Test

1. Mathematical models
2. Computer representations
3. Operations

Mathematical models

Combinatorial Cell Complexes (1/3)

Computer representations

Sparse matrix representations

Basic representations

A few basic representation of topology are used in LARCC.

They include some common sparse matrix representations:

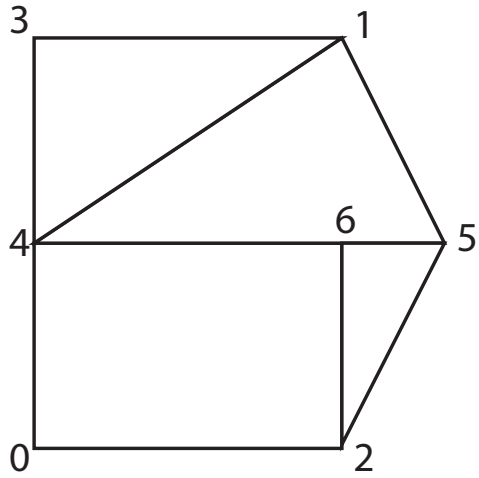
- CSR (Compressed Sparse Row),
- CSC (Compressed Sparse Column),
- COO (Coordinate Representation),
- and BRC (Binary Row Compressed)

Data definition

- The rank zero cells below a cell $x \in S$ are called the vertices of x, that is the least upper bound of its vertices.

Characteristic matrix $M_2 : C_0 \to C_2$

$$M_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

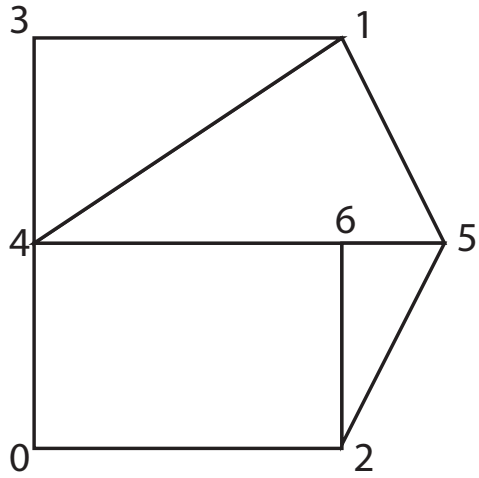


Data definition

- The rank zero cells below a cell $x \in S$ are called the vertices of x, that is the least upper bound of its vertices.
- So we can identify each cell with its set of vertices.

Characteristic matrix $M_2 : C_0 \to C_2$

$$M_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

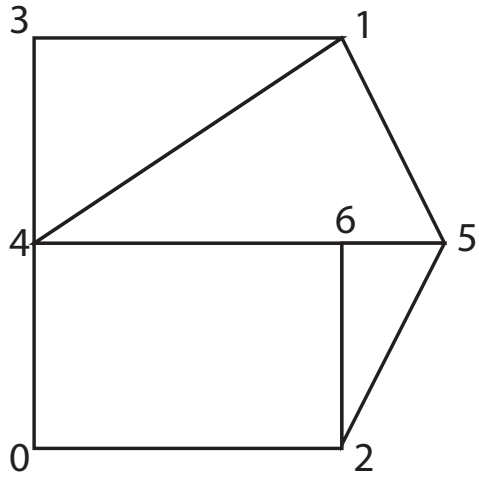


Data definition

- The rank zero cells below a cell $x \in S$ are called the vertices of x, that is the least upper bound of its vertices.
- So we can identify each cell with its set of vertices.
- Thus, to define S, we start from the vertex set S_0, and specify the vertex subsets which correspond to the cells, and the rank of each cell.

Characteristic matrix $M_2 : C_0 \to C_2$

$$M_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

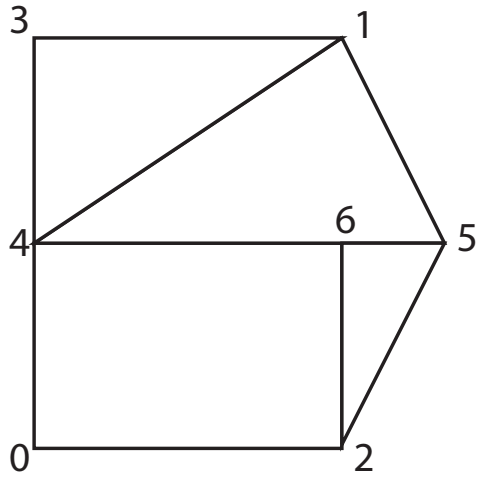


Data definition

- The rank zero cells below a cell $x \in S$ are called the vertices of x, that is the least upper bound of its vertices.
- So we can identify each cell with its set of vertices.
- Thus, to define S, we start from the vertex set S_0, and specify the vertex subsets which correspond to the cells, and the rank of each cell.
- The partial order is induced by set inclusion.

Characteristic matrix $M_2 : C_0 \to C_2$

$$M_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$



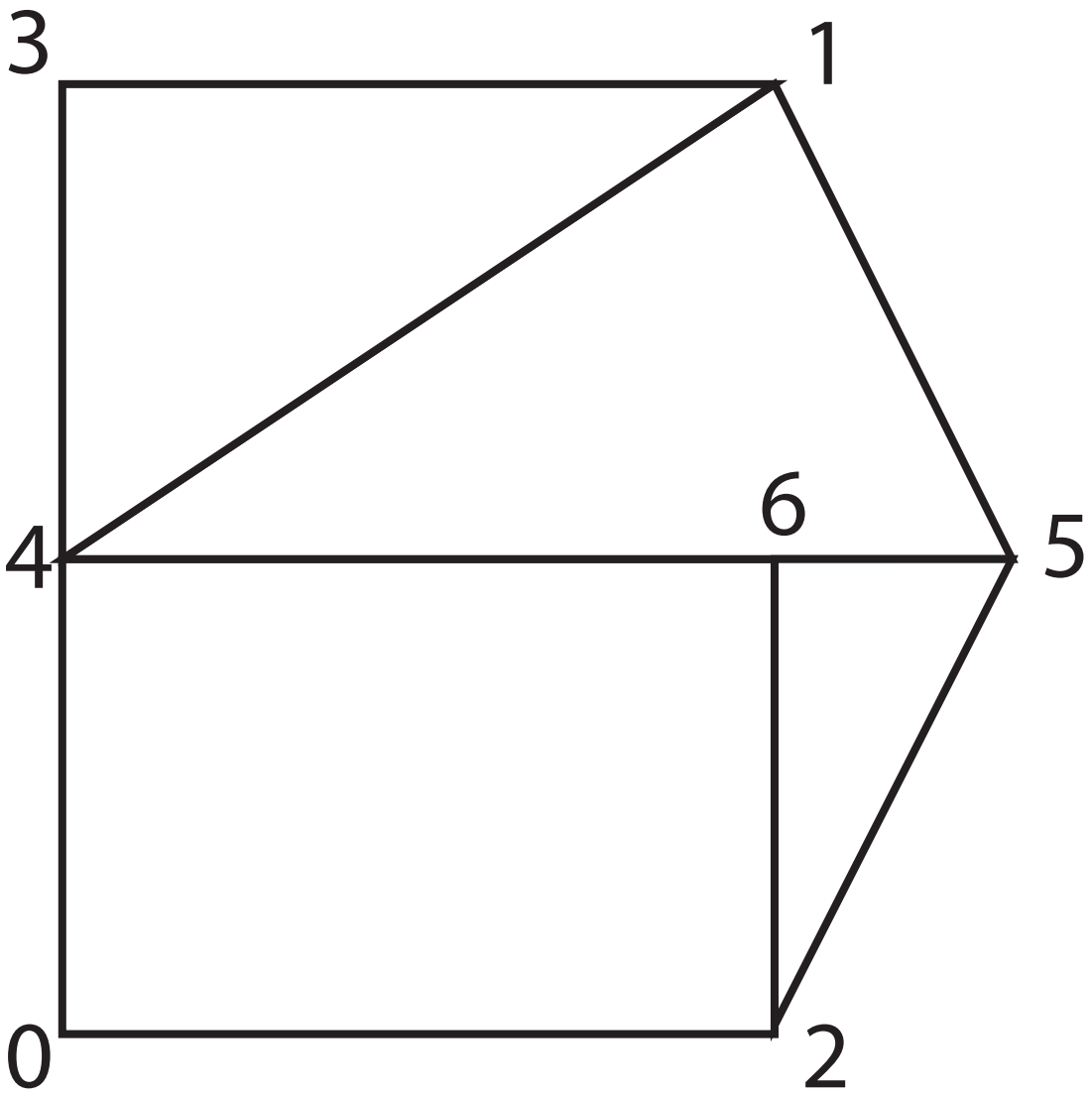
BRC (Binary Row Compressed) representation

List of list of integer typically used for input of cell complexes

Let $A = (a_{i,j} \in \{0,1\})$ be a characteristic matrix.

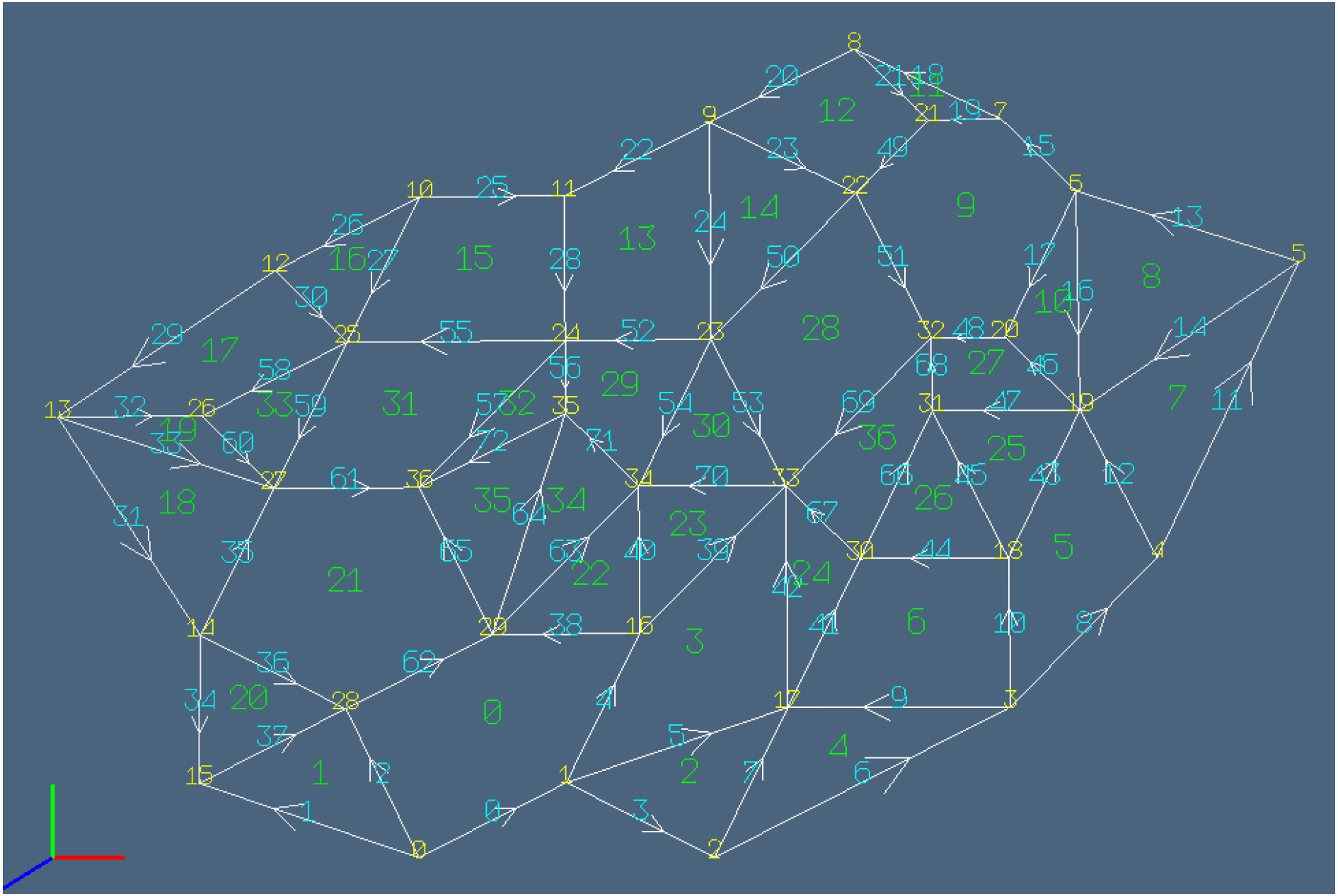
$$A = \begin{pmatrix} 0,1,0,0,0,0,0,1,0,0 \\ 0,0,1,0,0,0,0,0,0,0 \\ 1,0,0,1,0,0,0,0,0,1 \\ 1,0,0,0,0,0,1,0,0,0 \\ 0,0,0,0,0,1,1,1,0,0 \\ 0,0,1,0,1,0,0,0,1,0 \\ 0,0,0,0,0,0,0,0,0,0 \\ 0,1,0,0,0,0,0,1,0,1 \\ 0,0,0,1,0,0,0,0,1,0 \\ 0,1,1,0,1,0,0,0,0,0 \end{pmatrix} \quad \mapsto \quad \texttt{BRC}(A) = \begin{array}{l} \texttt{[[1,7],} \\ \texttt{[2],} \\ \texttt{[0,3,9],} \\ \texttt{[0,6],} \\ \texttt{[5,6,7],} \\ \texttt{[2,4,8],} \\ \texttt{[],} \\ \texttt{[1,7,9],} \\ \texttt{[3,8],} \\ \texttt{[1,2,4]]} \end{array}$$

The idea: generalisation of abstract simplicial sets



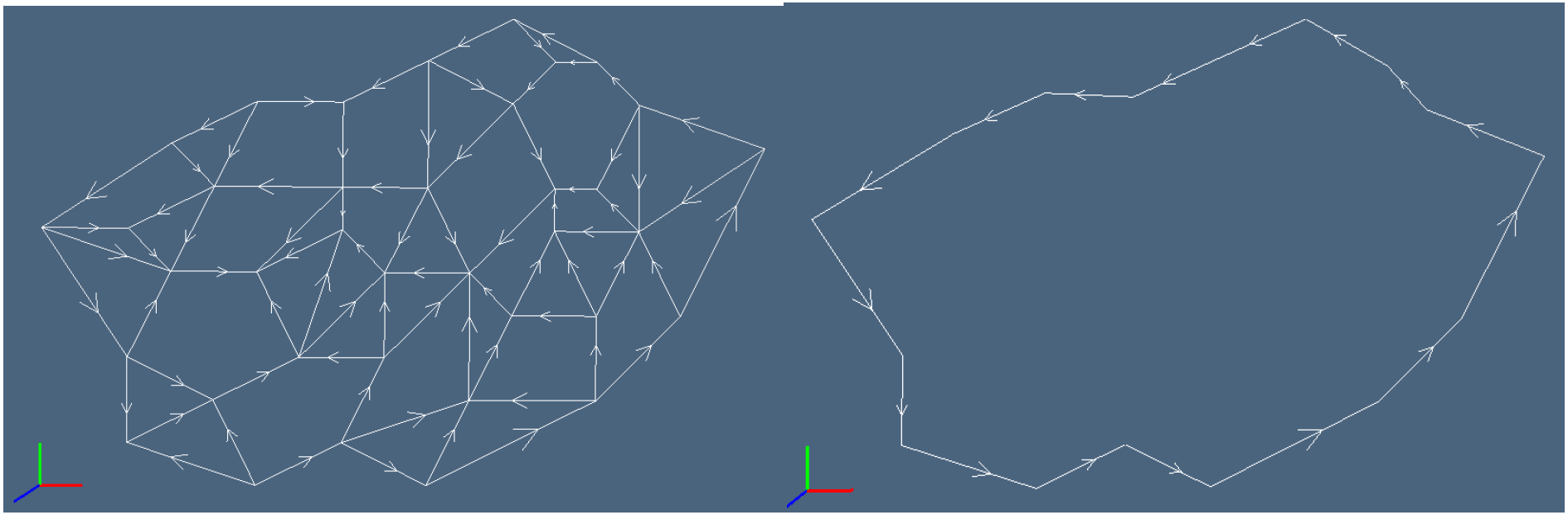
n-cells by their vertices:

```
[[0,2,4,6],
[1,4,5,6],
[1,5,6],
[2,5,6]]
```



The bare minimum data

with full awareness of topology



```
V=[[5,0],[7,1],[9,0],[13,2],[15,4],[17,8],[14,9],[13,10],[11,11],[9,10],[5,9],[7,
9],[3,8],[0,6],[2,3],[2,1],[8,3],[10,2],[13,4],[14,6],[13,7],[12,10],[11,9],[9,7],
[7,7],[4,7],[2,6],[3,5],[4,2],[6,3],[11,4],[12,6],[12,7],[10,5],[8,5],[7,6],[5,5]]
```

```
FV = [[0,1,16,28,29],[0,15,28],[1,2,17],[1,16,17,33],[2,3,17],[3,4,18,19],[3,17,
18,30],[4,5,19],[5,6,19],[6,7,20,21,22,32],[6,19,20],[7,8,21],[8,9,21,22],[9,11,
23,24],[9,22,23],[10,11,24,25],[10,12,25],[12,13,25,26],[13,14,27],[13,26,27],
[14,15,28],[14,27,28,29,36],[16,29,34],[16,33,34],[17,30,33],[18,19,31],[18,30,
31],[19,20,31,32],[22,23,32,33],[23,24,34,35],[23,33,34],[24,25,27,36],[24,35,36],
[25,26,27],[29,34,35],[29,35,36],[30,31,32,33]]
```

Compressed Sparse Row (CSR) matrix storage

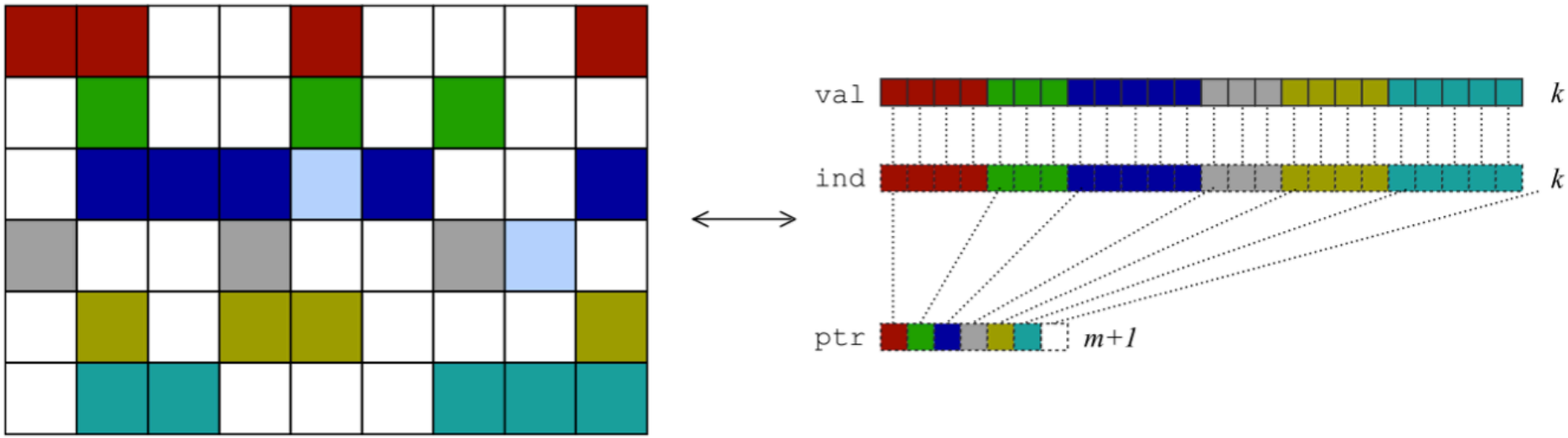
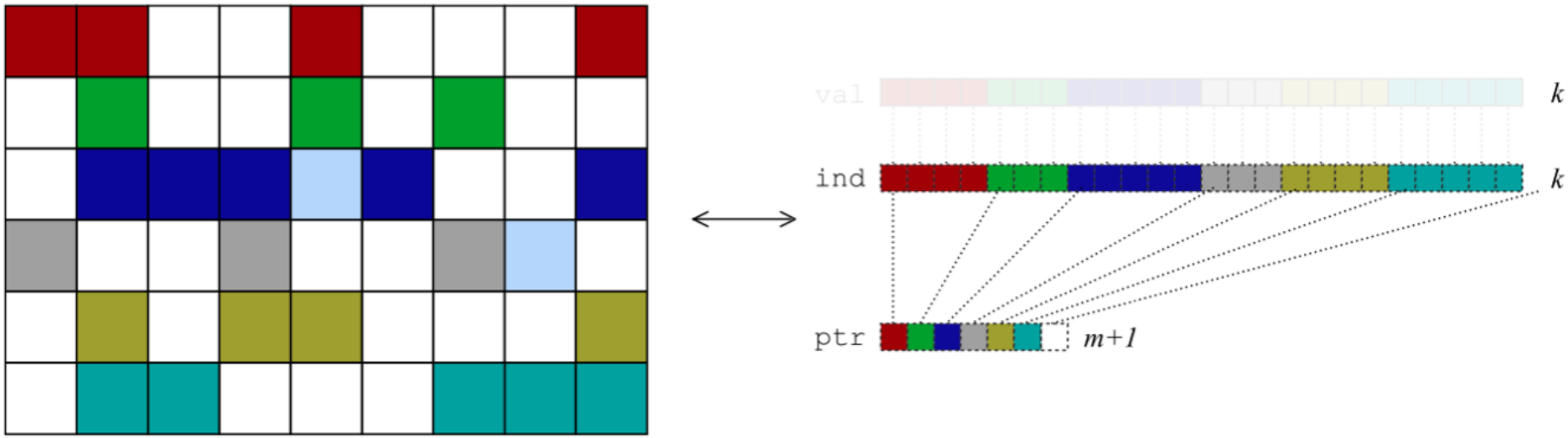


image from

Samuel Williams, Leonid Oliker, Richard Vuduc, John Shalf, Katherine Yelick, and James Demmel, Optimization of sparse matrix-vector multiplication on emerging multicore platforms, Proceedings of the 2007 ACM/IEEE conference on Supercomputing (New York, NY, USA), SC '07, ACM, 2007, pp. 38:1–38:12.

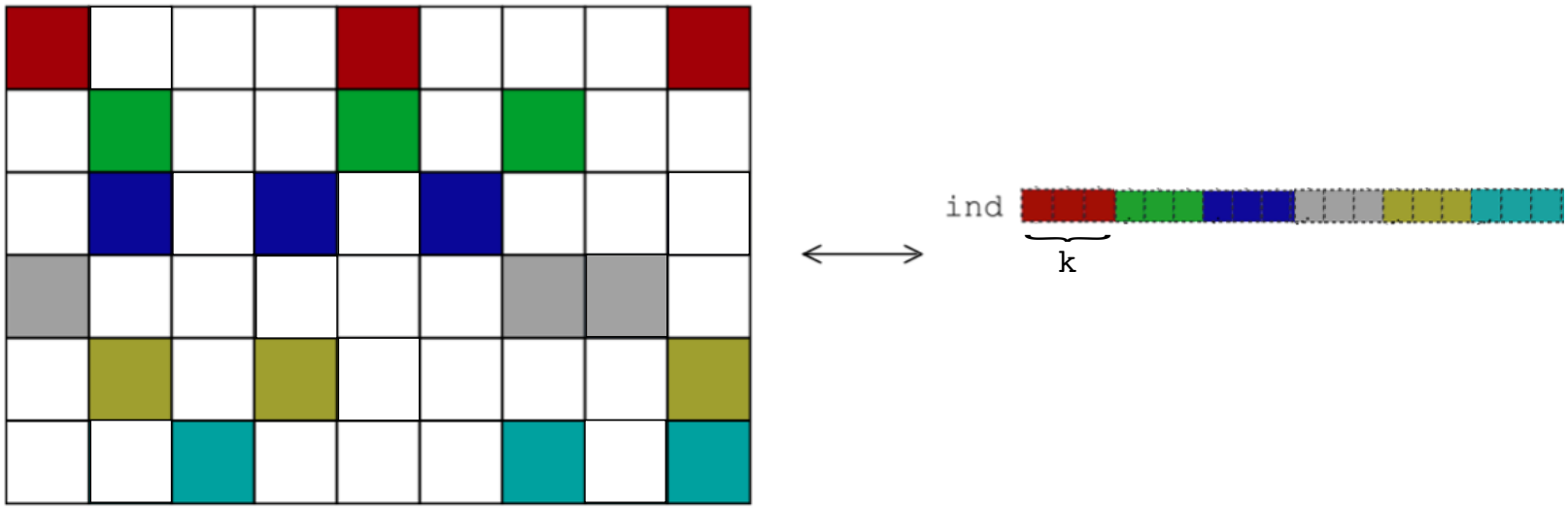
CSR storage of a BINARY matrix



of course, non-zero values (all ones) do not require storage

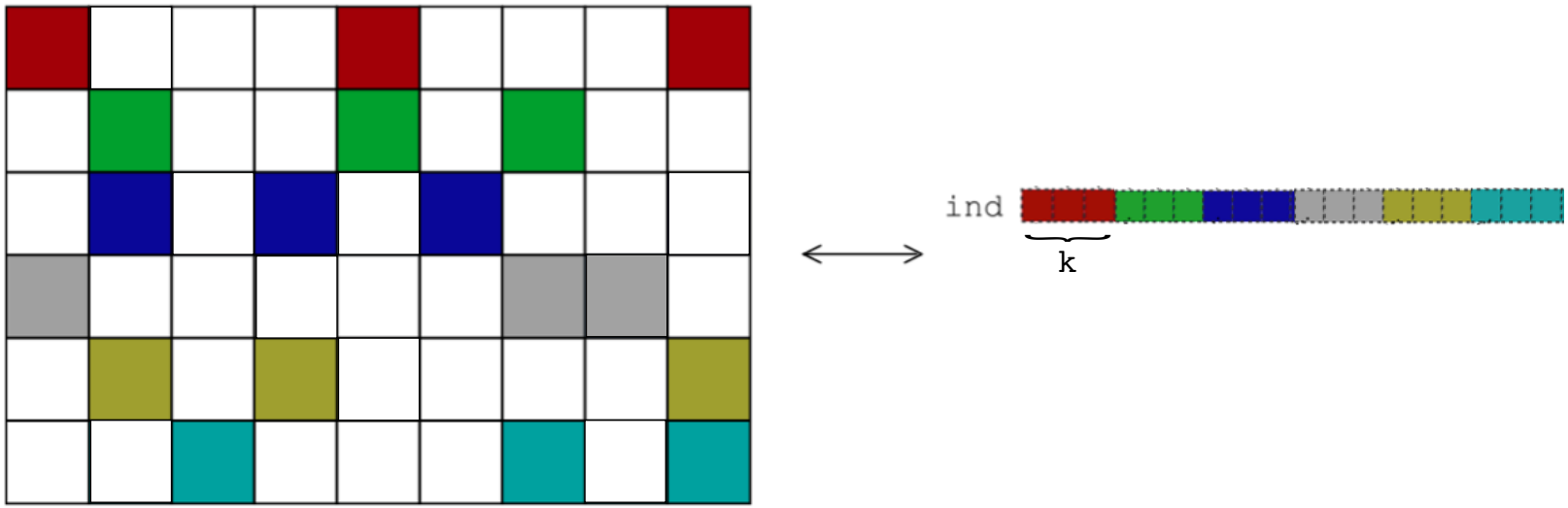
CSR storage of a binary matrix M_d such that

$$M_d \mathbf{1} = \mathbf{k}$$



CSR storage of a binary matrix M_d such that

$$M_d \mathbf{1} = \mathbf{k}$$



important cases:

- regular simplicial d-complexes: $k = d + 1$
- regular cuboidal d-complexes: $k = 2^d$

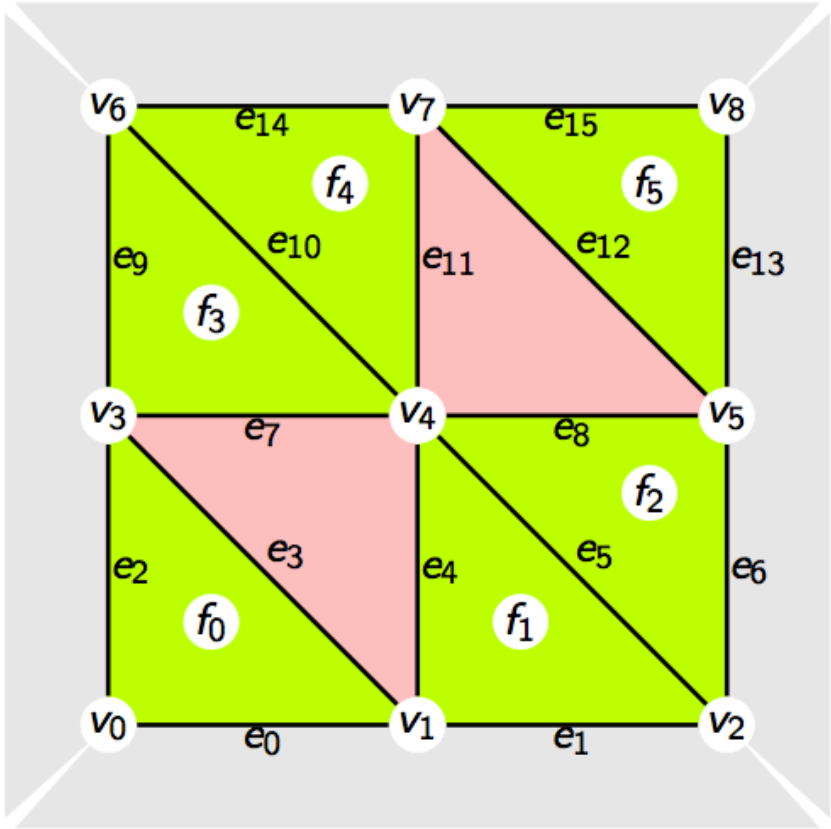
Operations

Facet extraction (1/5)

use the characteristic matrix M_d of a cellular partition Λ_d of $\mathbb{E}^d$, empty cells included

$$2\text{D} \Rightarrow d = 2$$

$$M_2 = \begin{pmatrix}
1\,1\,0\,1\,0\,0\,0\,0\,0 \\
0\,1\,1\,0\,1\,0\,0\,0\,0 \\
0\,0\,1\,0\,1\,1\,0\,0\,0 \\
0\,0\,0\,1\,1\,0\,1\,0\,0 \\
0\,0\,0\,0\,1\,0\,1\,1\,0 \\
0\,0\,0\,0\,0\,1\,0\,1\,1 \\
0\,1\,0\,1\,1\,0\,0\,0\,0 \\
0\,0\,0\,0\,1\,1\,0\,1\,0 \\
1\,1\,1\,0\,0\,0\,0\,0\,0 \\
0\,0\,1\,0\,0\,1\,0\,0\,1 \\
0\,0\,0\,0\,0\,0\,1\,1\,1 \\
1\,0\,0\,1\,0\,0\,1\,0\,0
\end{pmatrix}$$

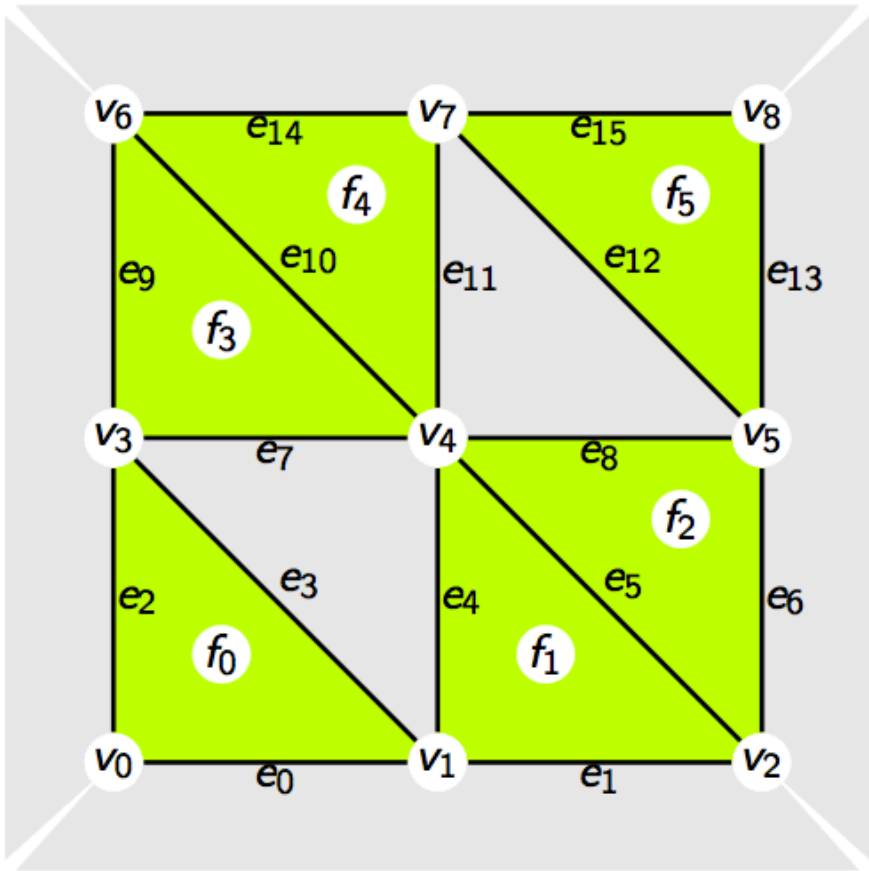


Facet extraction (2/5)

use the characteristic matrix M_d of a cellular partition Λ_d of $\mathbb{E}^d$, empty cells included

$$2D \Rightarrow d = 2$$

$$M_2 = \begin{pmatrix}
1&1&0&1&0&0&0&0&0\\
0&1&1&0&1&0&0&0&0\\
0&0&1&0&1&1&0&0&0\\
0&0&0&1&1&0&1&0&0\\
0&0&0&0&1&0&1&1&0\\
0&0&0&0&0&1&0&1&1\\
0&1&0&1&1&0&0&0&0\\
0&0&0&0&1&1&0&1&0\\
1&1&1&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&1\\
0&0&0&0&0&0&1&1&1\\
1&0&0&1&0&0&1&0&0
\end{pmatrix}$$



Facet extraction (3/5)

use the characteristic matrix M_d of a cellular partition Λ_d of $\mathbb{E}^d$, empty cells included

$$A = M_2 M_2^t = \begin{pmatrix}
3 & 1 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 2 & 2 & 0 \\
 & 3 & 2 & 1 & 1 & 0 & 2 & 1 & 0 & 0 & 2 & 1 \\
 & & 3 & 1 & 1 & 1 & 1 & 2 & 0 & 0 & 1 & 2 \\
 & & & 3 & 2 & 0 & 0 & 0 & 1 & 2 & 2 & 1 \\
 & & & & 3 & 1 & 0 & 0 & 2 & 1 & 1 & 2 \\
 & & & & & 3 & 0 & 2 & 2 & 0 & 0 & 2 \\
 & & & & & & 3 & 1 & 0 & 1 & 1 & 0 \\
 & & & & & & & 3 & 1 & 0 & 0 & 1 \\
 & & & & & & & & 3 & 1 & 0 & 1 \\
 & & & & & & & & & 3 & 1 & 0 \\
 & & & & & & & & & & 3 & 1 \\
 & & & & & & & & & & & 3
\end{pmatrix}$$

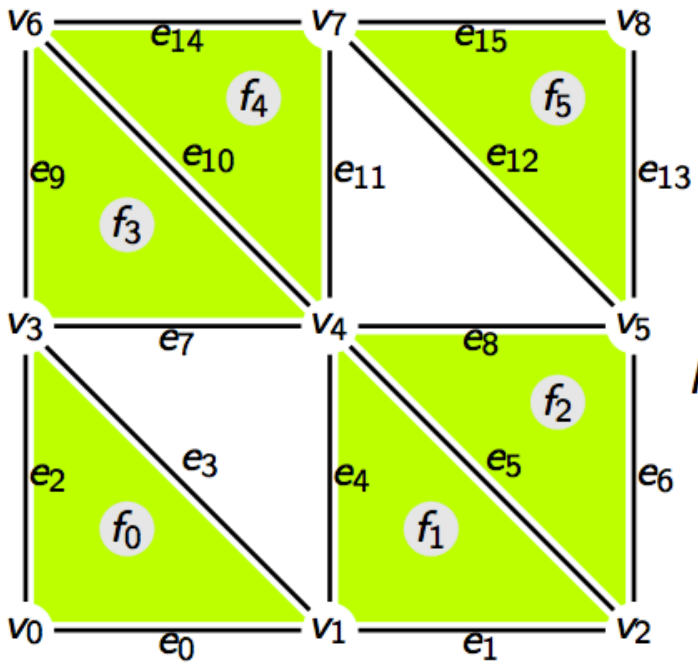
sym

$$\sharp(\lambda_d^i \cap \lambda_d^j) = A(i,j) \geq d \ (i \neq j) \ \Rightarrow \ \exists \ \lambda_{d-1} = M_d(i) \wedge M_d(j)$$

Facet extraction (4/5)

use the characteristic matrix M_d of a cellular partition Λ_d of $\mathbb{E}^d$, empty cells included

$$A(0,6) = 2 \quad \Rightarrow \quad (\,110100000\,) \wedge (\,010110000\,) = (\,010100000\,) \quad \Rightarrow \quad e = (v_1, v_3)$$

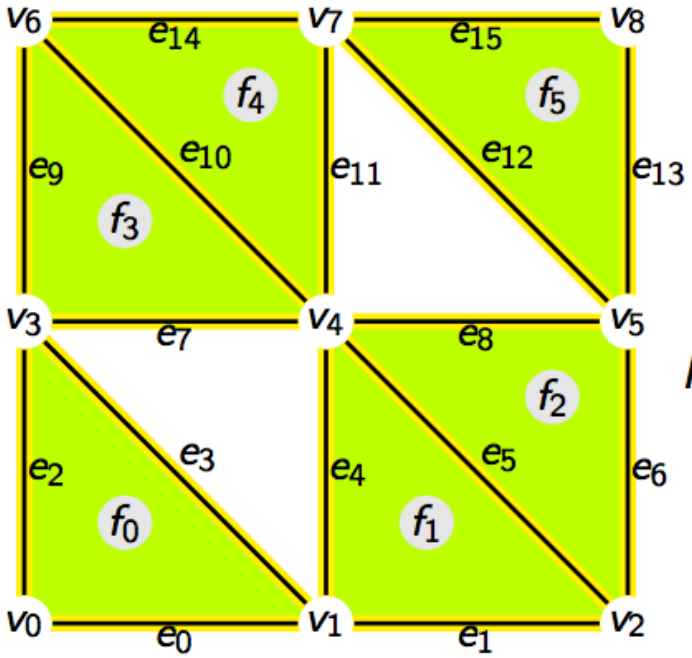


$$M_2 = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,1 \\ 0\,1\,0\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0\,1\,0 \\ 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0\,0 \end{pmatrix} \Rightarrow M_1 = \begin{pmatrix} 1\,1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,1 \end{pmatrix}$$

Facet extraction (5/5)

use the characteristic matrix M_d of a cellular partition Λ_d of $\mathbb{E}^d$, empty cells included

$$A(0,6) = 2 \quad \Rightarrow \quad (\,110100000\,) \wedge (\,010110000\,) = (\,010100000\,) \quad \Rightarrow \quad e = (v_1, v_3)$$



$$M_2 = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,1 \\ 0\,1\,0\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0\,1\,0 \\ 1\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \\ 1\,0\,0\,1\,0\,0\,1\,0\,0 \end{pmatrix} \Rightarrow M_1 = \begin{pmatrix} 1\,1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,1\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,1 \end{pmatrix}$$

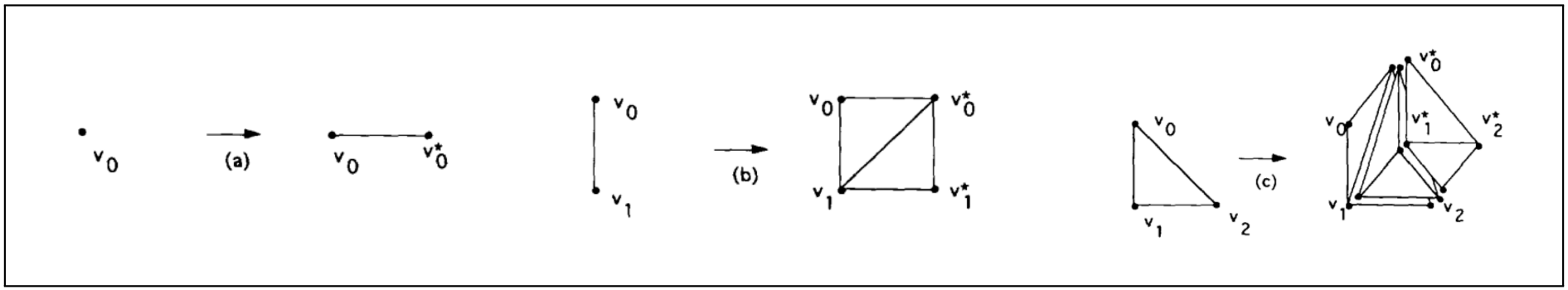
Simplicial extrusion (1/3)

Optimal combinatorial algorithm [Ferrucci & Paoluzzi, CAD, 1991]

for each d-simplex

$$\sigma^d = (v_0, v_1, \ldots, v_d)$$

in the input complex $S(d)$, we generate combinatorially a chain of $d+1$ coherently-oriented simplexes of dimension $d+1$:



So we have, with $|\gamma^{d+1}| = \sigma^d \times I$, and $I = [0, 1]$:

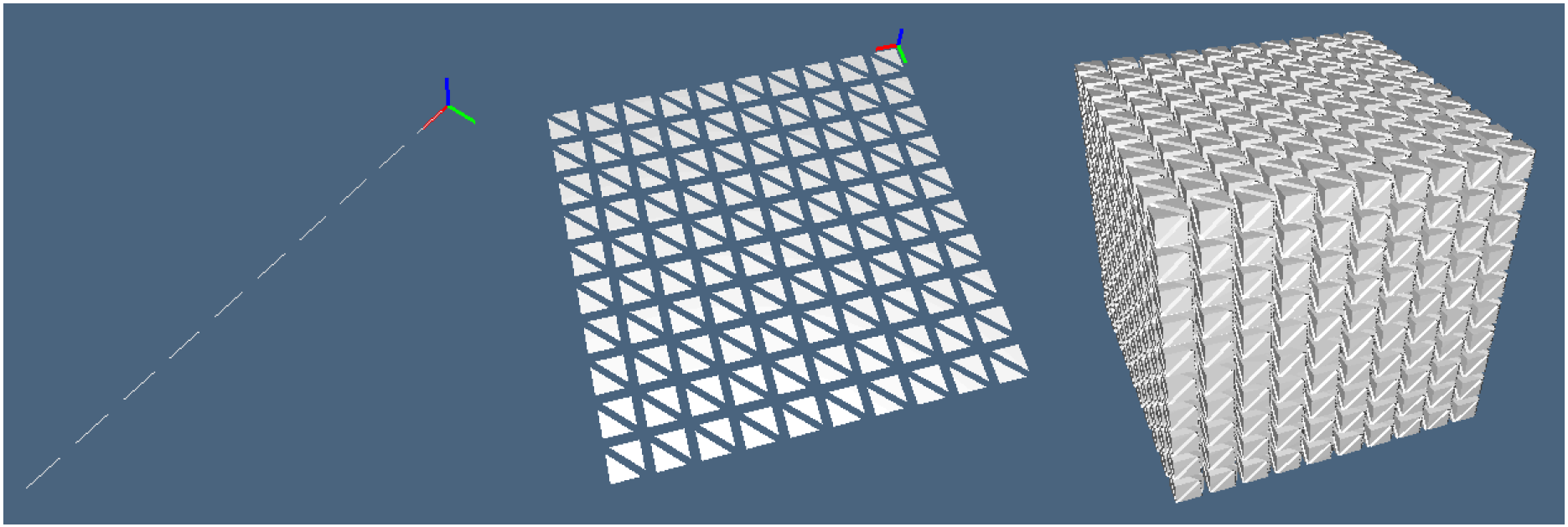
$$\gamma^{d+1} = \sum_{k=0}^{d} (-1)^{kd} \langle v_k, \ldots v_d, v_0^*, \ldots v_k^* \rangle$$

with $v_k \in \sigma^d \times \{0\}$ and $v_k^* \in \sigma^d \times \{1\}$,

Simplicial extrusion (2/3)

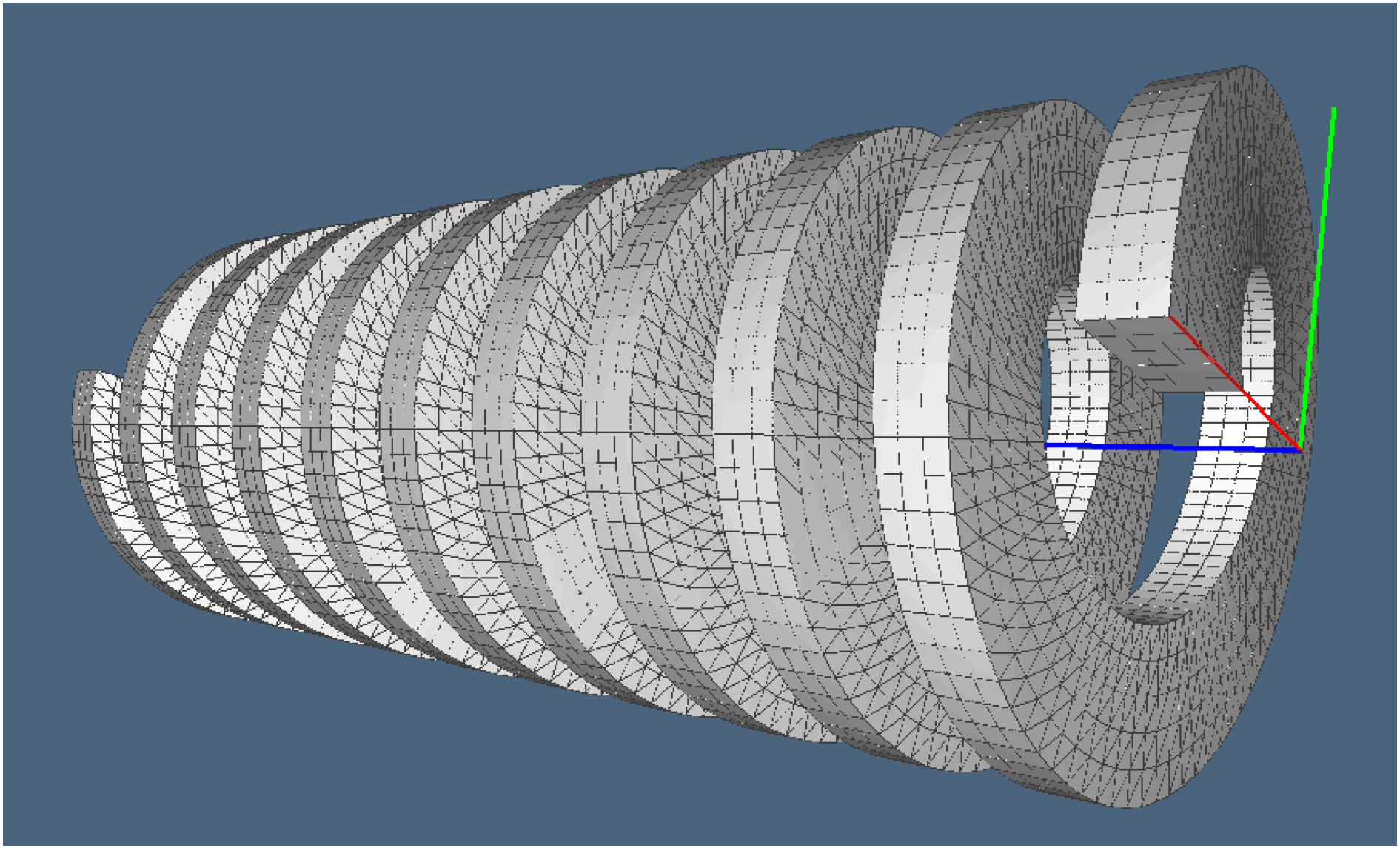
Let $\mathfrak{S}$ be the set of simplicial c.c.c., and $S_d, S_1 \in \mathfrak{S}$:

$$S_d \times S_1 =: S_{d+1} \in \mathfrak{S}$$



```
model1 = larExtrude( VOID, 10*[1] )
model2 = larExtrude( model1, 10*[1] )
model3 = larExtrude( model2, 10*[1] )
```

Simplicial extrusion (3/3)



Cartesian product of complexes (1/4)

Let $\mathfrak{C}$ be the set of c.c.c's. $X, Y \in \mathfrak{C} \Rightarrow X \times Y \in \mathfrak{C}$ [Basak, Geom. dedicata, 2010]

Let

$$X = S(m) \in \mathfrak{S}, \quad \text{and} \quad Y = S(n) \in \mathfrak{S};$$

Then

$$X \times Y = S(X \times Y) \in \mathfrak{S}$$

with

$$x \times y = (x_1 y_1, \ldots, x_1 y_n, \ldots, x_m y_1, \ldots, x_m y_n) \in S(m+n).$$

Cartesian product of complexes (2/4)

Implementation

```
def larModelProduct(twoModels):
    (V, cells1), (W, cells2) = twoModels
    vertices = collections.OrderedDict(); k = 0
    for v in V:
        for w in W:
            id = tuple(v+w)
            if not vertices.has_key(id):
                vertices[id] = k
                k += 1
    cells = [ [vertices[tuple(V[v] + W[w])] for v in c1 for w in c2 ]
             for c1 in cells1 for c2 in cells2]
    model = [list(v) for v in vertices.keys()], cells
    return model
```

Cartesian product of complexes (4/4)