Journal of Applied Crystallography

ISSN 0021-8898

Received 0 XXXXXXX 0000
Accepted 0 XXXXXXX 0000
Online 0 XXXXXXX 0000

CrystalPlan: an Experiment Planning Tool for Crystallography

Janik Zikovsky,[a]* Peter Peterson,[a] Xiaoping Wang,[a] Matthew Frost[a] and Christina Hoffmann[a]

[a]Spallation Neutron Source, Oak Ridge National Laboratory, P.O. Box 2008 MS-6477, Oak Ridge, TN 37831-6477 USA. Correspondence e-mail: zikovskyjl@ornl.gov

Beam time at large x-ray and neutron scattering facilities is always at a premium. The CrystalPlan program can calculate the data coverage of a crystal in reciprocal space in a single-crystal diffraction time-of-flight experiment. CrystalPlan can help a user build an experiment plan that will acquire the most data possible, with sufficient coverage but limited redundancy, therefore increasing scientific productivity. An attractive GUI including a 3D viewer and an automated coverage optimizer are among its useful features. A sample use case of the program with the TOPAZ beamline at SNS will be presented.

1. Introduction

The TOPAZ beamline at the Spallation Neutron Source (SNS) at Oak Ridge National Lab (ORNL), currently finishing commissioning, is a neutron time-of-flight instrument designed to acquire single-crystal diffraction data of small-to-moderate sized unit cells with high throughput. The final design calls for an array of 48 detectors covering a large fraction of 2π steradians; however as of this writing, only 14 of the 48 detectors are installed. In order to have sufficient data to perform a fit, an experimenter will typically want to measure $> 85\%$ of the peaks within a certain range of d-spacings. However, TOPAZ's complex geometry makes it very difficult to come up with an effective set of sample orientations that will cover all the required peaks without unnecessary redundancy.

Existing software tools [ADD SOME NAMES HERE see my notes from ACA2010] were designed for X-ray experiments where a single incident wavelength and a large number of sample orientations or sweeps are used to collect data. These programs cannot be used for TOPAZ, since as a time-of-flight instrument, it is able to collect data from a wide bandwidth of incident neutrons. Additionally, this means that relatively few sample orientations are needed in order to complete a measurement: whereas an X-ray measurement might have hundreds or thousands of individual frames, a TOPAZ run may have as few as 10.

The goal of CrystalPlan was to make an easy-to-use software tool to aid users in planning their experiments. By quickly making an experimental plan that will be sufficient for the needs of the scientist (enough peaks measured), but without spending time needlessly measuring the same region in reciprocal space, CrystalPlan will greatly increase the scientific productivity and throughput of TOPAZ. More users will be able to measure more samples, faster.

2. Architecture

The program needs to be cross-platform, and run equally well on Linux, Mac OS and Windows, although Linux is our primary target operating system. Another design goal was to use of open-source software and toolkits as much as possible.

The calculation code of CrystalPlan is placed in a model layer and kept completely separate from eventual GUI elements. This makes it possible to run calculations using Python scripts, if desired. Unit tests for each part of the calculations are used to ensure consistent results on different platforms and systems.

2.1. Instrument Configuration

CrystalPlan supports area detectors with flat faces and arbitrary rectangular dimensions; this handles the detector types in use at SNS (Anger cameras and 8-pack tube detectors), but the software could easily be extended for more complicated detector shapes as needed. The detector configuration is loaded from a text file containing the location, size, orientation and number of pixels of each detector.

2.2. Sample Orientation

CrystalPlan is designed to be able to use a variety of sample orientation goniometers. A base Goniometer class is subclassed for specific types of goniometers, allowing a unified API: in a way, it is a universal goniometer, which uses the conventional goniometer angles of phi, chi, omega as defined in [REF XXX]. Each class inheriting from Goniometer can convert the desired phi, chi, omega angles to the internal motor positions it requires. The API allows for limited goniometers: for example, TOPAZ is currently equipped with a goniometer where chi is fixed at 45 degrees. CrystalPlan handles this by limiting the available sample orientation angles accessible to the user.

The data analysis program ISAW is used to calculate the the sample mounting orientation UB matrix, which is saved to a text file along with the lattice parameters. CrystalPlan can load this UB matrix file and will use that UB matrix in coverage calculations.

2.3. Volume coverage

2.3.1. Reciprocal Space Representation The reciprocal space to be measured is divided into a cubic 3D matrix, the size of which is determined by the user, who specifies the minimum d-spacing (d_{min}, in $\AA$) that he or she is interested in. The resolution of the 3D matrix is also specified (in $\AA^{-1}$), which determines the number of points to be modeled. Machine memory limits the resolution that can be achieved.

Each point of the reciprocal space 3D matrix holds an integer representing the number of times that voxel of q-space has been measured this is the total coverage matrix. To speed up the calculation of the total, a coverage matrix for each sample orientation is calculated and saved in memory. This means that the total coverage matrix can be computed by simply adding multiple 3D matrices corresponding to each sample orientation.

2.3.2. Calculating Coverage of One Sample Orientation For each sample orientation, a rotation matrix combining the sample's mounting orientation (the U matrix loaded from ISAW) and the goniometer rotation (Φ, Ξ, Ω) is computed. The coverage in reciprocal space is computed in this way: for each pixel in the face of a detector, the direction of the scattered beam is known. Then, two incident beam vectors are considered, corresponding to the low and high wavelength limits of the detectors and/or source. The scattered beam direction and the two possible incident beam vectors are used to to find two q vectors, which point in the same direction. At the high wavelength we find qmin, and the low wavelength limit scattering off of qmax. Since the instrument has a wide bandwidth, we fill in the 3D q-space matrix by filling all points from qmin to the qmax, for all pixels in the face of the detector. This quickly defines a volume of coverage, as shown in Figure X.

To reduce memory usage yet allow for quickly trying different detector configurations, the 3D matrix of each sample orientation is saved a 32- or 64-bit integer, with each bit representing a particular detector. A simple binary mask can then be applied to quickly compute the coverage if some detectors are disabled, for example.

2.4. Crystal Parameters and Single Reflection Coverage

Crystal lattice parameters (lengths a,b,c and angles alpha, beta, gamma) are typed in or loaded from the ISAW UB matrix file. CrystalPlan then generates a list of HKL peaks with d-spacings larger than the d_{min} specified. For each sample orientation being simulated and for each peak, a function calculates at what wavelength lambda and in which direction it scatters. If the wavelength is within range of the limits of the instrument, the beam direction is projected onto each detector face to determine where, on each detector surface, it will be measured.

Each HKL peak is a separate object that holds a list of simulated measurements, including which detector measured it, at what position on the detector face, at what wavelength, and for which sample orientation. These data will be used to graphically display coverage and calculate statistics such as redundancy.

2.5. Crystal Symmetry

When entering the lattice parameters, the crystal's point group symmetry can also be entered. This information is then used to calculate the increased measurement redundancy. A list of n 3x3 multiplication matrices is generated based on the point group; any hkl vector can be multiplied by these matrices to find the n equivalent hkl reflections; for example, for orthorombic (mmm) symmetry, n=8 and 8 multiplication matrices are generated, which multiply hkl to give h, k, l.

For the single reflections, each hkl is tracked separately, but a primary hkl is found that points to a list of equivalent hkls. The GUI can then display either all hkl, ignoring crystal symmetry; or combine them to display only part of the reflections.

In volume coverage mode, a 4D matrix is generated where each voxel has a list of n corresponding equivalent voxels. This allows CrystalPlan to quickly compute a volume coverage map that takes into account crystal symmetry: at each voxel, n original voxels are checked to see if any of them were measured.

2.6. Automatic Coverage Optimizer

CrystalPlan allows the user to manually enter lists of sample orientations to simulate; however, finding an optimal plan by trial and error would be very time consuming. To make the process easier, CrystalPlan includes an Automatic Coverage Optimizer, which takes as its inputs the desired number of sample orientations and uses a genetic algorithm to find a solution that satisfies a user-specified coverage criterion (for example, the user may ask for 85peaks to be measured, or for 90

Genetic algorithms have been covered in the literature before, see e.g ref [INSERT REFERENCE HERE]. In this application, the genes consist of goniometer angles. For instance, if a goniometer has freedom in 3 angles and the user requests a 10 different sample orientations, then each individual has 30 genes. The initial population is created randomly and is made to evolve using mutations and crossover. For each individual set of sample orientations, the coverage statistics are computed (taking crystal symmetry into account if desired), and this is used as the fitness of the given individual in the normal genetic algorithm process.

The Automatic Coverage Optimizer can find an acceptable solution (if any is possible) in times ranging from a few seconds to several minutes, depending on the difficulty of the problem and the computing hardware. Given that at a neutron facility, a measurement at a single sample orientation may take a few hours, this feature can result in significant beam-time savings.

3. Implementation

3.1. Language and Libraries

CrystalPlan was written in Python, a powerful, open, cross-platform scripting language. Python was chosen for its ease of cross-platform deployment, and for its attractive language features. The numpy and scipy libraries (two open-source, commonly installed Python libraries) were used for many of the calculations. Some of the critical calculations were also written with inline C code, which speed up some calculations by up to 400$\times$ as compared with the pure Python version (which can still be used, in case of incompatibilities).

3.2. Graphical User Interface

3.2.1. Design An attractive and easy-to-use GUI was considered an important part of CrystalPlan, since the program is meant to be used interactively as a guide to the experimenter. The user interface was written using wxPython, a mature cross-platform GUI toolkit, as well as the Enthought Traits GUI. Some of the 2D plots use matplotlib, and the 3D visualization elements use the Mayavi2 toolkit for Python, itself based on VTK. All of these libraries are open-source.

3.2.2. Main Program Workflow Some screenshots of the GUI are shown in Figure Y. The main window consists of a series of tabs, with workflow proceeding from left to right:

- Q-Space tab: define the volume of interest (minimum d-spacing) and resolution of reciprocal space. Detectors tab: load instrument detector geometry files, and enable or disable individual detectors. Goniometer tab: to choose the goniometer and degrees of orientation freedom of the instrument.
- Sample tab: enter or load the crystal's lattice parameters and UB matrix.
- Try an Orientation tab: interactively rotate a sample and observe the changes in coverage in real-time.
- Add Orientations tab: type in a list of sample orientation angles to add to the experiment plan.
- Experiment Plan tab: a list of the previously calculated sample orientations. Here you can manage the orientations you will use in the experiment: add, delete, or temporarily disable them to see the effect on coverage. Once complete, the plan can be sent to the SNS data acquisition computers in order to begin a run.

3.2.3. Reciprocal Space Coverage Visualization Figures Z -a shows a screenshot of the reciprocal space coverage 3D interface. The 3D view can represent q-space in two ways: volume coverage, or single-peak coverage.

The volume coverage shows a solid isosurface representing the volume measured at least once. Redundant regions can also be shown using semi-transparent isosurfaces; or the coverage can be inverted, showing non-measured volumes only; or sliced between user-selectable q_{min} and q_{max}.

The reflections view (Figure ZZ-a) displays each hkl as a small sphere; the color of which indicates the number of times the hkl is predicted to be measured in the current experiment plan.

Both views can take into crystal symmetry in their display (Figures Z-b and ZZ-b), using the techniques described previously.

Finally, both views display quick coverage statistics, showing the percentage of volume (or of the number of peaks) measured, as well as the proportion of redundancy (voxels or peaks measured 2 or more times).

4. Sample Use Case

5. Conclusions and Future Development

Live data processing interface with ISAW InelasCasetic instruments ??

Acknowledgements

This research is supported by UT Battelle, LLC under Contract No. DE-AC05-00OR22725 for the U.S. Department of Energy, Office of Science.

References

Author, A. & Author, B. (1984). *Journal* **Vol**, first page–last page.

Table 1
Caption to table

HEADING	FOR	EACH	COLUMN
entry	entry	entry	entry
entry	entry	entry	entry
entry	entry	entry	entry

Figure 1
Caption describing figure.