Persistence decomposition algorithms

Barbara Giunti
Joint work with
W. Chachólski, A. Jin, C.Landi

TDA Journal Club - February 15, 2021

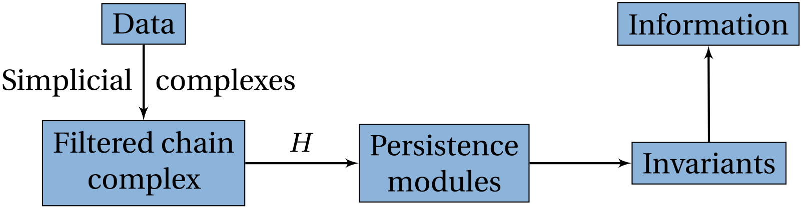
Graz University of Technology

Persistence pipeline

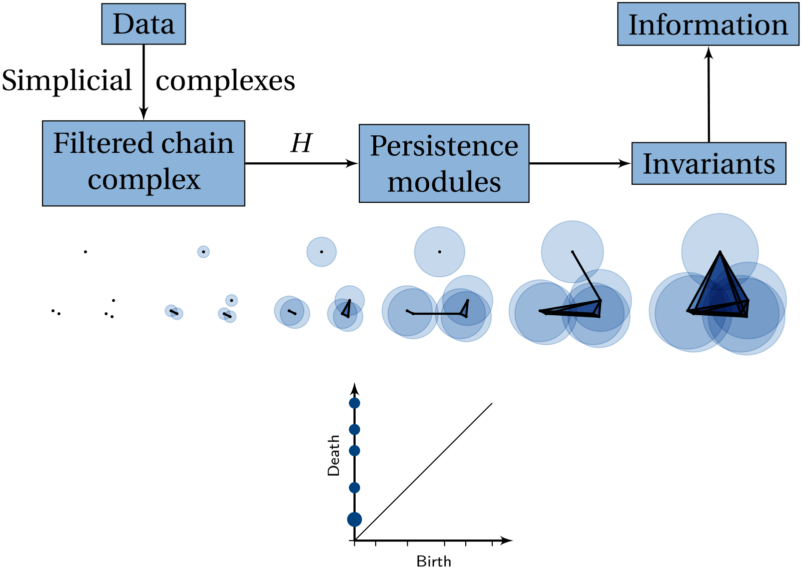
Data

Information

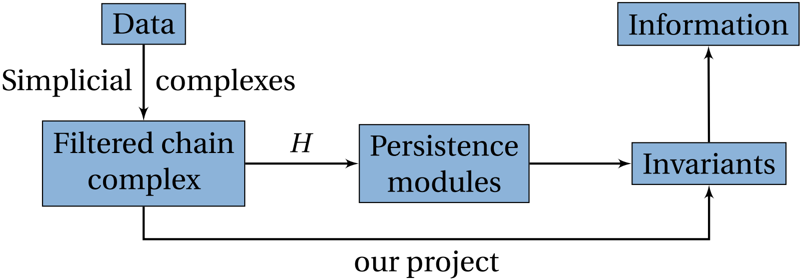
Persistence pipeline



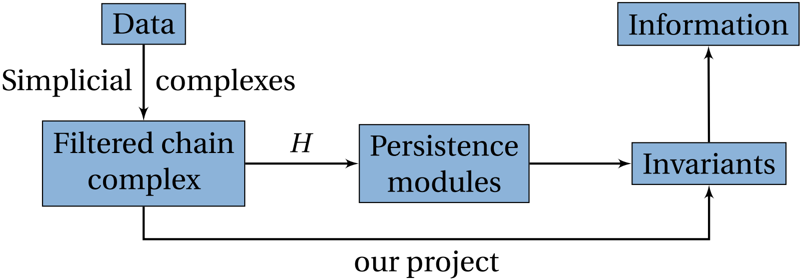
Persistence pipeline



Persistence pipeline

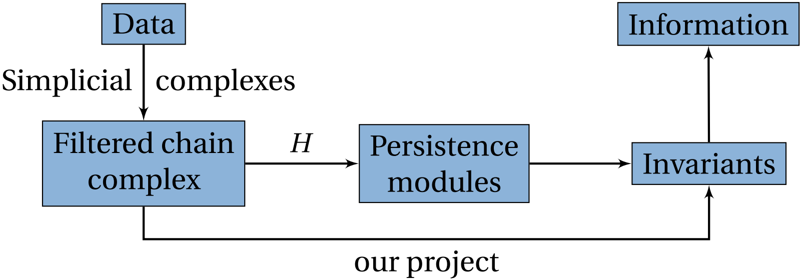


Persistence pipeline



Studying persistence at the filtered chain complex level give use insights (and possibly new information).

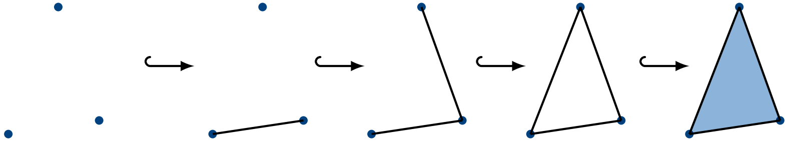
Persistence pipeline



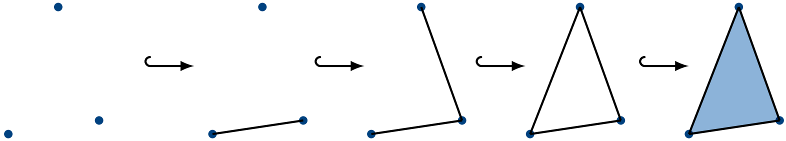
Studying persistence at the filtered chain complex level give use insights (and possibly new information).

http://www.zotero.org/groups/TDA-Applications

Filtered chain complexes



Filtered chain complexes



Definition

A **filtered chain complex** is a finite sequence of chain complexes and inclusions between them.

$$
\begin{tikzcd}
\vdots \arrow[d, "\partial_n^0"] & \vdots \arrow[d, "\partial_n^{\tau_1}"] & \cdots & \vdots \arrow[d, "\partial_n^{\tau_i}"] & \vdots \arrow[d, "\partial_n^{\tau_{i+1}}"] & \\
X_n^0 \arrow[r, hook] \arrow[d, "\partial_{n-1}^0"] & X_n^{\tau_1} \arrow[r, hook] \arrow[d, "\partial_{n-1}^{\tau_1}"] & \cdots \arrow[r, hook] & X_n^{\tau_i} \arrow[r, hook] \arrow[d, "\partial_{n-1}^{\tau_i}"] & X_n^{\tau_{i+1}} \arrow[r, hook] \arrow[d, "\partial_{n-1}^{\tau_{i+1}}"] & \cdots \\
X_{n-1}^0 \arrow[r, hook] \arrow[d, "\partial_{n-2}^0"] & X_{n-1}^{\tau_1} \arrow[r, hook] \arrow[d, "\partial_{n-2}^{\tau_1}"] & \cdots \arrow[r, hook] & X_{n-1}^{\tau_i} \arrow[r, hook] \arrow[d, "\partial_{n-2}^{\tau_i}"] & X_{n-1}^{\tau_{i+1}} \arrow[r, hook] \arrow[d, "\partial_{n-2}^{\tau_{i+1}}"] & \cdots \\
\vdots & \vdots & \cdots & \vdots & \vdots &
\end{tikzcd}
$$

Chain Complex

Parameter

Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.

Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, e)$ is the filtered chain complex given by:

$$
\begin{array}{ccccc}
t<s & & s \leq t<e & & e \leq t \\[1em]
0 & \longrightarrow & S^n & \hookrightarrow & D^{n+1}
\end{array}
$$

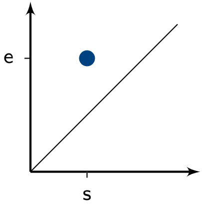
$$
\begin{array}{ccc}
0 & n+1 & \mathbb{F} \\
\big\downarrow & & \big\downarrow{\scriptstyle 1} \\
\mathbb{F} & n & \mathbb{F}
\end{array}
$$

Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, e)$ is the filtered chain complex given by:

$$\begin{array}{ccc} t<s & s\le t<e & e\le t \\ 0 & \longrightarrow \; S^n \; \hookrightarrow & D^{n+1} \end{array}$$

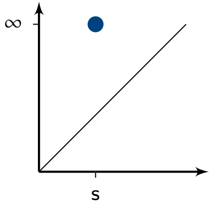
$$\begin{array}{ccc} 0 & n+1 & \mathbb{F} \\ \big\downarrow & & \big\downarrow{\scriptstyle 1} \\ \mathbb{F} & n & \mathbb{F} \end{array}$$



Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, \infty)$ is the filtered chain complex given by:

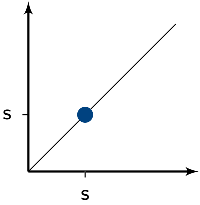
$$\begin{array}{ccc} \scriptstyle t<s & & \scriptstyle s\leq t \\ 0 & \longrightarrow & S^n \end{array}$$



Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, s)$ is the filtered chain complex given by:

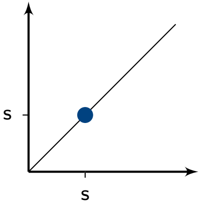
$$\begin{array}{ccc} t<s & & s\leq t \\ 0 & \longrightarrow & D^{n+1} \end{array}$$



Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, s)$ is the filtered chain complex given by:

$$\begin{array}{ccc} t<s & & s \leq t \\ 0 & \longrightarrow & D^{n+1} \end{array}$$



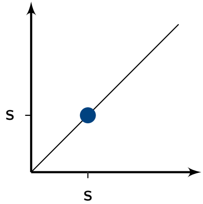
Theorem

Every filtered chain complex decomposes uniquely up to isomorphism into a finite direct sum of interval spheres.

Decomposition of filtered chain complexes

Let $n \in \mathbf{N}$, $s \in [0, \infty)$ and $e \in [s, \infty]$.
The **interval sphere** $I^n\,[s, s)$ is the filtered chain complex given by:

$$\begin{array}{ccc} t<s & & s\leq t \\ 0 & \longrightarrow & D^{n+1} \end{array}$$



Theorem

Every filtered chain complex decomposes uniquely up to isomorphism into a finite direct sum of interval spheres.

$\Longrightarrow$ applying homology, we get the persistence diagram of the associated persistence module

Generators

Let X be a filtered chain complex. There is a finite collection of interval spheres $\{I^{n_i}[s_i, s_i)\}_{i=1,\dots,m}$ and an epimorphism

$$\varphi\colon \oplus_{i=1}^{m} I^{n_i}[s_i, s_i) \to X$$

φ uniquely defines $\varphi_i\colon I^{n_i}[s_i, s_i) \to X$.

Generators

Let X be a filtered chain complex. There is a finite collection of interval spheres $\{I^{n_i}[s_i, s_i)\}_{i=1,\dots,m}$ and an epimorphism

$$\varphi\colon\ \oplus_{i=1}^{m} I^{n_i}[s_i, s_i) \to X$$

φ uniquely defines $\varphi_i\colon I^{n_i}[s_i, s_i) \to X$. Each φ_i can be identified with $x_i := (\varphi_i)_{n_i+1}^{s_i}(1) \in X_{n_i+1}^{s_i}$, with a degree and an entrance time function:

$$\deg(x_i) = n_i+1 \qquad \mathrm{ent}(x_i) = s_i$$

Generators

Let X be a filtered chain complex. There is a finite collection of interval spheres $\{I^{n_i}[s_i, s_i)\}_{i=1,\ldots,m}$ and an epimorphism

$$\varphi\colon \oplus_{i=1}^{m} I^{n_i}[s_i, s_i) \to X$$

φ uniquely defines $\varphi_i\colon I^{n_i}[s_i, s_i) \to X$. Each φ_i can be identified with $x_i := (\varphi_i)_{n_i+1}^{s_i}(1) \in X_{n_i+1}^{s_i}$, with a degree and an entrance time function:

$$\deg(x_i) = n_i{+}1 \qquad \operatorname{ent}(x_i) = s_i$$

Definition

The set $\mathcal{G} = \{x_1, \ldots, x_m\}$ is called a **set of generators of** X. If $\mathcal{G}$ is minimal degreewise, then it is called a **quasi-minimal** set of generators.

Generators

The simplices of a filtered simplicial complexes form a quasi-minimal set of generators.

Generators

The simplices of a filtered simplicial complexes form a quasi-minimal set of generators.

Consider the interval sphere $I^n\,[s,e)$. A quasi-minimal set of generators of it is $\mathcal{G} = \{x_s, x_e\}$, with $\deg(x_s) = n$, $\deg(x_e) = n{+}1$.

$$\begin{tikzcd}
\cdots & i{=}s{-}\varepsilon & i{=}s & \cdots & i{=}e{-}\varepsilon & i{=}e & \cdots & \\
\cdots \arrow[r] & 0 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & \cdots \arrow[r] & 0 \arrow[r] \arrow[d] & \mathbb{F} \arrow[r] \arrow[d] & \cdots & n{+}1 \\
\cdots \arrow[r] & 0 \arrow[r] & \mathbb{F} \arrow[r] & \cdots \arrow[r] & \mathbb{F} \arrow[r] & \mathbb{F} \arrow[r] & \cdots & n
\end{tikzcd}$$

Generators

The simplices of a filtered simplicial complexes form a quasi-minimal set of generators.

Consider the interval sphere $I^n[s,e)$. A quasi-minimal set of generators of it is $\mathcal{G} = \{x_s, x_e\}$, with $\deg(x_s) = n$, $\deg(x_e) = n{+}1$.

$$\begin{tikzcd}
\cdots & i{=}s{-}\varepsilon & i{=}s & i{=}s{+}\varepsilon & \cdots & \\
\cdots \arrow[r] & 0 \arrow[r] \arrow[d] & \mathbb{F} \arrow[r] \arrow[d] & \mathbb{F} \arrow[r] \arrow[d] & \cdots & n{+}1 \\
\cdots \arrow[r] & 0 \arrow[r] & \mathbb{F} \arrow[r] & \mathbb{F} \arrow[r] & \cdots & n
\end{tikzcd}$$

Generators

The simplices of a filtered simplicial complexes form a quasi-minimal set of generators.

Consider the interval sphere $I^n\,[s,e)$. A quasi-minimal set of generators of it is $\mathcal{G} = \{x_s, x_e\}$, with $\deg(x_s) = n$, $\deg(x_e) = n{+}1$.

Consider the interval sphere $I^n\,[s,\infty)$. A quasi-minimal set of generators of it is $\mathcal{G} = \{x_s\}$, with $\deg(x_s) = n$.

$$
\begin{tikzcd}
\cdots & i{=}s{-}\varepsilon & i{=}s & i{=}s{+}\varepsilon & \cdots & \\
\cdots \arrow[r] & 0 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & 0 \arrow[r] \arrow[d] & \cdots & n{+}1 \\
\cdots \arrow[r] & 0 \arrow[r] & \mathbb{F} \arrow[r] & \mathbb{F} \arrow[r] & \cdots & n
\end{tikzcd}
$$

Standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

Standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the j-th column of D, $low\,(j)$ denotes the row index of the lowest element of such row. If column j is zero, set $low\,(j) = 0$.

Standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the j-th column of D, $low\,(j)$ denotes the row index of the lowest element of such row. If column j is zero, set $low\,(j) = 0$.

Definition

A matrix C is called **column reduced** if $low\,(j) \neq low\,(j')$ for all non-zero columns $j \neq j'$. The element C_j^i is called a **column pivot** (of C) if $i = low\,(j)$.

Standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the j-th column of D, $low\,(j)$ denotes the row index of the lowest element of such row. If column j is zero, set $low\,(j) = 0$.

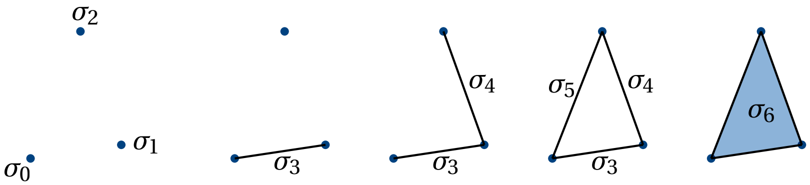
Definition

A matrix C is called **column reduced** if $low\,(j) \neq low\,(j')$ for all non-zero columns $j \neq j'$. The element C^i_j is called a **column pivot** (of C) if $i = low\,(j)$.

Theorem

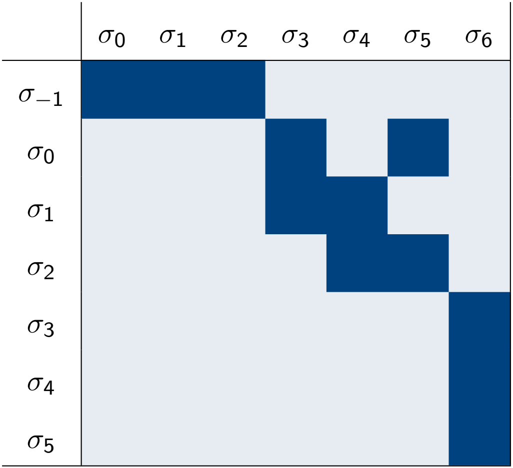
The column pivot pairing of a reduced boundary matrix provides the barcode decomposition of the associated persistent module.

Example



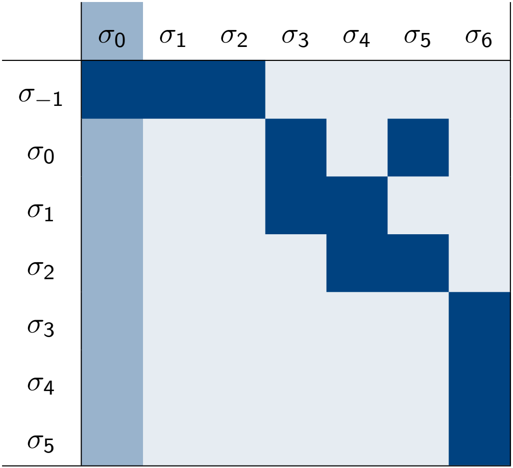
σ_2
σ_1
σ_0
σ_3
σ_4
σ_3
σ_5
σ_4
σ_3
σ_6

Example

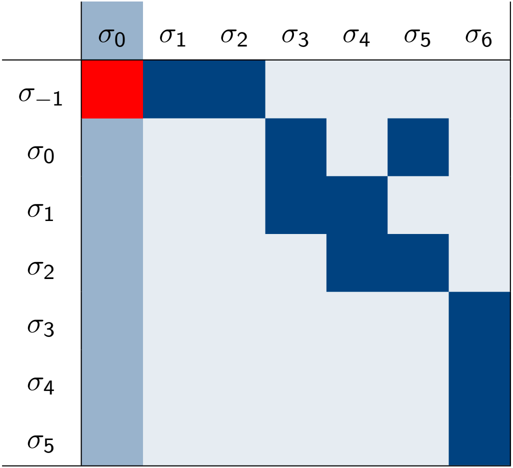


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

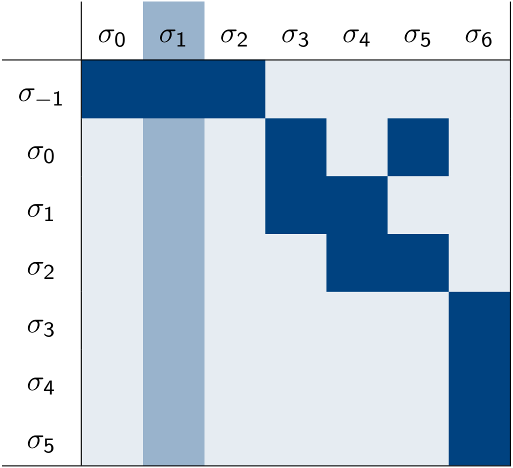
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

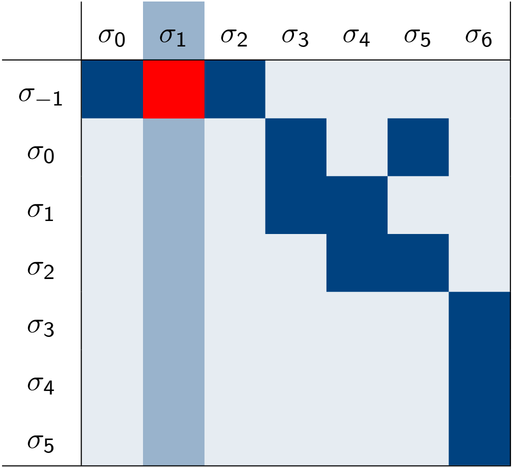

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example



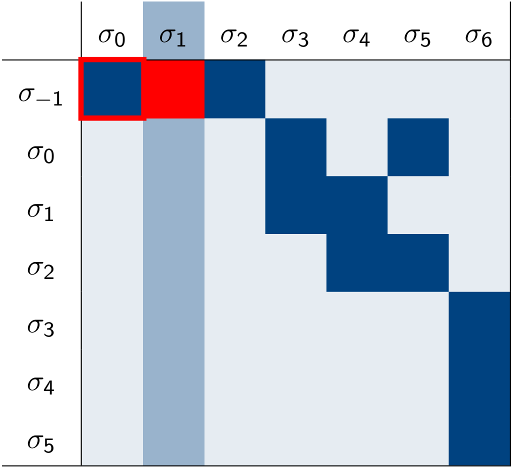
σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example



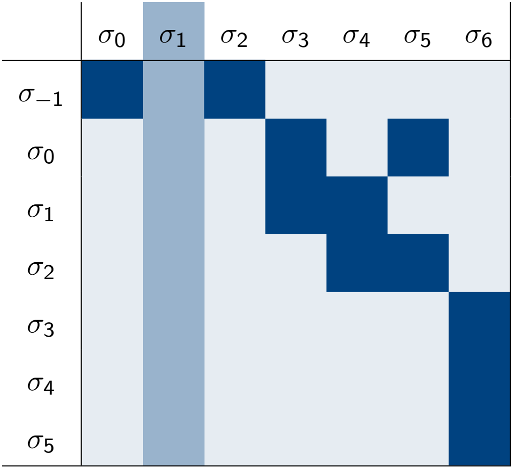
σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example



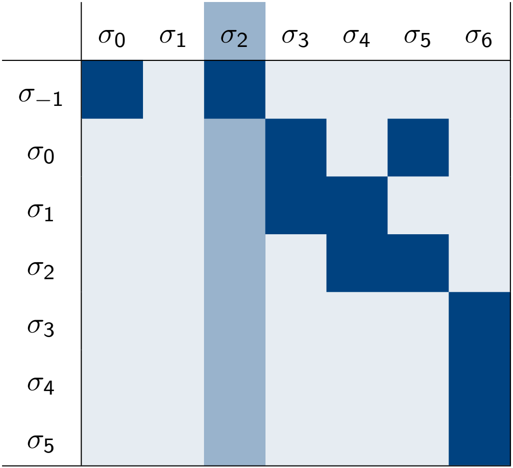
σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

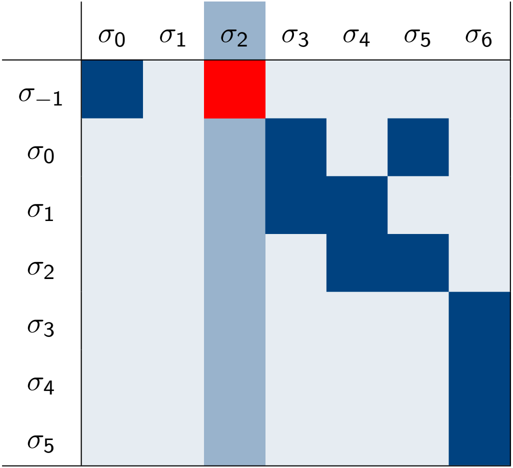


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

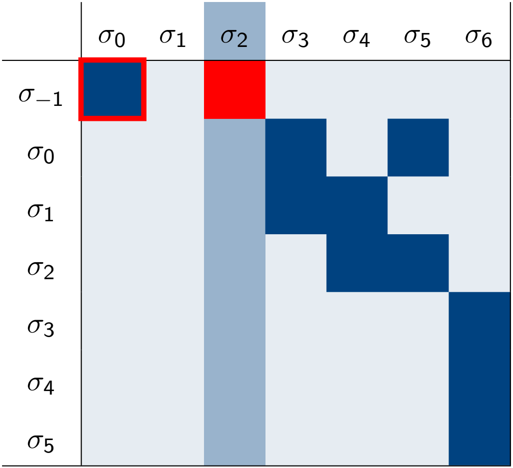

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

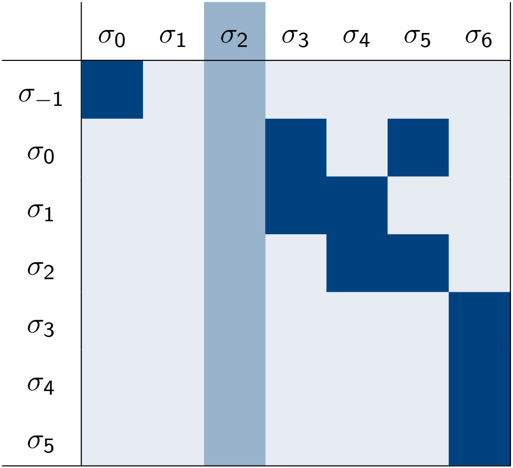


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

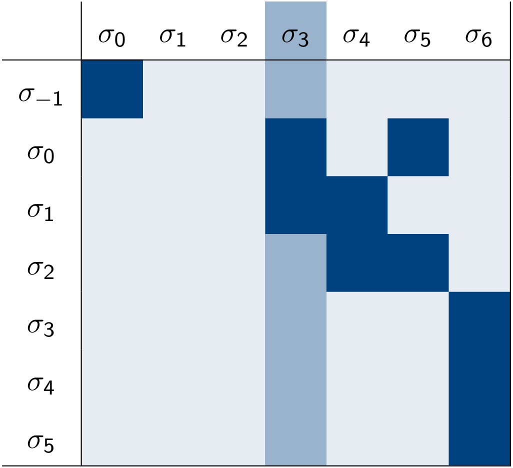
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

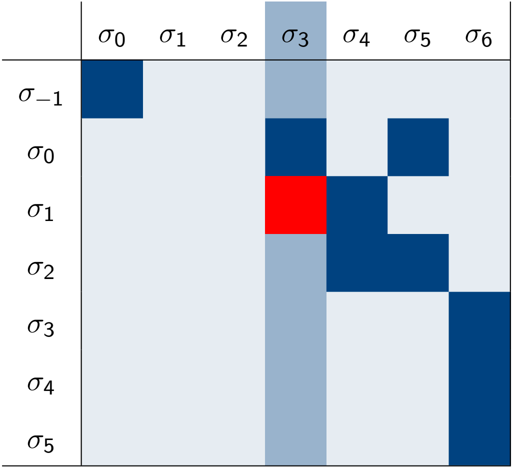
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

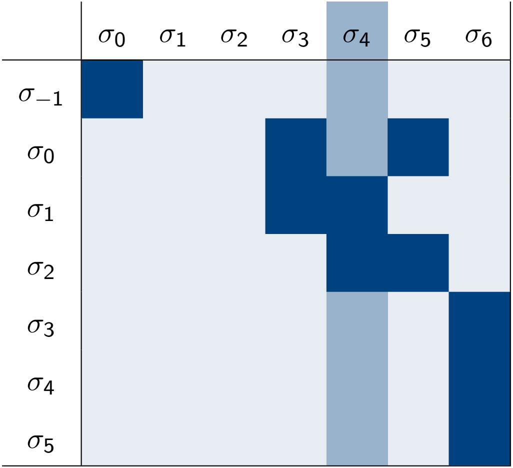

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

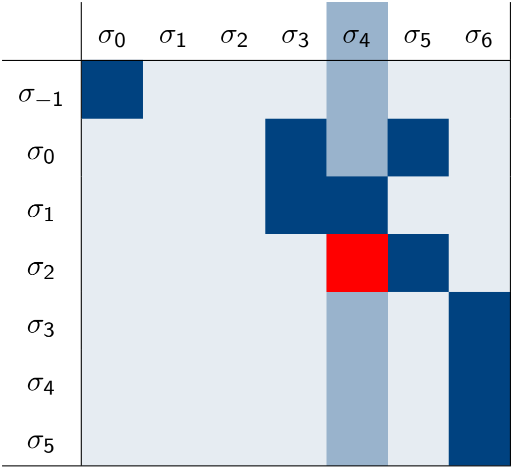


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

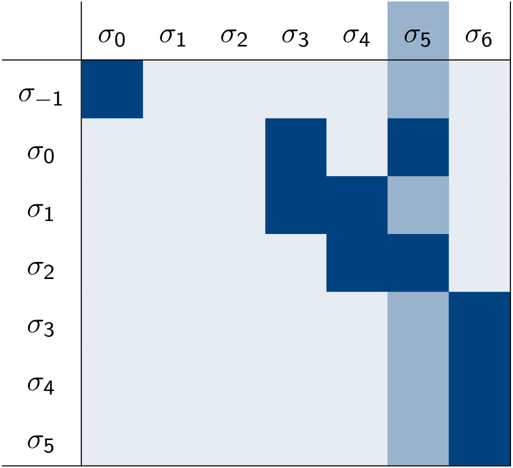

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

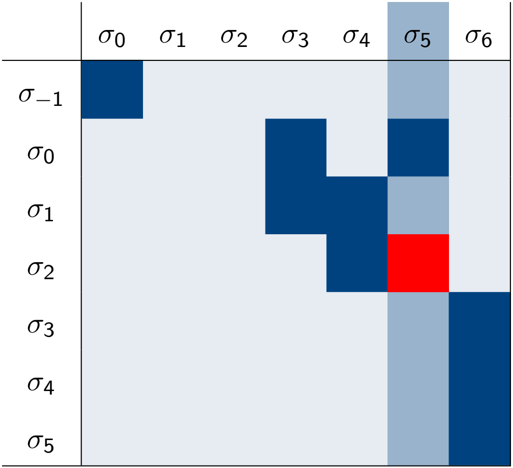


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

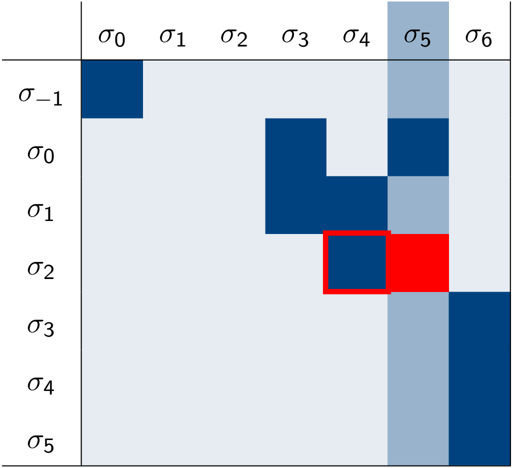
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

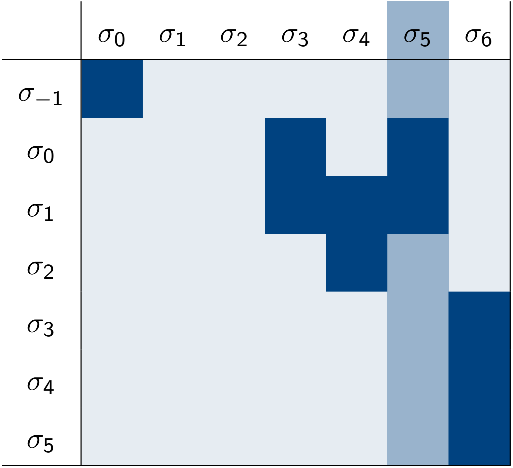
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

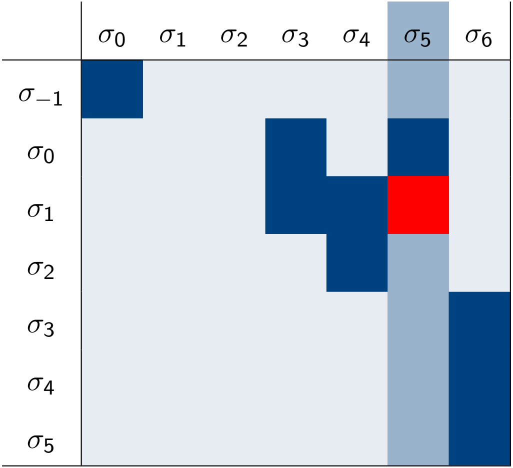
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

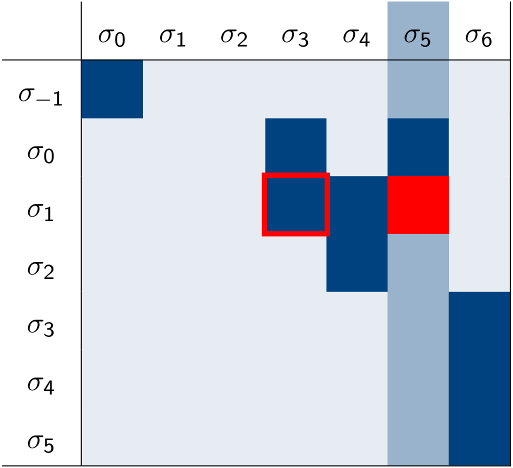
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

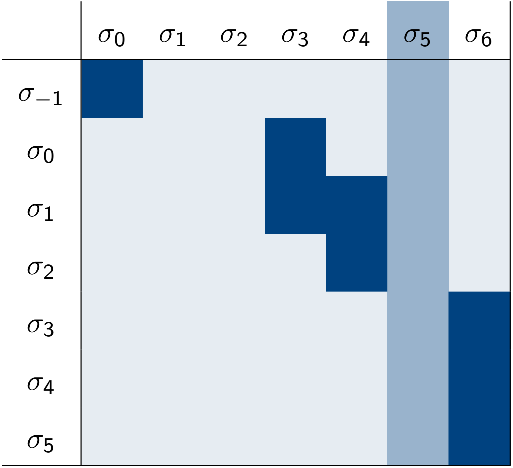
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

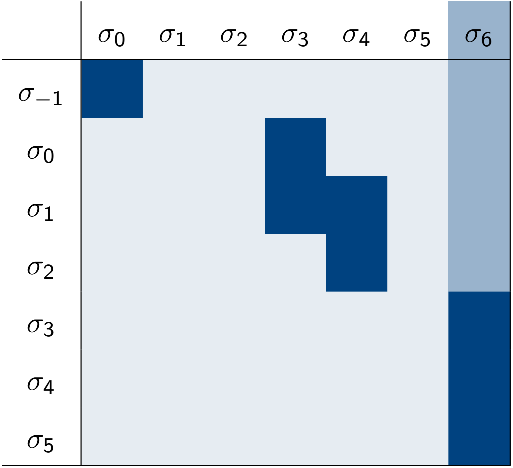

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

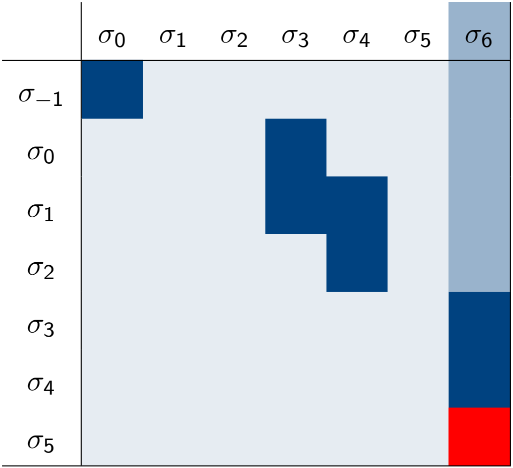


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

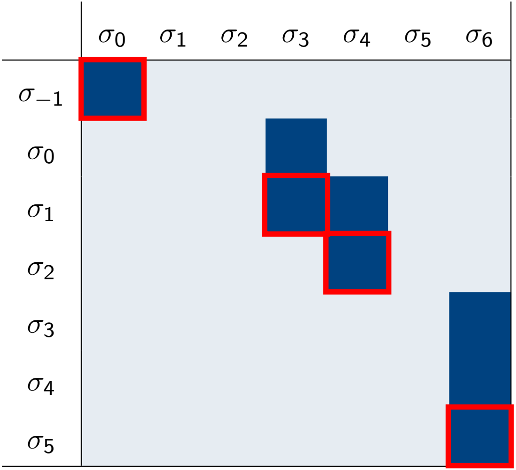
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

CLEAR: A generator that gives birth to an homological class cannot kill one.

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

CLEAR: A generator that gives birth to an homological class cannot kill one.
COMPRESS: A generator that kills an homological class cannot give birth to one.

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

CLEAR: A generator that gives birth to an homological class cannot kill one.
COMPRESS: A generator that kills an homological class cannot give birth to one.

$$\left\{ \begin{array}{c} \text{coboundary matrix} \\ \text{with clear} \end{array} \right\} \not\Longleftrightarrow \left\{ \begin{array}{c} \text{boundary matrix} \\ \text{with compress} \end{array} \right\}$$

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

CLEAR: A generator that gives birth to an homological class cannot kill one.
COMPRESS: A generator that kills an homological class cannot give birth to one.

$$\left\{ \begin{array}{c} \text{column operations on} \\ \text{coboundary matrix} \\ \text{with clear} \end{array} \right\} \not\Longleftrightarrow \left\{ \begin{array}{c} \text{column operations on} \\ \text{boundary matrix} \\ \text{with compress} \end{array} \right\}$$

Clear and compress

Two time optimisations of the standard persistence algorithm are the clear and compress.

CLEAR: A generator that gives birth to an homological class cannot kill one.
COMPRESS: A generator that kills an homological class cannot give birth to one.

$$\left\{ \begin{array}{c} \text{column operations on} \\ \text{coboundary matrix} \\ \text{with clear} \end{array} \right\} \Longleftrightarrow \left\{ \begin{array}{c} \text{row operations on} \\ \text{boundary matrix} \\ \text{with compress} \end{array} \right\}$$

Dualisation of the standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

Dualisation of the standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the i-th row of D, $left(i)$ denotes the column index of the leftmost element of such row. If row i is zero, set $left(i) = 0$.

Dualisation of the standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the i-th row of D, $left\,(i)$ denotes the column index of the leftmost element of such row. If row i is zero, set $left\,(i) = 0$.

Definition

A matrix R is called **row reduced** if $left\,(i) \neq left\,(i')$ for all non-zero rows $i \neq i'$. The element R^i_j is called a **row pivot** (of R) if $j = left\,(i)$.

Dualisation of the standard persistence algorithm

Let X be a filtered chain complex with quasi-minimal set of generators $\{x_1, \ldots, x_m\}$, totally ordered first by entrance time. Let D be its boundary matrix.

For the i-th row of D, $left(i)$ denotes the column index of the leftmost element of such row. If row i is zero, set $left(i) = 0$.

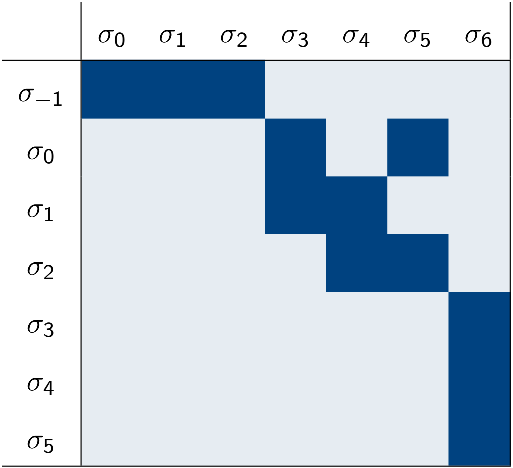
Definition

A matrix R is called **row reduced** if $left(i) \neq left(i')$ for all non-zero rows $i \neq i'$. The element R^i_j is called a **row pivot** (of R) if $j = left(i)$.

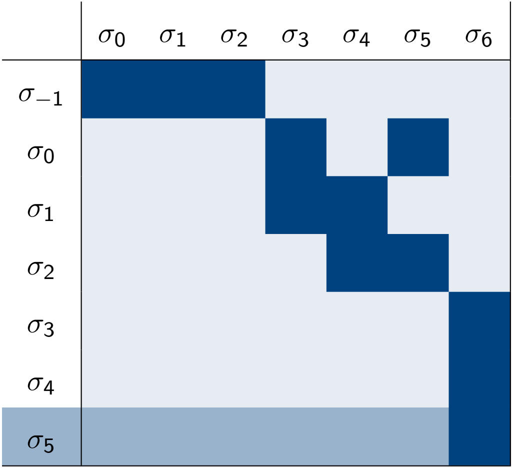
Theorem

The row pivot pairing of a reduced boundary matrix provides the barcode decomposition of the associated persistent module.

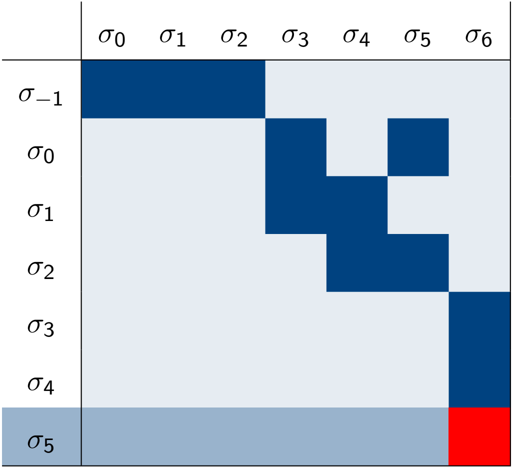
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

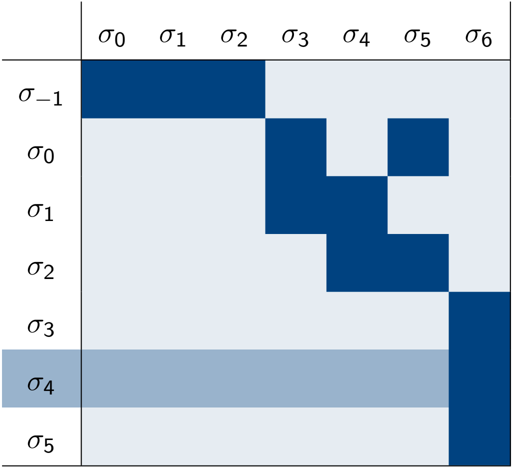
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

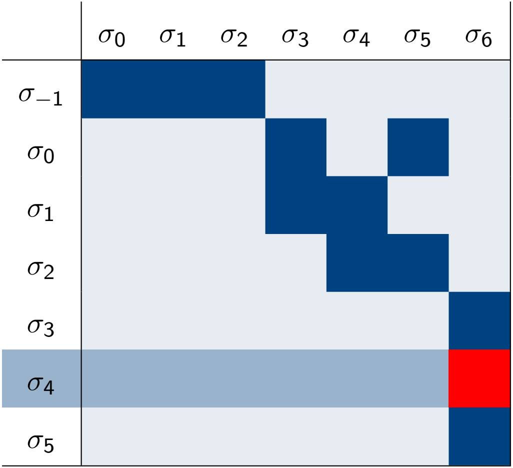
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

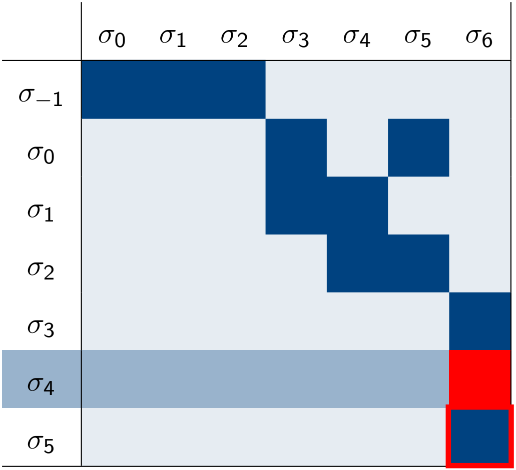
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

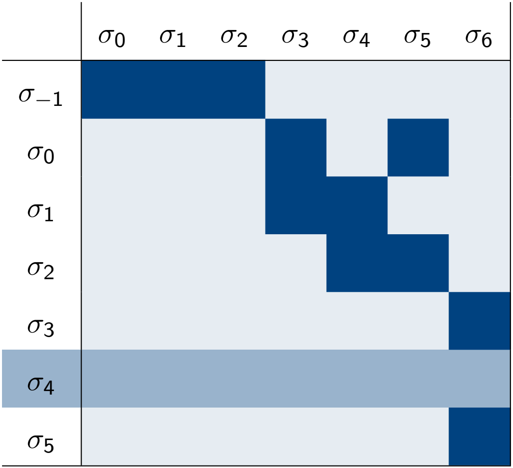
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

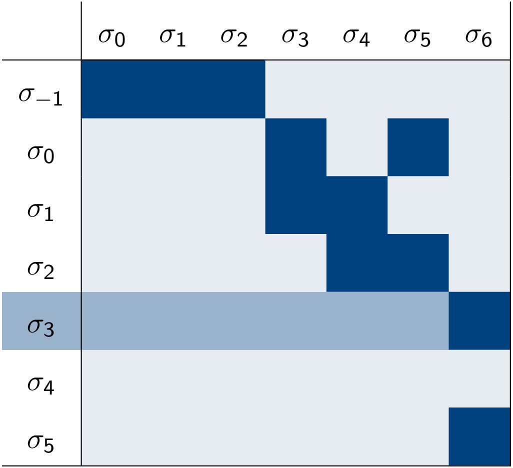
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

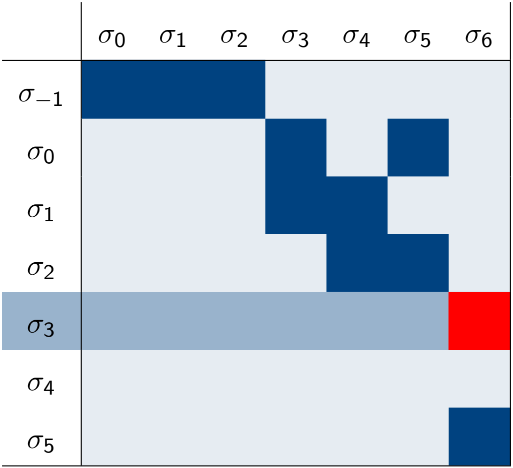
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

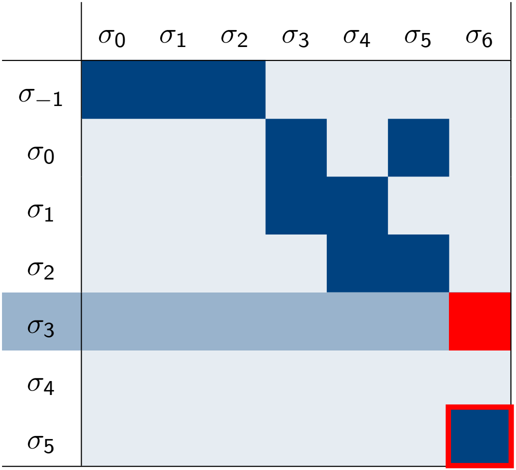
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

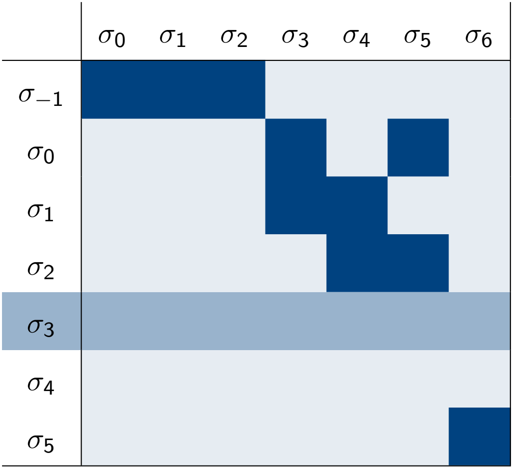
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

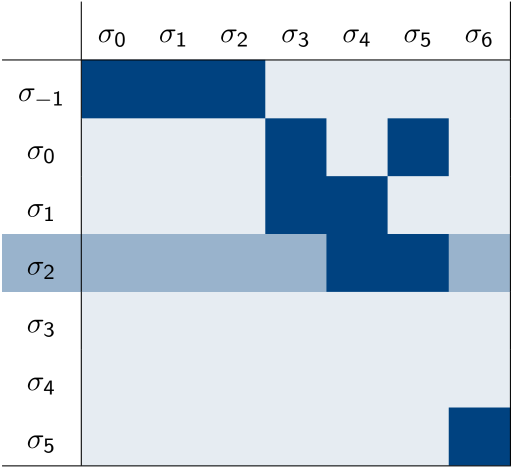
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

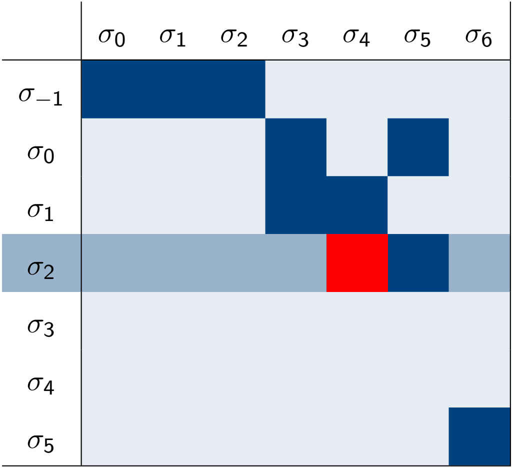
Example



Example

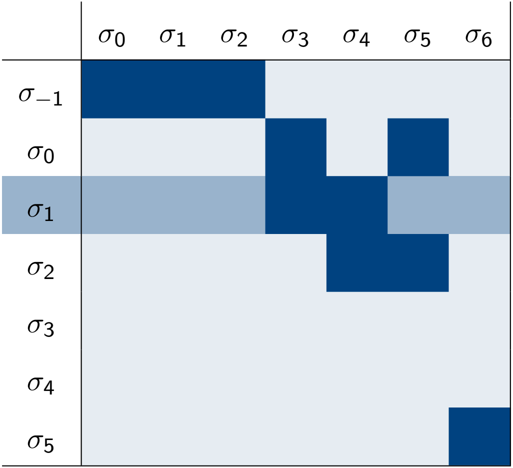

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example



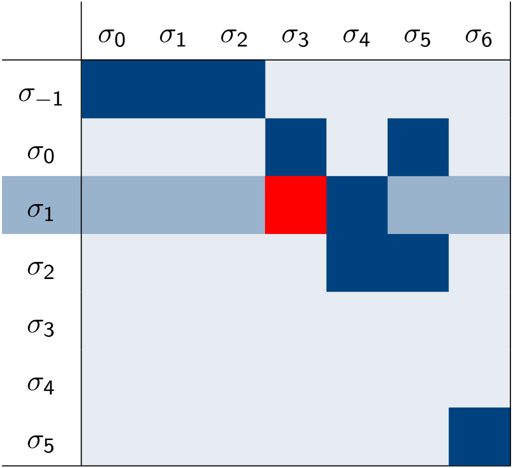
σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

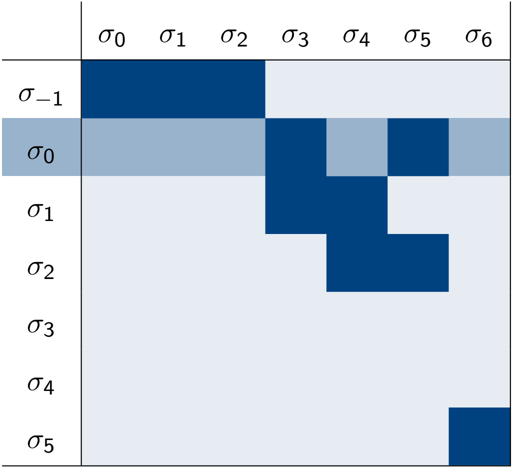


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

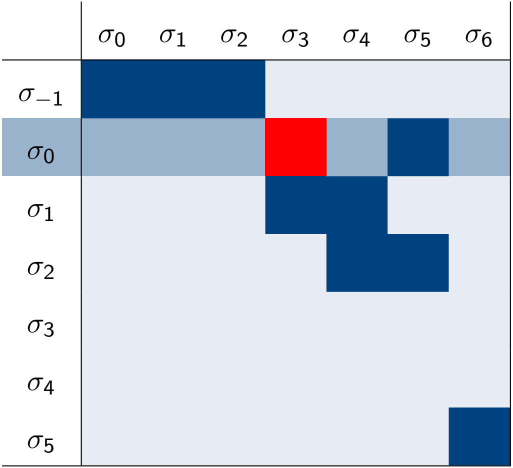
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

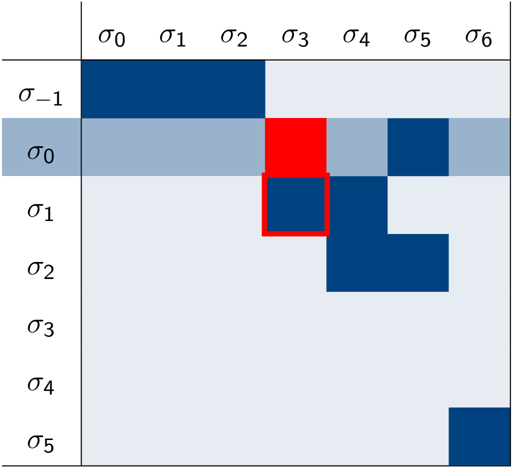
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

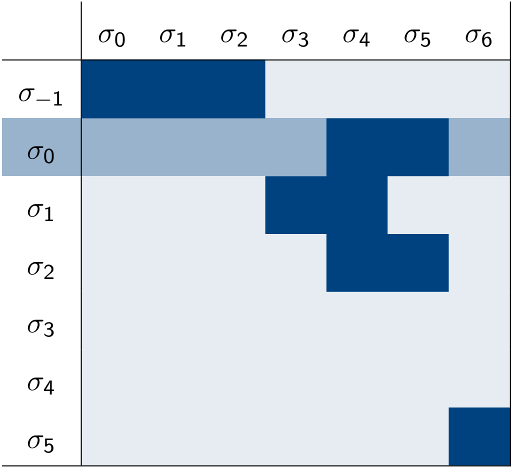

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

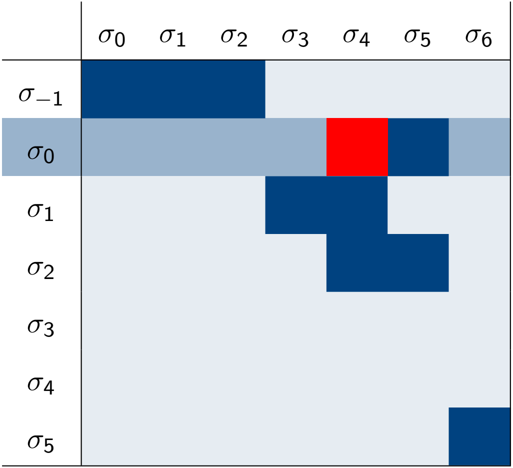


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

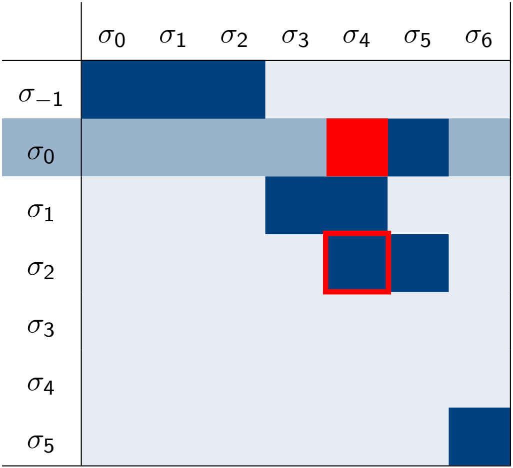
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

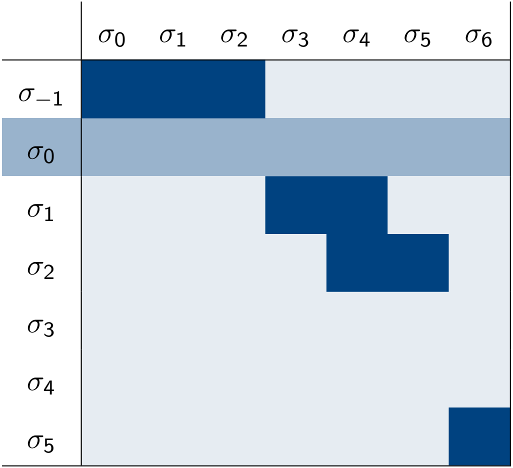
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

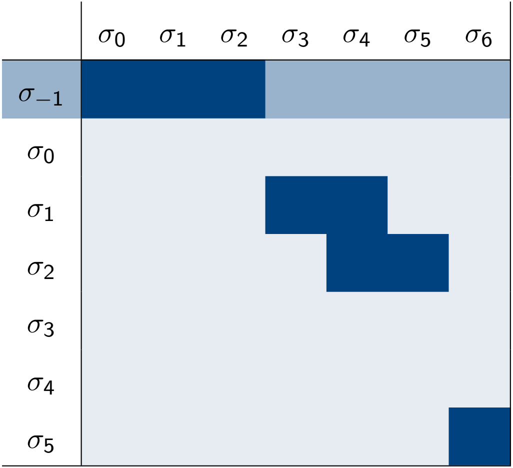
Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

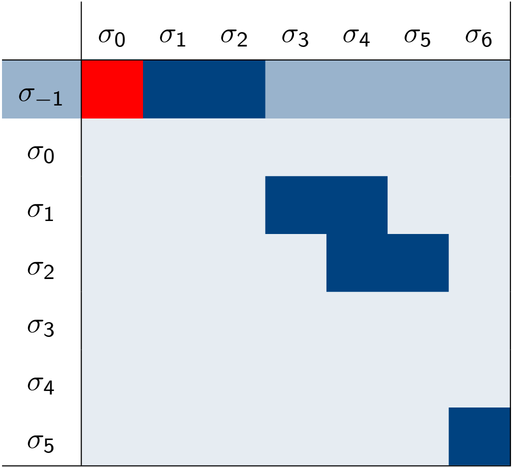

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

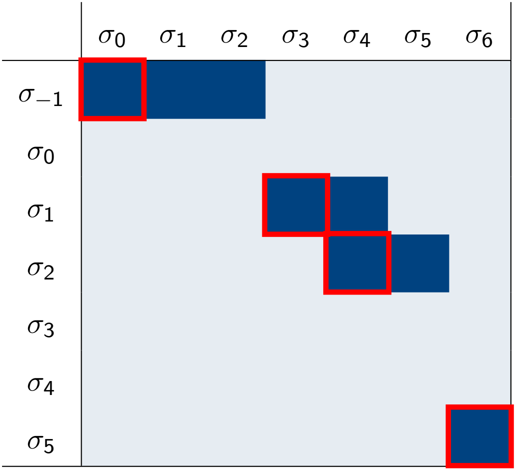


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Number of operations

Let X be a filtered Vietoris-Rips complex, whose quasi-minimal set of generators is ordered first by degree. Let v be the number of generators in degree 0 and N the maximal non-trivial degree of X.

Number of operations

Let X be a filtered Vietoris-Rips complex, whose quasi-minimal set of generators is ordered first by degree. Let v be the number of generators in degree 0 and N the maximal non-trivial degree of X.
The number of rows to be processed is:

$$\sum_{n=0}^{N} \underbrace{\binom{v}{n+1}}_{\dim(C_n X)} = \sum_{n=0}^{N} \underbrace{\binom{v-1}{n}}_{\dim(B_{n-1} X)} + \sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(Z_n X)}$$

Number of operations

Let X be a filtered Vietoris-Rips complex, whose quasi-minimal set of generators is ordered first by degree. Let v be the number of generators in degree 0 and N the maximal non-trivial degree of X.

The number of rows to be processed is:

$$\sum_{n=0}^{N} \underbrace{\binom{v}{n+1}}_{\dim(C_n X)} = \sum_{n=0}^{N} \underbrace{\binom{v-1}{n}}_{\dim(B_{n-1} X)} + \sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(Z_n X)}$$

Using the compress optimisation, this decreases to:

$$\underbrace{\binom{v-1}{0}}_{\dim(B_{-1} X)} + \sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(Z_n X)} = \sum_{n=0}^{N+1} \binom{v-1}{n}$$

Number of operations

Let X be a filtered Vietoris-Rips complex, whose quasi-minimal set of generators is ordered first by degree. Let v be the number of generators in degree 0 and N the maximal non-trivial degree of X.

The number of columns to be processed is:

$$\sum_{n=1}^{N+1} \underbrace{\binom{v}{n+1}}_{\dim(C^n X)} = \sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(B^{n+1} X)} + \sum_{n=0}^{N} \underbrace{\binom{v-1}{n}}_{\dim(Z^n X)}$$

Number of operations

Let X be a filtered Vietoris-Rips complex, whose quasi-minimal set of generators is ordered first by degree. Let v be the number of generators in degree 0 and N the maximal non-trivial degree of X.
The number of columns to be processed is:

$$\sum_{n=1}^{N+1} \underbrace{\binom{v}{n+1}}_{\dim(C^n X)} = \sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(B^{n+1} X)} + \sum_{n=0}^{N} \underbrace{\binom{v-1}{n}}_{\dim(Z^n X)}$$

Using the clear optimisation, this decreases to:

$$\sum_{n=0}^{N} \underbrace{\binom{v-1}{n+1}}_{\dim(B^{n+1} X)} + \underbrace{\binom{v-1}{0}}_{\dim(Z^0 X)} = \sum_{n=0}^{N+1} \binom{v-1}{n}$$

Algorithm for filtered chain complexes

Decomposition into interval spheres

Input: Totally ordered quasi-minimal set of generators $\{x_1, \ldots, x_m\}$
Output: List of interval spheres

```
List = ∅
D = BOUNDARY MATRIX(x_1, ..., x_m)
while exists non-zero row i in D do
    x_i, x_j = PAIR(D, i)
    Append I^{deg(x_i)} [ent(x_i), ent(x_j)) to List
    SPLIT(x_i, x_j, D)
for all indices i of remaining rows in D do
    Append I^{deg(x_i)} [ent(x_i), ∞) to List
Return List
```

Algorithm for filtered chain complexes

SPLIT

Input: Boundary matrix D, pair of splittable generators x_i, x_j

Output: Reduced boundary matrix D

for k *with* $D_j^k \neq 0$ **do**
    Add to the k-th row the i-th row multiplied by $-(D_j^i)^{-1} D_j^k$
Delete row j from D
Delete row i and column j from D
Delete column i from D

Algorithm for filtered chain complexes

SPLIT

Input: Boundary matrix D, pair of splittable generators x_i, x_j

Output: Reduced boundary matrix D

for k *with* $D_j^k \neq 0$ **do**
  Add to the k-th row the i-th row multiplied by $-(D_j^i)^{-1} D_j^k$

Delete row j from D $\Longrightarrow$ COMPRESS

Delete row i and column j from D

Delete column i from D

Algorithm for filtered chain complexes

SPLIT

Input: Boundary matrix D, pair of splittable generators x_i, x_j

Output: Reduced boundary matrix D

for k *with* $D_j^k \neq 0$ **do**

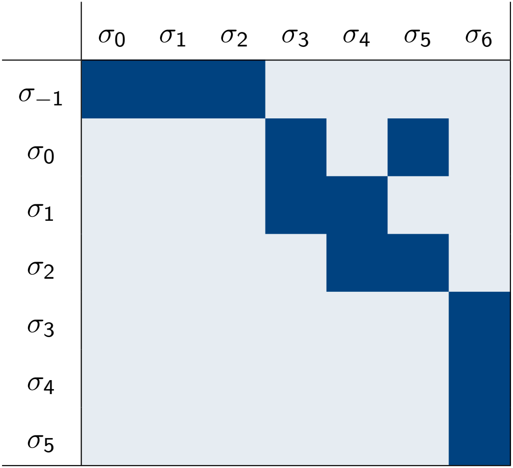
  Add to the k-th row the i-th row multiplied by $-(D_j^i)^{-1} D_j^k$

Delete row j from D $\Longrightarrow$ COMPRESS

Delete row i and column j from D

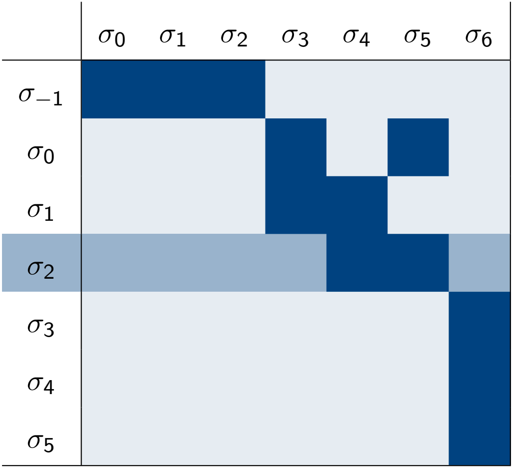
Delete column i from D $\Longrightarrow$ CLEAR

Example

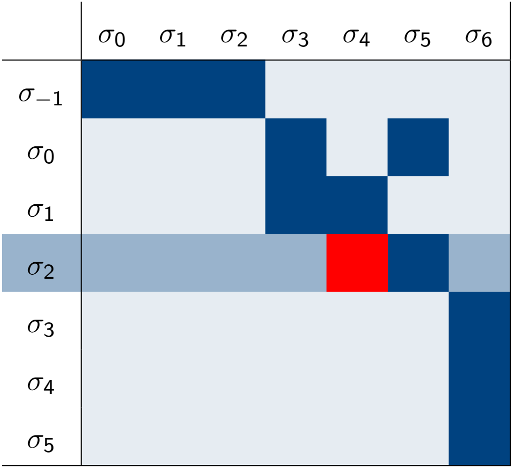


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

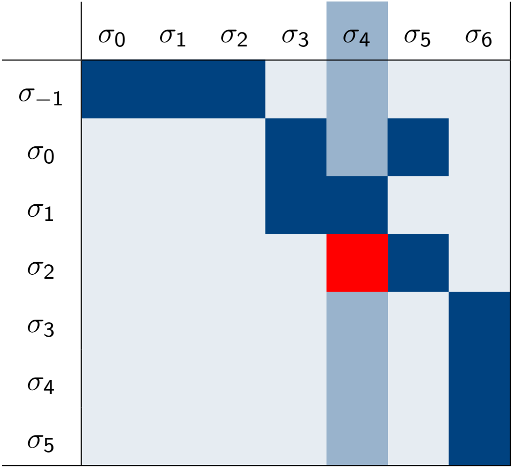


Example

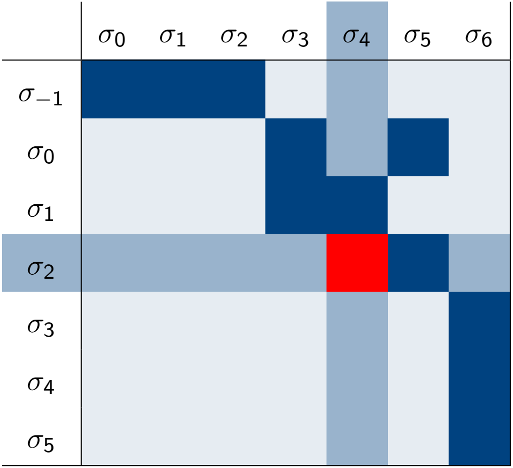


σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example

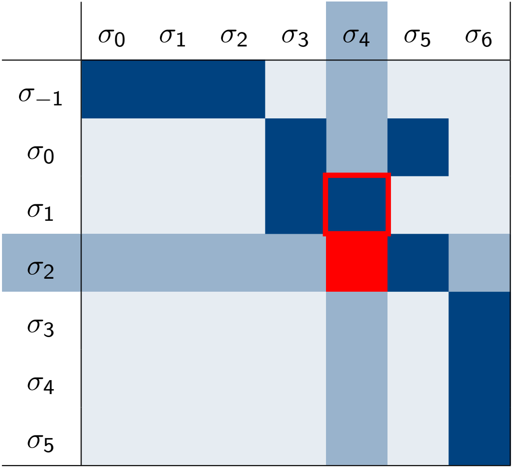

σ_0 σ_1 σ_2 σ_3 σ_4 σ_5 σ_6
σ_{-1}
σ_0
σ_1
σ_2
σ_3
σ_4
σ_5

Example



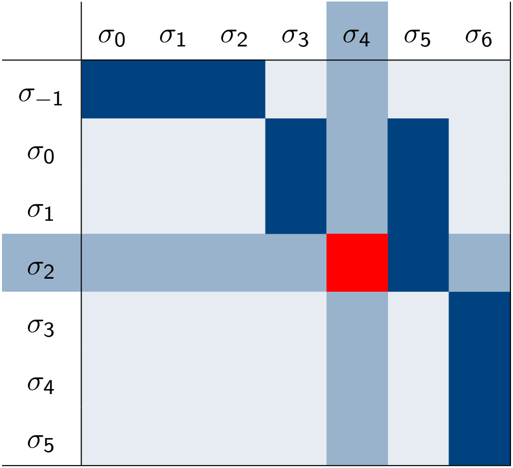
I^0 $[2, 4)$

Example



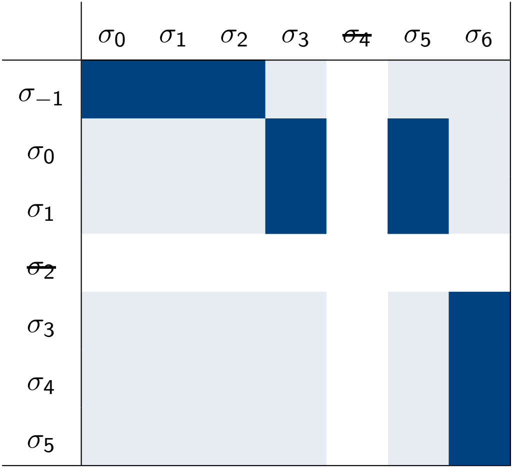
$I^0\ [2,4)$

Example



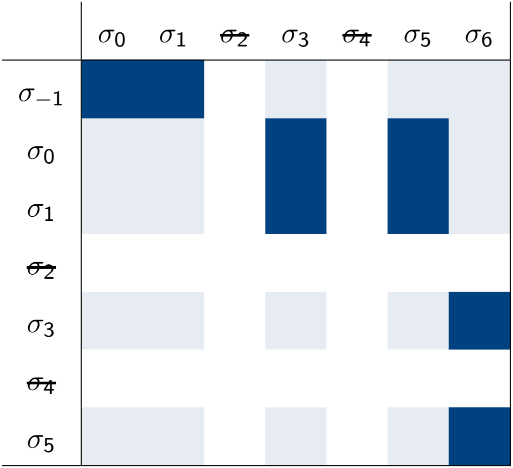
$I^0\ [2, 4)$

Example



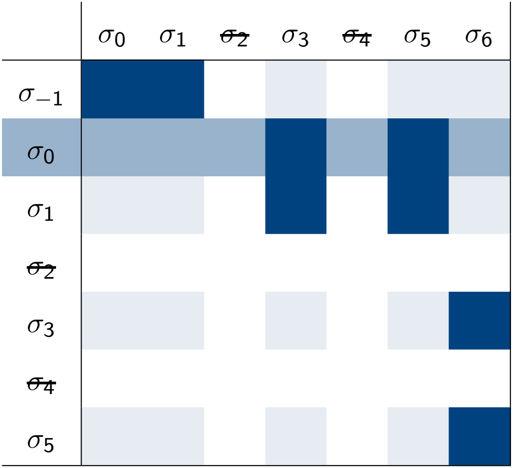
$I^0\ [2,4)$

Example



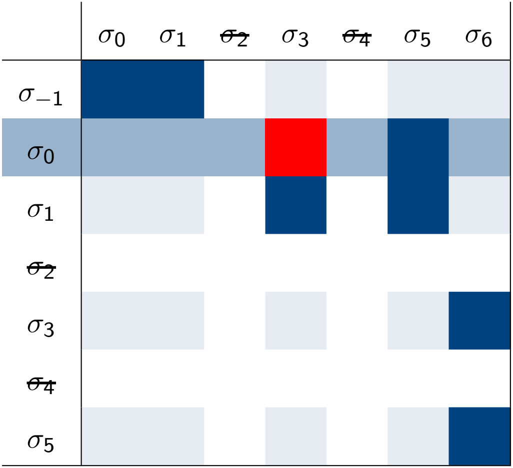
$I^0\ [2,4)$

Example



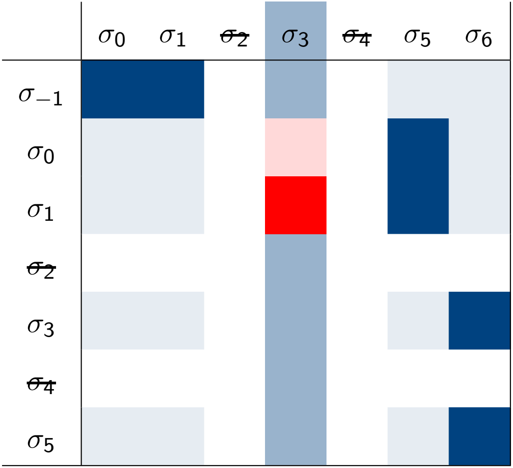
$I^0\ [2, 4)$

Example



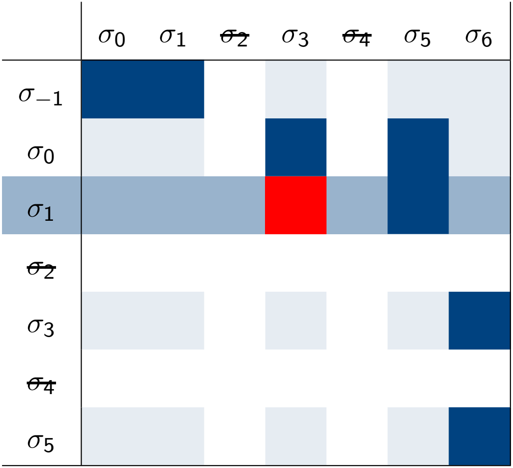
$I^0 \; [2, 4)$

Example



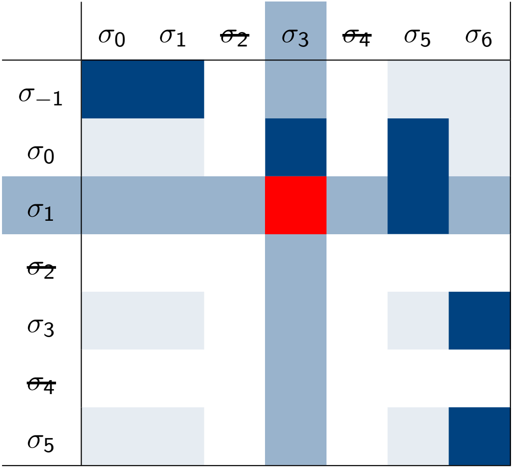
$I^0\ [2,4)$

Example



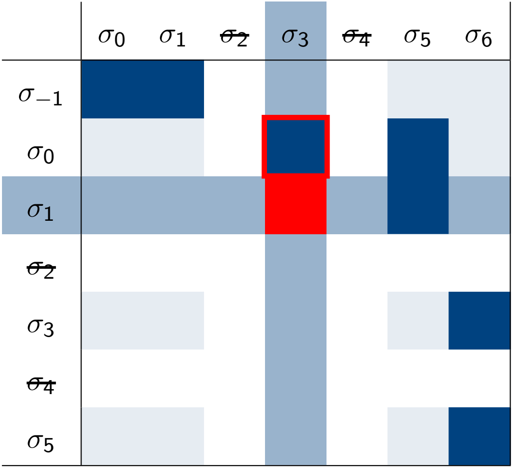
$I^0\ [2,4)$

Example



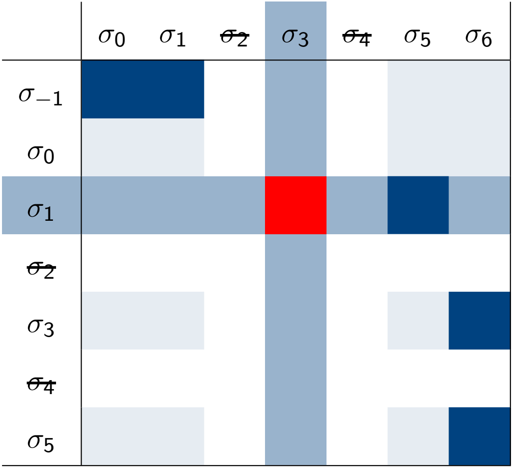
$I^0\left[2,4\right), I^0\left[1,3\right)$

Example



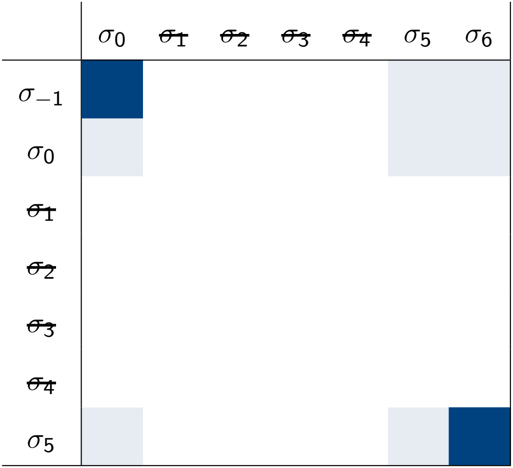
$I^0\left[2,4\right), I^0\left[1,3\right)$

Example



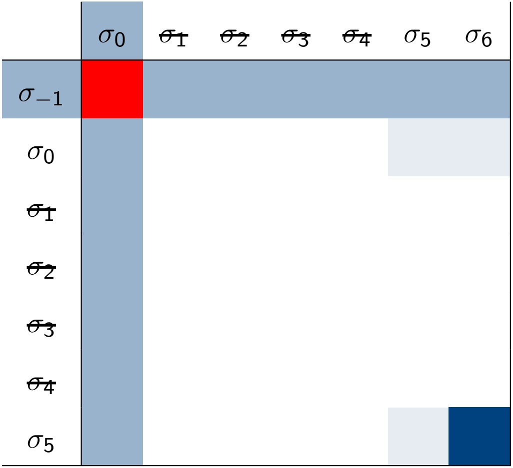
$I^0\,[2,4)\,,\,I^0\,[1,3)$

Example



$I^0\,[2,4)\,,\,I^0\,[1,3)$

Example



$I^0\left[2,4\right), I^0\left[1,3\right), I^0\left[0,\infty\right)$

Example

	$\cancel{\sigma_0}$	$\cancel{\sigma_1}$	$\cancel{\sigma_2}$	$\cancel{\sigma_3}$	$\cancel{\sigma_4}$	σ_5	σ_6
$\cancel{\sigma_{-1}}$							
$\cancel{\sigma_0}$							
$\cancel{\sigma_1}$							
$\cancel{\sigma_2}$							
$\cancel{\sigma_3}$							
$\cancel{\sigma_4}$							
σ_5							

$I^0\,[2,4)\,,\; I^0\,[1,3)\,,\; I^0\,[0,\infty)$

Example

	$\cancel{\sigma_0}$	$\cancel{\sigma_1}$	$\cancel{\sigma_2}$	$\cancel{\sigma_3}$	$\cancel{\sigma_4}$	σ_5	σ_6
$\cancel{\sigma_{-1}}$							
$\cancel{\sigma_0}$							
$\cancel{\sigma_1}$							
$\cancel{\sigma_2}$							
$\cancel{\sigma_3}$							
$\cancel{\sigma_4}$							
σ_5							

$I^0\ [2,4)\ ,\ I^0\ [1,3)\ ,\ I^0\ [0,\infty)\ ,\ I^1\ [5,6)$

Example

	$\cancel{\sigma_0}$	$\cancel{\sigma_1}$	$\cancel{\sigma_2}$	$\cancel{\sigma_3}$	$\cancel{\sigma_4}$	$\cancel{\sigma_5}$	$\cancel{\sigma_6}$
$\cancel{\sigma_{-1}}$							
$\cancel{\sigma_0}$							
$\cancel{\sigma_1}$							
$\cancel{\sigma_2}$							
$\cancel{\sigma_3}$							
$\cancel{\sigma_4}$							
$\cancel{\sigma_5}$							

$I^0\left[2,4\right), I^0\left[1,3\right), I^0\left[0,\infty\right), I^1\left[5,6\right)$

Conclusions

- Algorithm to decompose filtered chain complexes into interval spheres
- The algorithm does not require a fixed order on generators, intrinsically adopts both the clear and compress, and produces the barcode decomposition as a byproduct
- "Novel" algorithm to retrieve the barcode decomposition using row operations, dualising the standard persistence algorithm

Conclusions

- Algorithm to decompose filtered chain complexes into interval spheres
- The algorithm does not require a fixed order on generators, intrinsically adopts both the clear and compress, and produces the barcode decomposition as a byproduct
- "Novel" algorithm to retrieve the barcode decomposition using row operations, dualising the standard persistence algorithm

Conclusions

- Algorithm to decompose filtered chain complexes into interval spheres
- The algorithm does not require a fixed order on generators, intrinsically adopts both the clear and compress, and produces the barcode decomposition as a byproduct
- "Novel" algorithm to retrieve the barcode decomposition using row operations, dualising the standard persistence algorithm

Conclusions

- Algorithm to decompose filtered chain complexes into interval spheres
- The algorithm does not require a fixed order on generators, intrinsically adopts both the clear and compress, and produces the barcode decomposition as a byproduct
- "Novel" algorithm to retrieve the barcode decomposition using row operations, dualising the standard persistence algorithm

Invariants for tame parametrised chain complexes,
W. Chachólski, B. Giunti, and C. Landi, 2020

Algorithmic decomposition of filtered chain complexes,
W. Chachólski, B. Giunti, A. Jin, and C. Landi, 2020

Notes on pivot pairings, B. Giunti, 2021

Conclusions

Future directions:

Test the performances of the different algorithms

Optimise the PAIR algorithm

- Using local properties of filtered simplicial complexes
- Parallelising

Explore different orders

- To improve memory access
- To improve (co)faces generation and retrieval

Conclusions

Future directions:

Test the performances of the different algorithms

Optimise the PAIR algorithm

- Using local properties of filtered simplicial complexes
- Parallelising

Explore different orders

- To improve memory access
- To improve (co)faces generation and retrieval

Thank you for your attention!