JOURNAL OF
APPLIED
CRYSTALLOGRAPHY

ISSN: 1600-5767
journals.iucr.org/j

X-Seed 4: updates to a program for small-molecule supramolecular crystallography

Leonard J. Barbour

J. Appl. Cryst. (2020). **53**, 1141–1146

JOURNAL OF APPLIED CRYSTALLOGRAPHY

ISSN 1600-5767

Received 30 May 2020
Accepted 3 June 2020

Edited by S. Moggach, The University of Western Australia, Australia

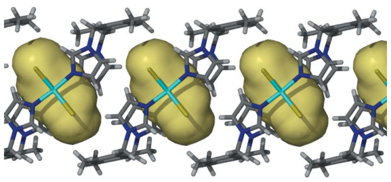
Keywords: graphical user interfaces; *SHELX*; *POV-Ray*; *MSRoll*; presentation-quality figures.

X-Seed 4: updates to a program for small-molecule supramolecular crystallography

Leonard J. Barbour*

Department of Chemistry and Polymer Science, University of Stellenbosch, Matieland, 7600, South Africa. *Correspondence e-mail: ljb@sun.ac.za

X-Seed is a native Microsoft Windows program with three primary functions: (i) to serve as a graphical user interface to the *SHELX* suite of programs, (ii) to facilitate exploration of crystal packing and intermolecular interactions, and (iii) to generate high-quality molecular graphics artwork suitable for publication and presentation. Development of *X-Seed* Version 1.0 began in 1998, when point-and-click crystallographic software was still limited in scope and power. Considerable enhancements have been implemented within *X-Seed* over the past two decades. Of particular importance are support for the *SHELX2019* programs (*SHELXS*, *SHELXD*, *SHELXT* and *SHELXL*) for structure solution and refinement, and *MSRoll* for rendering void spaces in crystal structures. The current version (*i.e.* Version 4) of *X-Seed* has a new interface designed to be more interactive and user friendly, and the software can be downloaded and used free of charge.



1. Introduction

Since the pioneering contributions of von Laue and the Braggs just over a century ago, the field of small-molecule crystallography has benefitted from many significant innovations. Given the vast number of calculations involved, two of the most important milestones occurred in the 1960s, with the advent of computer-controlled diffractometers and the use of digital computers for structure solution and refinement. Indeed, crystallographers were among the first scientists to embrace the use of electronic computers (Frenz, 1988).

Another landmark innovation took place in the 1990s, with the introduction of CCD-based area detectors to replace point detectors. Over the past 30 years substantial competition between instrument manufacturers has prompted dramatic advances in beam intensity (*e.g.* microfocus sources with improved beam optics), detector sensitivity and data processing algorithms. These developments, coupled with the increased availability of laboratory-based X-ray diffractometers and continuous improvements in computing power, have resulted in an explosion of crystal structure reports in the literature. One of the consequences of progress towards accessibility and high throughput is that the average crystallographer is no longer a specialist with a deep understanding of the subject. Productivity demands that we focus mainly on the chemistry while relying on advanced interactive software to navigate the process of structure solution, refinement and validation.

1.1. Structure solution and refinement using *SHELX* in the 1990s

Since the dawn of crystallographic computing, a number of different programs have been employed for crystal structure solution and refinement. Arguably, by far the most commonly used of these belong to Sheldrick's *SHELX* suite (Sheldrick, 2008). The *SHELX* programs are coded in standard Fortran such that they can be implemented on a wide range of operating systems as console applications. Free-form instructions and data are retrieved from text files, and the programs also produce text files as output. The script-like input provides the user with a powerful but (by current standards) user-unfriendly interface, particularly for the iterative process of structure refinement.

In 1990 the author was first introduced to small-molecule crystal structure solution and refinement using the programs *SHELXS86* (Sheldrick, 1985*a*,*b*) and *SHELX76* (Sheldrick, 1976), respectively. In subsequent years these programs were superseded by the solution and refinement suites *SHELX90*, *SHELX93* and *SHELX97* (Sheldrick, 1990, 1993; Robinson & Sheldrick, 1988; Sheldrick, 1997). However, all versions of *SHELX* have retained the well established mode of employing textual input and output.

A typical crystal structure analysis proceeded as follows. Intensity data were recorded, processed and then output to a `name.hkl` file. Possible space groups were inferred from unit-cell parameters and inspection of reflection intensities to assess Laue symmetry and systematic absences. Using a suitable text editor, a corresponding *SHELXS* input file `name.ins` was prepared manually for structure solution by one of three methods: direct methods, application of the Patterson function or partial-structure expansion. *SHELXS* was then invoked using `name` as a command-line parameter. Upon completion, *SHELXS* produced output in the form of `name.lst` and `name.res` text files. The LST file could be printed (preferably using a 136 column printer) since it contained useful feedback pertaining to the structure solution. In particular, the LST file included a crude projection of the electron density peaks; using a pencil, a 2D projection of the crystal structure could be generated by drawing lines between the numbered peaks, based on distance and angle information in the LST file. This primitive form of molecular graphics allowed the user to associate specific electron density peaks with molecular geometry and to thus assign atom types to the peaks. For convenience, the RES file was formatted as a *SHELXL* INS file, but with electron density peaks tentatively listed as carbon atoms with names (or codes) consisting of 'Q' followed by a number (the lower the number, the more intense the peak). On the basis of the molecular geometry deduced from the line-printer plot, the user manually edited the peak list in the RES file by changing atom codes and element types. If desired, the atoms list was sorted using the cut-and-paste method. The RES file was then renamed to `name.ins` and used as input for refinement using *SHELXL* (with the addition of suitable *SHELXL* instructions). Structure refinement then entailed gradual improvement of the model by means of several successive calls to *SHELXL*; using information embedded in the *SHELXL* LST file, the user could expand the structure by assigning additional peaks as atoms (*e.g.* modelling of solvent or disorder). Other gradual refinements consisted of modelling atoms with anisotropic displacement parameters, and adding hydrogen atoms using the well defined geometrical 'riding' models implemented by *SHELXL*. Each of these iterations of *SHELXL* consisted of editing the current RES file and renaming it to `name.ins` for further refinement.

1.2. *RES2INS*: the precursor to *X-Seed*

The most tedious part of the iterative procedure required by *SHELXL* entailed the use of static 2D projections in the form of line-printer plots to visualize 3D molecules, and then manual editing and sorting of atoms in a text file (*i.e.* converting a RES file to the next INS file). In 1997 the author developed a simple DOS program to ease this process. *RES2INS* (Barbour & Atwood, 1998) read a RES file produced by *SHELXS93* or *SHELXL93* and displayed the model on screen as a colour-coded wireframe drawing. The only other graphical user interface (GUI) to *SHELX* at that time was *XP*, which was developed by Sheldrick as part of the Bruker proprietary software package *SHELXTL* (Bruker, 1998). Using *RES2INS*, the model could be rotated on screen using the mouse, and the point-and-click philosophy was used to assign element types and to delete unwanted atoms/peaks (a process commonly referred to as 'tidying the structure'). Other point-and-click editing features included assigning atoms for isotropic/anisotropic refinement, adding hydrogen atoms in calculated positions, and generation of centroids that could be converted to atoms. To ease visualization and understanding of molecular geometry, the program also implemented a simple 'grow' routine that allowed expansion of a molecular fragment using space-group symmetry. The atom list could be sorted according to the numeric components of the atom names. Once the model had been edited, the user could save the atom list and associated *SHELXL* commands as an INS file, which could be edited manually (if necessary) before the next call to *SHELXL*. *RES2INS* could be implemented either as a standalone Microsoft DOS program or in a full-screen DOS window within Microsoft Windows. In these DOS environments, the *SHELX* executable programs were launched manually using the command line. Although *RES2INS* did not take advantage of the full power of *SHELXL*, it was highly successful as a rudimentary GUI to facilitate some of the most tedious model editing tasks. *RES2INS* was developed as a native DOS program using Borland's Turbo Pascal Version 5.5 integrated development environment (IDE). The Turbo Pascal graphics libraries were quite limited in scope, and coding the graphics engine and user-interface dialogues of *RES2INS* was an onerous task. Thus, the author's enthusiasm for developing *RES2INS* into a more useful program was limited, especially given that native Windows programs were beginning to supersede DOS programs on IBM-compatible PC platforms.

1.3. *X-Seed 1.0*

By the mid- to late-1990s, Microsoft Windows had evolved into an operating system rather than simply being a program running within DOS (at least, it became less obvious that Windows 95 and 98 were still DOS programs). The important consequences were that multitasking became possible on increasingly powerful IBM-compatible PCs, and the mouse became an indispensable peripheral device. Moreover, IDEs such as Borland's Delphi and Microsoft's Visual C++ (among others) simplified the development of native Windows applications with standardized libraries for menus, dialogue boxes and event handlers. In 1998, *X-Seed 1.0* emerged as a Windows version of *RES2INS*, rewritten using the Delphi 3 IDE. The author's objective for *X-Seed* was to completely eliminate the user's need to manually prepare *SHELX* INS files; *X-Seed* even launched *SHELXS* and *SHELXL* and, upon completion of these processes, automatically loaded the outcome from the RES file. The structural model was mainly edited using mouse operations, and provision was also made for supporting almost all of the *SHELX* commands. It soon became apparent that the utility of *X-Seed* could be extended significantly by adding several features that were relatively simple to implement. Major additions included (i) growing the structure on demand (*i.e.* application of relevant symmetry operators to the asymmetric unit according to the assigned space group) to aid the user in understanding packing modes and intermolecular interactions, and (ii) production of molecular graphics (Atwood & Barbour, 2003). These adaptations occurred concurrently with the early development of *WinGX* (Farrugia, 1999), and several years before the first release of *Olex* (Dolomanov *et al.*, 2003), which later became the highly popular *Olex2* package (Dolomanov *et al.*, 2009).

1.4. Evolution of *X-Seed*: updates and availability of Version 4

Since a brief description of *X-Seed 1.0* was published in 2001 (Barbour, 2001) a number of significant updates have been implemented. The two most important of these are the addition of an *MSRoll* (Connolly 1983*a*,*b*; Connolly, 1993) interface and support for the significantly updated versions of the *SHELX* suite (*i.e.* *SHELX2013* through *SHELX2019*), including the two programs *SHELXD* (Usón & Sheldrick, 2018) and *SHELXT* (Sheldrick, 2015). Although *X-Seed 1.0* was distributed commercially, Versions 2 and above have been distributed free of charge since 2010.

2. Utilization

The purpose of this contribution is to provide a succinct overview of the *X-Seed 4* package, and the reader is referred to the software manual for a more comprehensive description of features. *X-Seed* reads a *SHELX* INS or RES file and displays the model as a line drawing that can be rotated or zoomed using the mouse or keyboard. A more aesthetically pleasing OpenGL display of the model was considered. However, one of the important features of *X-Seed* involves growing extended structures that may include tens of thousands of atoms; in such cases, an OpenGL display would become unacceptably slow in responding to on-screen rotation and manipulation. A decision was therefore made to keep the representation of the model as simple and responsive as possible by implementing bitmap double buffering.

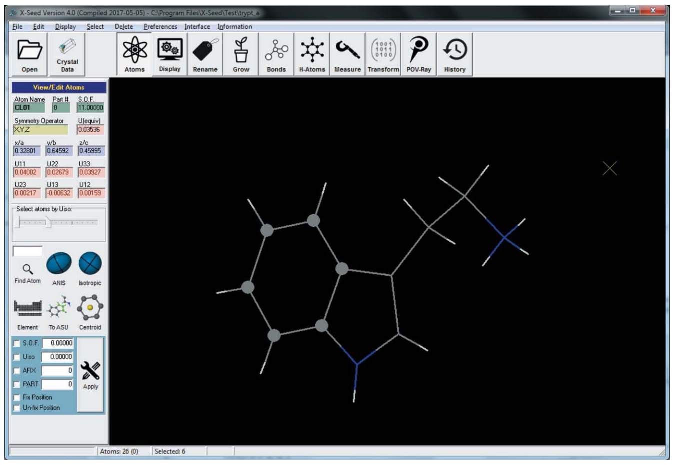


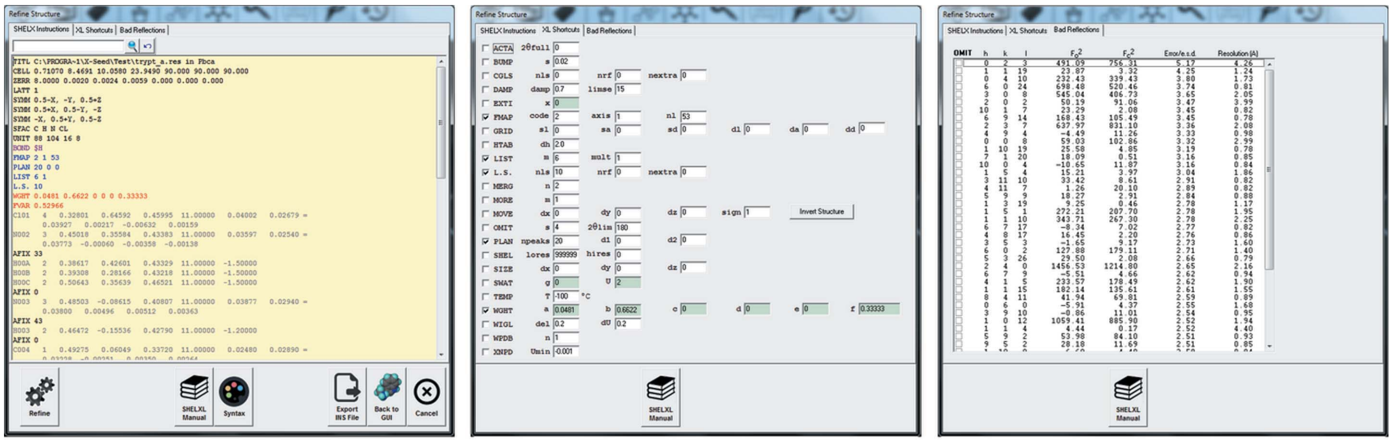
Figure 1
The *X-Seed 4* window, displaying the model of the test structure. The six carbon atoms of the benzo group are selected.

X-Seed uses the 'select . . . apply settings' philosophy. When a structural model is displayed, the user can select or deselect an atom or group of atoms. A large number of different operations can then be applied to the selected atoms using standard Windows controls such as drop-down menus and dialogue box elements – some examples include deleting or hiding atoms, creating centroids, marking atoms for anisotropic refinement, changing element types, generating symmetry-equivalent atoms, setting colours and radii for *POV-Ray* (Persistence of Vision Team, 1994) output, *etc*. Since the ability to select atoms is central to the effective use of *X-Seed*, there are several ways in which atoms or groups of atoms can be selected or deselected (*e.g.* individually, or collectively by element type, *SHELX* PART number, molecular fragments, symmetry relationship to the asymmetric unit *etc*.). Atoms can also be selected using a mouse lasso function, and bonds can be selected for various bond-related operations. Fig. 1 shows the *X-Seed* window with six atoms selected.

The three primary functions of *X-Seed* are as follows: (i) to serve as a GUI for the *SHELX* suite of programs, (ii) to facilitate exploring crystal structures, particularly with regard to crystal packing and intermolecular interactions, and (iii) to generate high-quality molecular graphics artwork suitable for publication and presentation. These features, particularly with regard to more recent additions, are described below.

2.1. *SHELX* Interface

The first three versions of *X-Seed* only implemented the *SHELX97* programs *SHELXS* and *SHELXL*. The *SHELX* instruction files created by *X-Seed* could not be edited directly; instead, *SHELX* commands were accessed using dialogue boxes and memo controls, and the structural model was formulated on the basis of the settings applied to the individual or groups of atoms. *X-Seed 4* makes provision for new *SHELX* commands from *SHELX2013* through *SHELX2019*.



```
TITL C:\PROGRA~1\X-Seed\Test\trypt_a.res in Pbca
CELL 0.71070 8.4691 10.0580 23.9490 90.000 90.000 90.000
ZERR 8.0000 0.0020 0.0024 0.0059 0.000 0.000 0.000
LATT 1
SYMM 0.5-X, -Y, 0.5+Z
SYMM 0.5+X, 0.5-Y, -Z
SYMM -X, 0.5+Y, 0.5-Z
SFAC C H N CL
UNIT 88 104 16 8
BOND $H
FMAP 2 1 53
PLAN 20 0 0
LIST 6 1
L.S. 10
WGHT 0.0481 0.6622 0 0 0 0.33333
FVAR 0.52966
C101   4   0.32801   0.64592   0.45995  11.00000   0.04002   0.02679 =
           0.03927   0.00217  -0.00632   0.00159
N002   3   0.45018   0.35584   0.43383  11.00000   0.03597   0.02540 =
           0.03773  -0.00060  -0.00358  -0.00138
AFIX 33
H00A   2   0.38617   0.42601   0.43329  11.00000  -1.50000
H00B   2   0.39308   0.28166   0.43218  11.00000  -1.50000
H00C   2   0.50643   0.35639   0.46521  11.00000  -1.50000
AFIX 0
N003   3   0.48503  -0.08615   0.40807  11.00000   0.03877   0.02940 =
           0.03800   0.00496   0.00512   0.00363
AFIX 43
H003   2   0.46472  -0.15536   0.42790  11.00000  -1.20000
AFIX 0
C004   1   0.49275   0.06049   0.33720  11.00000   0.02480   0.02890 =
```

```
[ ] ACTA  2θfull [0]
[ ] BUMP       s [0.02]
[ ] CGLS     nls [0]       nrf [0]      nextra [0]
[ ] DAMP    damp [0.7]   limse [15]
[ ] EXTI       x [0]
[✓] FMAP    code [2]      axis [1]          nl [53]
[ ] GRID      sl [0]        sa [0]          sd [0]       dl [0]       da [0]       dd [0]
[ ] HTAB      dh [2.0]
[✓] LIST       m [6]      mult [1]
[✓] L.S.     nls [10]      nrf [0]      nextra [0]
[ ] MERG       n [2]
[ ] MORE       m [1]
[ ] MOVE      dx [0]        dy [0]          dz [0]     sign [1]
[ ] OMIT       s [4]    2θlim [180]
[✓] PLAN  npeaks [20]       d1 [0]          d2 [0]
[ ] SHEL   lores [999999] hires [0]
[ ] SIZE      dx [0]        dy [0]          dz [0]
[ ] SWAT       g [0]         U [2]
[ ] TEMP       T [-100] °C
[✓] WGHT       a [0.0481]    b [0.6622]      c [0]        d [0]        e [0]        f [0.33333]
[ ] WIGL     del [0.2]      dU [0.2]
[ ] WPDB       n [1]
[ ] XNPD    Umin [-0.001]
```

OMIT	h	k	l	F_o^2	F_c^2	Error/e.s.d.	Resolution (A)
☐	0	2	3	491.09	756.31	5.17	4.26
☐	1	1	19	23.87	3.32	4.25	1.24
☐	0	4	10	232.43	339.43	3.80	1.73
☐	6	0	24	698.48	520.46	3.74	0.81
☐	3	0	8	545.04	406.73	3.65	2.05
☐	2	0	2	50.19	91.06	3.47	3.99
☐	10	1	7	23.29	2.08	3.45	0.82
☐	6	9	14	168.43	105.49	3.45	0.78
☐	2	3	7	637.97	831.10	3.36	2.08
☐	4	9	4	-4.49	11.26	3.33	0.98
☐	0	0	8	59.03	102.86	3.32	2.99
☐	1	10	19	25.58	4.85	3.19	0.78
☐	7	1	20	18.09	0.51	3.16	0.85
☐	10	0	4	-10.65	11.87	3.16	0.84
☐	1	5	4	15.21	3.97	3.04	1.86
☐	3	11	10	33.42	8.61	2.91	0.82
☐	4	11	7	1.26	20.10	2.89	0.82
☐	5	9	9	18.27	2.91	2.84	0.88
☐	1	3	19	9.25	0.46	2.78	1.17
☐	1	5	1	272.21	207.70	2.78	1.95
☐	1	1	10	343.71	267.30	2.78	2.25
☐	6	7	17	-8.34	7.02	2.77	0.82
☐	4	8	17	16.45	2.20	2.76	0.86
☐	3	5	3	-1.65	9.17	2.73	1.60
☐	6	0	2	127.88	179.11	2.71	1.40
☐	5	3	26	29.50	2.08	2.66	0.79
☐	2	4	0	1456.53	1214.80	2.65	2.16
☐	6	7	9	-5.51	4.66	2.62	0.94
☐	4	1	5	233.57	178.49	2.62	1.90
☐	1	1	15	182.14	135.61	2.61	1.55
☐	8	4	11	41.94	69.81	2.59	0.89
☐	0	6	0	-5.91	4.37	2.55	1.68
☐	3	9	10	-0.86	11.01	2.54	0.95
☐	1	0	12	1059.41	885.90	2.52	1.94
☐	1	1	4	4.44	0.17	2.52	4.40
☐	5	9	2	53.98	84.10	2.51	0.93
☐	9	5	2	28.18	11.69	2.51	0.85

Figure 2
The *SHELXL* dialogue box, showing tabs for (*a*) the *SHELX* instruction set for the next refinement, (*b*) checkboxes and fields for one-off commands, and (*c*) a list of the 50 worst reflections derived from the previous *SHELXL* run, which can be added to the 'OMIT hkl' list.

Importantly, the dialogue boxes from which the *SHELX* programs are launched now display the INS file instructions in text boxes (with syntax highlighting) as they will be parsed to the relevant *SHELX* programs, and the instruction sets are fully user editable.

As before, a typical structure solution and refinement procedure begins with the availability of an HKL file, as well as a *SHELX* INS file containing (at least) unit-cell parameters, space-group information and a molecular formula. After reading the INS file, the user can elect to solve the structure using any of the three programs *SHELXS*, *SHELXD* or *SHELXT*. When the 'Solve' button is pressed, *X-Seed* saves the displayed instructions to an INS file, launches the requested *SHELX* program and parses the real-time *SHELX* screen output to the dialogue box. Upon completion of the *SHELX* process, salient parameters are extracted from the LST file and displayed in a user-friendly format. The user can then either accept or reject the *SHELX* results. If the solution is accepted, the user can proceed to tidying the structure using the point-and-click interface before employing *SHELXL* iteratively to complete the model to convergence.

The *SHELXL* interface (Fig. 2) is similar to that for the three structure solution programs – *i.e.* the instruction set can be edited prior to launching *SHELXL*. A tabbed sheet is also provided for easy access to one-off *SHELX* commands (*e.g.* ACTA, L.S., SIZE *etc.*). Multiple-use (*e.g.* BIND, FLAT, DFIX, EADP *etc.*) and structured (FRAG…FEND) commands can be typed into the instruction set manually, and they will be retained by *X-Seed* during subsequent refinements. Some of the other new features of *X-Seed 4* that are pertinent to the *SHELX* interface include

(i) a direct link to the relevant online *SHELX* manual;

(ii) support for neutron diffraction data;

(iii) the ability to retrieve the *SHELX* solution and refinement history (*i.e.* loading any previous INS or RES file).

X-Seed 4 provides complete support for the commands relevant to *SHELXS*, *SHELXD* and *SHELXT*. It also supports almost all of the *SHELXL* commands, with the exceptions of ANSC, ANSR, LAUE, RESI and SPEC.

2.2. Structure exploration

The related fields of solid-state supramolecular chemistry and crystal engineering often focus on the close relationship between the structure and the properties of a crystal. In order to describe this relationship, the crystallographer is required to explore crystal packing and intermolecular interactions in detail. Since 2000, several excellent computer programs have become available for this purpose in particular; *e.g.* the program *Mercury* (distributed by the Cambridge Crystallographic Data Centre; Macrae *et al.*, 2006, 2008).

Over the past two decades there has been much interest in porous crystalline materials possessing channels or voids that are accessible to potential guest molecules. In describing such materials, it is useful to generate surface maps that represent the space available to a spherical probe of a given radius. The program *MSRoll* was originally developed by Connolly to map solvent-accessible surfaces in protein structures. Since *MSRoll* can also be employed to map guest-accessible space in porous or 'virtually porous' (Barbour, 2006) small-molecule crystal structures, a simple interface was added to *X-Seed* that allows the user to specify a probe radius and then launch *MSRoll*. Upon completion, the guest-occupiable volume is reported, and it is also possible to use the *POV-Ray* interface of *X-Seed* to incorporate the surface map into a high-quality packing diagram (see Fig. 3).

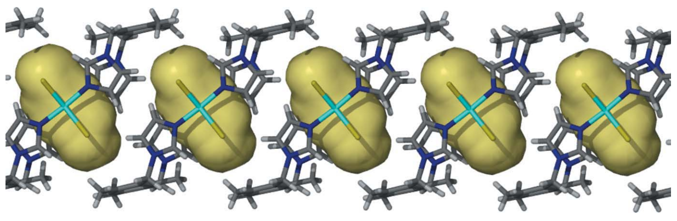


Figure 3
Capped-stick illustration showing the stacking of $[Cu_2Cl_4L1_2]$ metallocycles, where $L1$ = 1,3-bis(imidazol-1-ylmethyl)-2,4,6-trimethylbenzene (Nikolayenko *et al.*, 2017). The yellow surfaces represent intrinsic cavities of 97 Å^3 each, which are accessible to a virtual spherical probe of radius 1.5 Å. The surface maps were generated using the *MSRoll* interface within *X-Seed*, and the image was rendered using *POV-Ray*.

2.3. Molecular graphics

Well conceived metaphorical representations of molecules are critical to communicating the salient features of crystal structures. Crystallographers now have easy access to a variety of user-friendly computer programs for the production of molecular graphics, which are routinely included in journal articles, books, slide presentations, posters and theses. Indeed, we have become accustomed to the 'instant gratification' inherent in using fast computers employing software with interactive point-and-click interfaces. High-resolution colour reproductions are accessible and relatively inexpensive, and large raster graphics files in various formats (JPG, BMP, TIF, PNG *etc.*) are handled effortlessly by what-you-see-is-what-you-get applications such as graphics editors, word processors and slide presentation programs. However, it has not always been this easy to produce molecular graphics images. For the reader to appreciate the context in which this particular capability of *X-Seed* was conceived, it is useful to briefly discuss the state of molecular graphics in the late 1990s.

In the 1960s, Johnson (1965) developed the *Oak Ridge Thermal Ellipsoid Plot* (*ORTEP*) program to produce line drawings of molecules with the atoms displayed as ellipsoids defined by their anisotropic displacement parameters. During the following decade, Motherwell & Clegg (1978) developed *PLUTO* to generate packing diagrams (also as line drawings) in the space-filling ('CPK'), ball-and-stick, stick (straw) and wireframe metaphors. At the time, both *ORTEP* and *PLUTO* implemented efficient vector-graphics routines with hidden-line removal to draw the atoms and bonds. The user could code an image using a program-specific script file, which contained the crystal data as well as a selection of keywords that instructed the program on how to construct the 'scene'. Output took the form of a text file in Hewlett-Packard Graphics Language (HPGL) format, which consists of a set of instructions for a pen plotter. During the decade following 1988, the capabilities of *ORTEP* and *PLUTO* were also incorporated into other more interactive program packages such as *PLATON* (Spek, 2003), *NRCVAX* (Gabe *et al.*, 1989), *ORTEP-3 for Windows* (Farrugia, 1997) and *XP* (Bruker, 1998).

The earliest computer-generated raster-graphics renditions of macromolecular crystal structures were published in 1977/8 (Knowlton & Cherry, 1977; Porter, 1978; Feldmann *et al.*, 1978). In 1979, Keller developed *SCHAKAL* (Keller, 1980, 1989), a program more suited to generating both vector- and raster-graphics images of small-molecule crystal structures (user interaction was based on typing commands). These programs used ray-tracing algorithms to generate on-screen photorealistic images resembling plastic CPK models (and variations thereof). The screen could then be photographed and the prints used as figures for publication. Photographs of the screen could also be taken using 35 mm colour-reversal film for use with slide projectors. Popularization of digital images traces its origins to the 1990s.

The free, open-source and cross-platform program *POV-Ray* (for *Persistence of Vision Ray Tracer*) uses a text-based description of a scene to produce a high-quality digital image. It implements a sophisticated scene-description language that defines geometrical shapes, colours, textures and lighting effects as imaged by a user-defined 'camera'. Community-based development and maintenance of *POV-Ray* began in 1991; the software has undergone significant enhancements since then and it is still maintained by The POV-Ray Team community (Persistence of Vision Team, 1994). Since molecular graphics images require the use of rather simple primitives such as spheres and cylinders, rendering a 'molecular scene' is a trivial task for a program such as *POV-Ray*. Although a complete molecular graphics scene would be tedious to code manually, the text-based scene-description file is well suited to automatic generation by a program that can interpret crystal data (ideally, without even requiring the user to be familiar with the *POV-Ray* scene-description language).

First conceived in 1995, *POVChem* (Thiessen, 1996) uses crystal structure data in the Brookhaven Protein Data Bank (PDB) format to generate a *POV-Ray* scene. Although *POVChem* did not offer the ability to create packing diagrams, it was, to the best of the author's knowledge, the first program to harness the power of *POV-Ray* for molecular graphics. Although it was designed for macromolecular crystallographers, exploratory use of *POVChem* in 1998 alerted the author to the potential of *POV-Ray* for illustrating small-molecule crystal structures. A program was required that could read crystal data (*e.g.* in *SHELX* INS or RES file format) and, using space-group information, allow the user to interactively expand the structure and to set display attributes for the atoms. The program should then use this information

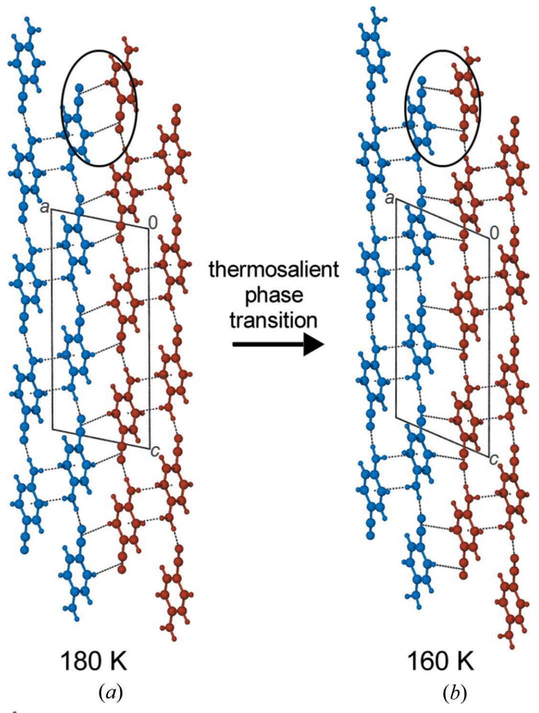


Figure 4
Ball-and-stick projections along [010] of two monoclinic forms of aminobenzonitrile. The crystal structures were determined for the same crystal at (*a*) 180 K and (*b*) 160 K. The side-by-side comparison of the two projections shows that the temperature-induced phase transition involves a shearing translation of adjacent bilayers parallel to (100), with a notable change in the unit-cell β angle (Alimi *et al.*, 2018).

interactively to generate a complete description of the scene in *POV-Ray* format. At that stage *X-Seed* was still in its earliest stages of development and the author had been contemplating how to write ray-tracing code *ab initio*. The *POV-Ray* revelation immediately rendered such efforts obsolete and *X-Seed* thus gained a *POV-Ray* interface. Independently, and at approximately the same time, a *POV-Ray* interface was also being added to *ORTEP-3 for Windows* (Farrugia, 1997).

The field of crystal engineering often requires crystal structures to be compared with one another as side-by-side illustrations (*e.g.* Fig. 4). The *POV-Ray*-based molecular graphics capability of *X-Seed* was designed with this in mind. It is relatively simple to select a direction for any projection (*i.e.* along any crystallographic vector or perpendicular to any crystallographic plane as specified using Miller indices, along a specified bond, or perpendicular to a plane through any three atoms) or to orient the structure using the mouse. While the right mouse button can be used to zoom in and out, the user is also able to enter a number for the zoom factor, thereby setting an arbitrary but reproducible scale. Two images generated using the same zoom factor will thus be rendered at the same scale for the same image size. Pressing the 'Render' button of the *POV-Ray* interface causes the scene to be generated as a POV file, which is then automatically parsed to *POV-Ray* for on-screen rendering. The general procedure to follow involves setting various attributes (*e.g.* background colour, atomic/bond radii and colours, *etc.*) and then rendering the scene with antialiasing turned off, without saving the image file. Antialiasing uses an edge-blending algorithm and is therefore computationally expensive – turning antialiasing off causes more rapid rendering but compromises image quality. Once the user is satisfied with the scene, a final antialiased version of the image can be saved to a high-quality PNG file for incorporation into a figure.

3. Resources and installation

The *X-Seed* Version 4 installation file can be downloaded at http://academic.sun.ac.za/barbour/Software.html and used free of charge. *X-Seed 4* has been tested under Windows 7 and 10. Owing to required setting up of registry entries, the installation procedure must be followed before the first use of *X-Seed*. Thereafter, ongoing updates (as announced on the web site) can be implemented by downloading only the latest executable file. A comprehensive manual is included in the installation, and updated versions of the manual are available for download. A simple test structure (INS and HKL files) is also included, and its solution and refinement are provided as a tutorial in the manual.

Acknowledgements

The author is grateful to the many *X-Seed* users who have provided valuable feedback over the past two decades.

References

Alimi, L., van Heerden, D. P., Lama, P., Smith, V. J. & Barbour, L. J. (2018). *Chem. Commun.* **54**, 6208–6211.
Atwood, J. L. & Barbour, L. J. (2003). *Cryst. Growth Des.* **3**, 3–8.
Barbour, L. J. (2001). *J. Supramol. Chem.* **1**, 189–191.
Barbour, L. J. (2006). *Chem. Commun.* **2006**, 1163–1168.
Barbour, L. J. & Atwood, J. L. (1998). *J. Appl. Cryst.* **31**, 963–964.
Bruker (1998). *SHELXTL*. Version 5.10. Bruker AXS Inc., Madison, Wisconsin, USA.
Connolly, M. L. (1983*a*). *Science*, **221**, 709–713.
Connolly, M. L. (1983*b*). *J. Appl. Cryst.* **16**, 548–558.
Connolly, M. L. (1993). *J. Mol. Graph.* **11**, 139–141.
Dolomanov, O. V., Blake, A. J., Champness, N. R. & Schröder, M. (2003). *J. Appl. Cryst.* **36**, 1283–1284.
Dolomanov, O. V., Bourhis, L. J., Gildea, R. J., Howard, J. A. K. & Puschmann, H. (2009). *J. Appl. Cryst.* **42**, 339–341.
Farrugia, L. J. (1997). *J. Appl. Cryst.* **30**, 565.
Farrugia, L. J. (1999). *J. Appl. Cryst.* **32**, 837–838.
Feldmann, R. J., Bing, D. H., Furie, B. C. & Furie, B. (1978). *Proc. Natl Acad. Sci. USA*, **75**, 5409–5412.
Frenz, B. (1988). *Comput. Phys.* **2**, 42–48.
Gabe, E. J., Le Page, Y., Charland, J.-P., Lee, F. L. & White, P. S. (1989). *J. Appl. Cryst.* **22**, 384–387.
Johnson, C. K. (1965). *ORTEP*. Report ORNL-3794. Oak Ridge National Laboratory, Tennessee, USA.
Keller, E. (1980). *Chem. Zeit.* **14**, 56–60.
Keller, E. (1989). *J. Appl. Cryst.* **22**, 19–22.
Knowlton, K. & Cherry, L. (1977). *Comput. Chem.* **1**, 161–166.
Macrae, C. F., Bruno, I. J., Chisholm, J. A., Edgington, P. R., McCabe, P., Pidcock, E., Rodriguez-Monge, L., Taylor, R., van de Streek, J. & Wood, P. A. (2008). *J. Appl. Cryst.* **41**, 466–470.
Macrae, C. F., Edgington, P. R., McCabe, P., Pidcock, E., Shields, G. P., Taylor, R., Towler, M. & van de Streek, J. (2006). *J. Appl. Cryst.* **39**, 453–457.
Motherwell, W. D. S. & Clegg, W. (1978). *PLUTO*. University of Cambridge, UK.
Nikolayenko, V. I., Heyns, A. & Barbour, L. J. (2017). *Chem. Commun.* **53**, 11306–11309.
Persistence of Vision Team (1994). *POV-RAY – the Persistence of Vison Raytracer*, http://www.povray.org.
Porter, T. K. (1978). *Comput. Graph.* **12**, 282–285.
Robinson, W. T. & Sheldrick, G. M. (1988). *Crystallographic Computing 4: Techniques and New Technologies*, edited by N. W. Isaacs & M. R. Taylor, pp. 366–377. IUCr/Oxford University Press.
Sheldrick, G. M. (1976). *SHELX76*. University of Göttingen, Germany.
Sheldrick, G. M. (1985*a*). *Crystallographic Computing 3: Data Collection, Structure Determination, Proteins and Databases*, edited by G. M. Sheldrick, C. Krüger & R. Goddard, pp. 175–189. IUCr/Oxford University Press.
Sheldrick, G. M. (1985*b*). *J. Mol. Struct.* **130**, 9–16.
Sheldrick, G. M. (1990). *Acta Cryst.* A**46**, 467–473.
Sheldrick, G. M. (1993). *Crystallographic Computing 6: a Window on Modern Crystallography*, edited by H. D. Flack, L. Párkányi & K. Simon, pp. 100–122. IUCr/Oxford University Press.
Sheldrick, G. M. (1997). *SHELX97*. University of Göttingen, Germany.
Sheldrick, G. M. (2008). *Acta Cryst.* A**64**, 112–122.
Sheldrick, G. M. (2015). *Acta Cryst.* A**71**, 3–8.
Spek, A. L. (2003). *J. Appl. Cryst.* **36**, 7–13.
Thiessen, P. A. (1996). *PovChem*. Version 1.00. https://www.chemicalgraphics.com/paul/PovChem.html.
Usón, I. & Sheldrick, G. M. (2018). *Acta Cryst.* D**74**, 106–116.