Computational Homology in Topological Dynamics

ACAT School, Bologna, Italy
MAY 26, 2012

Marian Mrozek

Jagiellonian University, Kraków

Outline

- Dynamical systems
- Rigorous numerics of dynamical systems
- Homological invariants of dynamical systems
- Computing homological invariants
- **Homology algorithms for subsets of $\mathbb{R}^d$**
- Homology algorithms for maps of subsets of $\mathbb{R}^d$
- Applications

Goal

Input: representation of a subset $X \subset \mathbb{R}^d$
Output: Betti numbers, torsion coefficients, homology generators

Goal 3

Input: representation of a subset $X \subset \mathbb{R}^d$
Output: Betti numbers, torsion coefficients, homology generators

Input: representation of a continuous map $f : X \to Y$ of subsets of $\mathbb{R}^d$
Output: matrix of map induced in homology

Prototype homology algorithm

In principle, the homology algorithm may look as follows:

Prototype homology algorithm

In principle, the homology algorithm may look as follows:

(1) Triangulate the space

Prototype homology algorithm

In principle, the homology algorithm may look as follows:

(1) Triangulate the space

(2) Construct the matrices of boundary maps

Prototype homology algorithm

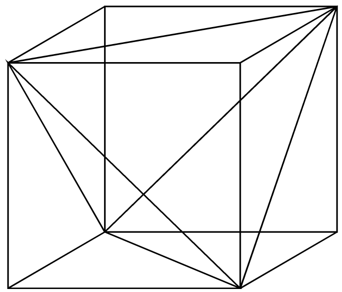
In principle, the homology algorithm may look as follows:

(1) Triangulate the space
(2) Construct the matrices of boundary maps
(3) Compute the kernel and the image

Prototype homology algorithm

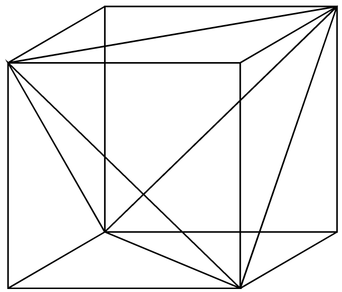
In principle, the homology algorithm may look as follows:

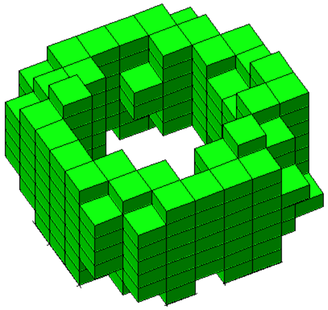
(1) Triangulate the space
(2) Construct the matrices of boundary maps
(3) Compute the kernel and the image
(4) Compute the quotient space



Simplicial complex

- classical





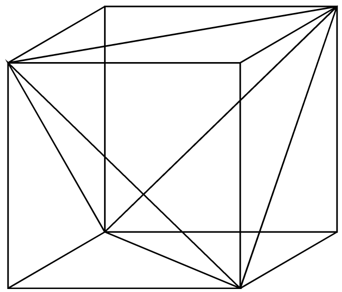
Simplicial complex

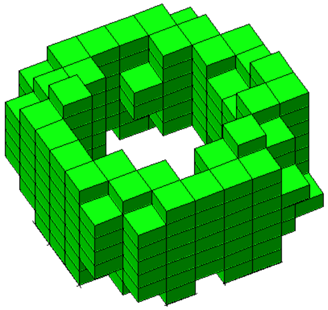
- classical

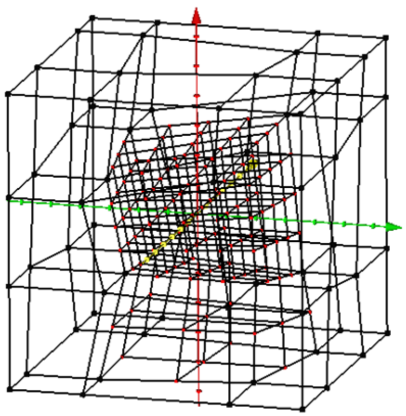
Cubical set

- typical in imaging and rigorous numerics
- very efficient and fast representation (bitmaps)

Set representation







Simplicial complex

- classical

Cubical set

- typical in imaging and rigorous numerics
- very efficient and fast representation (bitmaps)

General polyhedrons

- most general
- obtaining the chain complex is not straightforward

Cube triangulation

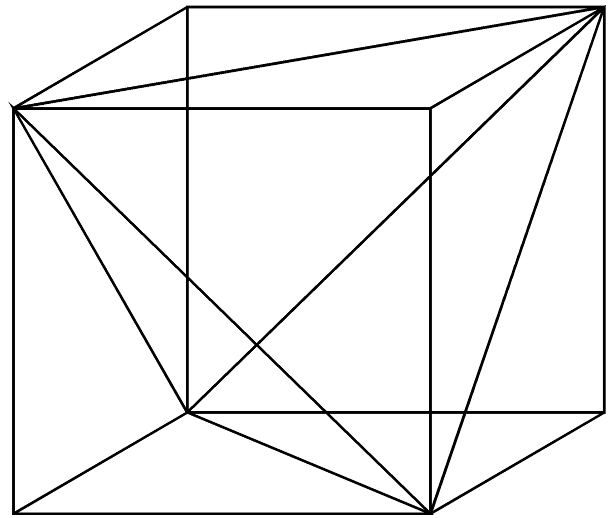
- How many simplices do we need to triangulate a d-cube?

Cube triangulation

- How many simplices do we need to triangulate a d-cube?
- Not more than $d!$ but can we do better?

Cube triangulation

- How many simplices do we need to triangulate a d-cube?
- Not more than $d!$ but can we do better?



Cube triangulation

Theorem. Hughes, Anderson (1995), Bliss, Su (2005)

d	1	2	3	4	5	6	7
$T^v(d)$	1	2	5	16	67	308	1493
$T(d)$	1	2	5	16	?	?	?

Theorem. Smith (2000)

$$C(d) \geq \frac{6^{d/2} d!}{2(d+1)^{(d+1)/2}}$$

Input

Algebraists expect matrices of boundary maps as input of homology computations.

Input

Algebraists expect matrices of boundary maps as input of homology computations.

- On input: a set represented as a list of top dimensional cells (cubes, simplices, ...)
- Generation of faces, incidence coefficients and boundary maps, whenever necessary, must be considered a part of the job!

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is
$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is

$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$

- The dimension of Q is the number of nondegenerate I_i.

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is
$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$
- The dimension of Q is the number of nondegenerate I_i.
- $\mathcal{K}$ — the set of all elementary cubes in $\mathbb{R}^d$

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is
$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$
- The dimension of Q is the number of nondegenerate I_i.
- $\mathcal{K}$ — the set of all elementary cubes in $\mathbb{R}^d$
- $\mathcal{K}_k$ — the set of all elementary cubes in $\mathbb{R}^d$ of dimension k

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is

$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$

- The dimension of Q is the number of nondegenerate I_i.
- $\mathcal{K}$ — the set of all elementary cubes in $\mathbb{R}^d$
- $\mathcal{K}_k$ — the set of all elementary cubes in $\mathbb{R}^d$ of dimension k
- An elementary cube is full if its dimension is d.

Elementary intervals and cubes

- An elementary interval is an interval $[k, l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k + 1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is

$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$

- The dimension of Q is the number of nondegenerate I_i.
- $\mathcal{K}$ — the set of all elementary cubes in $\mathbb{R}^d$
- $\mathcal{K}_k$ — the set of all elementary cubes in $\mathbb{R}^d$ of dimension k
- An elementary cube is full if its dimension is d.
- For $\mathcal{A} \subset \mathcal{K}$ we use notation $|\mathcal{A}| := \bigcup \mathcal{A}$.

Elementary intervals and cubes

- An elementary interval is an interval $[k,l] \subset \mathbb{R}$ such that $l = k$ (degenerate) or $l = k+1$ (nondegenerate).
- An elementary cube Q in $\mathbb{R}^d$ is
$$I_1 \times I_2 \times \cdots \times I_d \subset \mathbb{R}^d.$$
- The dimension of Q is the number of nondegenerate I_i.
- $\mathcal{K}$ — the set of all elementary cubes in $\mathbb{R}^d$
- $\mathcal{K}_k$ — the set of all elementary cubes in $\mathbb{R}^d$ of dimension k
- An elementary cube is full if its dimension is d.
- For $\mathcal{A} \subset \mathcal{K}$ we use notation $|\mathcal{A}| := \bigcup \mathcal{A}$.
- For $A \subset \mathbb{R}^d$ we use notation $\mathcal{K}(A) := \{\, Q \in \mathcal{K} \mid Q \subset A \,\}$.

- Given an elementary cube Q we define the associated elementary chain by

$$\widehat{Q}(P) = \begin{cases} 1 & \text{if } P = Q \\ 0 & \text{otherwise.} \end{cases}$$

Cubical Chains

- Given an elementary cube Q we define the associated elementary chain by
$$\widehat{Q}(P) = \begin{cases} 1 & \text{if } P = Q \\ 0 & \text{otherwise.} \end{cases}$$
- A cubical chain is a finite linear combination of elementary chains of the same dimension, called the dimension of the chain.

Cubical Chains

- Given an elementary cube Q we define the associated elementary chain by

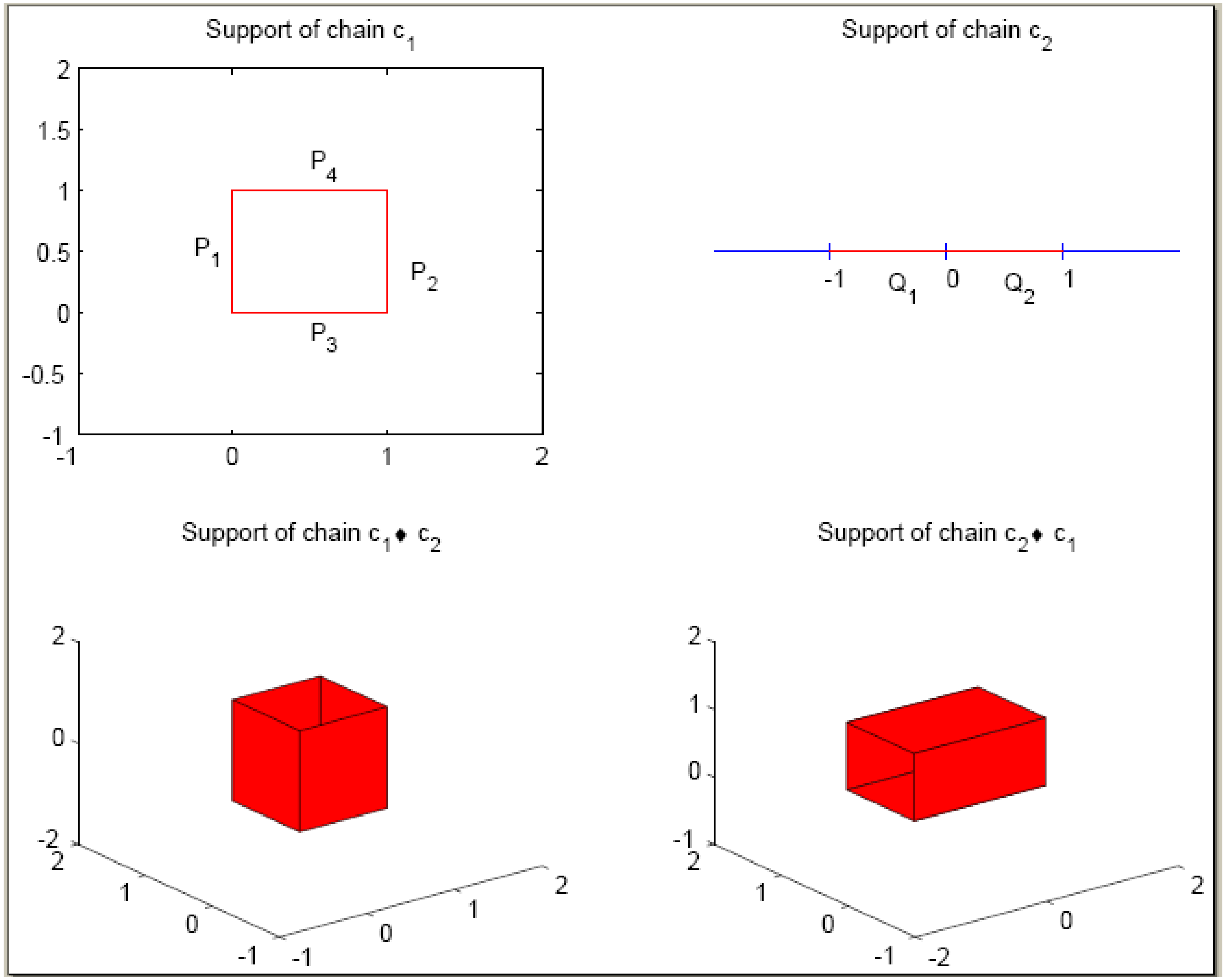
$$\widehat{Q}(P) = \begin{cases} 1 & \text{if } P = Q \\ 0 & \text{otherwise.} \end{cases}$$

- A cubical chain is a finite linear combination of elementary chains of the same dimension, called the dimension of the chain.
- All cubical chains of dimension q form an Abelian group, denoted C_q and called the group of q-chains.

Cubical Product

- Given two elementary chains $\widehat{P}, \widehat{Q}$, we define their cubical product by
$$\widehat{P} \diamond \widehat{Q} := \widehat{P \times Q}.$$
and we extend this definition linearly to arbitrary chains.

Cubical Product



Boundary Operator

- Boundary operator is a homomorphism $\partial : C_q \to C_{q-1}$ given on generators by

$$\partial\widehat{Q} := \begin{cases} 0 & \text{if } Q = [l], \\ \widehat{[l+1]} - \widehat{[l]} & \text{if } Q = [l, l+1]. \\ \partial\widehat{I} \diamond \widehat{P} + (-1)^{\dim I}\widehat{I} \diamond \partial\widehat{P} & \text{if } Q = I \times P. \end{cases}$$

Boundary Operator

- Boundary operator is a homomorphism $\partial : C_q \to C_{q-1}$ given on generators by

$$\partial \widehat{Q} := \begin{cases} 0 & \text{if } Q = [l], \\ \widehat{[l+1]} - \widehat{[l]} & \text{if } Q = [l, l+1]. \\ \partial \widehat{I} \diamond \widehat{P} + (-1)^{\dim I} \widehat{I} \diamond \partial \widehat{P} & \text{if } Q = I \times P. \end{cases}$$

Theorem.

$$\partial \circ \partial = 0$$

Chain groups of a cubical set

- For an elementary chain $c = \sum_{i=1}^{n} \alpha_i \widehat{Q}_i$ we define its support by
$$|c| := \bigcup \{ Q_i \mid \alpha_i \neq 0 \}$$

Chain groups of a cubical set

- For an elementary chain $c = \sum_{i=1}^{n} \alpha_i \widehat{Q}_i$ we define its support by
$$|c| := \bigcup \{\, Q_i \mid \alpha_i \neq 0 \,\}$$
- Given a cubical set X we define the group of q-chains of X by
$$C_q(X) := \{\, c \in C_q \mid |c| \subset X \,\}.$$

Chain groups of a cubical set

- For an elementary chain $c = \sum_{i=1}^{n} \alpha_i \widehat{Q}_i$ we define its support by
$$|c| := \bigcup \{\, Q_i \mid \alpha_i \neq 0 \,\}$$
- Given a cubical set X we define the group of q-chains of X by
$$C_q(X) := \{\, c \in C_q \mid |c| \subset X \,\}.$$
- Is is easy to verify that we have the induced boundary operator
$$\partial_q^X : C_q(X) \to C_{q-1}(X).$$

- The kernel of ∂_q^X is called the group of q-cycles of X and denoted by $Z_q(X)$.

- The kernel of ∂_q^X is called the group of q-cycles of X and denoted by $Z_q(X)$.
- The image of ∂_{q+1}^X is called the group of q-boundaries of X and denoted by $B_q(X)$.

- The kernel of ∂_q^X is called the group of q-cycles of X and denoted by $Z_q(X)$.
- The image of ∂_{q+1}^X is called the group of q-boundaries of X and denoted by $B_q(X)$.
- One can verify that $B_q(X) \subset Z_q(X)$, which allows us to define the qth homology group of X by

$$H_q(X) := Z_q(X)/B_q(X)$$

Cubical Homology

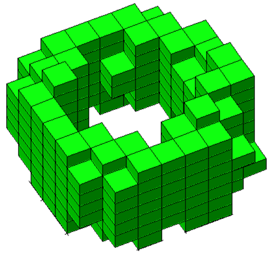
- The kernel of ∂_q^X is called the group of q-cycles of X and denoted by $Z_q(X)$.
- The image of ∂_{q+1}^X is called the group of q-boundaries of X and denoted by $B_q(X)$.
- One can verify that $B_q(X) \subset Z_q(X)$, which allows us to define the qth homology group of X by

$$H_q(X) := Z_q(X)/B_q(X)$$

- By homology of X we mean the collection of all homology groups $H(X) := \{H_q(X)\}$.

Immediate algebraization:

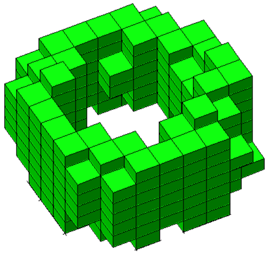
Standard approach



Immediate algebraization:

- generate the faces

Standard approach

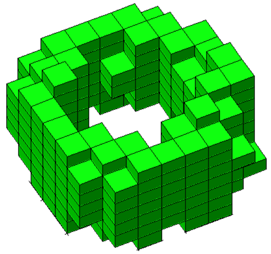


Immediate algebraization:

- generate the faces
- construct the boundary maps

$$D_k = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & 0 & 0 & \dots \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \dots \\ \mathbf{-1} & \mathbf{1} & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Standard approach 21



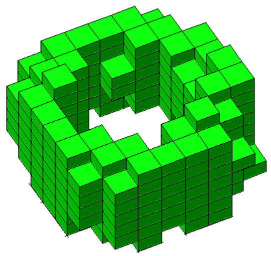
Immediate algebraization:

- generate the faces
- construct the boundary maps
- find Smith diagonalization and read Betti numbers

$$D_k = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & 0 & 0 & \dots \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \dots \\ \mathbf{-1} & \mathbf{1} & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$

$$B_k = Q^{-1} D_k R$$

$$B_k = \begin{bmatrix} \mathbf{2} & 0 & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ 0 & 0 & \mathbf{1} & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$



$$D_k = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & 0 & 0 & \dots \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \dots \\ \mathbf{-1} & \mathbf{1} & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$

$$B_k = Q^{-1} D_k R$$

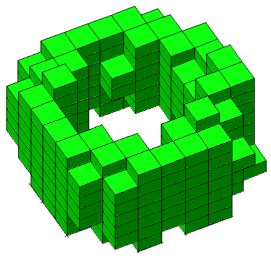
$$B_k = \begin{bmatrix} \mathbf{2} & 0 & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ 0 & 0 & \mathbf{1} & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$

Immediate algebraization:

- generate the faces
- construct the boundary maps
- find Smith diagonalization and read Betti numbers

Advantages:

- standard linear algebra
- may be easily adapted to homology generators



$$D_k = \begin{bmatrix} \mathbf{1} & \mathbf{-1} & 0 & 0 & \dots \\ \mathbf{1} & 0 & \mathbf{1} & 0 & \dots \\ \mathbf{-1} & \mathbf{1} & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$

$$B_k = Q^{-1} D_k R$$

$$B_k = \begin{bmatrix} \mathbf{2} & 0 & 0 & 0 & \dots \\ 0 & \mathbf{1} & 0 & 0 & \dots \\ 0 & 0 & \mathbf{1} & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ . & . & . & . & . \end{bmatrix}$$

Immediate algebraization:

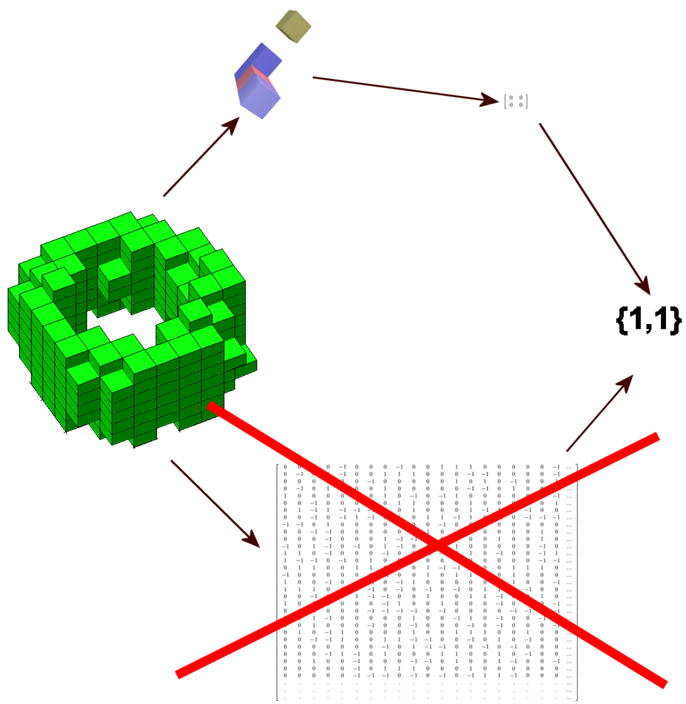
- generate the faces
- construct the boundary maps
- find Smith diagonalization and read Betti numbers

Advantages:

- standard linear algebra
- may be easily adapted to homology generators

Problems:

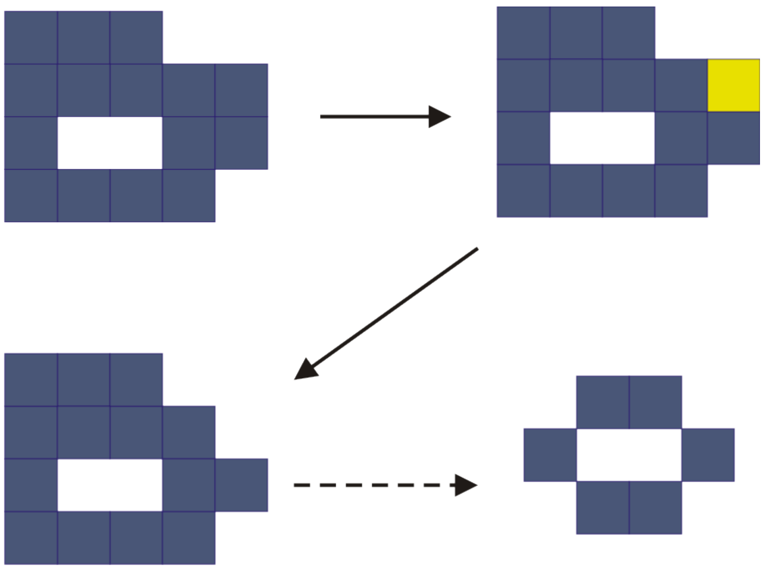
- constructing faces immediately may increase data size
- complexity: Cn^3
- sparseness of matrices may not help (fill-in process)
- C large for sparse matrices (dynamic storage allocation)



Geometric Reductions

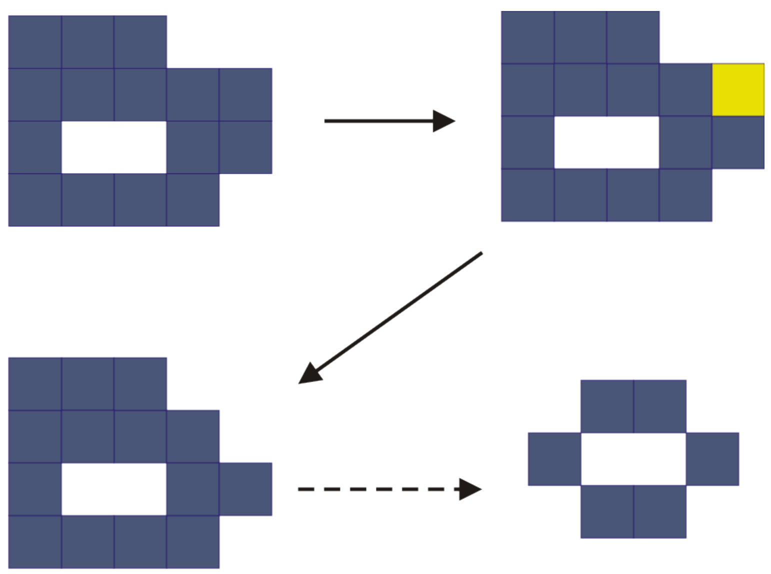
- Reduce the set so that
  - the representation used is preserved
  - the homology is not changed
- build chain complex
- compute homology

Shaving [23]



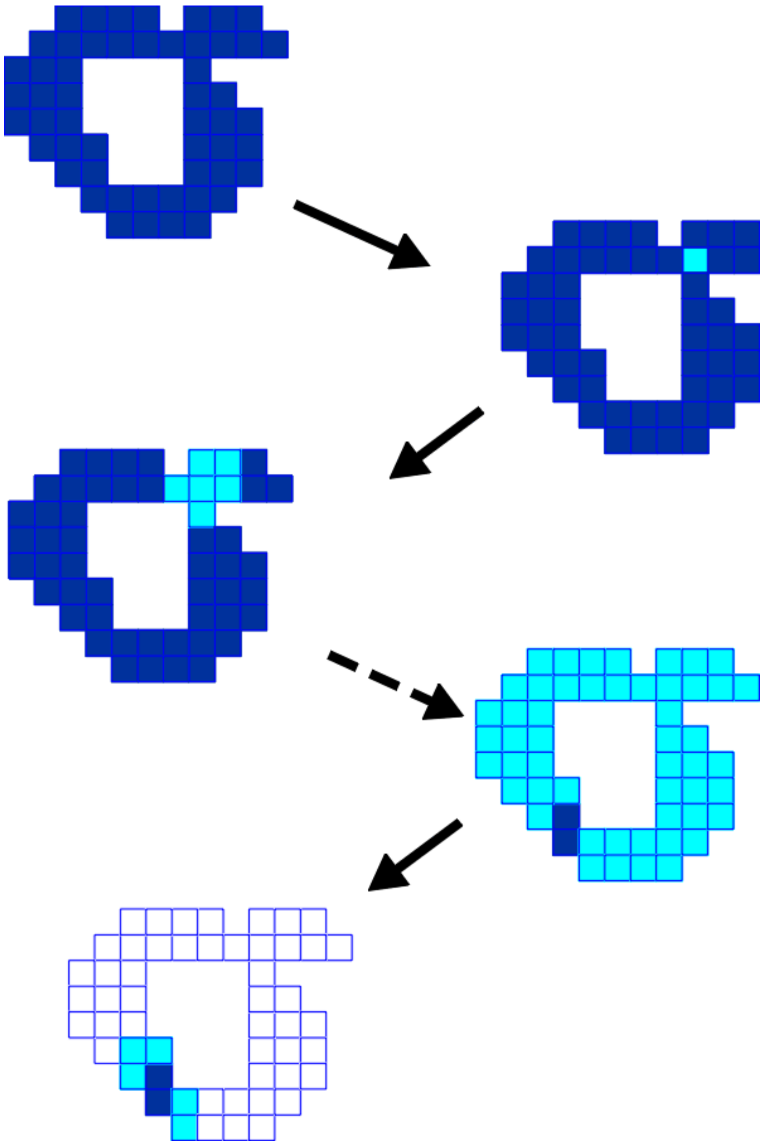
- If $X = \bigcup \mathcal{X}$ is cubical and $Q \in \mathcal{X}$ is an elementary cube such that $Q \cap X$ is acyclic and $X' = \bigcup(\mathcal{X} \setminus \{Q\})$ then
$$H(X) \cong H(X')$$
- full cubes representation is used!
- acyclicity tests via lookup tables:
  - 2^{3^d-1} entries
  - extremely fast in dimension 2 and 3

Shaving [23]



- If $X = \bigcup \mathcal{X}$ is cubical and $Q \in \mathcal{X}$ is an elementary cube such that $Q \cap X$ is acyclic and $X' = \bigcup(\mathcal{X} \setminus \{Q\})$ then
$$H(X) \cong H(X')$$
- full cubes representation is used!
- acyclicity tests via lookup tables:
  - 2^{3^d-1} entries
  - extremely fast in dimension 2 and 3
  - not enough memory for dimension above 3
- partial acyclicity tests in higher dimensions

Acyclic subspace

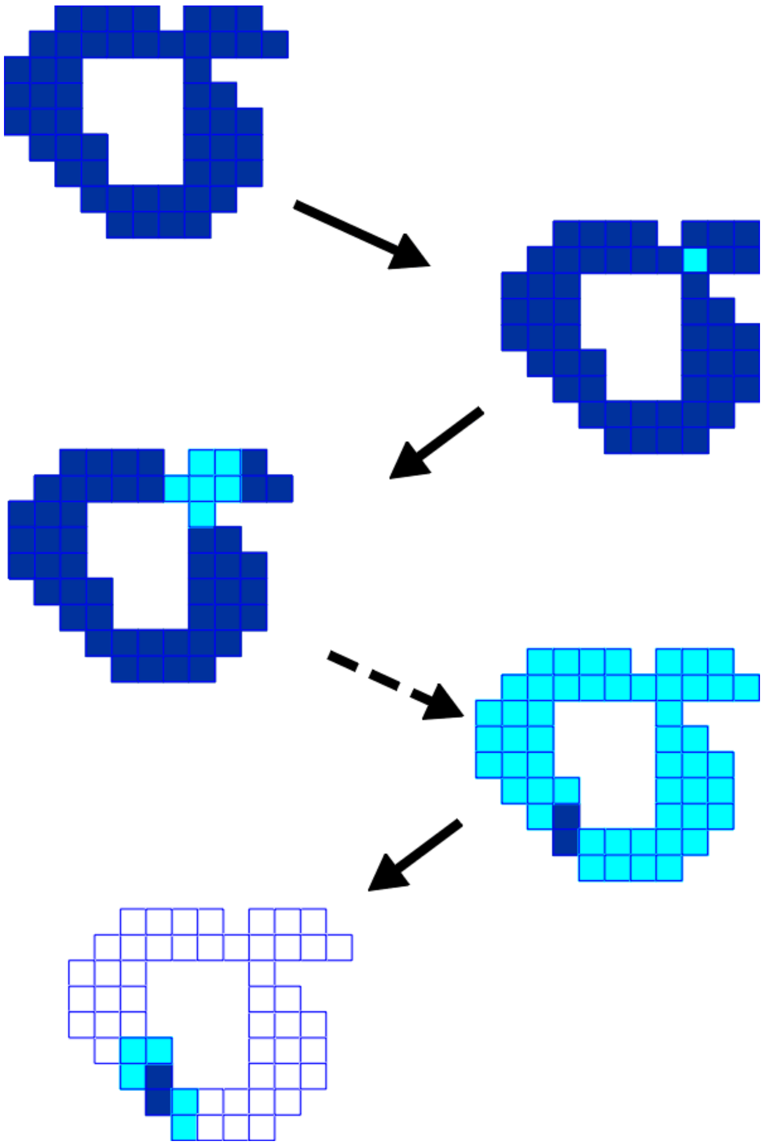


- If X is cubical and $A \subset X$ is acyclic then

$$H_n(X) \cong \begin{cases} H_n(X,A), & \text{for } n \geq 1 \\ \mathbb{Z} \oplus H_n(X,A), & \text{for } n = 0 \end{cases}$$

- breadth-first search construction

Acyclic subspace



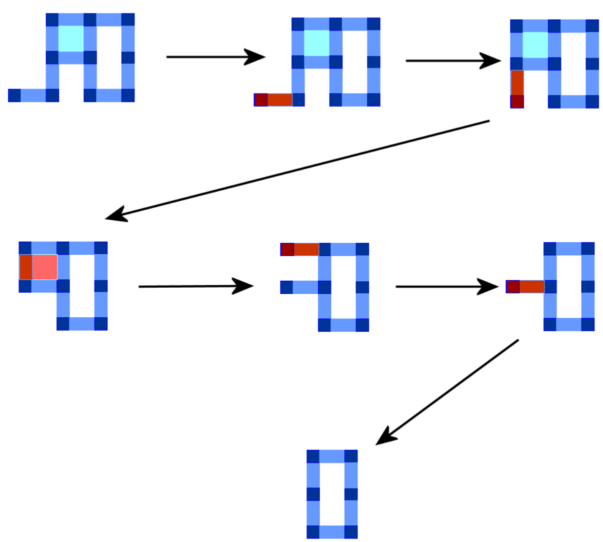
- If X is cubical and $A \subset X$ is acyclic then

$$H_n(X) \cong \begin{cases} H_n(X, A), & \text{for } n \geq 1 \\ \mathbb{Z} \oplus H_n(X, A), & \text{for } n = 0 \end{cases}$$

- breadth-first search construction

Free face reductions 25

foreach σ **do**
  if $\mathrm{cbd}(\sigma) = \{\tau\}$ **then**
    remove(σ);
    remove(τ);
  endif;
endfor;



- free face - a generator with exactly one generator in coboundary
- a combinatorial counterpart of deformation retraction
- on algebraic level:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \end{bmatrix}$$

Dual reductions?

$$\begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \cdots \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \cdots \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \cdots \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots \\
. & . & . & . & . & . & . & . & \cdots \\
. & . & . & . & . & . & . & . & \cdots \\
. & . & . & . & . & . & . & . & \cdots
\end{bmatrix}$$

Dual reductions?

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \end{bmatrix}$$

- free coface - a generator with exactly one generator in boundary

Dual reductions?

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \\ . & . & . & . & . & . & . & . & \dots \end{bmatrix}$$

- free coface - a generator with exactly one generator in boundary
- one space homology theory with compact supports for locally compact sets (Steenrod 1940, Massey 1978)

Dual reductions?

$$\begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & \dots \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \dots \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \dots \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots \\
. & . & . & . & . & . & . & . & \dots \\
. & . & . & . & . & . & . & . & \dots \\
. & . & . & . & . & . & . & . & \dots
\end{bmatrix}$$

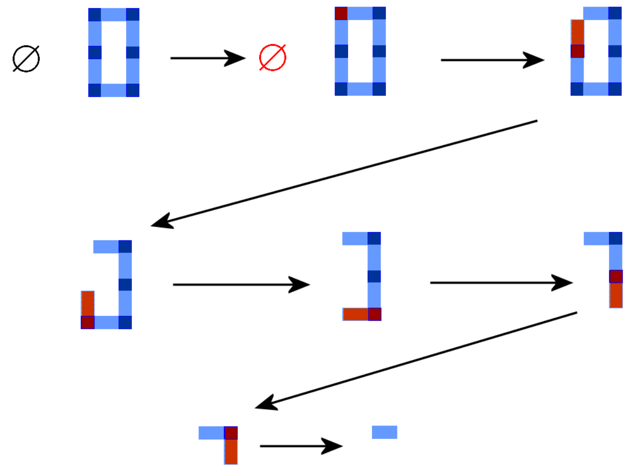
- free coface - a generator with exactly one generator in boundary
- one space homology theory with compact supports for locally compact sets (Steenrod 1940, Massey 1978)
- combinatorial version (MM, B. Batko, 2006)

Coreduction algorithm

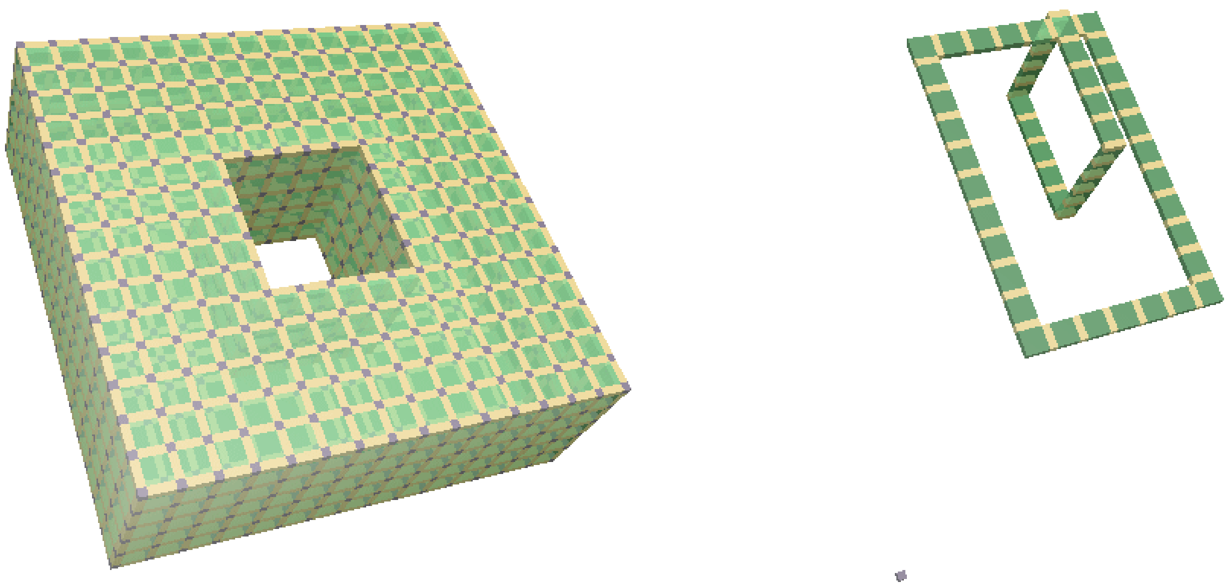

```
Q := empty queue;
enqueue(Q,s);
while Q ≠ ∅ do
  s:=dequeue(Q);
  if bd_S s = {t} then
    remove(s);
    remove(t);
    enqueue(Q, cbd_K t);
  else if bd_S s = ∅ then
    enqueue(Q, cbd_K s);
  endif;
endwhile ;
```

Coreduction algorithm

Q := empty queue;
enqueue(Q,s);
while $Q \neq \emptyset$ **do**
  s:=dequeue(Q);
  if $\mathrm{bd}_S\, s = \{t\}$ **then**
    remove(s);
    remove(t);
    enqueue($Q, \mathrm{cbd}_{\mathcal{K}}\, t$);
  else if $\mathrm{bd}_S\, s = \emptyset$ **then**
    enqueue($Q, \mathrm{cbd}_{\mathcal{K}}\, s$);
  endif;
endwhile ;



Coreduction algorithm



Coreductions for S-complexes

- S-complex - a free chain complex with a fixed basis S which allows computation of incidence coefficients $\kappa(s,t)$ directly from the coding of the basis

Coreductions for S-complexes

- S-complex - a free chain complex with a fixed basis S which allows computation of incidence coefficients $\kappa(s,t)$ directly from the coding of the basis
- Examples: cubical complexes, simplicial complexes

Coreductions for S-complexes

- S-complex - a free chain complex with a fixed basis S which allows computation of incidence coefficients $\kappa(s,t)$ directly from the coding of the basis
- Examples: cubical complexes, simplicial complexes
- Rectangular CW-complexes (P. Dłotko, T. Kaczynski, MM, T. Wanner, 2010)

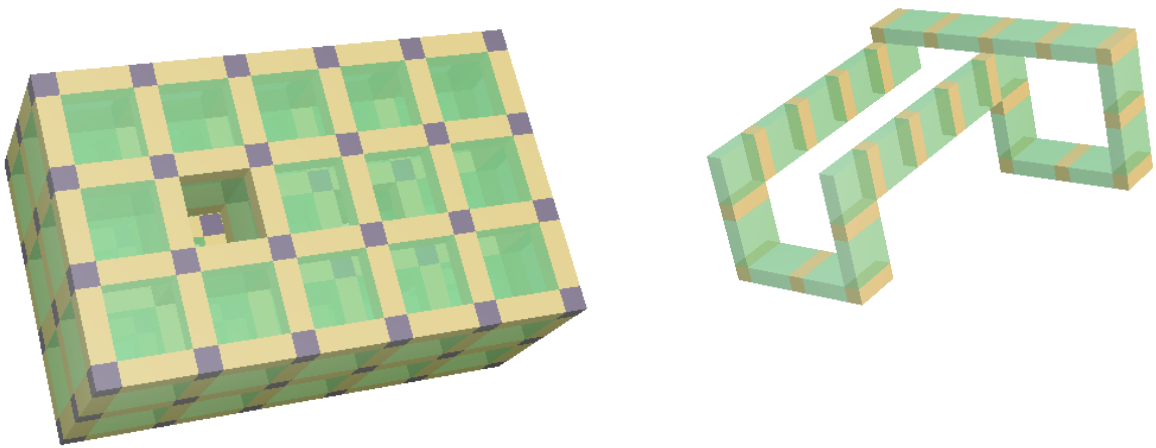
Augmentible S-complexes

Definition. An S-complex is augmentible iff there exists $\epsilon : S_0 \to R$ (augmentation) such that

- $\epsilon(t) \neq 0$ for $t \in S_0$
- $\sum_t \kappa(s, t)\epsilon(t) = 0$ for $s \in S_1$

Coreductions may be applied to any augmentible S-complexes.

Coreduction algorithm



Unlike torus, coreductions of Bing's House result in a non-augmentible S-complex.

- http://redhom.ii.uj.edu.pl

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

- AS, CR, DMT algorithms

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

- AS, CR, DMT algorithms
- Betti and torsion numbers, homology generators, homology maps, persistence intervals

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

- AS, CR, DMT algorithms
- Betti and torsion numbers, homology generators, homology maps, persistence intervals
- $\mathbb{Z}$ and $\mathbb{Z}_p$ coefficients

CAPD::RedHom

- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

- AS, CR, DMT algorithms
- Betti and torsion numbers, homology generators, homology maps, persistence intervals
- $\mathbb{Z}$ and $\mathbb{Z}_p$ coefficients
- generic but efficient: for cubical sets, simplicial sets, cubical CW complexes, ...

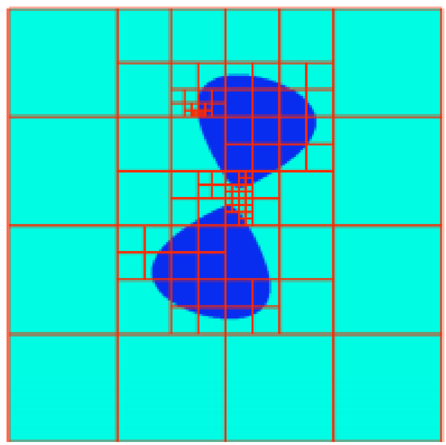
CAPD::RedHom

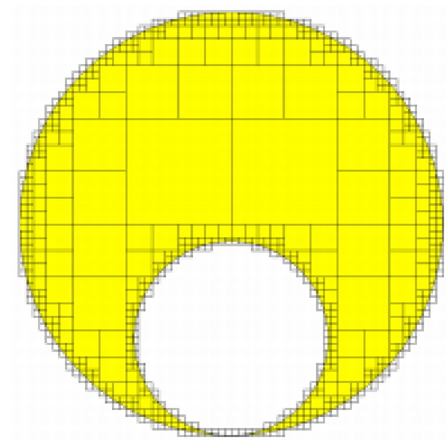
- http://redhom.ii.uj.edu.pl
- A subproject of CAPD (http://capd.ii.uj.edu.pl)
- A sister project of CHomP (http://chomp.rutgers.edu)

Generic homology software based on geometric reductions

- AS, CR, DMT algorithms
- Betti and torsion numbers, homology generators, homology maps, persistence intervals
- $\mathbb{Z}$ and $\mathbb{Z}_p$ coefficients
- generic but efficient: for cubical sets, simplicial sets, cubical CW complexes, ...
- written in C++, based on C++ templates and generic programming
- Authors: P. Dłotko, M. Juda, A. Krajniak, MM, H. Wagner, ...

Rectangular CW-complexes





Rectangular CW-complexes

Theorem. (P. Dłotko, T. Kaczynski, MM, T. Wanner, 2010) Consider a rectangular CW-complex given by a rectangular structure $\mathcal{Q}$. Let P and Q denote two arbitrary rectangles in $\mathcal{Q}$ with $\dim Q = 1 + \dim P$, and define the number α_{QP} as follows. For $d = 1$ and $Q = [a, b]$ let

$$\alpha_{QP} := \begin{cases} -1 & \text{if} \quad P = [a] \ , \\ 1 & \text{if} \quad P = [b] \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

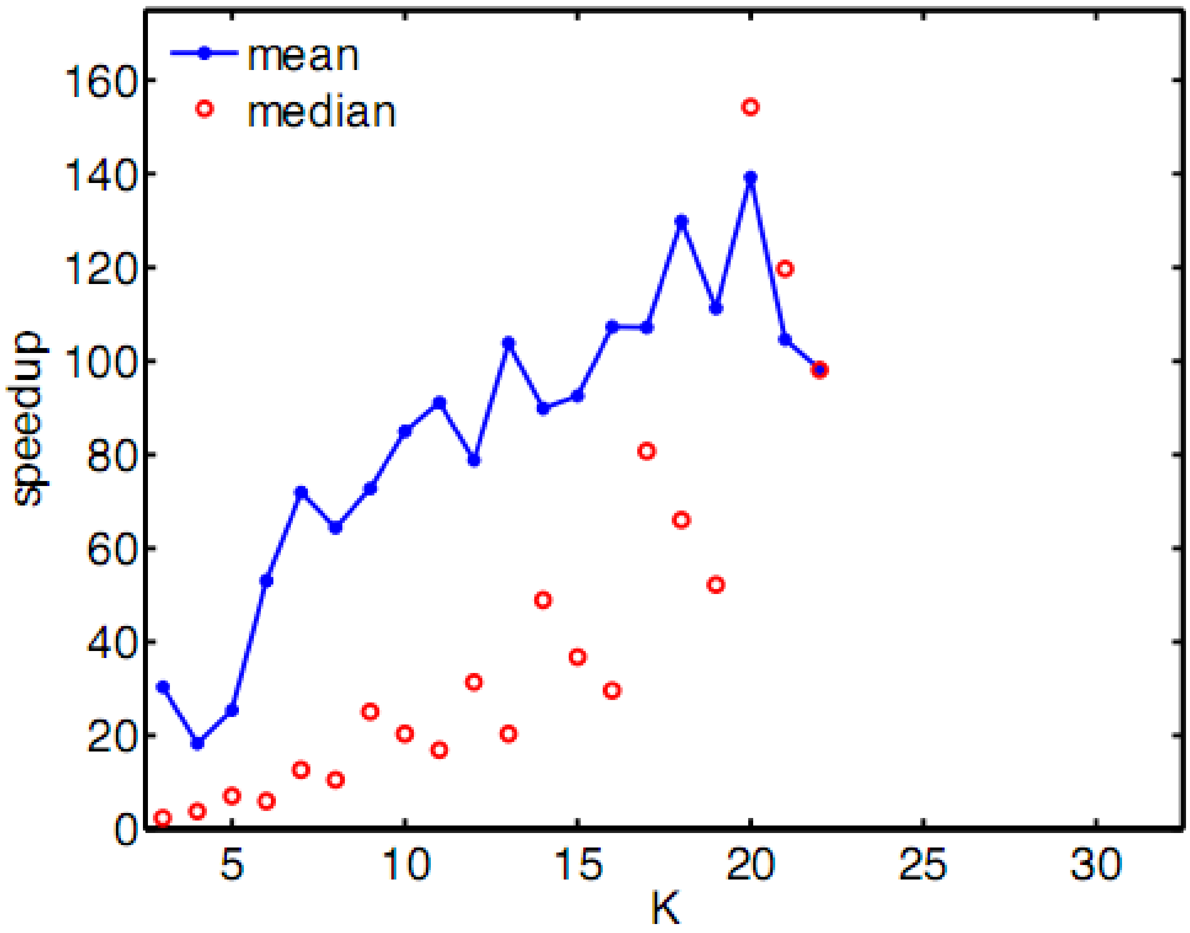
and for $d > 1$ set

(1)

$$\alpha_{QP} := \begin{cases} (-1)^{\sum_{i=1}^{j-1} \dim Q_i} \, \alpha_{Q_j P_j} & \text{if } P < Q \text{ and } P_j < Q_j \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Then the numbers α_{QP} are incidence numbers for the given rectangular CW-complex.

Rectangular CW complex versus cubical approach



Numerical examples - manifolds

	$T \times S^1$	$(S^1)^3$	$S^1 \times K$	$T \times T$	$P \times K$
dim	5	6	6	6	8
size in millions	0.07	0.10	0.40	2.36	32.05
H_0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
H_1	$\mathbb{Z}^3$	0	$\mathbb{Z}^2 + \mathbb{Z}_2$	$\mathbb{Z}^4$	$\mathbb{Z} + \mathbb{Z}_2^2$
H_2	$\mathbb{Z}^3$	0	$\mathbb{Z} + \mathbb{Z}_2$	$\mathbb{Z}^6$	$\mathbb{Z}_2^2$
H_3	$\mathbb{Z}$	$\mathbb{Z}$		$\mathbb{Z}^4$	$\mathbb{Z}_2$
H_4				$\mathbb{Z}$	
Linbox::Smith	130	350	> 600	> 600	-
CHomP::homcubes	1.3	1.7	10	56	17370
RedHom::CR	0.03	0.04	0.26	2.5	34
RedHom::CR+DMT	0.02	0.08	0.5	1.1	-

Numerical examples - Cahn-Hillard

	P0001	P0050	P0100
dim	3	3	3
size in millions	75.56	73.36	71.64
H_0	$\mathbb{Z}^7$	$\mathbb{Z}^2$	$\mathbb{Z}$
H_1	$\mathbb{Z}^{6554}$	$\mathbb{Z}^{2962}$	$\mathbb{Z}^{1057}$
H_2	$\mathbb{Z}^2$		
Linbox::Smith	> 600	> 600	> 600
CHomP::homcubes	400	360	310
RedHom::CR	18	16	15
RedHom::CR+DMT	8	7	6
RedHom::AS	10	5	3.5

Numerical examples - random sets

	d4s8f50	d4s12f50	d4s16f50	d4s20f50
dim	4	4	4	4
size in millions	0.07	0.34	1.04	2.48
H_0	$\mathbb{Z}^2$	$\mathbb{Z}^2$	$\mathbb{Z}^2$	$\mathbb{Z}^2$
H_1	$\mathbb{Z}^2$	$\mathbb{Z}^{17}$	$\mathbb{Z}^{30}$	$\mathbb{Z}^{51}$
H_2	$\mathbb{Z}^{174}$	$\mathbb{Z}^{1389}$	$\mathbb{Z}^{5510}$	$\mathbb{Z}^{15401}$
H_3	$\mathbb{Z}^2$	$\mathbb{Z}^{15}$	$\mathbb{Z}^{71}$	$\mathbb{Z}^{179}$
Linbox::Smith	120	> 600	> 600	> 600
CHomP::homcubes	1	8.3	41	170
RedHom::CR	0.08	1.4	15	140
RedHom::CR+DMT	0.03	0.16	0.58	2.9

Numerical examples - simplicial sets

	random set	S^2	S^5
dim	4	2	5
size in millions	4.8	1.9	4.3
H_0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$
H_1	$\mathbb{Z}^{39}$	0	0
H_2	$\mathbb{Z}^{84}$	$\mathbb{Z}$	0
H_3			0
H_4			$\mathbb{Z}$
CHomP::homcubes	830	310	2100
RedHom::CR+DMT	65	11	100

Theorem. Assume S' is an S-complex resulting from removing an coreduction pair (a,b) in an S-omplex S. Then the chain maps $\psi^{(a,b)} : R(S) \to R(S')$ and $\iota^{(a,b)} : R(S') \to R(S)$ given by

$$\psi_k^{(a,b)}(c) \;:=\; \begin{cases} c - \frac{\langle c,a\rangle}{\langle \partial b,a\rangle}\partial b & \text{for } k = \dim b - 1 \;, \\ c - \langle c,b\rangle b & \text{for } k = \dim b \;, \\ c & \text{otherwise} \;, \end{cases}$$

and

$$\iota_k^{(a,b)}(c) \;:=\; \begin{cases} c - \frac{\langle \partial c,a\rangle}{\langle \partial b,a\rangle} b & \text{for } k = \dim b \;, \\ c & \text{otherwise} \;, \end{cases}$$

are mutually inverse chain equivalences.

Theorem. Assume S' is an S-complex resulting from removing an coreduction pair (a, b) in an S-omplex S. Then the chain maps $\psi^{(a,b)} : R(S) \to R(S')$ and $\iota^{(a,b)} : R(S') \to R(S)$ given by

$$\psi_k^{(a,b)}(c) := \begin{cases} c - \frac{\langle c,a\rangle}{\langle \partial b,a\rangle}\partial b & \text{for } k = \dim b - 1 \ , \\ c - \langle c, b\rangle b & \text{for } k = \dim b \ , \\ c & \text{otherwise} \ , \end{cases}$$

and

$$\iota_k^{(a,b)}(c) := \begin{cases} c - \frac{\langle \partial c,a\rangle}{\langle \partial b,a\rangle} b & \text{for } k = \dim b \ , \\ c & \text{otherwise} \ , \end{cases}$$

are mutually inverse chain equivalences.

Homology model

- the reduced S-complex S^f — the homology model of S as a convenient model to solve the problems of decomposing homology classes on generators.

Homology model

- the reduced S-complex S^f — the homology model of S as a convenient model to solve the problems of decomposing homology classes on generators.
- $\pi^f : R(S) \to R(S^f)$ and $\iota^f : R(S^f) \to R(S)$ mutually inverse chain equivalences obtained by composing the maps $\pi^{(a,b)}$ and $\iota^{(a,b)}$.

Homology model

- the reduced S-complex S^f — the homology model of S as a convenient model to solve the problems of decomposing homology classes on generators.
- $\pi^f : R(S) \to R(S^f)$ and $\iota^f : R(S^f) \to R(S)$ mutually inverse chain equivalences obtained by composing the maps $\pi^{(a,b)}$ and $\iota^{(a,b)}$.
- used to transport homology classes between $H_*(S)$ and $H_*(S^f)$

Homology model

- the reduced S-complex S^f — the homology model of S as a convenient model to solve the problems of decomposing homology classes on generators.
- $\pi^f : R(S) \to R(S^f)$ and $\iota^f : R(S^f) \to R(S)$ mutually inverse chain equivalences obtained by composing the maps $\pi^{(a,b)}$ and $\iota^{(a,b)}$.
- used to transport homology classes between $H_*(S)$ and $H_*(S^f)$

The cost of transporting one generator through π^f or ι^f in general may be quadratic.

Homology model

- the reduced S-complex S^f — the homology model of S as a convenient model to solve the problems of decomposing homology classes on generators.
- $\pi^f : R(S) \to R(S^f)$ and $\iota^f : R(S^f) \to R(S)$ mutually inverse chain equivalences obtained by composing the maps $\pi^{(a,b)}$ and $\iota^{(a,b)}$.
- used to transport homology classes between $H_*(S)$ and $H_*(S^f)$

The cost of transporting one generator through π^f or ι^f in general may be quadratic.

In the case of Free Face Reduction Algorithm and Free Coface Reduction Algorithm it is linear!

Homology generators

- $S_q = \{ s^q_1, s^q_2, \ldots s^q_{r_q} \}$.

Homology generators

- $S_q = \{\, s^q_1, s^q_2, \ldots s^q_{r_q} \,\}$.
- $\{\, [u_1], [u_2], \ldots [u_n] \,\}$ – generators of the homology group $H_q(S)$.

Homology generators

- $S_q = \{\, s_1^q, s_2^q, \ldots s_{r_q}^q \,\}$.
- $\{\, [u_1], [u_2], \ldots [u_n] \,\}$ – generators of the homology group $H_q(S)$.
- Task: decompose $[z] \in H_q(X)$ on homology generators

$$[z] = \sum_{i=1}^{n} x_i [u_i]$$

Homology generators

- $S_q = \{\, s_1^q, s_2^q, \ldots s_{r_q}^q \,\}$.
- $\{\, [u_1], [u_2], \ldots [u_n] \,\}$ – generators of the homology group $H_q(S)$.
- Task: decompose $[z] \in H_q(X)$ on homology generators

$$[z] = \sum_{i=1}^{n} x_i [u_i]$$

- Linear algebra problem

$$z = \sum_{i=1}^{n} x_i u_i + \partial c$$

with unknown variables $x_1, x_2, \ldots, x_n \in \mathbb{Z}$ and $c \in R_{q+1}(S)$.

Homology generators

$$z = \sum_{j=1}^{r_q} z_j s_j^q, \quad u_i = \sum_{j=1}^{r_q} u_{ij} s_j^q, \quad c = \sum_{k=1}^{r_{q+1}} y_k s_k^{q+1}$$

$$\partial s_k^{q+1} = \sum_{j=1}^{r_q} a_{kj} s_j^q$$

$$\partial c = \sum_{j=1}^{r_q} \left(\sum_{k=1}^{r_{q+1}} a_{kj} y_k \right) s_j^q$$

Homology generators

$$z = \sum_{j=1}^{r_q} z_j s_j^q, \quad u_i = \sum_{j=1}^{r_q} u_{ij} s_j^q, \quad c = \sum_{k=1}^{r_{q+1}} y_k s_k^{q+1}$$

$$\partial s_k^{q+1} = \sum_{j=1}^{r_q} a_{kj} s_j^q$$

$$\partial c = \sum_{j=1}^{r_q} \left(\sum_{k=1}^{r_{q+1}} a_{kj} y_k \right) s_j^q$$

Thus, we get a system of r_q linear equations with $n + r_{q+1}$ unknowns

$$z_j = \sum_{i=1}^{n} u_{ij} x_i + \sum_{k=1}^{r_{q+1}} a_{kj} y_k \text{ for } j = 1, 2, \ldots r_q.$$

- In case of large S the cost is huge!
- **Solution**: transport the problem via π^f to homology model and solve it there

Homology of cubical maps

- X, Y – cubical complexes

Homology of cubical maps

- X, Y – cubical complexes
- $g : X \to Y$ is cubical if maps elementary cubes to elementary cubes.
- g induces chain map $g_{\#} : R(X) \to R(Y)$

Homology of cubical maps

- X, Y – cubical complexes
- $g : X \to Y$ is cubical if maps elementary cubes to elementary cubes.
- g induces chain map $g_{\#} : R(X) \to R(Y)$
- examples of interest: inclusions and projections

Homology of cubical maps

- X, Y – cubical complexes
- $g : X \to Y$ is cubical if maps elementary cubes to elementary cubes.
- g induces chain map $g_\# : R(X) \to R(Y)$
- examples of interest: inclusions and projections
- $U := \{ [u_1], [u_2], \ldots [u_m] \}$ and $W := \{ [w_1], [w_2], \ldots [w_n] \}$ — bases of $H_q(X)$ and $H_q(Y)$

Homology of cubical maps

- X, Y – cubical complexes
- $g : X \to Y$ is cubical if maps elementary cubes to elementary cubes.
- g induces chain map $g_\# : R(X) \to R(Y)$
- examples of interest: inclusions and projections
- $U := \{ [u_1], [u_2], \ldots [u_m] \}$ and $W := \{ [w_1], [w_2], \ldots [w_n] \}$ — bases of $H_q(X)$ and $H_q(Y)$
- To find the matrix of g_* ecompose $g_\#(u_i)$ on generators in W

Homology of cubical maps

- X, Y – cubical complexes
- $g : X \to Y$ is cubical if maps elementary cubes to elementary cubes.
- g induces chain map $g_\# : R(X) \to R(Y)$
- examples of interest: inclusions and projections
- $U := \{ [u_1], [u_2], \ldots [u_m] \}$ and $W := \{ [w_1], [w_2], \ldots [w_n] \}$ — bases of $H_q(X)$ and $H_q(Y)$
- To find the matrix of g_* ecompose $g_\#(u_i)$ on generators in W
- Using the diagram

$$
\begin{tikzcd}
R(X) \arrow[r, "g_\#"] & R(Y) \arrow[d, shift right, "\pi^f"'] \\
 & R(Y^f) \arrow[u, shift right, "\iota^f"']
\end{tikzcd}
$$

we can solve the problem in the homology model Y^f, where it is much simpler.

Computing Homology of Maps

In principle, computing homology of a map $f : X \to Y$ is a three step procedure:

(1) Find a finite representation of f
(2) Use it to build the chain map
(3) Compute the map in homology from the chain map

Computing Homology of Maps

In principle, computing homology of a map $f : X \to Y$ is a three step procedure:

(1) Find a finite representation of f
(2) Use it to build the chain map
(3) Compute the map in homology from the chain map

Combinatorial representations

Let $X = |\mathcal{X}|$ be a full cubical set with representation $\mathcal{X}$ and let $f : X \to X$ be a continuous map. We say that $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$ is a representation of f if

$$f(Q) \subset \operatorname{int} |\mathcal{F}(Q)|.$$

Combinatorial representations

Let $X = |\mathcal{X}|$ be a full cubical set with representation $\mathcal{X}$ and let $f : X \to X$ be a continuous map. We say that $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$ is a representation of f if

$$f(Q) \subset \operatorname{int} |\mathcal{F}(Q)|.$$

and for each $x \in X$ the set

$$\lceil \mathcal{F} \rceil (x) := \bigcup \{ |\mathcal{F}(Q)| \mid x \in Q \in \mathcal{X} \}$$

is acyclic.

Let $X = |\mathcal{X}|$ be a full cubical set with representation $\mathcal{X}$ and let $f : X \to X$ be a continuous map. We say that $\mathcal{F} : \mathcal{X} \overset{\to}{\to} \mathcal{X}$ is a representation of f if

$$f(Q) \subset \operatorname{int} |\mathcal{F}(Q)|.$$

and for each $x \in X$ the set

$$\lceil \mathcal{F} \rceil(x) := \bigcup \{|\mathcal{F}(Q)| \mid x \in Q \in \mathcal{X}\}$$

is acyclic.

A representation for a given $\mathcal{X}$ may not exist. However, under a sufficiently good subdivision, the represntation always exists.

Let $X = |\mathcal{X}|$ be a full cubical set with representation $\mathcal{X}$ and let $f : X \to X$ be a continuous map. We say that $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$ is a representation of f if

$$f(Q) \subset \operatorname{int} |\mathcal{F}(Q)|.$$

and for each $x \in X$ the set

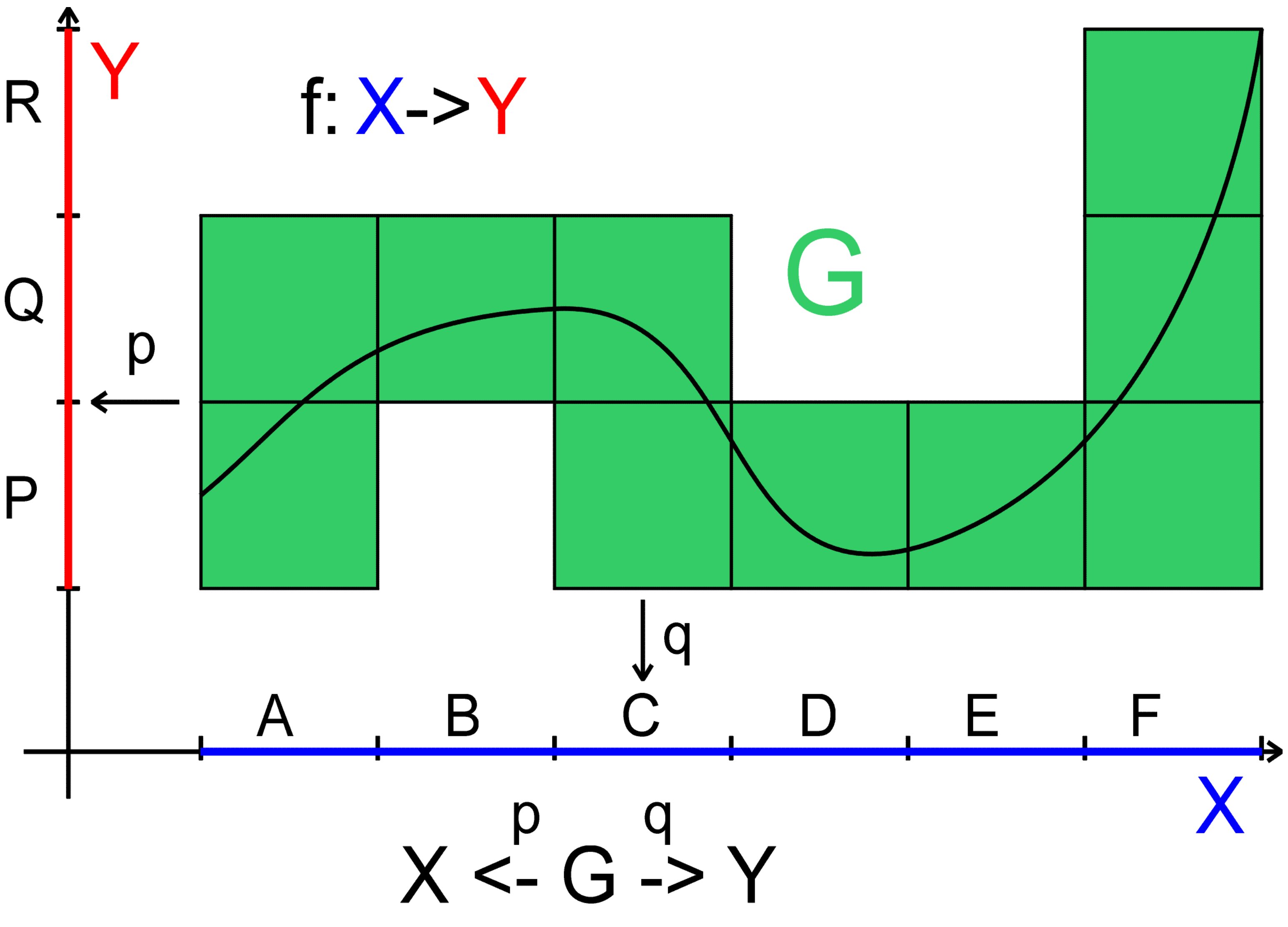
$$\lceil \mathcal{F} \rceil(x) := \bigcup \{ |\mathcal{F}(Q)| \mid x \in Q \in \mathcal{X} \}$$

is acyclic.

A representation for a given $\mathcal{X}$ may not exist. However, under a sufficiently good subdivision, the represntation always exists.

Theorem. (Allili, Kaczynski, 2000, Kaczynski, Mischaikow, MM 2004) If $\mathcal{F}$ is a representation of f, then there is an algorithm which transforms $\mathcal{F}$ into a chain map whose homology is the homology of f.

Graph approach (Granas and L. Górniewicz, 1981)



The homology map algorithm (Mischaikow, MM, Pilarczyk 2005)

(1) Construct a representation $\mathcal{F}$ of $f : X \to Y$.
(2) Construct the graph G of $F := \lceil \mathcal{F} \rceil$.
(3) If the homologies of the values of F are not trivial, refine the grid and go to 1.
(4) Apply shaving to X, Y and G in such a way that the shaved G' is the graph of an acyclic mv map $F' : X' \to Y'$
(5) Find the homologies of the projections $p : G \to X$ and $q : G \to Y$.
(6) Return $H_*(q)H_*(p)^{-1}$

The homology map algorithm (Mischaikow, MM, Pilarczyk 2005)

(1) Construct a representation $\mathcal{F}$ of $f : X \to Y$.
(2) Construct the graph G of $F := \lceil \mathcal{F} \rceil$.
(3) If the homologies of the values of F are not trivial, refine the grid and go to 1.
(4) Apply shaving to X, Y and G in such a way that the shaved G' is the graph of an acyclic mv map $F' : X' \to Y'$
(5) Find the homologies of the projections $p : G \to X$ and $q : G \to Y$.
(6) Return $H_*(q)H_*(p)^{-1}$

- Pilarczyk (2005) — implementation
- satisfactorily fast for a class of practical problems
- remains computationally most expensive part in applications in dynamics
- preserving the acyclicity of values when applying reductions is computationally expensive

Coreduction model approach, MM 2010

- Using coreductions construct homology models of X, Y, and G
- Using homology models find the homology of the projections $p : G \to X$ and $q : G \to Y$
- Compute the inverse of p_* and return $q_* p_*^{-1}$

> - No need to preserve the acyclicity under reductions
> - significantly faster than the previous graph approach

Graph approach versus coreduction model approach

Set	Emb Dim	Size $\times 10^6$	CHomP homcubes	RedHom CR	speedup
z2torus8.map	6	3.65	10.3	1.5	7
z2torus12.map	6	7.75	25.6	3.2	8
z2torus16.map	6	14.13	54.5	6.8	8
z2torus19.map	6	23.29	121.0	11.7	10

Alternative based on Čech structures (outline)

- choose a Čech structure $\mathcal{X}$ on X
- for $Q \in \mathcal{X}$ take $\mathcal{F}(Q)$ as a convex enclosure of $f(Q)$ obtained via rigorous numerics
- $\mathcal{F} : K(\mathcal{X}) \to K(\mathcal{X} \cup \mathcal{F}(\mathcal{X}))$ acts as a simplicial map
- the homology of f is computed when $K(\mathcal{X}) \subset K(\mathcal{X} \cup \mathcal{F}(\mathcal{X}))$ induces an isomorphism
- this is guaranteed when the enclosure is good enough

Alternative based on Čech structures (outline)

- choose a Čech structure $\mathcal{X}$ on X
- for $Q \in \mathcal{X}$ take $\mathcal{F}(Q)$ as a convex enclosure of $f(Q)$ obtained via rigorous numerics
- $\mathcal{F} : K(\mathcal{X}) \to K(\mathcal{X} \cup \mathcal{F}(\mathcal{X}))$ acts as a simplicial map
- the homology of f is computed when $K(\mathcal{X}) \subset K(\mathcal{X} \cup \mathcal{F}(\mathcal{X}))$ induces an isomorphism
- this is guaranteed when the enclosure is good enough

Alternative based on Čech structures

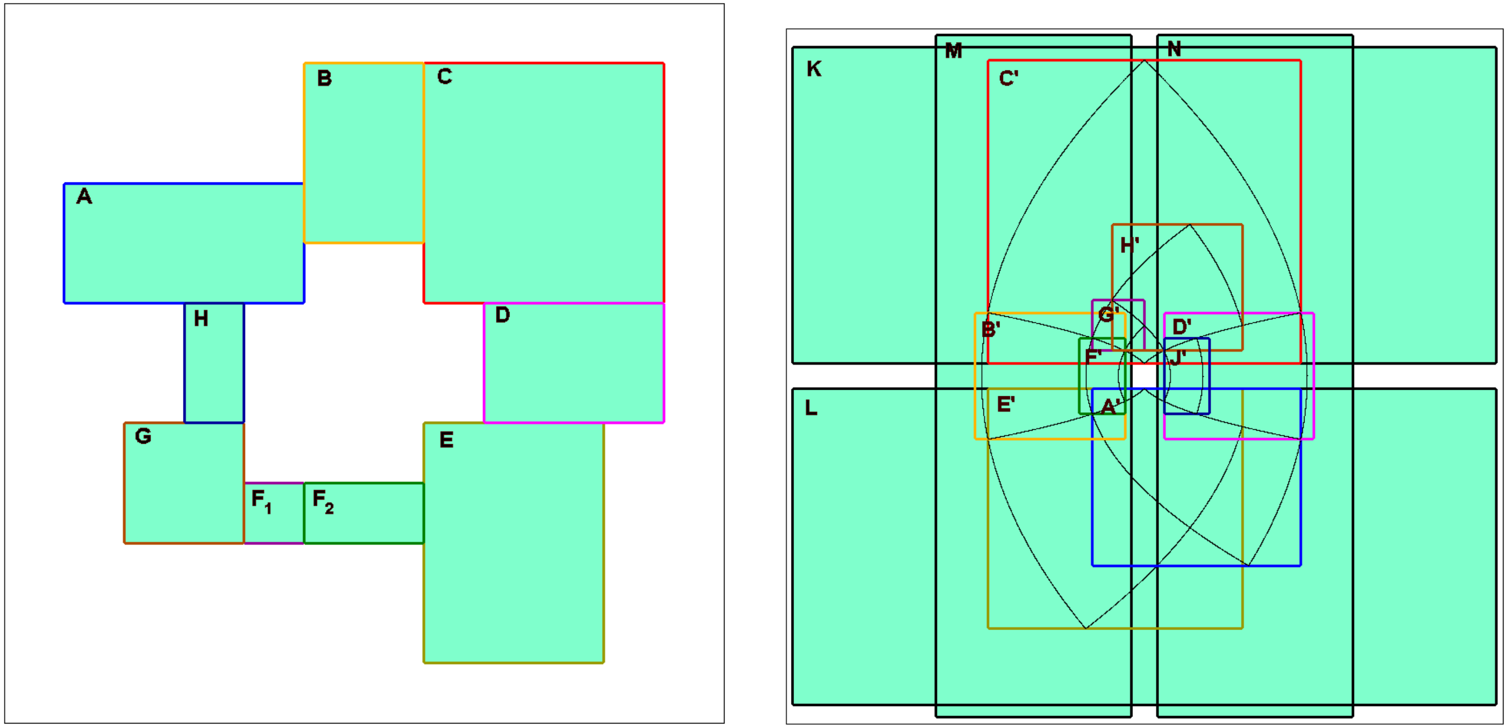
Example: consider

$$f : \mathbb{C} \ni z \to z^2 \in \mathbb{C}$$

Alternative based on Čech structures

Example: consider

$$f : \mathbb{C} \ni z \to z^2 \in \mathbb{C}$$



Cubical persistence via coreductions and inclusions

- Assume field coefficients
- Compute the maps induced in homology by inclusions
- Find the compositions and ranks of the respective matrices

$$\beta_q^{i,j} := \mathsf{rank}\ \iota_q^{i,j},$$

- Compute the number of (i,j)-persistence intervals from formula

$$\mathsf{pi}_q(i,j) = \left(\beta_q^{i,j-1} - \beta_q^{i-1,j-1}\right) - \left(\beta_q^{i,j} - \beta_q^{i-1,j}\right)\ .$$

Direct coreduction approach to cubical persistence

- levelwise coreductions

Direct coreduction approach to cubical persistence

- levelwise coreductions
- seperate queue for BFS on each level

Direct coreduction approach to cubical persistence

- levelwise coreductions
- seperate queue for BFS on each level
- selection always from the lowest level non-empty queue

Direct coreduction approach to cubical persistence

- levelwise coreductions
- seperate queue for BFS on each level
- selection always from the lowest level non-empty queue
- result: coreductions for all sublevel sets together in $O(n \log^* n)$ time

Direct coreduction approach to cubical persistence

- levelwise coreductions
- seperate queue for BFS on each level
- selection always from the lowest level non-empty queue
- result: coreductions for all sublevel sets together in $O(n \log^* n)$ time
- Complexity of finding persistence intervals on the plane is $O(n \log^* n)$

Timings

Grid	Levels	classical approach (*)	**RedHom** CR-incl.	**RedHom** CR-direct
1024×1024	17	3299.0	470.0	3.4
2048×2048	18	36187.0	8012.0	13.0
$100 \times 100 \times 100$	25	60407.0	4025.0	-

(*) - implementation of Edelsbrunner-Letscher-Zomorodian algorithm for cubical sets by V. Nanda

References

- **MM, P. Pilarczyk, N. Żelazna**, Homology algorithm based on acyclic subspace, *Computers and Mathematics with Applications* (**2008**).
- **MM, B. Batko**, Coreduction homology algorithm, *Discrete and Computational Geometry* (**2009**).
- **K. Mischaikow, MM, P. Pilarczyk**, Graph Approach to the Computation of the Homology of Continuous Maps, *Foundations of Computational Mathematics* (**2005**).
- **MM**, Čech Type Approach to Computing Homology of Maps, *Discrete and Computational Geometry* (**2010**).
- **P. Dłotko, T. Kaczynski, MM, T. Wanner**, Coreduction Homology Algorithm for Regular CW-Complexes, *Discrete and Computational Geometry* (**2011**).
- **MM, T. Wanner**, Coreduction Homology Algorithm for Inclusions and Persistence, *Computers and Mathematics with Applications* (**2010**).
- **S. Harker, K. Mischaikow, MM, V. Nanda, H. Wagner, M. Juda, P. Dłotko**, The Efficiency of a Homology Algorithm based on Discrete Morse Theory and Coreductions, *Proceedings CTIC 2010* (**2010**).
- **A. Krajniak, MM, T. Wanner**, Direct coreduction approach to persistence, **in preparation**.