**First PCC standard**

***Standard’s ID*: pcc1s**

**Date: *20 April 2024***

*Dr Elijah Borodin, The University of Manchester*

The standard for ***computation representation of Polytopal Cell Complexes (PCC***) as is used in the MATERiA codes projects (<https://github.com/PRISBteam>) and the PCC database (<https://materia.team>).

**Table of Contents**

1. **Introductory comments**
2. **Combinatorial data**
3. **Algebraic data**

**2.1 Adjacency matrices (sparse)**

**2.2 Incidence matrices (sparse)**

**2.3 Combinatorial Laplacians**

1. **Geometrical data**

**3.1 Coordinates of the corresponding tessellation of space**

**3.2 Measures of the corresponding tessellation of space**

1. **APPENDIX A: Definition of adjacency and incidence matrices with their sparse representations**
2. **APPENDIX B: How to create a space tessellation?**
3. **Introductory comments**

The standard contains the description of PCC files and their directories reflecting diversity in the information related to a PCC and the corresponding tessellation of continuous space into polytopes. This standard is a completely computational thing and has nothing to do with the mathematical definitions of cell complexes as the combinatorial discrete manifolds in algebraic topology ([book](https://link.springer.com/book/10.1007/978-1-84996-290-2), books..).

* In algebraic topology, a discrete topological n-complex is a collection of cells of dimensions *k ≤ n*, where every k-cell for any *0 < k ≤ n*has a boundary formed by (k-1)-cells belonging to the complex. The co-boundary of every k-cell for *0 ≤ k < n*is the set of (k+1)-cells whose boundaries contain the k-cell. In this terminology, 1-complex is a graph. Polyhedral complexes are a special class of regular quasi-convex discrete topological complexes, in the geometric realisation of which 0-cells are identified with points or vertices, 1-cells with line segments or edges, 2-cells with planar polygons or faces, 3-cells with polyhedra or simply cells, etc. We restrict our consideration to the polyhedral 3-complexes whose 3-cells are convex polyhedra with 2-cells in the boundary of exactly two 3-cells. An assembly of polyhedrons is a geometric realisation of a combinatorial structure referred to as a cell complex in algebraic topology.

The **key concept** of the standard is the using of ***minimal files provided necessary computational freedom and effectiveness***. It does mean including in the standard the files which are relatively hard/costly to compute (such as barycentre coordinates of grains and grain boundaries) and not including the data which can be calculated simply (such as the middle point coordinates of edges).

1. **Combinatorial data**
2. **Algebraic data**

**2.1 Adjacency matrices (sparse)**

**2.2 Incidence matrices (sparse)**

**2.3 Combinatorial Laplacians**

1. **Geometrical data**

**3.1 Coordinates of the corresponding tessellation of space**

**3.2 Measures of the corresponding tessellation of space**

1. **APPENDIX A: Definition of a PCC, its adjacency and incidence matrices with their sparse representations**

An excellent simple introduction to the area of DCCs with their various applications is given in the [book](https://link.springer.com/book/10.1007/978-1-84996-290-2)of Leo Grady and Jonathan Polimeni *“Discrete Calculus. Applied Analysis on Graphs for Computational Science. (2010)* Below are just a few notes necessary for understanding the output of the code.

* The geometric properties of the DCC are encoded in the volumes of different cells: 1 for 0-cells, length for 1-cells, area for 2-cells, and volume for 3-cells. The topological properties of the DCC are encoded in the boundary operator Bk, which maps all (k+1)-cells to the k-cells in their boundaries, taking into account cell orientations. The algebraic realisation of the operator for the [k,(k+1)] pair of cells is referred to as the k-th incidence matrix, Bk, which has Nk rows (where Nk denote the number of k-cells in a complex) and Nk+1 columns and contains 0, 1, -1, indicating non-adjacency, adjacency with agreeing and with opposite orientations, respectively, between k-cells and (k+1)-cells. The transpose of the k-th incidence matrix, bk = BkT, is a matrix representing the k-th co-boundary operator, which maps all k-cells to the (k+1)-cells in their co-boundaries.
* A standard way is to decide on a consistent orientation of all top-dimensional cells, e.g., to select the positive orientation to be from interior to exterior of the 3-cells and assign arbitrary orientations for all lower-dimensional cells. There are exactly three options for the relation between k-cell and (k+1)-cell in an oriented complex: they are not coincident - encoded by 0; the k-cell is on the boundary (k+1)-cell, and they have consistent orientations, encoded by 1; the k-cell is on the boundary (k+1)-cell and they have opposite orientations, encoded by -1. The transpose of the k-th incidence matrix is a matrix representing the k-th co-boundary operator, which maps all k-cells to the (k+1)-cells in their co-boundaries.
* The k-th combinatorial *Laplacian* (Laplace–de Rham operator) can be written as  
  *Lk = bk-1 Bk-1 + Bk bk*  
  and it maps all k-cells to themselves, collecting local connectivity information. One important application of the combinatorial Laplacians is in calculating [combinatorial curvatures](https://link.springer.com/article/10.1007/s00454-002-0743-x). Since the Laplacians are symmetric positive semi-definite matrices, their eigenvalues are real. The spectra of eigenvalues can be used to classify discrete topologies, with two topologies considered equivalent when they have the same Laplacians’ spectra.

1. **APPENDIX B: How to create a space tessellation (2024 year)?**

**Tessellations of space provided by Neper software**

The Voronoi tessellation provided by Neper is supposed to be a *dual* complex and so all the other tessellations provided by the Neper output with the [morphology](https://neper.info/doc/neper_t.html#morphology-options)option *-morpho*like a *cube, square, tocta, lamellar, etc.*different from *Voronoi*.

Please, see more [examples](https://neper.info/doc/neper_t.html#examples)on the Neper webpage.

**Tips and tricks**

* The metric information like the volumes of all 3-cells and areas of all 2-cells can be obtain directly from the Neper output using [statcell](https://neper.info/doc/neper_t.html" \l "cmdoption-statcell)and statface options with the corresponding [keys](https://neper.info/doc/exprskeys.html#tessellation-keys)like "-statcell vol -statface area" or providing the corresponding values for every k-cell in the complex. In this case, the terminal command may look like
* neper -T -n 300 -id 1 -dim 3 -statcell vol -statface area

Please, see more [examples](https://neper.info/doc/neper_t.html#examples)on the Neper webpage.

* Using the file ``seeds.txt`` with some specific set of seed points a new Neper tessellation can be performed. The terminal command creating a complex with coordinates of the seed points as the centres of 3-cells may looks like
* neper -T -n <number of seeds> -id 1 -statcell vol -statface area -domain "cube(1.0,1.0,1.0)" -morphooptiini "coo:file(seeds.txt)"

You must call Neper from the folder (cd <path to the directory containing the file "seeds.txt">) containing the seeds.txt file, or write the whole path instead of the file name in the *coo:file()* command.

* More flexibility in the tesselation provide the [transformation](https://neper.info/doc/neper_t.html#examples)options of the Neper. In particular, for the creation of a 2D complex as a plane cut of the 3D one, the *slice(d,a,b,c)* function can be used as it is shown below for the half-cut of the Voronoi complex containing 1000 grains:
* neper -T -n 1000 -id 1 -domain "cube(1.0,1.0,1.0)" -transform "slice(0.5,0,0,1)" -dim 3 -statcell area; \
* neper -V n1000-id1.tess -datacelltrs 0.5 -print DCC\_slice

Here *d, a, b*, and *c* are parameters in the corresponding equation of a plane *ax + by + cz = d* and it is worth to be mentioned here that the normal vector of this plane is *n = (a,b,c)*.

**Acknowledgements**

The basic methodology has been developed as a part of the EPSRC-funded projects [EP/V022687/1](https://gow.epsrc.ukri.org/NGBOViewGrant.aspx?GrantRef=EP/V022687/1)*“Patterns recognition inside shear bands: tailoring microstructure against localisation”* (PRISB) and [EP/N026136/1](https://gow.epsrc.ukri.org/NGBOViewGrant.aspx?GrantRef=EP/N026136/1)*"Geometric Mechanics of Solids: new analysis of modern engineering materials"* (GEMS).