Polytopal Cell Complex (PCC) Processing Design (CPD code\*)

Quick Start with the CPD code

Manual version 0.1.0 27-Jan-2024

**The code online source repository:** <https://github.com/PRISBteam/PCC_Processing_Design>

**The code reference example:**

Elijah Borodin (2024) *Polytopal Cell Complex (PCC) Processing Design (CPD code)*, (Version 3.0) [Computer software]. Zenodo. <https://github.com/PRISBteam/PCC_Processing_Design>

Contents:

1. Where to take a complex?

2. Tutorial and Scientific execution modes

3. Test simulation mode

4. List simulation mode

5. Task simulation mode

1. **Where to take a complex (go above)**

The polytopal cell complex is a pretty well-known object that originated from the field of algebraic topology, so it can be obtained in many various ways Below is just a concise review of a couple of flexible tools developed in the Mechanics and Physics of Solids research group in the University of Manchester providing PCCs based on Voronoi and a few others tessellations of space by convex polygons.

<h3> Tessellations of space provided by Neper software </h3>

The Voronoi tesselation provided by Neper supposed to be a <i>dual</i> complex and so all the other tessellations provided by the Neper output with the <a href="https://neper.info/doc/neper\_t.html#morphology-options" target="\_blank"> morphology </a> option <i> -morpho <morphology> </i> like <i> cube, square, tocta, lamellar, etc. </i> different from <i>voronoi</i>.

Please, see more <a href="https://neper.info/doc/neper\_t.html#examples" target="\_blank"> examples </a> on the Neper webpage.

<h3> PCC Generator Tool </h3>

Based on the Poisson-Voronoi tessellation of space provided by the <a href="https://neper.info" target=”\_blank”> Neper </a> software the code creates discrete (combinatorial) cell complex (DCC) as the set of sparse matrices. The DCC Generator Tool generates a sparse representation of matrices: for any matrix element \_a\_(\_i\_, \_j\_) = \_c\_, the files of the matrices contain the list of triplets in the form (\_i\_, \_j\_, \_c\_). The enumeration of indices starts from 0, and, for instance, the line "5, 7, 1" in the adjacency matrix A<sub>k</sub> means that the \_k\_-cell #6 is the neighbour of the \_k\_-cell #8. For any incidence matrices B<sub>k</sub>, the same triplet "5, 7, 1" means that the (\_k\_-1)-cell #6 is on the boundary of the \_k\_-cell #8, and their orientations coincide (\_c\_ = -1 for the opposite orientations).

All the other information on the GitHub page of the <a href="https://github.com/PRISBteam/Voronoi\_DCC\_Analyser/blob/main/README.md" target=”\_blank”> project </a>

The latest release of the code can be downloaded from the <a href="[https://github.com/PRISBteam/Voronoi\_DCC\_Analyser/blob/main/README.md](https://github.com/PRISBteam/Voronoi\_DCC\_Analyser/tags)" target=”\_blank”> DCGT </a> project page.

<h3> FCC and BCC primal slip planes </h3>

The package DCC\_Structure contains Python modules to build a discrete cell complex (DCC) based on the slip planes of crystal nanostructures (simple cubic, FCC, BCC; HCP not yet available). The script execute.py is a summarised execution of the whole package. It takes as input:

…