two approaches is illustrated in Fig. 2. Since GUIs offer a more intuitive approach for the simulation setup, `GUIB` software [8] has recently been implemented. The purpose of `GUIB` is to provide a simple mechanism (i.e. interface) for constructing of the GUIs that aims solely at the creation of input files for the numerical simulation programs (see Section 4).

3. The `XCrySDen` program

`XCrySDen` [2,3] is a crystalline- and molecular-structure visualisation program. It facilitates the display of isosurfaces and contours which can be superimposed on crystalline structures and interactively rotated and manipulated. It also possesses some tools for analysing properties in reciprocal space, such as visualisation of Fermi surfaces and interactive selection of *k*-paths in the Brillouin zone for band-structure plots. It can run on most UNIX platforms, without any special hardware requirements.

The graphical user interface of `XCrySDen` was developed specially to provide an easy to use and learn interface. Casual users should be able to



```
water molecule
 &parameters
   job_type   = optimization
   functional = B3LYP
   basis_set  = 6-311G
 &end
Input_Geometry
H    0.000    0.773    -0.555
O    0.000    0.000     0.139
H    0.000   -0.773    -0.555
End
```

Simulation Setup (No.0)

File

Line: Title

Job title: Help

Namelist: Parameters

Job type:	Structural optimization	Help
DFT Functional:	B3LYP	Help
Gaussian basis set:	6-311G	Help

Enter atomic cooordinates

	Atomic symbol	X-Coordinate	Y-Coordinate	Z-Coordinate
1	H	0.000	0.773	-0.555
2	O	0.000	0.000	0.139
3	H	0.000	-0.773	-0.555

Fig. 2. Traditional "text-only" (left) vs. GUI (right) approaches in seting up input parameters for atomistic simulation. Creating the input file with text-editors requires the knowledge of the input format (syntax). The GUI shown in the right panel was created using the `GUIB` software (see Section 4).

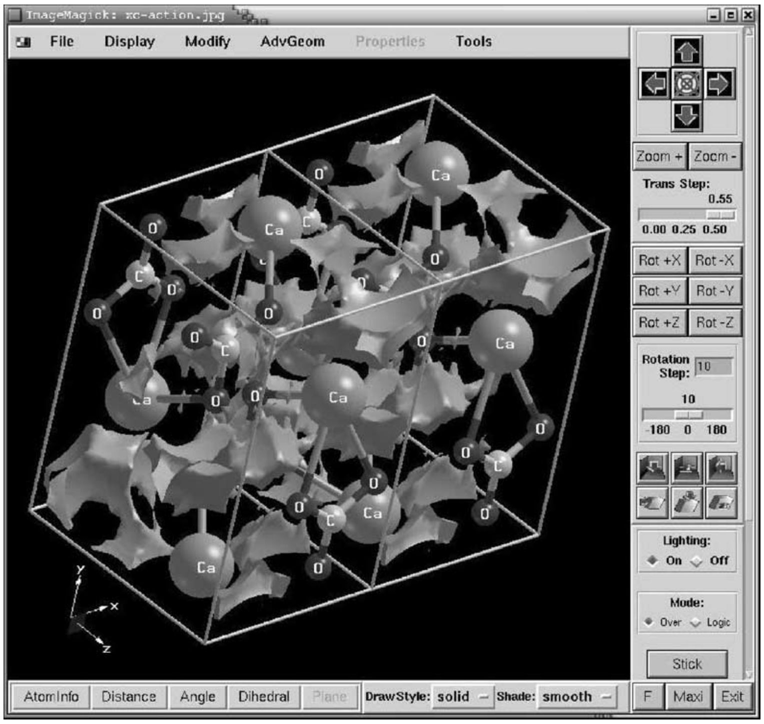


Fig. 3. Main window of `XCrySDen` program.

exploit more than just the basic functionality without devoting more than a few hours to learning to use the program. A special toolbox panel is located on the right part of the `XCrySDen` main window (see Fig. 3) with a collection of command buttons. 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.

3.1. Programming considerations

The programming tasks for XCrySDen can be divided into three groups: (i) molecular graphics, (ii) GUI, and (iii) numerical tasks. For each group, different programming languages and libraries are used. The routines associated with the molecular graphics are written in C-language using an OpenGL [9] library. Since XCrySDen has been developed on the Linux platform, the freeware Mesa [10] implementation of OpenGL was used. The GUI is written in Tcl/Tk language. Tcl/Tk is an interpreted scripting language, with an easy capability to extend the interpreter by including custom application-specific commands. In this way the link between the C-code and Tcl/Tk in XCrySDen was made by creating a custom interpreter with all XCrySDen-specific commands and OpenGL support. For the purpose of integrating OpenGL with Tcl/Tk, Togl [11]—a Tk OpenGL widget—was used. The numerical part of the program is partly written in C and partly in Fortran. In addition, some auxiliary single-purpose programs are written in Fortran and a few Bourne shell and Awk scripts are provided for text manipulation, such as conversion between different input-file formats. The scheme of XCrySDen's programming structure is shown in Fig. 4.

3.2. *XCrySDen: a molecular- and crystalline-structure visualisation program*

XCrySDen possesses most of the features that are expected from a modern molecular visualiser, i.e.: (i) representing molecular models in several display modes (wireframe, stick, ball-and-stick, spacefill, ...), (ii) displaying forces acting on atoms, (iii) real time operations such as rotation, translation and zooming of the displayed structure, (iv) animation, (v) measurement of distances, angles, and dihedrals, (vi) setting various attributes, such as colours of atoms and bonds, radii of atoms, thicknesses of bonds and lines, and sizes of points, balls and spacefills. XCrySDen can read a molecular structure from a few standard file formats such as *XYZ* [12], *PDB* [13], and its own internal *XSF* format (see Appendix A). In addition, it can visualise chemical structures for GAUSSIAN [14], CRYSTAL [4], PWscf [6], WIEN [5], and FHI98MD [7] programs.

In addition to these "standard" molecular visualising features, XCrySDen comes with an additional set of features that are necessary for visualising crystalline structures. Thus, XCrySDen is able to (i) switch between primitive and conventional cell settings, (ii) change the number of the displayed unit cells, that is, display smaller or

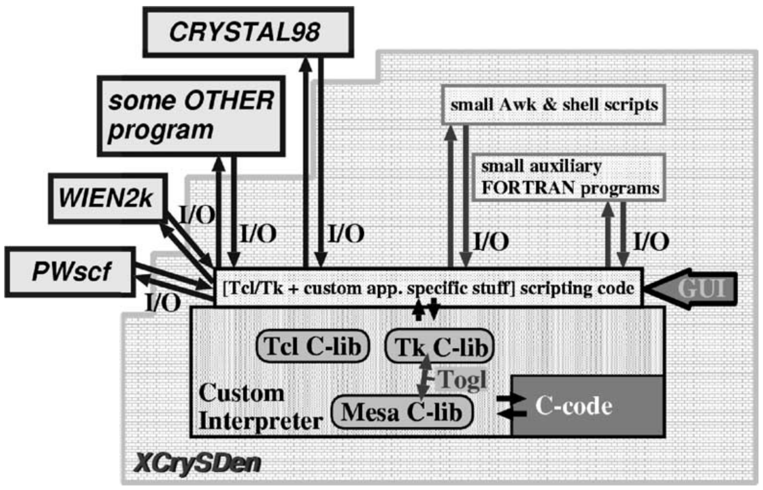


Fig. 4. Programming structure of XCrySDen shown schematically. The labels I/O in the figure stand for the input/output communication.

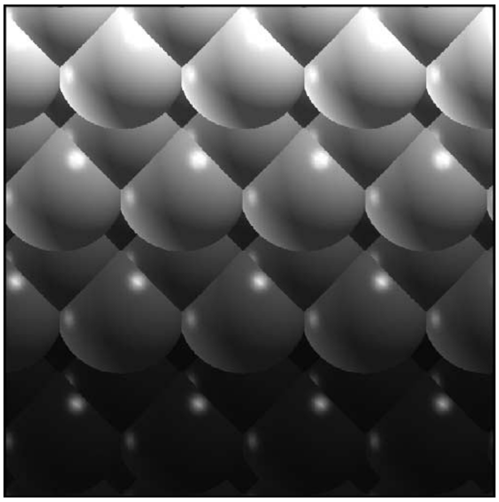


Fig. 5. An example of *slab-colour-scheme* used to emphasise the stepped structure of the fcc-(2 1 0) surface.

larger portions of a crystal, (iii) display the crystal lattice, (iv) visualise the Wigner–Seitz cell and Brillouin-zone, and so on. It is also possible to orient the structure by pressing the *orientation* buttons to the $\widehat{xy}$, $\widehat{xz}$ or $\widehat{yz}$ plane. In addition, there are also $\widehat{ab}$, $\widehat{ac}$ and $\widehat{bc}$ *orientation* buttons, which orient the structure to the basal three planes spanned by the lattice vectors **a**, **b**, and **c**. Such orientations are important for crystals with non-orthogonal lattice vectors.

In some cases it is useful to colour the displayed structure according to some structure-attribute in order to emphasise a particular structural aspect. `XCrySDen` comes with the following colour-schemes: (i) *distant-colour-scheme*—a site or atom is chosen as the centre and all other atoms are then coloured according to the distance from the selected site; (ii) *slab-colour-scheme* colours the atoms according to a selected direction, which can be useful for examining stepped surfaces (see Fig. 5).

3.3. *XCrySDen: a structure builder*

Currently the geometrical manipulations performed by `XCrySDen` are achieved through input/output (I/O) communication with the ab initio `CRYSTAL` [4] program, [1] and only a few are driven solely by `XCrySDen` itself. An enormous advantage provided by `XCrySDen` is the potential for yielding graphical feedback to the user. Every change of structure is visualised immediately and, to enhance the usefulness of `XCrySDen` even further, an UNDO/REDO option makes every false move recoverable. The following geometrical manipulations can be achieved: (i) adding, substituting, displacing, and deleting an atom or group of atoms; (ii) cutting a two-dimensional slab out of bulk crystal and possibly generating a multi-slab; (iii) cutting a cluster out of a higher dimensional structure (i.e. slab or bulk-crystal), (iv) generating a supercell, (v) rotating a Cartesian frame of the cell, and (vi) elastically deforming the unit cell.

`XCrySDen` possesses several interactive graphical procedures to aid building chemical structures and manipulating them. Some of the manipulations, such as *adding an atom* and *cutting a cluster* require selection of a site. The site can be selected in three different ways: (i) *hole selection*—the site is a geometrical centre of several atoms, and is selected by mouse-clicking these atoms; (ii) *line selection*—the site is on the line between two atoms. A fraction of the distance from the first atom must be specified; (iii) *cell selection*—the site is chosen according to specified fractions of the unit cell vectors. For molecules, the cell vectors are chosen from its bounding box. In addition, for polymers and for slabs, vectors of non-periodical directions are chosen from the bounding box.

Other groups of manipulations, such as *substitution*, *deletion*, and *displacement* of an atom, are achieved simply by mouse-clicking a given atom on the displayed structure. For the *manipulating-the-cell* family of options both the newly manipulated unit cell and the old unit cell are rendered simultaneously in transparent mode and are superimposed with atoms that belongs to the new unit cell. Using this family of options one can (i) generate a supercell, (ii) rotate a Cartesian frame,

[1] The I/O communications with the `CRYSTAL` program are managed automatically by `XCrySDen` and are hence hidden from the user. However, they require the installed `CRYSTAL` program.

and (iii) perform an elastic deformation to the lattice.

3.4. XCrySDen: a property analyser

A special advantage of XCrySDen is its ability to visualise properties. One of the main goals of the program is to make the analysis of computed data (properties) easier. XCrySDen possesses a few graphical procedures for selecting a region of space or plane on which a given property is to be computed and displayed. Two types of such selection exist. For crystals (3D periodicity) the whole unit cell can be selected, whereas for slabs (2D periodicity) and polymers (1D periodicity) the non-periodic dimensions have to be specified explicitly. For molecules a bounding box with some margins can be specified. The second selection possibility is to select interactively the region following some sequence of tasks. Fig. 6 shows feedback of XCrySDen to the user during selection of a region. The selected region is shown as a transparent parallelepiped.

3.4.1. An example: charge density plots

Rendering a property such as electronic charge density is an insolvable problem because it requires four dimensions, and the computer can provide only a 2D projection of 3D objects. To foresee the problem explicitly, let us write electronic charge density, ρ, as:

$$0 = F[x, y, z; \rho], \tag{1}$$

where x, y, z span the whole space, while ρ goes from its minimal to its maximal value. Usually the electronic charge density is represented as an isosurface:

$$\rho_0 = f(x, y, z), \tag{2}$$

where ρ_0 has a chosen constant value. Another commonly useful representation of the charge density is a contour plot on a selected plane:

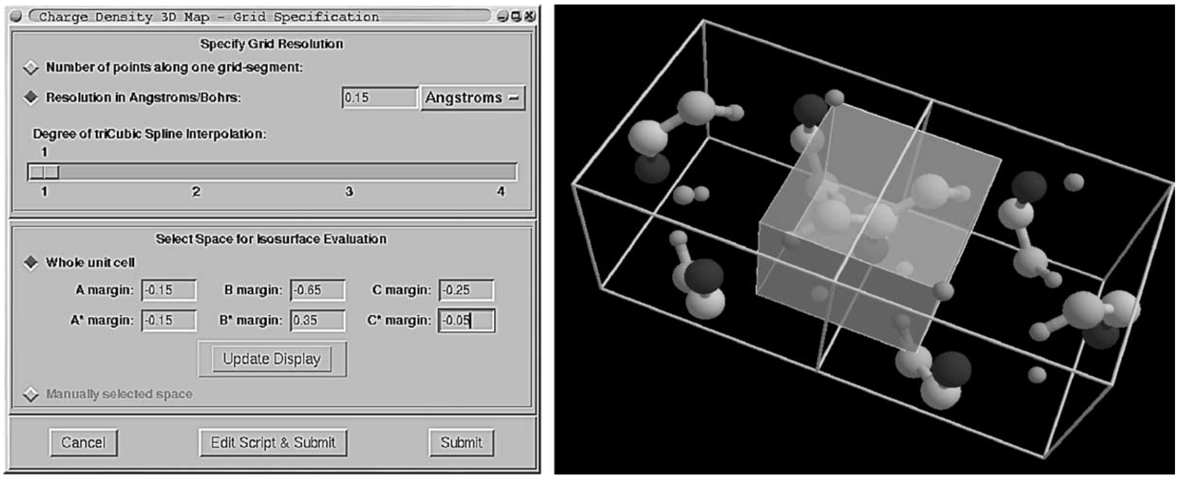
$$z'_0 = f(x', y'; \rho), \tag{3}$$

where z'_0 has a given constant value. Here x', y', and z' constitute the transformed coordinate system, chosen in such a way that the selected plane is the $\widehat{x'y'}$ plane. In Fig. 7 both representations (Eqs. (2) and (3)) are superimposed on the chemical structure.

In both cases, animation can be used to provide an impression of the fourth dimension, which was set as a constant (ρ_0 or z'_0). Such animation is easily performed with XCrySDen program.

3.4.2. Displaying the properties on a selected plane

Some properties are particularly suited to be displayed as contours (isolines) or colourplanes.


Charge Density 3D Map - Grid Specification

Specify Grid Resolution

Number of points along one grid-segment: [ ]
[✓ Resolution in Angstroms/Bohrs:] 0.15 Angstroms

Degree of triCubic Spline Interpolation: 1

Select Space for Isosurface Evaluation

[✓ Whole unit cell]

A margin:	-0.15	B margin:	-0.65	C margin:	-0.25
A* margin:	-0.15	B* margin:	0.35	C* margin:	-0.05

Manually selected space [ ]

Fig. 6. Snapshot of selection of a region for grid calculation of a given property for a molecular crystal of urea. The selected region is shown as a transparent box and comprises one complete urea molecule.

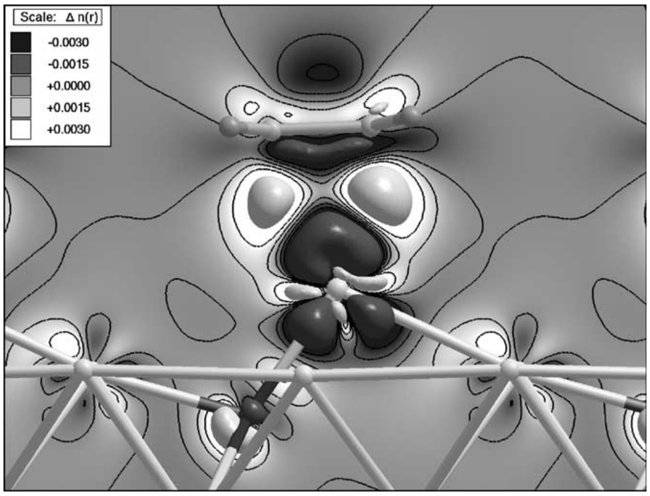


Fig. 7. 2D (contours) and 3D (isosurfaces) graphical representations of the electron deformation density, $\Delta n(\mathbf{r}) = n_{\text{Eth/Substrate}}(\mathbf{r}) - [n_{\text{Substrate}}(\mathbf{r}) + n_{\text{Eth}}(\mathbf{r})]$, of ethylene adsorbed on a defective site of a silver surface [15]. The charge flows from dark to bright regions.

By colourplane is meant a plane where a property is not represented by isolines, but by colours. Several colour schemes are available (black-and-white, rainbow, red–green–blue, cyan–magenta–yellow, and geographic—i.e. blue–green–yellow–brown–violet–white). Contour plots can easily be achieved, for example, by a function and data plotting program such as *gnuplot* [16]. But the advantages of XCrySDen are: (i) a contour plane can be graphically selected, (ii) contour plots can be superimposed on the crystalline structure in order to enhance the readability of the plot; (iii) the whole picture can be interactively rotated and zoomed in real time, and displayed in the desired perspective; (iv) contours can be displayed as colourplanes or colourplanes with isolines; and (v) in the case of crystal structures, such plots have a periodic attribute.

3.4.3. Displaying the isosurfaces

XCrySDen can display an isosurface from any uniform 3D grid of points—the grid does not need to be orthogonal. The algorithm used to achieve the polygonalisation of the scalar field (i.e. triangulation) is the well known *Marching Cubes* [17]. XCrySDen uses the algorithm following the recipe of Bourke [18]. The isosurfaces can be visualised in several display modes including solid, wire, and dotted representations. It is possible to set the isosurface as transparent. XCrySDen also allows various OpenGL parameters such as materials properties to be set.

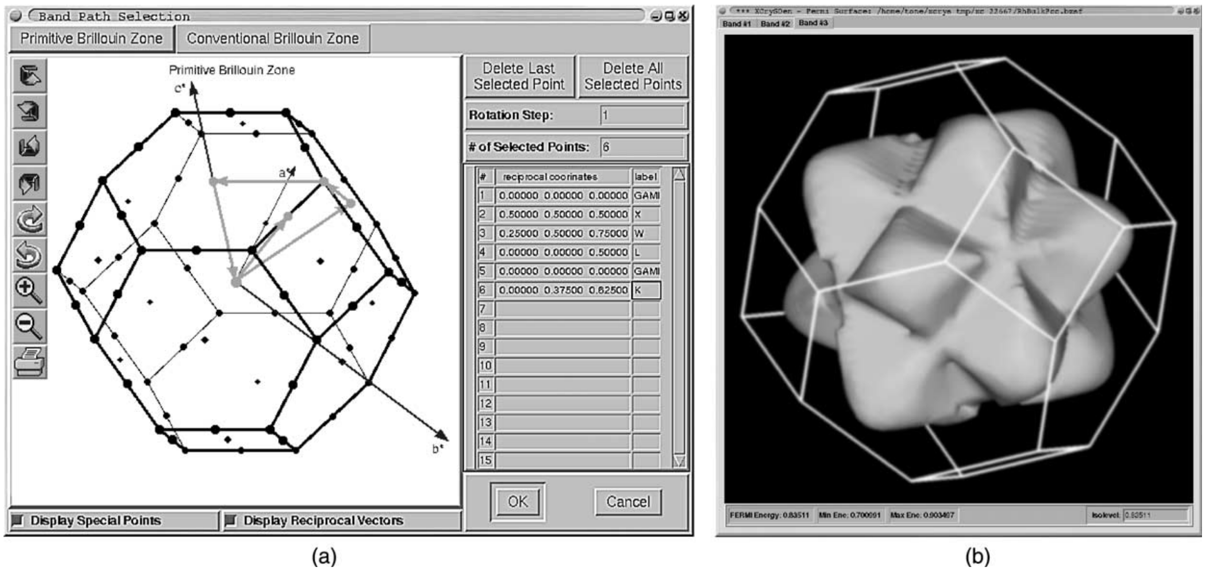
Again, as in the case of 2D contour/colourplane plots, the isosurfaces can be superimposed on the crystalline structure. XCrySDen can display isosurfaces of two different isovalues and three different contour/colourplane planes simultaneously. This is more than enough for most cases, because more information on a single plot would result in complete unreadability.

3.4.4. Analysis in reciprocal space

It is possible to perform some reciprocal space analyses with XCrySDen. In particular, two useful features are implemented. The first is the possibility of selecting a k-path in the Brillouin zone for the band structure plots. For this purpose a special widget is provided which displays a Brillouin zone, and the path along the high symmetry points can be selected interactively by clicking the points with a mouse (see Fig. 8a). The second feature is the visualisation of Fermi surfaces (see Fig. 8b).

3.5. Additional information

Additional up-to-date information about the XCrySDen package is available at the XCrySDen WWW home page, http://www.xcrysden.org/.


Band Path Selection
Primitive Brillouin Zone
Conventional Brillouin Zone
Primitive Brillouin Zone
c*
a*
b*
Delete Last Selected Point
Delete All Selected Points
OK
Cancel
Display Special Points
Display Reciprocal Vectors
(a)
*** XCrySDen - Fermi Surface: /home/tone/xcrys tmp/xc 22667/RhBulkFcc.bxsf
Band #1
Band #2
Band #3
(b)

Rotation Step: 1

# of Selected Points: 6

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

FERMI Energy: 0.83511 Min Ene: 0.700091 Max Ene: 0.903497 Isolevel: 0.83511

Fig. 8. (a) A widget specialised for k-path selection; (b) an example of Fermi surface plotting. The Fermi surface of fcc rhodium crystal associated with one of the bands that cross the Fermi level is shown.

4. The GUIB project

As already stated in Section 2, the utilisation of the computer's graphics capabilities in chemistry, physics and materials science is not solely to visualise properties and structures, but also to provide intuitive graphical user interfaces, GUIs, that facilitate management of the daily use of numerical simulation programs. Despite the fact that several computer languages and libraries exist that enable easy GUI creation and that specialised graphical programs aimed at GUI construction are available, the creation of GUIs for the numerical simulation software is still a task that requires major effort. This is one of the reasons why only a minority of free or low-cost numerical simulation software packages devoted to the electronic structure of periodic systems have the GUI.

A GUI that is aimed solely at creating input files for numerical simulation programs is a simple yet very powerful and useful one. It is appreciated particularly by newcomers, who have to struggle with the new input syntax. A significant fraction of numerical simulation software is written in Fortran. The input files for such programs consist of several records, which usually follow one another sequentially and have well defined formats. In some cases such inputs consist of the sequence of records composed of digits—each digit having a special meaning to the program. Hence, one is forced, either to memorise the input syntax by heart, or to browse the manual to ascertain their meaning. This situation is not well suited for users, but parsing of such inputs is easily programmed. The `GUIB` project [8]—the name stands for graphical user interface builder—has emerged exactly on the basis of these considerations, with the goal of providing a solution to the problem. Since the structure of input files for numerical simulation programs are usually simple from the computer perspective, this can be well exploited, as done in the `GUIB`. The `GUIB` software was written in Incr Tcl/Incr Tk [19]—an object oriented extension of Tcl. The idea behind the `GUIB` project was to construct a special meta-language with two purposes. The first is to define the syntax of the input file of a given numerical program, and the second is to simultaneously provide automatic construction of the GUI on the basis of this definition. This is achieved by providing a definition file which defines the two entities (input syntax and GUI). Such a GUI then facilitates the management of the input files (creation of new inputs, editing of existing ones). Since `GUIB` package is nothing more than a set of keywords (more precise methods) on top of the Tcl language, the syntax of the `GUIB` definition file is that of the Tcl language. In addition to `GUIB` methods the whole set of Tcl commands can used. The following example illustrates the idea. Consider the following input for a particular program aimed at simulating electronic structure of crystalline structures:

```
K_POINTS automatic
8 8 8
1 1 1
```

This input requests the generation of the *k*-point mesh (first line of input) using an $8 \times 8 \times 8$ uniform grid (second line of input) with the shift of the mesh in all three directions (third line of input). To create an appropriate GUI with the `GUIB` software is fairly simple. The corresponding `GUIB` definition file is shown in Example 1, and the resulting GUI is shown in Fig. 9. The definition file of Example 1 defines both the syntax of the input file for our program as well as the GUI itself. This definition file starts with the `module` keyword, which includes the whole definition. In the definition file three `GUIB` keywords are used: `line`, `keyword`, and `var`. The `line` keyword stands for the line of input. The first line of the above input example consists of `K_POINTS` keyword and a variable. The value of the variable can be either `automatic` or `gamma`. The corresponding `GUIB` definition is specified by the keyword `line` (i.e. `line -name` "`K-point input`" ...). The `K_POINTS` keyword is specified by `keyword K_POINTS`, and the variable by the `var` keyword and its options. Although the value of the variable can be either `automatic` or `gamma`, more user readable strings were specified for the GUI (compare `-textvalue` and `-value` options). The option `-widget` requests the type of widget to be displayed in GUI for this variable. The second line of input consists of three positive integer numbers that define the *k*-mesh (one number per dimension). The corresponding `GUIB` definition is again specified by the keyword `line` (i.e. `line -name` "`K-point mesh`"...). Inside the line-definition three `var` keywords are specified. The program (as well as the GUI) expects positive integer numbers, therefore the `-validate posint` option is specified. A similar `GUIB` definition follows for the

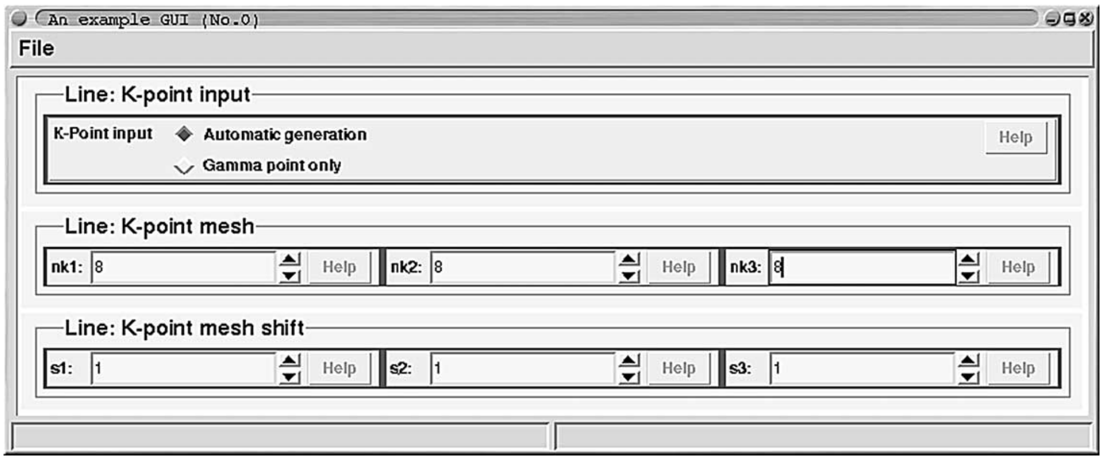
Example 1
A simple GUIB definition file

```
module example1\#auto -title "An example GUI" -script {

    line -name "K-point input" {
        keyword K_POINTS
        var {
            -label     "K-Point input"
            -variable  kpoint_type
            -textvalue { "Automatic generation" "Gamma point only" }
            -value     { automatic gamma }
            -widget    radiobox
        }
    }

    line -name "K-point mesh" {
        var { -label "nk1:"  -variable nk1  -widget spinint  -validate posint  -default 1 }
        var { -label "nk2:"  -variable nk2  -widget spinint  -validate posint  -default 1 }
        var { -label "nk3:"  -variable nk3  -widget spinint  -validate posint  -default 1 }
    }

    line -name "K-point mesh shift" {
        var { -label "s1:"  -variable s1  -widget spinint  -validate posint  -default 1 }
        var { -label "s2:"  -variable s2  -widget spinint  -validate posint  -default 1 }
        var { -label "s3:"  -variable s3  -widget spinint  -validate posint  -default 1 }
    }
}
```



An example GUI (No.0)

File

Line: K-point input

K-Point input ◆ Automatic generation

◇ Gamma point only

Line: K-point mesh

nk1: 8

nk2: 8

nk3: 8

Line: K-point mesh shift

s1: 1

s2: 1

s3: 1

Fig. 9. A snapshot of the GUI constructed by the GUIB software using the code from Example 1.

third line of input (see line -name "K-point mesh shift"...).

The GUIB provides more keywords than just the three above mentioned. It also supports an event-driven mechanism, and special keywords are provided for this purpose. A typical use of this is the situation where, on setting a particular value of a given variable, the program expects one type of proceeding input, while for another value of the variable the input is of another type. In the previous example, on selecting the "*Gamma point only*" option, one does not need to input the parameters for the automatic *k*-mesh generation. Hence, by the event-driven mechanism GUIB can enable certain widgets and disable others.

The GUIB software has been already used in implementing GUI for the program PWscf. [2] Fig. 10 shows the main window of the PWscf's GUI. Although PWscf is a complex program, with very rich functionality and a lot of options, the code for GUI was easily and quickly written and does not exceed 2000 lines.

[2] The first release of TkPWscf [20]—a GUI of the PWscf—was made available during the Winter College on Numerical Methods in Electronic Structure Theory, taking place in Abdus Salam International Centre for Theoretical Physics in Trieste, Italy, in January 2003.

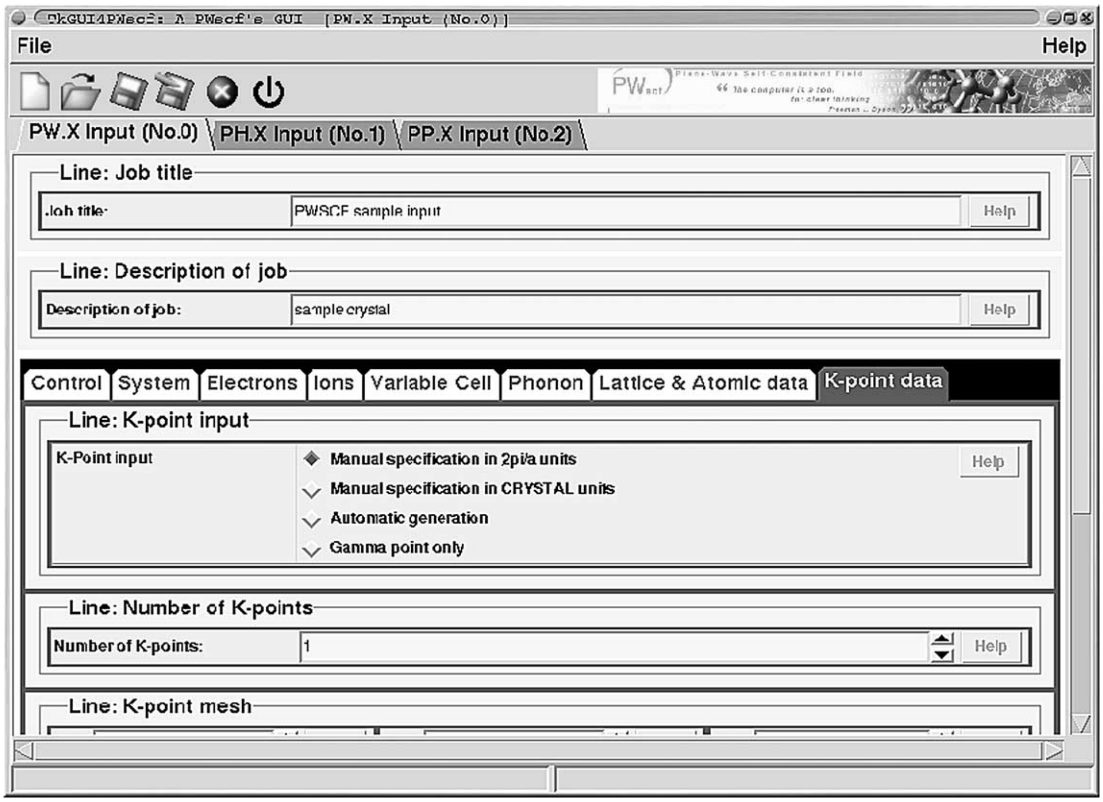
5. Conclusions

The role of computer graphics in different aspects of simulating matter at the atomic scale has been addressed. The computer graphics is useful not only to visualise chemical structures, but also for their specification and examination. Computer graphics is an indispensable tool in the analysis of computed data and makes the interpretation of results easier. In this context the XCrySDen program was presented.

The utilisation of the computer's graphics capabilities in providing graphical user interfaces is of great importance. Graphical interfaces are powerful and versatile means of communication between a user and a computer. GUIs offer an intuitive approach in the simulation setup. A useful, though simple, example is the use of GUIs for creating input files for the numerical simulation programs. Therefore, GUIB has recently been developed. Its purpose is to provide a simple mechanism for constructing GUIs that aims solely at the creation of input files for numerical simulation programs.

Acknowledgements

The author thanks Prof. Mauro Causà for motivation and suggestions in the implementation of XCrySDen. The author also thanks Prof. Ivan


TkGUI4PWscf: A PWscf's GUI [PW.X Input (No.0)]

File Help

PW.X Input (No.0) | PH.X Input (No.1) | PP.X Input (No.2)

Line: Job title
Job title: PWSCF sample input

Line: Description of job
Description of job: sample crystal

Control | System | Electrons | Ions | Variable Cell | Phonon | Lattice & Atomic data | K-point data

Line: K-point input
K-Point input: [✓ Manual specification in 2pi/a units] Manual specification in CRYSTAL units / Automatic generation / Gamma point only

Line: Number of K-points
Number of K-points: 1

Line: K-point mesh

Fig. 10. Main window of the *TkPWscf* application—a GUI of the `PWscf` program.

Kobal for useful discussions and careful reading of the manuscript. This work was supported financially by the Ministry of Education, Science and Sport of Slovenia. The work is also associated with activities in INFM DEMOCRITOS National Simulation Center from Trieste, Italy, as a part of research "Information technology for molecular modelling and computation".

Appendix A. Short description of `XCrySDen` XSF format

The XSF format is internal `XCrySDen` structure format. The name XSF stands for **X**crysden **S**tructure **F**ile. It is used to describe (i) molecular and crystal structures, (ii) forces acting on constituent atoms, and (iii) scalar fields (for example: electronic charge density, electrostatic potential). Here the specification of the first two items is briefly described. The main attributes of XSF format are:

- all records are in free format;
- the XSF file is composed of various sections;
- each section begins with the keyword (keywords are case-sensitive);
- the length unit is Å, and that for force is Hartree/Å.

A.1. Specification of molecular structure

For molecules the XSF format is very simple. The first line begins with the `ATOMS` keyword and then one specifies the atomic numbers (or symbols) and Cartesian coordinates and possibly forces for all atoms. An entry for an atom looks like:

```
AtNum  X  Y  Z  [Fx  Fy  Fz]
```

Example 2
Molecular XSF file for the water molecule

```
ATOMS
1        0.000000     0.772651    -0.555141
8        0.000000     0.000000     0.138785
1        0.000000    -0.772651    -0.555141
```

where AtNum stands for atomic number (or symbol), while X, Y, and Z are Cartesian coordinates in Å units. The symbols Fx, Fy, and Fz stands for force components for the three Cartesian directions. The force records are optional (indicated by the square braces above). A simple example of molecular XSF file is shown in Example 2 for water molecule.

A.2. Specification of crystal structure

In order to explain the specification of crystal structure with XSF format, we start with the example of fcc ZnS crystal (see Example 3). The example file begins with the CRYSTAL keyword, which indicates that the structure is a crystal (i.e. three-dimensional periodic structure). Two other possibilities are SLAB, and POLYMER, which represent 2D and 1D periodic structures, respectively. The lattice vectors are then specified. XSF format accepts two different settings of the unit cell. These are usually called the *primitive* and the *conventional* unit cell. The corresponding keywords are PRIMVEC for primitive lattice vectors and CONVVEC for conventional lattice vectors. [3] Nine real numbers written in three lines—one lattice vector per line—are following the PRIMVEC and CONVVEC keywords. The PRIMVEC section is mandatory, whereas the CONVVEC section is optional.

After specifying the two sets of lattice vectors, atoms belonging to the primitive unit cell are specified by the PRIMCOORD keyword. The line following consists of two digits. The first is the number of atoms in the primitive cell, while the second is always 1 for PRIMCOORD coordinates. Then follows the specification of each atom in the primitive cell (the same format as for the atom definition in the ATOMS section described above).

[3] In the XSF file the primitive and conventional settings of the unit cell are presumed to have the same origin. To describe a given crystal structure with two settings of the unit cell that have different origins, two XSF files should be created—one for each cell setting.

Example 3
Crystal XSF file for ZnS crystal structure

```
CRYSTAL
PRIMVEC
   0.0000000    2.7100000    2.7100000
   2.7100000    0.0000000    2.7100000
   2.7100000    2.7100000    0.0000000
CONVVEC
   5.4200000    0.0000000    0.0000000
   0.0000000    5.4200000    0.0000000
   0.0000000    0.0000000    5.4200000
PRIMCOORD
2 1
16      0.0000000     0.0000000     0.0000000
30      1.3550000    -1.3550000    -1.3550000
```

A.3. Animated XSF

XSF format allows several structures to be specified. A typical use of this is an animation of chemical structure evolving in, for example, a molecular dynamics simulation. A simple example of an animated XSF file is shown in Example 4 for the hydrogen molecule, and in Example 5 for a ZnS crystal. The file begins with the keyword ANIMSTEPS, and an integer number follows. This number specifies the number of structures in the

Example 4
Animated XSF file containing several snapshots of a hydrogen molecule (aimed at animation)

```
ANIMSTEPS 3
ATOMS 1
1          0.000000      0.000000     -0.300000
1          0.000000      0.000000      0.300000
ATOMS 2
1          0.000000      0.000000     -0.315000
1          0.000000      0.000000      0.315000
ATOMS 3
1          0.000000      0.000000     -0.335000
1          0.000000      0.000000      0.335000
```

Example 5
Animated XSF file containing two snapshots of ZnS crystal structure (aimed at animation)

```
ANIMSTEPS 2
CRYSTAL
PRIMVEC
   0.0000000    2.7100000    2.7100000
   2.7100000    0.0000000    2.7100000
   2.7100000    2.7100000    0.0000000
PRIMCOORD 1
2 1
16       0.0000000     0.0000000     0.0000000
30       1.3550000    -1.3550000    -1.3550000
PRIMCOORD 2
2 1
16       0.0000000     0.0000000     0.0000000
30       1.2550000    -1.2550000    -1.2550000
```

file. In animated XSF, the `ATOM` and `PRIMCOORD` keywords are postfixed with an integer number, which is the sequential index of the structure.

References

[1] A.R. Leach, Molecular Modelling: Principles and Applications, Addison Wesley Longman Limited, Edinburgh, England, 1996.

[2] A. Kokalj, J. Mol. Graphics Modell. 17 (1999) 176.

[3] A. Kokalj, M. Causà, XCrySDen: (X-window) CRYstalline Structures and DENsities, 2003. Available from: <http://www.xcrysden.org/>.

[4] V.R. Saunders, R. Doves, C. Roetti, M. Causá, N.M. Harrison, R. Orlando, C.M. Zicovich Wilson, Crystal98 1.0—user's manual, University of Torino, Torino, 1999.

[5] P. Blaha, K. Schwarz, G. Madsen, D. Kvasnicka, J. Luit, WIEN2k, an augmented plane wave + local orbitals program for calculating crystal properties, Karlheinz Schwarz, Techn. Universität Wien, Austria, 2001.

[6] S. Baroni, A. Dal Corso, S. de Gironcoli, P. Giannozzi, PWSCF and PHONON: plane-wave pseudo-potential codes, 2001. Available from: <http://www.pwscf.org/>.

[7] M. Bockstedte, A. Kley, J. Neugebauer, M. Scheffler, Comput. Phys. Commun. 107 (1997) 187.

[8] A. Kokalj, GUIB: Graphical User Interface Builder. Available from: <http://www-k3.ijs.si/kokalj/guib/>.

[9] OpenGL—high performance 2d/3d graphics, The official WEB site of OpenGL is <http://www.opengl.org/>.

[10] B. Paul, The Mesa 3d graphics library, The official WEB site of Mesa is <http://www.mesa3d.org/>.

[11] B. Paul, B. Bederson, Togl—a Tk Open GL widget, The official WEB site of Togl is <http://togl.sourceforge.net/>.

[12] XYZ format was the native format of the XMOL program developed by the Network Computing Services, Inc. The XMOL program is not developed anymore.

[13] Brookhavens protein databank PDB format, The official WEB site of PDB is <http://www.rcsb.org/pdb/> 1999.

[14] M.J. Frisch, G.W. Trucks, H.B. Schlegel, P.M.W. Gill, B.G. Johnson, M.A. Robb, J.R. Cheeseman, T. Keith, G.A. Petersson, J.A. Montgomery, K. Raghavachari, M.A. Al-Laham, V.G. Zakrzewski, J.V. Ortiz, J.B. Foresman, C.Y. Peng, P.Y. Ayala, W. Chen, M.W. Wong, J.L. Andres, E.S. Replogle, R. Gomperts, R.L. Martin, D.J. Fox, J.S. Binkley, D.J. Defrees, J. Baker, J.P. Stewart, M. Head-Gordon, C. Gonzalez, J.A. Pople, Gaussian, Inc., Pittsburgh PA, 1995.

[15] A. Kokalj, A. Dal Corso, S. de Gironcoli, S. Baroni, J. Phys. Chem. B 106 (2002) 9839;
A. Kokalj, A. Dal Corso, S. de Gironcoli, S. Baroni, Surf. Sci. 532–535 (2003) 191.

[16] T. Williams, C. Kelley, Gnuplot—an interactive plotting program, The official WEB site of Gnuplot is <http://www.gnuplot.info/>.

[17] W. Lorensen, H. Cline, Comput. Graphics 21 (1987) 163.

[18] P. Bourke, Modelling, Accessible on WEB site: <http://astronomy.swin.edu.au/~pbourke/modelling/>.

[19] Information about [Incr Tcl] can be found on the Web site: <http://incrtcl.sourceforge.net/itcl/> while information about [Incr Tk] can be found on Web site: <http://incrtcl.sourceforge.net/itk/>.

[20] A. Kokalj, TkPWscf—a GUI for the PWscf, Available from: <http://www-k3.ijs.si/kokalj/tkpwscf/>.