



# A user manual for CrystalView

Philip Sternberg

**ABSTRACT.** A description of CrystalView, a software package for visualizing crystal bases and carrying out calculations on them.

**RÉSUMÉ.** Une description de CrystalView, un progiciel pour visualizer les bases cristallines et mise en oeuvre des calculs sur eux.

## 1. Overview

CrystalView is a software package for visualizing crystals for irreducible highest weight modules of simple Lie algebras. Based on user input, the program will produce an image file with the requested crystal graph. The program will automatically produce an .epsf file, which can be included directly in a  $\text{\LaTeX}$  file. Using the web interface, the user can see the tableau associated a vertex of the crystal by moving the mouse pointer over it.

When running this program locally (as opposed to via the web), the program produces a list of all tableaux in the specified crystal. This can be used to carry out calculations on crystals, including searches for tableaux with specified properties. Additionally, Kashiwara operators may be applied to the tableaux.

All calculations on tableaux are carried out using python. Image files are automatically generated PostScript. The web interface uses html, javascript, and css.

## 2. Requirements

CrystalView may be run using a web interface or from source code. The web interface can, in principle, be used from any browser that supports form input (any “modern” browser). However, some aspects of the interface will only work with a browser that complies with standard html, javascript, and css. In particular, Internet Explorer is known to have issues with some dynamic aspects of the web interface. Firefox is a recommended alternative.

To run the software from source code, the user must have Python 2.4 installed on their local machine. Other versions of Python (both older and newer) may not run CrystalView properly. See <http://www.python.org> for further information regarding python. Python is available free of charge for all major operating systems, and is included pre-installed on many modern computers, including almost all distributions of Unix/Linux.

If used to generate image files locally, the user is advised that for large crystals, these images can get quite large. See section 7.

## 3. User input

The user may specify the following Lie theoretic data:

- symmetry type of the algebra being represented;
- rank of the algebra being represented;

---

2000 *Mathematics Subject Classification.* Primary 17B37; Secondary 05C75.

*Key words and phrases.* crystal bases, Lie algebras, software.

Supported in part by NSF grants DMS-0135345, DMS-0200774, and DMS-0501101.

- highest weight of the representation.

Additionally, the user may specify how the edges of the crystal graph will be drawn. There are two default settings, color and grayscale, as well as an option for custom settings. If the custom option is selected, the user may specify the following:

- red/green/blue values for each edge color on a scale from 0 to 1 (Default = 1),
- line width on a scale from 1 to 5 (Default = 5),
- dash pattern; none, short, long (Default = none).

There are numerous resources on the web and preinstalled on many computers to assist the user in finding red/green/blue values for their desired colors.

The user may also choose to have the output converted to .pdf, .jpg, .gif, and/or .tiff formats.

#### 4. Limitations

Currently, only types  $A$  and  $B$  (i.e.,  $\mathfrak{sl}_n$  and  $\mathfrak{so}_{2n+1}$ ) are supported. Furthermore, in the case of type  $B$ , only even multiples of the highest fundamental weight (corresponding to the short root) may be specified. These limitations are due to the current stage of the development cycle; future versions of the software will add support for types  $C$  and  $D$  and all dominant weights.

The rank of the algebra is currently restricted by the web interface to be no larger than 5. This is an artificial limitation; any rank of algebra may be specified when running CrystalView from source.

The web interface only allows crystals with as many as 4,000 vertices to be calculated to prevent excessive strain on the server. Considering the resolution at which these images can be viewed/printed, it is unlikely that producing crystals larger than this would be useful to most users. However, this limitation is artificial; when running the program on a local machine, the user is limited only by their own patience and hard disk space.

#### 5. How it works

The web interface for CrystalView is written in dhtml; i.e., html enhanced by javascript and css. The Weyl dimension formula is used to calculate the number of vertices in the currently specified crystal.

The tableaux are produced by generating the list of all column tableaux for the column lengths appearing in the shape specified by the dominant weight. To determine what constitutes a legal tableaux, the criteria of [1] are used. The columns are then compared pairwise in order of decreasing length to build the set of all legal tableaux. In the case of type  $B$  crystals, the “split form” criterion of [2] is used to determine which columns can be adjacent in a legal tableau.

The graph is ensured to have a reasonable number of edge crossings by ordering the vertices of the graph as follows, starting from the top and going down the rows, and proceeding through each row from left to right. First, the tableaux are collected into rows according to content. There is only one tableau in the first row (the highest weight tableau), so the first row is in order. Now, given that row  $n$  is in order, row  $n + 1$  is put in the following order. The leftmost vertices in row  $n + 1$  will be the non-zero images of the Kashiwara operators  $f_i$  on the leftmost vertex in row  $n$ , taken in order from  $f_1$  up through  $f_r$ , where  $r$  is the rank of the algebra. The next leftmost vertices in row  $n + 1$  are those tableaux that result from applying  $f_i$  to the next vertex from row  $n$ , excluding those that have been placed to the left already. This process continues until all vertices have been placed in their final position.

#### 6. How to get the software

This software can be accessed at the following url:

<http://www.math.ucdavis.edu/~sternberg/crystalview/>

The source code is available upon request from the author.

#### 7. Legal disclaimer

This software is provided without warranty; use is at the user’s sole risk. Under no circumstances will any user hold the author of this software liable for any damages that may result from the use of this software. Use of the software implies that the user agrees to these terms.

### References

- [1] J. Hong and S.-J. Kang, *Introduction to quantum groups and crystal bases*, Graduate Studies in Mathematics, 42. American Mathematical Society, Providence, RI, 2002. xviii+307 pp. ISBN: 0-8218-2874-6.
- [2] C. Lecouvey, *Schensted-type correspondences and plactic monoid for types  $B_n$  and  $D_n$* , J. Algebraic Combin. **18** (2003), no. 2, 99–133.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, ONE SHIELDS AVENUE, DAVIS, CA 95616-8633, U.S.A.

*E-mail address:* `sternberg@math.ucdavis.edu`

*URL:* <http://www.math.ucdavis.edu/~sternberg>