CHAIN COMPLEX REDUCTION USING STEEPNESS MATCHINGS

LEON LAMPRET

ABSTRACT. Computing the homology of a chain complex with algebraic Morse theory is like doing Gaussian elimination, but only *on certain columns/rows* and *with several pivots* (in all the matrices at once). This significantly reduces fill-in and coefficient growth of the matrices, but it requires the construction of the set of pivots (called a Morse matching).

On any chain complex of free modules of finite rank, we define a family of acyclic matchings in the context of AMT. These pairings have relatively small memory requirements. The downside is that applying such a matching gives a (significantly smaller) homotopy-equivalent chain complex, but not necessarily its homology directly (except when working over a field or a local PID).

Many acyclic matchings in the literature are special instances of this construction. In fact, we show that *every* acyclic matching is a *subset* of some matching from our family, so all maximal Morse matchings are of this type.

We provide MATHEMATICA code for this reduction and compare its performance with SAGEMATH algorithms. Using it, we computed (on a laptop) the homology (over $\mathbb{Z}_2$ and $\mathbb{Q}$, with generators) of a chain complex of Heisenberg Lie algebras, where the largest matrix is bigger than 20 million $\times$ 20 million.

Motivation. Since chain complexes have become an important part of many mathematical areas, there is a big demand for efficient methods (with regard to the processor as well as the memory load) for the computation of their homology. Unfortunately, this continues to be a hard problem, since it appears it is still infeasible to compute with sparse matrices ∂_k larger than $10^6 \times 10^6$.

Note that in general, *the bigger the matrices in C_* get, the sparser they become.* For instance, in the Poincaré / Eilenberg-MacLane / Hochschild / Chevalley chain complex, the k-th matrix has density:

- $= \frac{k \cdot f_k}{f_{k-1} \cdot f_k} \xrightarrow{f_{k-1} \to \infty} 0$, where f_k is the number of k-faces of the simplicial complex;
- $= \frac{(k+1) \cdot n^k}{n^{k-1} \cdot n^k} \xrightarrow{n \to \infty} 0$, where n is the cardinality of the group;
- $< \frac{(k+1) n \cdot n^{k+1}}{n^k \cdot n^{k+1}} \xrightarrow{n \to \infty} 0$, where n is the dimension of the associative algebra;
- $< \frac{\binom{k}{2} n \cdot \binom{n}{k}}{\binom{n}{k-1} \cdot \binom{n}{k}} \xrightarrow{n \to \infty} 0$, where n is the dimension of the Lie algebra.

Therefore, it makes sense to concentrate on methods that work best on sparse chain complexes, so we assume the matrices contain mainly zeros.

I offer a new algorithm, which reduces the size of all boundary matrices ∂_k at once, by deleting every row i and column j that contain an invertible entry $\partial_{k,i,j}$ which has zeros left and below it (i.e. $\partial_{k,i,<j} = 0 = \partial_{k,>i,j}$), and (possibly) alters

Date: March 2, 2019.

1991 *Mathematics Subject Classification.* 17B56, 13P20, 13D02, 55-04, 18G35, 58E05.

Key words and phrases. algebraic/discrete Morse theory, homological algebra, chain complex, acyclic matching, algebraic combinatorics, algorithm implementation, MATHEMATICA, code.

the remaining entries. The capabilities of my implementation are documented in 5.4. It requires no computation of the Smith normal forms or ranks of matrices.

Conventions. Throughout this article, R will be a commutative unital ring and

$$C_*\colon C_0 \xleftarrow{\partial_1} C_1 \leftarrow \ldots \leftarrow C_{N-1} \xleftarrow{\partial_N} C_N$$

a chain complex of free R-modules of finite rank. Also, $R^\times$ is the group of units (=invertible elements) of R, e.g. $\mathbb{Z}^\times = \{1,-1\}$ and $K^\times = K\setminus\{0\}$ for any field K.

1. Introduction to AMT

My algorithm uses algebraic Morse theory, so I include a concise review of it.

1.1. **Formulation.** Pick a basis I_k for each C_k. Let Γ_{C_*} be a graph, with vertex set the disjoint union $\bigsqcup_k I_k$, and for every nonzero matrix entry $\partial_{k,u,v} = w \in R\setminus\{0\}$ a directed weighted edge $u \xleftarrow{w} v$.

A *Morse matching* is any collection $\mathcal{M}$ of edges from Γ_{C_*}, such that:

(1) $\mathcal{M}$ is a *matching*, i.e. edges in $\mathcal{M}$ have no common vertex, i.e. whenever $u\leftarrow v, x\leftarrow y \in \mathcal{M}$ we have $|\{u,v,x,y\}| = 4$;
(2) for every edge $u \xrightarrow{w} v$ in $\mathcal{M}$, the corresponding weight w is invertible in R;
(3) denote by $\Gamma_{C_*}^{\mathcal{M}}$ the graph, obtained from Γ_{C_*} by replacing every $u \xleftarrow{w} v \in \mathcal{M}$ with $u \xrightarrow{-1/w} v$; then $\Gamma_{C_*}^{\mathcal{M}}$ must contain no directed cycles, and no infinite paths $u_1 \to v_1 \to u_2 \to v_2 \to \ldots$ with all $u_1, u_2, \ldots \in I_k$.

Let $\mathcal{M}$ be a Morse matching. Denote $\mathcal{M}_k = \mathcal{M} \cap \partial_k$, $m_k = |\mathcal{M}_k|$,
$I_k^+ = \{v \in I_k;\ \exists u \leftarrow v \in \mathcal{M}\} = \{\text{indices of columns in } \partial_k \text{ that contain some } e \in \mathcal{M}\}$,
$I_k^- = \{u \in I_k;\ \exists u \leftarrow v \in \mathcal{M}\} = \{\text{indices of rows in } \partial_{k+1} \text{ that contain some } e \in \mathcal{M}\}$,
$I_k' = I_k \setminus (I_k^+ \cup I_k^-) = \{v \in I_k;\ v \text{ is not incident to any } e \in \mathcal{M}\}$, the *critical* vertices.
Let $\Gamma_{v,u}$ denote the set of all directed paths γ in Γ_{C_*} from v to u (including paths of length 0 when $v = u$). Given such $\gamma = (v \xrightarrow{w_0} u_1 \xleftarrow{-1/w_1} v_1 \xrightarrow{w_2} u_2 \xleftarrow{-1/w_3} v_2 \to \ldots \leftarrow v_r \xrightarrow{w_{2r}} u)$, let $\partial_\gamma(v) = w_{2r} \cdots \frac{-1}{w_3} w_2 \frac{-1}{w_1} w_0\, u$ be the multiple of u by the product of weights. Let C_k' be the free module on I_k'. Define $\partial_k'\colon C_k' \to C_{k-1}'$, $f_k\colon C_k' \to C_k$, $g_k\colon C_k \to C_k'$ by

$$\partial_k'(v') = \sum_{\substack{u' \in I_{k-1}',\\ \gamma \in \Gamma_{v',u'}}} \partial_\gamma(v'), \qquad f_k(v') = \sum_{\substack{v \in I_k,\\ \gamma \in \Gamma_{v',v}}} \partial_\gamma(v'), \qquad g_k(v) = \sum_{\substack{v' \in I_k',\\ \gamma \in \Gamma_{v,v'}}} \partial_\gamma(v). \tag{4}$$

Theorem 1.2 ([6, 2, 3] 2005)**.** ***(a)*** *For any Morse matching $\mathcal{M}$, the induced $f_*\colon (C_*', \partial_*') \to (C_*, \partial_*)$ is a homotopy-equivalence of chain complexes, with h-inverse g_*.*

(b) *There exist bases $\widetilde{I}_k$ in which* $C_* = \ldots \xleftarrow{\begin{bmatrix} \partial'_{k-1} & 0 & 0 \\ 0 & \mathrm{id}_{m_{k-1}} & 0 \\ 0 & 0 & 0 \end{bmatrix}} C_{k-1} \xleftarrow{\begin{bmatrix} \partial'_{k} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathrm{id}_{m_k} \end{bmatrix}} C_k \xleftarrow{\begin{bmatrix} \partial'_{k+1} & 0 & 0 \\ 0 & \mathrm{id}_{m_{k+1}} & 0 \\ 0 & 0 & 0 \end{bmatrix}} \ldots.$

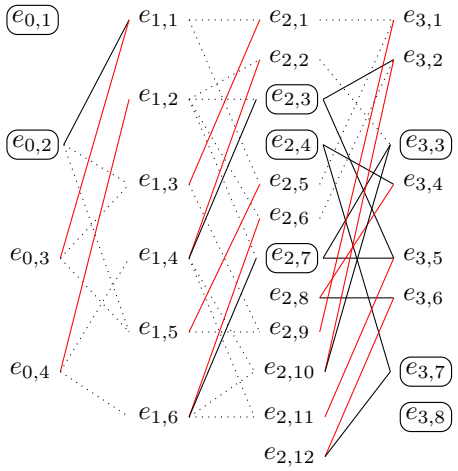
Part (a) of 1.2 holds even if C_* is unbounded (i.e. $C_k \neq 0$ for infinitely many $k \in \mathbb{Z}$) and C_k is any (not necessarily finite) direct sum of submodules $\bigoplus_{v \in I_k} C_{k,v}$ over a (not necessarily commutative) unital ring. Then the column-finitary boundary matrices ∂_k have for entries $\partial_{k,u,v}$ not weights $w \in R$ but morphisms $C_{k-1,u} \xleftarrow{\varphi} C_{k,v}$, condition (2) says that φ must be invertible as a morphism, and ∂_*, f_*, g_* are defined via $\partial_\gamma = \varphi_{2r} \circ \ldots \circ (-\varphi_3^{-1}) \circ \varphi_2 \circ (-\varphi_1^{-1}) \circ \varphi_0$. However, for our purposes, we shall work

within the confines of assumptions stated in the conventions above. Hence the part of condition (3) about no infinite paths is always satisfied.

1.3. **Example.** Consider the Khovanov chain complex for the trefoil knot

$$C_*\colon\ R^4 \xleftarrow{\begin{bmatrix} 0&0&0&0&0&0\\ 1&0&-1&0&1&0\\ 1&0&-1&0&1&0\\ 0&1&0&-1&0&1 \end{bmatrix}} R^6 \xleftarrow{\begin{bmatrix} -1&0&0&0&1&0&0&0&0&0&0&0\\ 0&-1&-1&0&0&1&1&0&0&0&0&0\\ -1&0&0&0&0&0&0&0&1&0&0&0\\ 0&-1&-1&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&-1&0&0&0&1&0&0&0\\ 0&0&0&0&0&-1&-1&0&0&1&1&0 \end{bmatrix}} R^{12} \xleftarrow{\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&1&0&0&1&0&0&0\\ 0&0&0&1&0&0&1&0\\ 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&1&0&0&0\\ 0&0&0&1&0&1&0&0\\ 1&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&1&0 \end{bmatrix}} R^8.$$

Let the Morse matching consist of the red entries/edges. The associated graph Γ_{C_*} (without displayed weights, and with $I_k = \{e_{k,1}, e_{k,2}, \ldots\}$) is below left. Gray entries / dotted edges are the ones that can be removed by remark 1.4 (c). I omitted the arrows, since black and dotted lines always point leftwards, red lines point rightwards. Below right are the paths between critical vertices, that give ∂'_*.



∂_γ:

$e_{2,10}$ — $e_{3,3}$, $e_{2,3}$ — $e_{3,2}$: $1 \cdot \frac{-1}{1} \cdot 1 = -1$

$e_{2,7}$ — $e_{3,3}$: 1

$e_{2,4}$ — $e_{3,7}$: 1

$e_{2,12}$ — $e_{3,7}$, $e_{2,8}$ — $e_{3,6}$, $e_{2,4}$ — $e_{3,4}$: $1 \cdot \frac{-1}{1} \cdot 1 \cdot \frac{-1}{1} \cdot 1 = 1$

From this, we get an h-equivalent complex $C'_*\colon R^2 \xleftarrow{0} R^0 \xleftarrow{0} R^3 \xleftarrow{\begin{bmatrix} -1&0&0\\ 0&2&0\\ 1&0&0 \end{bmatrix}} R^3$. Moreover,

$$f_* = \begin{bmatrix} 1&0\\ 0&1\\ 0&0\\ 0&0 \end{bmatrix}, 0, \begin{bmatrix} 0&0&0\\ -1&0&0\\ 1&0&0\\ 0&1&0\\ 0&0&0\\ 0&0&-1\\ 0&0&1\\ 0&0&0\\ 0&0&0\\ 0&0&0\\ 0&0&0\\ 0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0\\ -1&0&0\\ 1&0&0\\ 0&1&0\\ 0&0&0\\ 0&-1&0\\ 0&1&0\\ 0&0&1 \end{bmatrix} \text{ and } g_* = \begin{bmatrix} 1&0&0&0\\ 0&1&-1&0 \end{bmatrix}, 0, \begin{bmatrix} 0&0&1&0&0&0&0&0&0&-1&-1&0\\ 0&0&0&1&0&0&0&-1&0&0&0&1\\ 0&0&0&0&0&0&1&0&0&0&-1&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$

Thus over $R = \mathbb{Z}$, the homology and its generators are $H_0C_* = \langle e_{0,1}, e_{0,2} - e_{0,3} \rangle \cong \mathbb{Z}^2$, $H_1C_* = 0$, $H_2C_* = \langle e_{2,3} - e_{2,2}, e_{2,4} \rangle \cong \mathbb{Z} \oplus \mathbb{Z}_2$, $H_3C_* = \langle e_{3,8} \rangle \cong \mathbb{Z}$. ◊

1.4. **Remarks. (a)** If C_* is the Poincaré chain complex of a finite simplicial complex Δ (or regular CW-complex), then Γ_{C_*} is the Hasse diagram of Δ and AMT reduces to Forman's DMT [1] (discrete Morse theory, 1995). However, DMT gives a topological h-equivalence of CW-complexes $\Delta \simeq \Delta'$, not just an algebraic $C_* \simeq C'_*$. However, to determine the gluing maps, one needs to take into account paths between all critical simplices, not just those in consecutive degree; see [5].

(b) Notice that ∂'_k is determined by $\partial_{k-1}, \partial_k, \partial_{k+1}$, not just ∂_k. Indeed, I'_k is obtained from $\partial_{k-1}, \partial_k$ and I'_{k+1} is obtained from $\partial_k, \partial_{k+1}$, whilst ∂'_k is computed from $I'_k, I'_{k+1}, \mathcal{M}_k, \partial_k$. Also, ∂'_k and $f_k(v') = v' + \sum_{v\in I_k^+, \gamma\in\Gamma_{v',v}} \partial_\gamma(v')$ contain only edges from ∂_k. On the other hand, $g_k(v) = v$ if $v \in I'_k$, and $g_k(v) = 0$ if $v \in I_k^+$, and $g_k(v) = \sum_{v'\in I'_k, \gamma\in\Gamma_{v,v'}} \partial_\gamma(v)$ if $v \in I_k^-$, so g_k contains only edges from ∂_{k+1}.

(c) For every $u \leftarrow v \in \mathcal{M}_k$, no edge $x \leftarrow u$ or $v \leftarrow y$ can lie on a path between critical vertices in $\Gamma_{C_*}^{\mathcal{M}}$, since $\mathcal{M}$ is a matching. Thus we can delete all entries in the u-column (or I_{k-1}^--columns) of ∂_{k-1} and v-row (or I_k^+-rows) of ∂_{k+1}, without changing the computed complex C'_* and h-equivalence f_*, g_* (using the same $\mathcal{M}$). Also, for every $u \overset{w}{\leftarrow} v \in \mathcal{M}_k$ we can replace u with $\frac{-1}{w}u$, i.e. multiply the u-row of ∂_k by $\frac{-1}{w}$, and get an isomorphic complex, where edges of $\mathcal{M}$ in $\Gamma_{C_*}^{\mathcal{M}}$ have weight 1.

(d) As long as ∂_* contains at least one invertible entry, by the construction below we always have a nonempty $\mathcal{M}_{\leq}$; see 3. Hence there is a sequence of h-equivalences

$$C_* \xleftarrow{f'_*} C'_* \xleftarrow{f''_*} C''_* \longleftarrow \ldots \xleftarrow{f_*^{(r)}} C_*^{(r)}, \tag{5}$$

where all entries of $\partial_*^{(r)}$ are nonunits. If R is a field, we have $\partial_*^{(r)} = 0$, so $C_k^{(r)} \cong H_k C_*$; then $f_* := f'_* \circ f''_* \circ \ldots \circ f_*^{(r)}$ gives the actual generators of homology in C_*; see 5.1.

(e) If $R_{(p)}$ is a localization of a PID R at a prime p, then the only noninvertible entries are multiples of p. Hence $\partial_*^{(r)} = p^a \widetilde{\partial}_*^{(r)}$ for some $a \in \mathbb{N}$ and $\widetilde{\partial}_*^{(r)}$. Via 1.2 (b) we can reconstruct $H_*(C_*^{(r)}, \partial_*^{(r)})$ from $H_*(C_*^{(r)}, \frac{\partial_*^{(r)}}{p^a})$. Hence we continue with (5) on $(C_*^{(r)}, \frac{\partial_*^{(r)}}{p^a})$: for every Morse matching $\mathcal{M}$ on it, we add $(\frac{R}{Rp^a})^{|\mathcal{M}_k|}$ to $H_{k-1}(C_*, \partial_*)$. This process ends at $\partial_*^{(r_1+\ldots+r_t)} = 0$, and we obtain the p-torsion of $H_*(C_*; R)$; see 4.

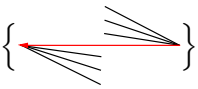
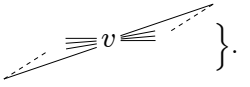
2. Matchings induced from orderings

In this section, we define a class of matchings on any chain complex. The construction requires (and is very dependent on) the ordering of basis elements.

2.1. **Formulation.** Pick a total order $\leq_k$ on I_k and let $\leq = \bigcup_k \leq_k$ be the combined partial order on the vertices of Γ_{C_*}. Visualize the elements of I_k positioned vertically, with u above v iff $u < v$ (as with row indices in the matrix ∂_{k+1}). Define

$$\mathcal{M}_{\leq} = \left\{ u \overset{w}{\leftarrow} v;\ \begin{smallmatrix} \forall x \leftarrow v\colon x < u, \\ \forall u \leftarrow y\colon v < y \end{smallmatrix} \right\} \cap \left\{ u \overset{w}{\leftarrow} v;\ w \in R^\times \right\},$$

the set of all steepest edges which also have invertible weights. The critical vertices are $I'_* = \left\{ v;\ \begin{smallmatrix} \text{if } v \overset{r}{\leftarrow} u \text{ is steepest into } v, & \text{then } r \in R\setminus R^\times \text{ or } \exists v' \leftarrow u \text{ with } v' > v, \\ \text{if } w \overset{s}{\leftarrow} v \text{ is steepest out of } v, & \text{then } s \in R\setminus R^\times \text{ or } \exists w \leftarrow v' \text{ with } v' < v \end{smallmatrix} \right\}$. Visually,

$\mathcal{M}_{\leq} = \Big\{$  $\Big\}$ and $I'_* = \Big\{ v;$  $\Big\}$.

Lemma 2.2. ***(a)*** *$\mathcal{M}_{\leq}$ is a Morse matching, the* steepness *pairing associated to $\leq$.*
(b) *Every Morse matching is a subset of some steepness matching, so*

$$\{\textit{maximal Morse matchings on } \Gamma_{C_*}\} \subseteq \{\textit{steepness pairings on } \Gamma_{C_*}\}.$$

Proof. **(a)** The three conditions from the above definition must be verified.

(1) $\mathcal{M}$ is a matching: For any edges $u \overset{x}{\underset{y}{\lessdot}}$, either $x<y$ or $y<x$, so both edges cannot be in $\mathcal{M}$. For any edges $\overset{x}{\underset{y}{\gtrdot}} v$, either $x<y$ or $y<x$, so both edges cannot be in $\mathcal{M}$. For any edges $x \overset{r}{\leftarrow} y \overset{s}{\leftarrow} z$, either $rs=0$ (then weights r and s are not units of R, so their edges cannot be in $\mathcal{M}$), or there exist edges $x \leftarrow v \leftarrow z$ with $y \neq v$ (otherwise $\partial_{k-1}\partial_k \neq 0$). Now, either $y<v$ or $v<y$, so $x \leftarrow y$ and $y \leftarrow z$ cannot be both in $\mathcal{M}$, because at least one of them would not be the steepest.

(2) $\mathcal{M}$ consists of isomorphisms: This is by the assumption on invertible weights.

(3) $\mathcal{M}$ induces no cycles: Every zig-zag $u \overset{x}{\underset{y}{\lessdot\gtrdot}}$ in which $u \leftarrow x \in \mathcal{M}$ implies that $x<y$. Thus every zig-zag strictly increases the initial vertex, so a zig-zag path cannot end at its starting point, because it would imply $y>y$.

(b) Let $\mathcal{M}$ be an arbitrary Morse matching on Γ_{C_*}. It suffices to find for every k a total order $\leq_k$ on I_k such that the edges in $\mathcal{M}$ are the steepest.

By assumption, the digraph $\Gamma_{C_*}^{\mathcal{M}}$ is acyclic, so it can be viewed as a poset: $u \preceq v$ iff there exists a directed path from u to v. Let $\preceq_k$ be the restriction of $\preceq$ to I_k. By construction, members of $\mathcal{M}$ are steepest w.r.t. $\preceq_k$. Since every finite partial order $\preceq_k$ can be extended to a linear order $\leq_k$, we conclude that $\mathcal{M} \subseteq \mathcal{M}_{\leq}$.

If $\mathcal{M}$ is also maximal, then $\mathcal{M} \subseteq \mathcal{M}_{\leq}$ for some $\leq$ implies $\mathcal{M} = \mathcal{M}_{\leq}$. □

Note that if ∂_k was infinite, with only diagonal and first supdiagonal entries which were all units, then $\mathcal{M}_{\leq}$ would not satisfy the part of condition (3) about infinite paths. Thus infinite steepness pairings are not always Morse matchings.

2.3. **Matrix interpretation.** If ∂_k are given as finite matrices, we may assume that $I_k = \{1, \dots, |I_k|\}$ and $\leq_k$ is the usual total order on $\mathbb{N}$, so the first row/column index is the smallest and the last one is the largest. Our $\mathcal{M}_{\leq}$ consists of those invertible matrix entries that have only zeros left in its row and below in its column,

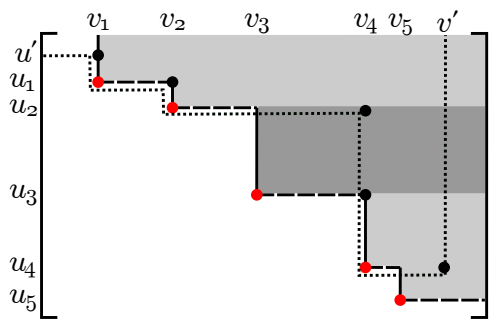
$$\mathcal{M}_{\leq} = \left\{ \begin{array}{c} \\ u \\ \\ \\ \end{array} \overset{v}{\begin{bmatrix} 0 \cdots 0 & w \\ & 0 \\ & \vdots \\ & 0 \end{bmatrix}} ;\ w \in R^{\times} \right\}. \tag{6}$$

Thus $\mathcal{M}$ is largest when the matrices ∂_k are in 'block upper-triangular' form. The set of critical vertices is $I'_k = I_k \setminus (\{\text{column indices of } \mathcal{M}_k\} \cup \{\text{row indices of } \mathcal{M}_{k+1}\})$.

Lemma 2.4. *Given $\mathcal{M}_{\leq}$, the u'-row r' of ∂'_k is obtained from the u'-row r of ∂_k in the following way. While $V := \{\text{nonzero positions of } r\} \cap I_k^+$ is nonempty, for $v = \min V$, $u \overset{w}{\leftarrow} v \in \mathcal{M}$, $u' \overset{w'}{\leftarrow} v \in \Gamma_{C_*}$, add $\frac{-w'}{w} \cdot (u\text{-row of } \partial_k)$ to r. Then remove all I_k^+ entries (which are zero) and I_{k+1}^--entries from r to get the (shorter) row r'.*

Hence AMT gives the same result as Gaussian elimination with several pivots. However, with AMT we have to compute $\partial'(v)$ only for critical vertices v (there may be very few of them, sometimes none at all). Also, AMT calculates the new complex recursively using (4), which seems to be much faster than row operations.

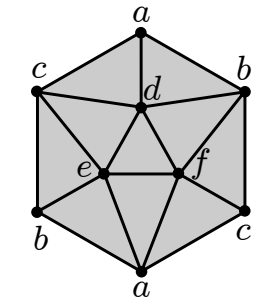
Proof. For the sake of clarity, we assume ∂_k have block upper-triangular form, though the arguments work for any $\leq$. This is depicted by the image below: white area are zeros, full (vertical) lines are nonzeros, dashed (horizontal) lines and gray area are zeros and nonzeros, $\mathcal{M}_{\leq}$ consists of those red entries that are invertible, black bullets are nonzero entries, the dotted line is a path γ from v' to u' in ∂_k.

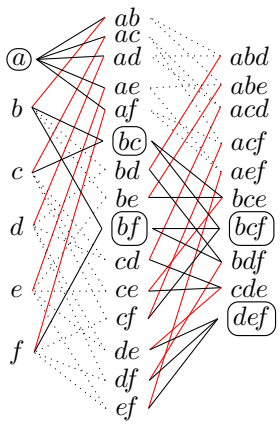


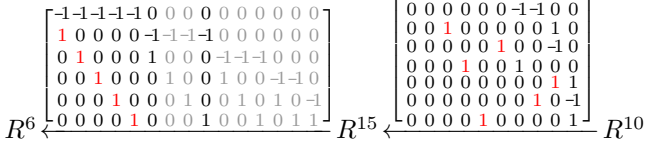
(7)

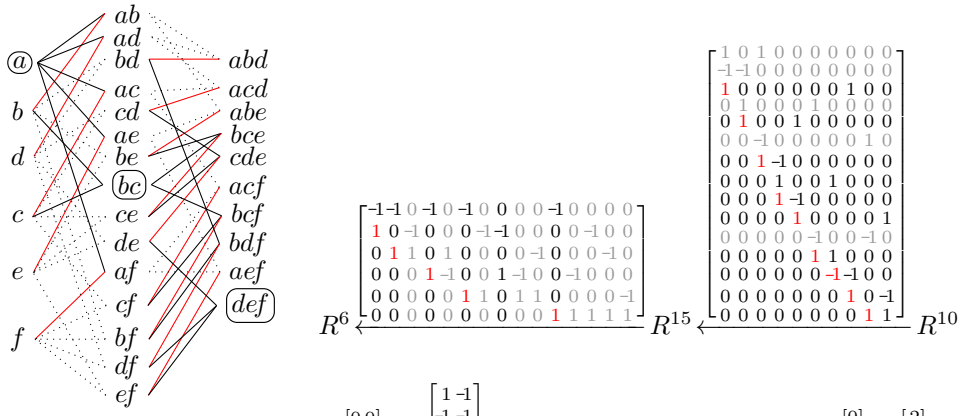
By the construction of $\Gamma^{\mathcal{M}}_{C_*}$, a path $\gamma \in \Gamma_{v',u'}$ in ∂_k consists of horizontal moves from black to red bullets, and vertical moves from red to black bullets. Let I_k^+ consist of $v_1, v_2, \ldots$. The u'-row of ∂_k corresponds to paths in $\Gamma_{\ldots,u'}$ of length 1. Adding $\frac{-w_1'}{w_1}\cdot(u_1\text{-row of } \partial_k)$ means replacing the path $u' \leftarrow v_1$ by all paths $u' \leftarrow v_1 \rightarrow u_1 \leftarrow v$ of length 3. Adding $\frac{-w_2'}{w_2}\cdot(u_2\text{-row of } \partial_k)$ means replacing the latter by all paths $u' \leftarrow v_1 \rightarrow u_1 \leftarrow v_2 \rightarrow u_2 \leftarrow v$ of length 5. This process ends when all $v_1, v_2, \ldots$-entries of r are zero (i.e. we used up all $\mathcal{M}_{\leq}$ to create paths), and then taking only the I_k'-entries (throwing away paths that do not begin in a critical v') produces r'. □

2.5. **Examples.** • Consider the triangulation Δ below for the real projective plane. Let C_* be the chain complex for simplicial homology of Δ. The digraph Γ_{C_*} (=Hasse diagram of Δ) with the lexicographic order on vertices is pictured below left, and Γ_{C_*} with a different total order is shown below that. The red edges are members of the steepness matching, the circled vertices are the critical simplices, and the dotted edges / gray entries can be deleted by remark 1.4 (c).









By the first ordering, $C'_* = R \xleftarrow{[0\,0]} R^2 \xleftarrow{\left[\begin{smallmatrix}1 & -1\\ -1 & -1\end{smallmatrix}\right]} R^2$, but by the second, $C'_* = R \xleftarrow{[0]} R \xleftarrow{[-2]} R$, with 3 critical simplices and 4 zig-zag paths, which is optimal.

• In the example 1.3, our $\mathcal{M}$ is the steepness matching.

• Sköldberg's matching in [6, p.48], which computes the homology of Heisenberg Lie algebras, is $\mathcal{M}_\leq$ w.r.t. lexicographic ordering for $x_1 < \ldots < x_n < y_1 < \ldots < y_n$.

• Forman's matching in [1, p.17], which computes the homology of the simplicial complex on $E(K_n)$ of non-connected subgraphs of the full graph K_n, is $\mathcal{M}_\leq$ w.r.t. the lexicographic order using $12 < 13 < 23 < 14 < 24 < 34 < 15 < \ldots < 45 < \ldots$.

• Kozlov's matching in [4, p.193], which computes the homology of the simplicial complex on $V(P_n)$ of all independent sets in the path graph P_n, is $\mathcal{M}_\leq$ w.r.t. the lexicographic order for $0 < 3 < 6 < 9 < \ldots < 2 < 5 < 8 < 11 < \ldots < 1 < 4 < 7 < 10 < \ldots$. ◊

2.6. **Remarks.** Consider the following chain complexes of $\mathbb{Z}$-modules:

$$(a)\ 0 \leftarrow \mathbb{Z}^2 \xleftarrow{\left[\begin{smallmatrix}3 & 4\\ 2 & 3\end{smallmatrix}\right]} \mathbb{Z}^2 \leftarrow 0 \qquad (b)\ 0 \leftarrow \mathbb{Z}^n \xleftarrow{\left[\begin{smallmatrix}1 & & & 1\\ & \ddots & \iddots & \\ 1 & 1 & & \\ 1 & & & \end{smallmatrix}\right]} \mathbb{Z}^n \leftarrow 0 \qquad (c)\ 0 \leftarrow \mathbb{Z}^2 \xleftarrow{\left[\begin{smallmatrix}0 & 1\\ 1 & 1\end{smallmatrix}\right]} \mathbb{Z}^2 \leftarrow 0$$

• The boundary matrices might have all entries non-invertible (so the steepness matching is empty w.r.t. any ordering), yet the chain complex can still be contractible. E.g. (a) is contractible (since $\det \partial_1 = 1 \in \mathbb{Z}^\times$), but $\mathcal{M}_\leq = \emptyset$.

• The boundary matrices may have all nonzero entries invertible and be very sparse, but if $\leq$ is badly chosen, the steepness matching can still be very small. E.g. for (b) our $\mathcal{M} = \{n \leftarrow 1\}$ has one edge. However, if we pick the same order on rows and reverse order on columns, ∂_1 becomes upper-triangular and $\mathcal{M} = \{i \leftarrow i;\ i \in [n]\}$.

• The inclusion in Lemma 2.2 (b) is not an equality. E.g. for (c) the steepness pairing $\mathcal{M} = \{2 \leftarrow 1\}$ is not maximal. It is strictly contained in the Morse Matching $\mathcal{M} = \{2 \leftarrow 1, 1 \leftarrow 2\}$, which is the steepness pairing when the rows are switched.

• The boundary matrices may have all nonzero entries invertible, but the steepness matching could be small no matter the choice of the ordering. E.g. for the complex $0 \leftarrow \mathbb{Z}^m \xleftarrow{\partial} \mathbb{Z}^n \leftarrow 0$ in which every entry of ∂ is ± 1, any choice of $\leq$ gives $\mathcal{M} = \{m \leftarrow 1\}$, though admittedly this matrix ∂ is not sparse.

• Needless to say, for most chain complexes in practice, e.g. those coming from simplicial complexes (Poincaré), (semi)groups (Eilenberg-MacLane), associative algebras (Hochschild), Lie algebras (Chevalley), knots and links (Khovanov), etc., the matrices are usually very sparse, with just entries 0 and ± 1. Hence the steepness matching is very large and useful, as it kills the majority of basis elements. ◊

3. Choice of ordering

When ∂_k are given as finite matrices, the bases and orders are already determined: $I_k = \{1, \dots, |I_k|\}$ and $\leq_k$=(usual total order on $\mathbb{N}$). We can get any other order by permuting the rows and/or columns of boundary matrices.

Even in very sparse complexes, bad orders lead to small matchings and therefore slow computation (as seen in remark 2.6 (b)), or they lead to increase in matrix densities and therefore memory overflow (as seen in the following example).

3.1. **Examples.** Consider the 3 orderings of the same chain complex:

$$C_*: \quad (a)\ 0 \leftarrow \mathbb{Z}^m \xleftarrow{\begin{bmatrix} 1 & & \\ \vdots & & 0 \\ 1 & & \\ -1 & 1 \cdots & 1 \end{bmatrix}} \mathbb{Z}^n \leftarrow 0 \qquad (b)\ 0 \leftarrow \mathbb{Z}^m \xleftarrow{\begin{bmatrix} 1 & 1 \cdots & 1 \\ 1 & & \\ \vdots & & 0 \\ -1 & & \end{bmatrix}} \mathbb{Z}^n \leftarrow 0 \qquad (c)\ 0 \leftarrow \mathbb{Z}^m \xleftarrow{\begin{bmatrix} -1 \cdots & 1 & 1 \\ & & 1 \\ 0 & & \vdots \\ & & -1 \end{bmatrix}} \mathbb{Z}^n \leftarrow 0$$

$$C'_*: \quad 0 \leftarrow \mathbb{Z}^{m-1} \xleftarrow{\begin{bmatrix} -1 & \cdots & -1 \\ \vdots & & \vdots \\ -1 & \cdots & -1 \end{bmatrix}} \mathbb{Z}^{n-1} \leftarrow 0 \qquad 0 \leftarrow \mathbb{Z}^{m-1} \xleftarrow{\begin{bmatrix} 1 \cdots 1 \\ 0 \end{bmatrix}} \mathbb{Z}^{n-1} \leftarrow 0 \qquad 0 \leftarrow \mathbb{Z}^{m-2} \xleftarrow{\begin{bmatrix} 0 \end{bmatrix}} \mathbb{Z}^{n-2} \leftarrow 0$$

If in (a) we switch the first and last row, we get (b); if we also switch the first and last column, we get (c). The starting matrix density is $\frac{m+n-1}{mn}$. After computing C'_*, in (a) the density increased to 1, in (b) it decreased to $\frac{1}{m-1}$, and in (c) it became 0. $\Diamond$

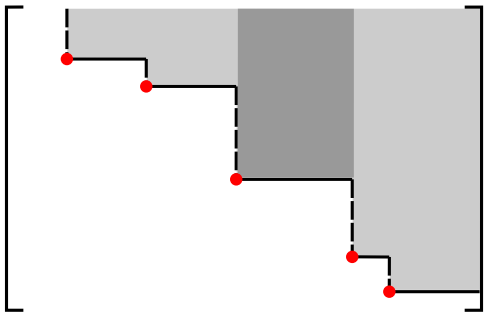
3.2. **Algorithm.** To meaningfully reduce C_*, we must create as many red entries like in (6), as few situations like in example 3.1 (a), and as few zig-zag paths like in (7) (i.e. new entries in C'_*) as possible. This is achieved in two steps:

For $k = 1, \dots, N$, permute the columns of ∂_k (and thus rows of ∂_{k+1}, with the same permutation), by lexicographically comparing, for every column index v, the tuple $(c_1, c_2, c_3, c_4)(v)$. Here we used:
$c_1 = 0$ if the v-column contains a unit entry and 1 otherwise,
$c_2 =$ position of the last nonzero entry in the v-column,
$c_3 =$ density of the v-column,
$c_4 = 0$ if the last nonzero entry in the v-column is a unit and 1 otherwise. (8)

For $k = N-a, N-a-b, N-a-2b, N-a-3b, \dots$ where $a \in \{0, 1\}$, permute the rows of ∂_k (and thus columns of ∂_{k-1}, with the same permutation), by lexicographically comparing, for every row index u, the tuple $(r_1, r_2, r_3, r_4)(u)$. Here we used:
$r_1 = 1$ if the u-row contains a unit entry and 0 otherwise,
$r_2 =$ position of the first nonzero entry in the u-row,
$r_3 =$ -(density of the u-row),
$r_4 = 1$ if the first nonzero entry in the u-row is a unit and 0 otherwise. (9)

Performing (9) for $a = 0$ and $b = 1$ by comparing just r_2 produces a complex, in which every matrix has the form (7). Similarly, applying (8) by comparing just c_2 produces a complex, in which every matrix has the form (10) below (white area are zeros, gray area and dashed lines are any entries, full lines are nonzeros, invertible red entries are members of $\mathcal{M}_{\leq}$). Further comparing $r_i, \dots$ (resp. $c_j, \dots$) means permuting the rows (columns) in the dark gray area of (7) (resp. (10)).

As an illustration, notice that in examples 2.5, the second ordering of the chain complex for the projective plane has the form (7).



(10)

If we apply (8) and then (9) for $a=0$ and $b=1$, then (9) will not be reordering matrices of the form (10), since it changes ∂_k as well as ∂_{k-1}. However, if we apply (8) and then (9) for $a=\{0,1\}$ and $b=2$, then (9) receives as input the form (10).

If we want to maximize the size of $\mathcal{M}_{\leq}$, we order w.r.t. (c_1,c_2,c_4) and (r_1,r_2,r_4). If we want to minimize the density of matrices in C'_*, we order w.r.t. (c_1,c_2,c_3) and (r_1,r_2,r_3). Note however, that this does not always produce the desired result, since (9) orders only half of the matrices, and may mess up the other half.

3.3. **Remark.** In my experience, if the matrices of C_* come from a typical homology theory (e.g. from simplicial complexes, groups, associative or Lie algebras, etc.), then the default lexicographic order is often already very effective. Without spending time on reordering, AMT produces a larger chain complex, but faster.

It would be interesting if one could derive some nice characterization that tells for which $\leq$ is our stepness Morse matching $\mathcal{M}_{\leq}$ maximal.

4. Noninvertible entries

Let R be a PID and $p\in R$ a prime. Hence for every k, we have $H_k(C_*;R)\cong \bigoplus_{t\in T_k} R/(t)$ for a unique finite multiset T_k of zeros and prime powers. Then over the localized ring, $H_k(C_*;R_{(p)})\cong \bigoplus_{t\in T_k\cap Rp} R_{(p)}/(t)$ gives p-torsion and free part. Computing $H_k(C_*;R_{(p)})$ for all (relevant) primes p then produces $H_k(C_*;R)$.

The only noninvertible elements in the subring $R_{(p)}=\{\frac{a}{b}\in Q(R); b\notin Rp\}$ of the field of fractions of R are $R^{\times}_{(p)}=\{\frac{a}{b}\in Q(R); a\in Rp, b\notin Rp\}$, the multiples of p.

4.1. **Algorithm.** As long as a chain complex contains a unit entry $u\overset{w}{\leftarrow}v$, there is a Morse matching $\mathcal{M}$ containing that entry (e.g. $\mathcal{M}=\{u\overset{w}{\leftarrow}v\}$ suffices).

Let $C_*^{[0]}:=C_*$. We have a sequence (5), where $C_*^{(r)}$ contains only nonunit entries (i.e. multiples of p). Let p^{a_1} be the largest power of p that divides all nonzero entries of $C_*^{(r)}$ (if there are any). Let $C_*^{[1]}$ be the complex $C_*^{(r)}$ in which all entries are divided by p^{a_1}; then $C_*^{[1]}$ contains unit entries. Thus we can apply (5) again, get a complex with all entries nonunits, divide it by the largest divisor p^{a_2}, and obtain a complex $C_*^{[2]}$ that contains unit entries. Doing the same again produces $C_*^{[3]}$ and so on. Denote by $m_{k,i}\in\mathbb{N}$ the sum of $|\mathcal{M}_k|$, where $\mathcal{M}$ are all the Morse matchings, used when applying (5) to $C_*^{[i-1]}$ to create $C_*^{[i]}$.

Lemma 4.2. *The above sequence of complexes terminates at some $C_*^{[t]}$ in which all matrices are zero. Then the p-torsion of $H_k(C_*;R)$ is $\bigoplus_{1\leq i<t}(R/Rp^{a_1+\ldots+a_i})^{m_{k+1,i}}$.*

Proof. The sequence terminates, since every $e \in \mathcal{M}_k$ reduces the finite rank of C_k and C_{k-1} by 1. Via using 1.2 (b) repetitively, the above procedure reconstructs the SNF of C_* over $R_{(p)}$. Indeed, by constructing $C_*^{[1]}, \dots, C_*^{[t]}$, we successively change the bases of all C_k, such that the boundary matrices have the form

$$\begin{aligned}
\partial_k &= \operatorname{diag}(\mathrm{id}_{m_{k,0}}, 0, p^{a_1}\partial_k^{[1]}),\\
\partial_k &= \operatorname{diag}(\mathrm{id}_{m_{k,0}}, 0, p^{a_1}\mathrm{id}_{m_{k,1}}, 0, p^{a_1+a_2}\partial_k^{[2]}),\\
\partial_k &= \operatorname{diag}(\mathrm{id}_{m_{k,0}}, 0, p^{a_1}\mathrm{id}_{m_{k,1}}, 0, p^{a_1+a_2}\mathrm{id}_{m_{k,2}}, 0, p^{a_1+a_2+a_3}\partial_k^{[3]}),\\
&\vdots\\
\partial_k &= \operatorname{diag}(\mathrm{id}_{m_{k,0}}, 0, p^{a_1}\mathrm{id}_{m_{k,1}}, 0, \dots, p^{a_1+\dots+a_{t-1}}\mathrm{id}_{m_{k,t-1}}, 0, p^{a_1+\dots+a_t}0),
\end{aligned}$$

where the last zero is the trivial map $C_{k-1}^{[t]} \leftarrow C_k^{[t]}$ that determines the free part. □

4.3. **Remark.** In my experience, if the matrices of C_* over $\mathbb{Z}$ have only entries 0 and ± 1 and if the torsion part of H_*C_* is small (relative to the ranks of C_k), then applying AMT produces C'_* in which most entries are still 0 and ± 1. Consequently, for many complexes it is effective to work over $\mathbb{Z}$ directly, not over $\mathbb{Z}_{(p)}$ for all p.

For complexes C_* with large torsion part in homology, it would really help if one could efficiently compute some finite (reasonably small) set of primes P, such that H_*C_* contains no p-torsion for $p \in \mathbb{N} \setminus P$.

5. Computer Implementation

5.1. **Algorithm.** We have a procedure to compute everything from 1.2.

Input: Matrices $\partial_* = (\partial_k)_{k=1}^N$ of the complex C_* over R. Let $f_* := (\mathrm{id}_k)_{k=0}^N$.
While at least one matrix ∂_k contains at least one invertible entry:

(a) Permute the basis elements (columns/rows of all ∂_k), so that as many invertible entries have zeros left and below them, e.g. like (8) and (9).
(b) Compute $\mathcal{M}_\le$ using (6). For all k, delete the I_{k-1}^--columns of ∂_{k-1} and I_k^+-rows of ∂_{k+1}, which are unnecessary by remark 1.4 (c).
(c) Compute ∂'_* and f'_* w.r.t. $\mathcal{M}_\le$ using formulas (4) and 1.4 (b) recursively (finding directed paths in a graph).
(d) Let $\partial_* := \partial'_*$ and $f_k := f_k f'_k$ (matrix multiplication).

Return: ∂'_* (the reduced chain complex) and f_* (the h-equivalence $C'_* \to C_*$).

Step (a) is optional: if we use it, then this algorithm returns ∂'_* in which all entries are nonunits. Also, there are many ways to perform (a), so it leaves a lot of room for optimization. If R is a field, the above process returns $\partial'_* = 0$; then the width of ∂'_k equals $\dim H_k C_*$ and the columns of f_k are the generators of $H_k C_*$, by (5).

When ∂_k are sparse and come from common homology theories, the default $\le$ is often quite good for computation, and additional reordering improves the calculation a little, but spends unnecessary time. Thus we often skip (a), unless we work over a local PID, which requires killing all unit entries to get torsion.

The part of (b) about column/row deletion is optional: if we skip it, the calculation will just require more time and memory. The part of (c) and (d) about f_* is optional: if we only want the isomorphism type of $H_k C_*$, we compute just ∂'_*.

5.2. **Explicit code.** Below is a rudimentary code in Mathematica that, given C_* over $\mathbb{Z}$ or $\mathbb{Q}$ or $\mathbb{Z}_p$ for prime p, changes C_* with the command `reduceChCx` (using the steepness matching) to a smaller h-equivalent complex according to AMT 1.2.

It is to be copied into MATHEMATICA and inspected there, hence the small font here.

```
$HistoryLength=0; SetAttributes[{steepM,orderC,orderR,reduceChCx},HoldAll];
cols[a_] := With[{l=GatherBy[Sort@Transpose@Join[Reverse@Transpose@a["NonzeroPositions"],{a["NonzeroValues"]}],
   First]}, AssociationThread[Map[First,l,2],Map[Rest,l,{2}]]];
orderC[bdrs_,dims_] := Module[{dim=Length@bdrs,cls,kys,v,c1,c2,c3,c4,pmt,pc,pr}, Do[
   If[dims[[k]]*dims[[k+1]]==0 || bdrs[[k]]["Density"]==0, Continue[]]; cls=cols@bdrs[[k]]; v=Values@cls;
   c1=If[MemberQ[Last/@#^2,1],0,1]&/@v; c2=Length/@v; c3=#[[-1,1]]&/@v; c4=If[#[[-1,2]]^2==1,0,1]&/@v;
   kys=Keys@cls; Clear[cls,v]; pmt=First/@SortBy[Transpose@{kys,c1,c3,c4},Rest]; pmt=AssociationThread[pmt,kys];
   pc=({u_,v_}->w_):>({u,pmt[v]}->w);   pr=({u_,v_}->w_):>({Lookup[pmt,u,u],v}->w);
   bdrs[[k]]  =SparseArray[ArrayRules[bdrs[[k  ]]][[1;;-2]] /.pc,dims[[k;;k+1]]];   If[k<dim,
   bdrs[[k+1]]=SparseArray[ArrayRules[bdrs[[k+1]]][[1;;-2]] /.pr,dims[[k+1;;k+2]]];], {k,dim}]];
orderR[bdrs_,dims_,a_,b_] := Module[{dim=Length@bdrs,rws,kys,v,r1,r2,r3,r4,pmt,pr,pc}, Do[
   If[dims[[k]]*dims[[k+1]]==0 || bdrs[[k]]["Density"]==0, Continue[]]; rws=cols@Transpose@bdrs[[k]]; v=Values@rws;
   r1=If[MemberQ[Last/@#^2,1],1,0]&/@v; r2=-Length/@v; r3=#[[1,1]]&/@v; r4=If[#[[1,2]]^2==1,1,0]&/@v;
   kys=Keys@rws; Clear[rws,v]; pmt=First/@SortBy[Transpose@{kys,r1,r3,r4},Rest]; pmt=AssociationThread[pmt,kys];
   pr=({u_,v_}->w_):>({pmt[u],v}->w); pc=({u_,v_}->w_):>({u,Lookup[pmt,v,v]}->w);
   bdrs[[k]]  = SparseArray[ArrayRules[bdrs[[k  ]]][[1;;-2]] /.pr, dims[[k;;k+1]]]; If[1<k,
   bdrs[[k-1]]= SparseArray[ArrayRules[bdrs[[k-1]]][[1;;-2]] /.pc, dims[[k-1;;k]]];], {k,dim-a,1,-b}]];
steepM[rws_,cls_,p_:"Z"] := Module[{v,w}, Association@DeleteCases[Table[{v,w}=rws[u][[1]]; If[cls[v][[-1]]=={u,w}
   && Which[p=="Z",w^2==1,p==0,w!=0,True,Mod[w,p]!=0], rws[u]=rws[u][[2;;-1]]; cls[v]=cls[v][[1;;-2]];
   u->{v,Which[p=="Z",-w,p==0,-1/w,True,ModularInverse[-w,p]]},""], {u,Keys@rws}],_String]];
zzsb[cls_,Mk_,Ik_,zz_List] := Flatten[Table[With[{e=Lookup[Mk,l[[1]],{}]}, Which[ KeyExistsQ[Ik,l[[1]]], {l},
   e=={}, {}, True, zzsb[cls,Mk,Ik,cls[e[[1]]]/.{u_Integer,w_}:>{u,w*e[[2]]*l[[2]]}]]], {l,zz}],1];
zzsb[cls_,Mk_,Ik_,v_Integer] := {#[[1,1]],v}->Plus@@(Last/@#)&/@ GatherBy[zzsb[cls,Mk,Ik,Lookup[cls,v,{}]],First];
zzsf[cls_,Mk_,v_Integer] := {#[[1]],v}->#[[2]]&/@ zzsf[cls,Mk,{},{{v,1}}];   zzsf[cls_,Mk_,res_,{}] := res;
zzsf[cls_,Mk_,res_,tmp_] := zzsf[cls,Mk,Join[res,tmp], DeleteCases[Flatten[
   Table[{#1,#2*u[[2]]*v[[2]]}& @@ Lookup[Mk,u[[1]],{0,0}], {v,tmp},{u,Lookup[cls,v[[1]],{}]}],1],{0,0}]];
reduceChCx[bdrs_,dims_,p_,ord_:{-1,0},heq_:{}] :=
   Module[{dim=Length@bdrs,Mk={},Mkk,Ik={},Ikk,rwsk={},clsk={},rwskk,clskk,ik,ikk,mk,entries,bdr},
   If[p!="Z" && p!=0, bdrs=Mod[bdrs,p]];  bdrs=Table[SparseArray[bdrs[[k]],dims[[k;;k+1]]],{k,dim}];
   If[ord[[1]]>=0, orderC[bdrs,dims]; orderR[bdrs,dims,ord[[1]],ord[[2]]]];   Do[
   If[ k==dim || dims[[k+1]]*dims[[k+2]]==0 || bdrs[[k+1]]["Density"]==0, {rwskk,clskk,Mkk}={{},{},{}};,
   {rwskk,clskk}=cols/@{Transpose@bdrs[[k+1]],bdrs[[k+1]]};   Mk=steepM[rwskk,clskk,p];  rwskk={};];
   Ikk=Complement[Range@dims[[k+1]],First/@Values[Mk],Keys@Mkk];   Ikk=AssociationThread[Ikk,Range@Length@Ikk];
   {ik,ikk,mk}=Length/@{Ik,Ikk,Mk};   entries=Flatten[If[mk*ik==0,{},Table[zzsb[clsk,Mk,Ik,v],{v,Keys@Ikk}]],1];
   bdr=If[ik*ikk>0&&mk==0,bdrs[[k,Keys@Ik,Keys@Ikk]],SparseArray[entries/.({u_,v_}->w_):>({Ik[u],Ikk[v]}->w),{ik,ikk}]];
   If[heq!={}, If[p!="Z"&&p!=0&&ikk!=0, heq[[k+1]]=SparseArray@Mod[heq[[k+1]],p]];
   entries=Flatten[Table[zzsf[clsk,Mk,v],{v,Keys@Ikk}],1];   heq[[k+1]]=If[ikk==0,SparseArray[{},{dims[[k+1]],0}],
   heq[[k+1]].SparseArray[entries/.({vv_,v_}->w_):>({vv,Ikk[v]}->w),{dims[[k+1]],ikk}]]];
   If[1<=k, bdrs[[k]]=bdr]; dims[[k+1]]=ikk; Ik=Ikk; Mk=Mkk; clsk=clskk; Clear[entries,Ikk,Mkk,clskk];, {k,0,dim}]];
```

In the code, we used the notation `bdrs`$=\partial_1,\dots,\partial_N$, `dim`$=N$, `dims`$=|I_0|,\dots,|I_N|$, `p`$=R$ (`"Z"`,`0`,`p` for $\mathbb{Z},\mathbb{Q},\mathbb{Z}_p$), `orderC`$=(8)$, `orderR`$=(9)$, `heq`$=f_0,\dots,f_N$, `Mk`$=\mathcal{M}_k$, `Mkk`$=\mathcal{M}_{k+1}$, `Ik`$=I'_{k-1}$, `Ikk`$=I'_k$, `bdr`$=\partial'_k$, `zzsb`=zig-zag paths for ∂'_*, `zzsf`=zig-zag paths for f_*, `rwsk,clsk,rwskk,clskk`=rows/columns of ∂_k and ∂_{k+1}.

5.3. **Usage.** The example 1.3 is constructed with the code below.

```
time=0.0; dims={4,6,12,8}; heq=Table[SparseArray[{i_,i_}->1,{r,r}],{r,dims}]; bdrs={
SparseArray[{{2,1}->1,{3,1}->1,{4,2}->1,{2,3}->-1,{3,3}->-1,{4,4}->-1,{2,5}->1,{3,5}->1,{4,6}->1},{4,6}],
SparseArray[{{1,1}->-1,{3,1}->-1,{2,2}->-1,{4,2}->-1,{2,3}->-1,{4,3}->-1,{1,5}->1,{5,5}->-1,{2,6}->1,
   {6,6}->-1,{2,7}->1,{6,7}->-1,{3,9}->1,{5,9}->1,{4,10}->1,{6,10}->1,{4,11}->1,{6,11}->1},{6,12}],
SparseArray[{{1,1}->1,{5,1}->1,{9,1}->1,{3,2}->1,{6,2}->1,{10,2}->1,{2,3}->1,{7,3}->1,{10,3}->1,{4,4}->1,
   {8,4}->1,{3,5}->1,{7,5}->1,{11,5}->1,{8,6}->1,{12,6}->1,{4,7}->1,{12,7}->1},{12,8}]}; MatrixForm/@bdrs
```

To use the reduction algorithm over $\mathbb{Z}$ once (without reordering), we run either the first command (it does not compute f_*) or second command (it computes f_*, whose columns form the basis of homology over any R when $\partial'_*=0$) below.

```
reduceChCx[bdrs,dims,time,"Z",{-1,2},{}]        (11)
reduceChCx[bdrs,dims,time,"Z",{-1,2},heq]       (12)
```

To see C'_* and f_*, run `MatrixForm/@bdrs` and `MatrixForm/@heq`. By running (11) twice on the example 1.3, the obtained result is $C'_*\colon R^2 \xleftarrow{0} R^0 \xleftarrow{0} R^2 \xleftarrow{\left[\begin{smallmatrix}0&0\\2&0\end{smallmatrix}\right]} R^2$.

To compute over $\mathbb{Q}$ or $\mathbb{Z}_p$, replace `"Z"` by `0` or `p`. To obtain homology over a field, run (11) or (12) until the matrices `bdrs` are zero. With the 5th argument `{a,b}`, if `a`<0 the algorithm does not reorder, but if `a`≥ 0, it permutes the columns of $\partial_1,\dots,\partial_N$ and rows of $\partial_{N-a},\partial_{N-a-b},\partial_{N-a-2b},\partial_{N-a-3b},\dots$, as in (8) and (9). By using `{0,2}` and `{1,2}` alternatingly, the end result ∂'_* has very few unit entries.

To see the effects of reordering, enter again `bdrs` and `dims` above, and then run:

```
MatrixForm/@bdrs
orderC[bdrs,dims]; MatrixForm/@bdrs
orderR[bdrs,dims,0,1]; MatrixForm/@bdrs
```

If we run (12) twice on the complex from 2.5 over $\mathbb{Z}_2$, we see that $H_*C_* \cong \mathbb{Z}_2, \mathbb{Z}_2, \mathbb{Z}_2$ are generated by a, $ab+ac+bc$, $abd+abe+acd+acf+aef+bce+bcf+bdf+cde+def$.

5.4. **Performance.** All computations were carried out on a ZBook 17 G5 laptop with a 6-core 2.6–4.3GHz i7-8850H CPU, 64GB DDR4 2666MT/s RAM + 64GB swap (part of SSD disk, functioning as RAM), 512GB PCIe NVMe TLC SSD disk, and Linux Mint 19 operating system with MATHEMATICA 11.3 and SAGEMATH 8.1.

We use the abbreviations s=second(s), m=minute(s), h=hour(s), d=day(s). For each experiment, we report the CPU and RAM performance (total time and maximum memory, needed for the calculation). SAGEMATH calculations did not involve the computation of generators. MATHEMATICA calculations did not involve reordering of rows/columns; (11) and (12) were used several times.

• Let Δ_n be the simplicial complex of all anticliques in the n-th hypercube graph (i.e. the independence complex of Q_n). When $n=5$, the dimension is 15, the largest matrix is $\partial_6 \in \mathbb{Z}^{48\,960\times 54\,304}$ with 380 128 (nonzero) entries; the complex uses up 24MB. For computing $H_*(\Delta_5;\mathbb{Z})$, i.e. the input is a simplicial complex, not a chain complex, SAGEMATH needs 6h34m and 7.0GB. For computing $H_*C_*(\Delta_5;\mathbb{Z})$, SAGEMATH used up all 64GB + 10GB swap and didn't finish after 3d, so I aborted the calculation; (11) computes it in 10s with 164MB; (12) needs 12s with 164MB.

Let $\Delta_{m,n}$ be the $m\times n$-chessboard complex. When $m=n=8$, the dimension is 7, the largest matrices are $\partial_5 \in \mathbb{Z}^{376\,320\times 564\,480}$ with 3 386 880 entries and $\partial_6 \in \mathbb{Z}^{564\,480\times 322\,560}$ with 2 257 920 entries; the complex uses up 122MB and contains 8 378 944 entries. Applying (11) 8 times spends 51m and 7.0GB, the new chain complex contains 2 613 430 entries, the largest matrices are $\partial'_5 \in \mathbb{Z}^{42\times 14\,584}$ with 36 842 entries and $\partial'_6 \in \mathbb{Z}^{14\,584\times 8\,333}$ with 2 576 588 entries. The density of ∂_6 increased from $1\cdot 10^{-6}$ to $2\cdot 10^{-2}$; due to this, another application of (11) takes forever to finish, so I aborted. A similar story happens for the matching complex (i.e. the independence complex of the line graph of the complete graph K_n).

Let $\Delta_{m,n}$ be the independence complex of the m,n-Kneser graph. When $m=8$ and $n=3$, the dimension is 20, the largest matrix is $\partial_9 \in \mathbb{Z}^{5\,772\,760\times 5\,372\,920}$ with 53 729 200 entries; the complex uses up 4.7GB. For computing $H_*C_*(\Delta_{8,3};\mathbb{Z})$, (11) requires 23m and 23GB; (12) needs 28m and 23GB.

• Let $\mathfrak{g}_n$ be the n-th general linear Lie algebra over $\mathbb{Z}$ (it has dimension n^2) and C_* its Chevalley chain complex. When $n=5$, the largest matrix is $\partial_{13} \in \mathbb{Z}^{5\,200\,300\times 5\,200\,300}$ with 66 339 260 entries; the complex uses up 5.9GB. For computing $H_*C_*(\mathfrak{g}_5;\mathbb{Z}_2)$, SAGEMATH operated at a constant 26GB and didn't finish after 30d, so I aborted; (11) computes it in 1h41m with 40GB; (12) needs 2h13m and 40GB. For $H_*C_*(\mathfrak{g}_5;\mathbb{Q})$, (11) needs 1h3m and 42GB; (12) needs 1h18m and 42GB.

Let $\mathfrak{g}_n$ be the n-th Heisenberg Lie algebra over $\mathbb{Z}$ (it has dimension $2n+1$) and C_* its Chevalley chain complex. When $n=11$, the largest matrix is $\partial_{12} \in \mathbb{Z}^{1\,352\,078\times 1\,352\,078}$ with 2 032 316 entries; the complex uses up 222MB. For computing $H_*C_*(\mathfrak{g}_{11};\mathbb{Z}_2)$, SAGEMATH needs 3h8m and 6.7GB; (11) computes it in 3m19s with 1.7GB; (12) finishes in 4m41s with 1.7GB. For $H_*C_*(\mathfrak{g}_{11};\mathbb{Q})$, SAGEMATH needs 2d6h44m and 12.4GB; (11) computes it in 7m15s with 1.7GB; (12) finishes in 13m54s with 1.7GB. When $n=13$, the largest matrix is $\partial_{14} \in \mathbb{Z}^{20\,058\,300\times 20\,058\,300}$

with 35 154 028 entries; the complex uses up 4.5GB. For computing $H_*C_*(\mathfrak{g}_{13};\mathbb{Z}_2)$, (11) requires 1h17m and 28GB; (12) needs 1h44m and 28GB. For $H_*C_*(\mathfrak{g}_{13};\mathbb{Q})$, (11) requires 3h36m and 28GB; (12) needs 6h26m and 28GB.

6. Conclusion

As we have seen in 5.4, the implementation 5.2 of our new method (11) succeeds in computing the homology of very large complexes on just a laptop and with no reordering of the columns/rows of matrices. In most cases, it worked very fast and with small memory requirements (for instance, it computed the homology over $\mathbb{Z}$ of the independence complex of the $8,3$-Kneser graph, with dimension 20 and largest matrix of size 5 million $\times$ 5 million). However, in some instances (usually when the ranks of C_k are large, N is small, and H_*C_* contains a lot of torsion), in the new complex the density of matrices increases to the amount that AMT slows down too much to be efficient. I suspect that for such cases, a very specific reordering would remedy the problem, though how to choose such orders is an open problem.

7. Acknowledgment

I wish to thank Aleš Vavpetič for the idea of considering edges with the most upward tilt. Its generality really sparked my interest in this topic.

This research was supported by the Slovenian Research Agency program P1-0292 and grants J1-7025 and J1-8131.

References

[1] R. Forman, *A user's guide to discrete Morse theory*, Sém. Lothar. Combin. 48 (2002), Article B48c.

[2] M. Jöllenbeck, *Algebraic discrete Morse theory and applications to commutative algebra*, Thesis, (2005).

[3] D.N. Kozlov, *Discrete Morse theory for free chain complexes*, C. R. Acad. Sci. Paris **340** (2005), 867–872.

[4] D.N. Kozlov, *Combinatorial Algebraic Topology*, Springer, ACM 21 (2005).

[5] V. Nanda, D. Tamaki, K. Tanaka, *Discrete Morse theory and classifying spaces*, (2018), arXiv:1612.08429v2.

[6] E. Sköldberg, *Morse theory from an algebraic viewpoint*, Trans. Amer. Math. Soc. 358 (2006), no. **1**, 115–129.

Faculty of Mathematics and Physics, Department of Mathematics, University of Ljubljana, Slovenia

E-mail address: lampretl@gmail.com, leon.lampret@fmf.uni-lj.si