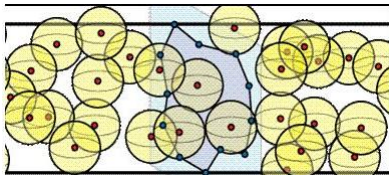


On some applications of the coreduction homology algorithm.

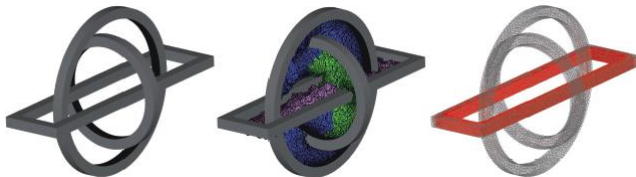
Paweł Dłotko
Institute of Computer Science, Jagiellonian University,
Kraków, Poland.

Two different stories, two different applications, one algorithm

- **Part 1:** The coverage problem in the sensor network.



- **Part 2:** Where Poincare meets Maxwell – how the co-reductins is used in the electrical engineering.

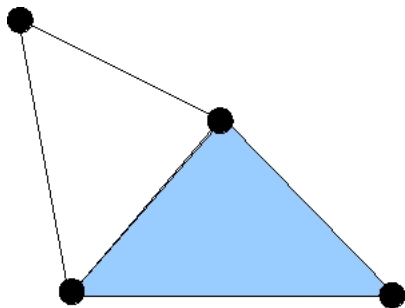


The Coreduction homology algorithm.

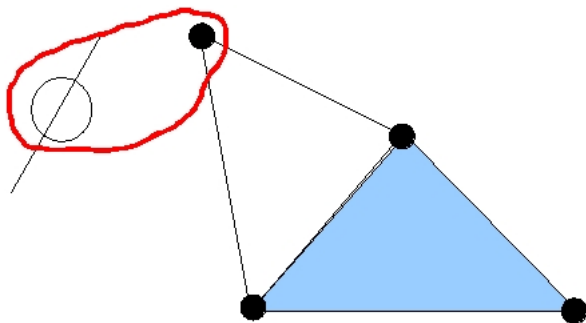
Part 1.

The Corecudtion homology algorithm.

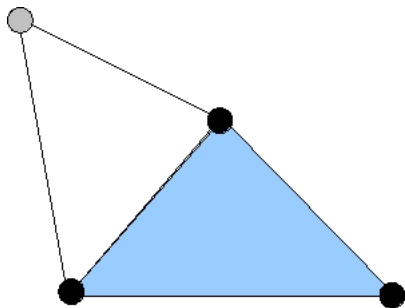
The Coreduction homology algorithm.



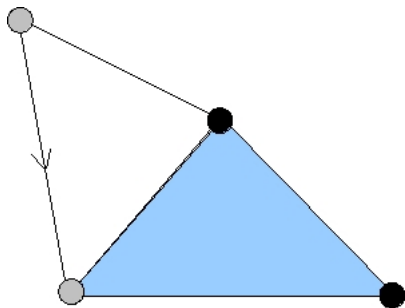
The Coreduction homology algorithm.



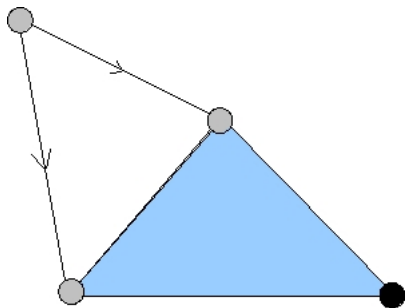
The Coreduction homology algorithm.



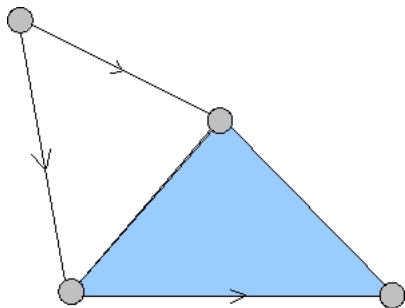
The Coreduction homology algorithm.



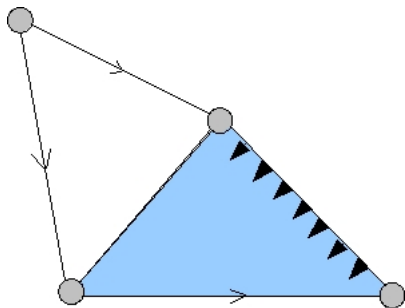
The Coreduction homology algorithm.



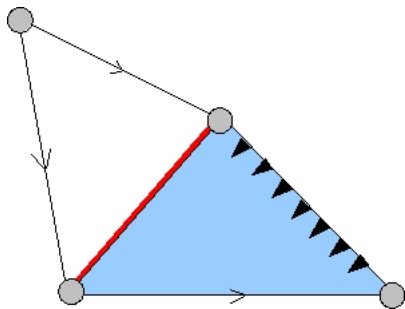
The Coreduction homology algorithm.



The Coreduction homology algorithm.



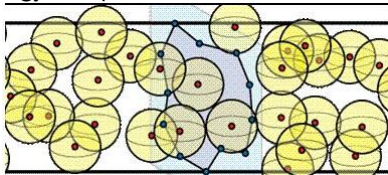
The Coreduction homology algorithm.



The coverage problem in the sensor network.

Part 2,

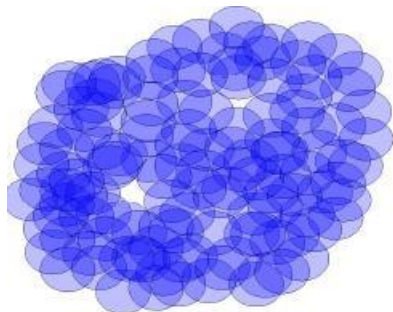
Homology computations on the sensor network,



(joint work with Robert Ghrist, Mateusz Juda and Marian Mrozek.)

Formulation of the problem.

- ▶ The importance of the coverage problems in the sensor networks (security, mine field sweeping, ad hoc communication networks).
- ▶ Systems with stationary nodes in \mathbb{R}^2 with radially symmetric coverage domains.



- ▶ Verify if the given set D such that $\partial D = F$ is covered by the disks of a given radius centered in the sensors.

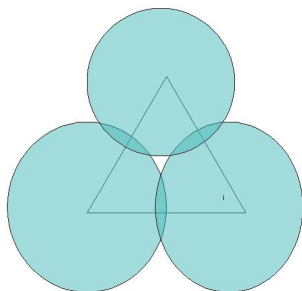
How the problem was solved so far?

- ▶ Computational geometry approach (Delaunay or Voronoi diagrams).
 - ▶ Precise geometry of the domain,
 - ▶ known localization of the nodes,
 - ▶ then we have exact solution.
- ▶ Probabilistic methods.
 - ▶ Big enough number of the sensors,
 - ▶ uniformly distributed on the region
 - ▶ implies coverage of the region with high probability (but no guarantee).
- ▶ What we wish to have?
 - ▶ Deterministic Yes/No or Yes/No/Do_Not_Know algorithm
 - ▶ arbitrary geometry of the region,
 - ▶ no assumptions about the placement of the sensors.

The Rips complexes

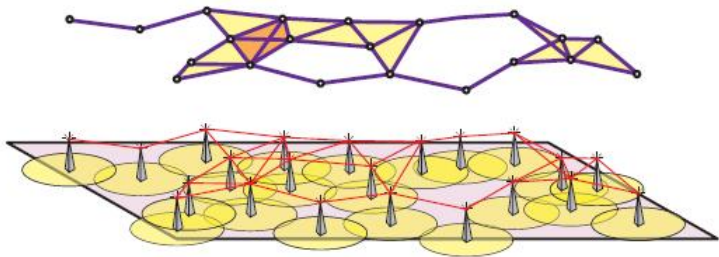
- ▶ Problem of computing the homology groups of a union of sets is classical (nerve of the cover).
- ▶ Let $\{X_1, \dots, X_n\}$ be the points where sensors are placed.
- ▶ Let r_{com} be the radius of communication of a sensor.
- ▶ The Rips complex associated with the set $X = \{X_1, \dots, X_n\}$ is a simplicial complex whose simplices are subsets $V \subset X$ such that for each $X, Y \in V$, $d(X, Y) \leq r_{com}$

The radius of a cover.



- ▶ Each sensor has some radius of coverage r_{cov} .
- ▶ Putting $r_{cov} \geq \sqrt{\frac{1}{3}}r_{com} \approx 0.577r_{com}$ enables us to "fill all the gaps".

Example of Rips complex (see V. Silva, R. Ghrist [3])



The homological criterion

- ▶ The coverage criterion based on the relative homology of the Rips complexes.

Theorem (Silva, Ghrist)

If there exists a homology class $[\alpha] \in H_2(\mathcal{R}_{r_{com}}, \mathcal{F}_{r_{com}})$ such that $\partial\alpha \neq 0$, then the considered region is covered by the balls of a radius r_{cov} centered in the given sensors.

- ▶ To use this (partial) criterion we need to compute the second relative homology generator...
- ▶ check its boundary...
- ▶ When this criterion holds, some sensors can be turned off (power-saving mode).

Rips complexes in the coverage problem

- ▶ $\mathcal{R}_{r_{com}}$ denotes the Rips complex based on the set of distributed sensors and radius r_{com} .
- ▶ The fence complex $\mathcal{F}_{r_{com}}$ is a subcomplex of $\mathcal{R}_{r_{com}}$.
- ▶ A geometric realization of the fence complex is assumed to be a Jordan curve in the plane.
- ▶ The Silva-Ghrist criterion is used to check the coverage of the interior of this curve.

What we should do to check this criterion

- ▶ The sensors need to be distributed in the given set.
- ▶ The fence sensors have to know that they belong to the fence.
- ▶ Each sensor has to get to know all its neighbors (it suffices to construct the Rips complex).
- ▶ Construct the Rips complex.
- ▶ Compute relative homology of the complex.
- ▶ Last two things can be done in theory after gathering all the information from all the sensors into one computer...
- ▶ ...which is not easy and not effective for a real-sized complexes.

So maybe we can try to make these computations in a distributed way on the sensors themselves?

- ▶ Doing algebraic homology computations in the distributed way is not easy.
- ▶ Why not to use the reduction procedures, that can be implemented in a purely combinatorial way?
- ▶ In some cases the (co)reductions can reduce the complex up to the homology generators.

What kind of sensors do we need to maintain the parallel computations?

- ▶ Each sensor is identified by unique integer label and equipped with a processor and radio transmitter.
- ▶ Each sensor can read the labels of its neighbors lying not farther than r_{com} (the communication range).
- ▶ The sensor can communicate with the sensors lying not farther then r_{com} .
- ▶ The sensor has the radius of detection r_{cov} such that
$$r_{cov} \geq \sqrt{\frac{1}{3}} r_{com}.$$
- ▶ The sensor does not have any metric information (neither coordinates, nor distance, nor direction to other sensors known).

What kind of sensors do we need to maintain the parallel computations?

- ▶ Each sensor can keep the list of simplices.
- ▶ Each simplex S from the list can keep the following information:
 - ▶ Labels of all the sensors that belong to S ,
 - ▶ labels of all the sensors that do not belong to S and are in the communication range with all sensors that belong to S ,
 - ▶ lists of faces and cofaces of the simplex.
- ▶ For the efficiency and simplicity we need to know exactly in which sensors each simplex from the Rips complex should be kept.

How the Rips complex can be effectively (without redundancy) represented in the sensor network?

- ▶ Each sensor has a unique integer label.
- ▶ Let $[X_1, \dots, X_n]$ be the representation of the simplex S in the Rips complex with sensor's labels.
- ▶ Convention – if the dimension of a simplex S is greater or equal 1, then the information about S is kept in the two sensors having the lowest labels among $[X_1, \dots, X_n]$.
- ▶ 0-dimensional simplices are kept only in one sensor.
- ▶ Convention – all the decisions about the simplex S are taken by the sensor with the smallest label in S .
- ▶ So for each simplex S all sensors that "rule" the boundary of S have the information about S .

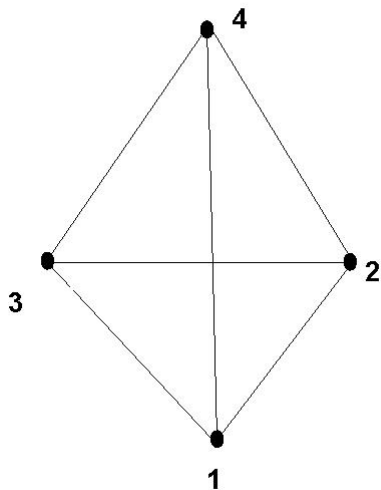
Construction of the Rips complex by the sensors alone

- ▶ Once the sensors are distributed, they need to communicate to get to know the labels of all its neighbors.
- ▶ Moreover every sensor needs to know the labels of the neighbors of its neighbors.
- ▶ No more communication is needed to construct the Rips complex.
- ▶ The communication graph is a 1-skeleton of the Rips complex we are looking for.
- ▶ From the presented convention each 1-simplex is kept in the both of its endpoints.

Construction of the Rips complex by the sensors...

- ▶ Each simplex S keeps the list of the sensors $\{X_1, \dots, X_n\}$ that are neighbors of each sensor involved in S .
- ▶ $S \cup X_i$ for $i \in \{1, \dots, n\}$ is a simplex in a Rips complex we are creating.
- ▶ We have to create the simplex $S \cup X_i$ in the right sensors to follow the given convention.
- ▶ The simplex is created in a sensor if the sensor is in charge of S in the following two situations:
 - ▶ The label of X_i is greater than any label of the sensor in S .
 - ▶ The label of X_i is smaller than any label of the sensor in S
- ▶ In this way all the simplices that belongs to the Rips complex will be created,
- ▶ each simplex will be created by at most two sensors, and only one of them will be in charge of the simplex.
- ▶ The presented procedure can be easily implemented on the parallel architecture like the sensor network.

Example.



Fence reduction procedure

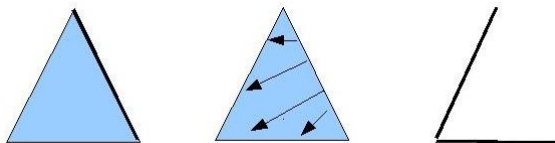
- ▶ Silva-Ghrist's criterion needs relative homology group generators $[\alpha] \in H_2(\mathcal{R}_{r_{com}}, \mathcal{F}_{r_{com}})$.
- ▶ Computation of relative homology is needed.
- ▶ The fence complex $\mathcal{F}_{r_{com}}$ is a closed subcomplex of the complex $\mathcal{R}_{r_{com}}$.
- ▶ In this case $H_2(\mathcal{R}_{r_{com}}, \mathcal{F}_{r_{com}})$ is isomorphic to $H_2(\mathcal{R}_{r_{com}} \setminus \mathcal{F}_{r_{com}})$.
- ▶ So we simply remove the 0 and 1 dimensional simplices that belong to the fence from the constructed Rips complex.
- ▶ This can be done by the sensors themselves if each sensor knows if it belongs to the fence or not (we assume they know that).

(Co)reduction algorithms

- ▶ The created Rips complex is usually a really big structure.
- ▶ The simplices of arbitrary dimension can appear in it.
- ▶ This is not a good input for the homology computation algorithm, we need to reduce the input as far as possible.
- ▶ To do so the combination of the simple reduction and coreduction algorithm will be used.

(Co)reduction algorithms

- ▶ Simple reduction – the retraction procedure:



- ▶ The (co)reduction algorithm is a kind of deal between a simplex and its (co)boundary.
- ▶ But from the construction of the Rips complex we know, that the sensors that have the information about a simplex S also have the information about its boundary.
- ▶ Once the technicalities are fixed the sequence of Simple and Co- reductions works perfectly.

How far can we reduce the complex? Do we need algebraic computations?

Conjecture







For the relative homology of a Rips complex built on the basis of the planar point set the sequence of Simple and Co – reductions reduce the complex up to its homology generators.

- ▶ By the generator in the reduced complex we mean the simplex without boundary elements.
- ▶ The idea behind this – even if the dimensions of simplexes in the Rips complex are high, the topology of it almost "planar" (in planar case the reductions can do all the job) ...
- ▶ this is what we see from the numerical experiments.
- ▶ Even if the Conjecture is not true, the complex that remains after (co)reductions is very small...
- ▶ and can be gathered by one of the sensors and the homology can be computed.

Can this work? Can we prove the algorithm is correct?

- ▶ In the papers dealing with Simple and Co – reductions only proofs of the correctness of the sequential algorithms are provided.
- ▶ We are using a parallel version of these algorithms.
- ▶ The parallel execution of the presented algorithm is equivalent to certain execution of the sequential algorithm.
- ▶ We demonstrate that the maps induced in homology by the parallel execution are isomorphic to the maps induced by the sequential algorithm...
- ▶ ... so both parallel (co)reductions and pulling back the generator are correct algorithms.

The bibliography for this section.

-  R. Ghrist, A. Muhammad, *Coverage and hole-detection in sensor networks via homology.*
-  V. de Silva, R. Ghrist, A. Muhammad, *Blind Swarms for Coverage in 2-D.*
-  V. de Silva and R. Ghrist, *Coordinate-free coverage in sensor networks with controlled boundaries via homology.*
-  B. Batko, M. Mrozek, *Coredution Homology Algorithm.*
-  Computer Assisted Proofs in Dynamics,
<http://capd.wsb-nlu.edu.pl>.
-  Computational Homology Project, <http://chomp.rutgers.edu>.

Where Maxwell meet Poincare.

Part 3,

Computational homology in electromagnetism, where Poincare meets Maxwell.



Henri Poincaré



James Clerk Maxwell



(joint work with Ruben Specogna and Francesco Trevisan.)

Modeling of real-world objects

- ▶ Tetrahedral mesh used in modeling real-world objects.
- ▶ Discretization of the Maxwell's equations used, computed values constant for each simplex in the mesh.
- ▶ Maxwell's equations can be formulated by using only (co)chains and (co)boundary operator [4], [5].
- ▶ In such approach from very beginning theory works on the discrete level. No approximation of continuous mathematical operators needed.
- ▶ $T - \Omega$ formulation uses this approach. However it requires some prior knowledge about the topology of the mesh.

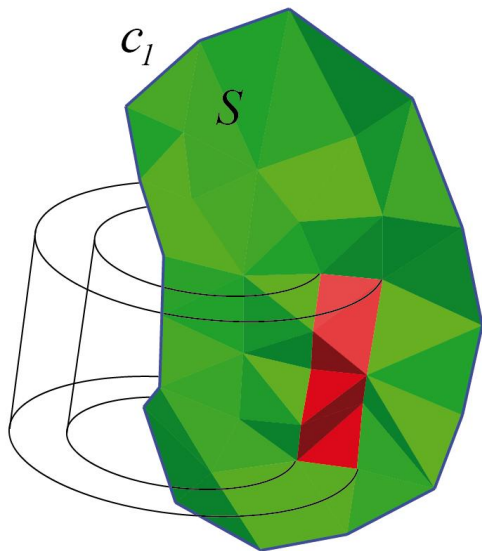
What do electrical engineers need the (co)homology for?

- ▶ Two regions present in the meshes:
 - ▶ Conducting region, D_c .
 - ▶ Non-conducting (air) region, D_a .
- ▶ Assumption: $D_a \cup D_c$ is acyclic.
- ▶ Extra topological information needed to enforce the Ampere's law in D_a .
- ▶ Ampere's law in the classical physics– integrated magnetic field around a closed loop is related to the electric current passing through any surface the boundary of which is the loop.
- ▶ Discrete level:
 - ▶ Cycles in D_a instead of the closed loops.
 - ▶ Chains having the cycle as a boundary instead of the surfaces.
 - ▶ The magnetomotive force (this is what we are looking for) d is a cochain.
 - ▶ The integral of a magnetic field over some chain c is the evaluation $\langle d, c \rangle$.

Non-trivial topology appears

- ▶ If D_a – topologically trivial, then each cycle is a boundary. Electric current passing through the boundary is zero (the current can be driven only by elements of D_c).
- ▶ It is getting more interesting when D_a is homologically non-trivial.
- ▶ If c_1 in D_a is not a boundary in D_a , then each chain S in $D_a \cup D_c$ such that $\partial S = c_1$ needs to "intersect" D_c .
- ▶ The information about cycles that are not boundaries in D_a is needed. The basis of such cycles is precisely the basis of the homology group $H_1(D_a)$.

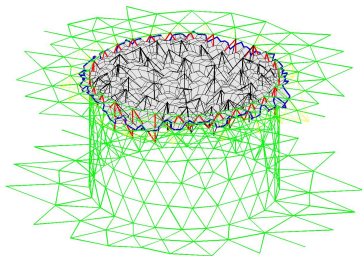
Topologically nontrivial D_a



What is the topological information we need, though?

How the magnetomotive force should be defined?

- ▶ Standard approach (used so far in the electrical engineering) – cochain called *thick cut* has to be found to make the $T - \Omega$ formulation computations in the homologically nontrivial regions possible.



- ▶ For example in case of simple torus the *thick cut* d is defined in such a way that for each cycle c that "goes n times around the branch of a torus" we have $|\langle d, c \rangle| = n$.

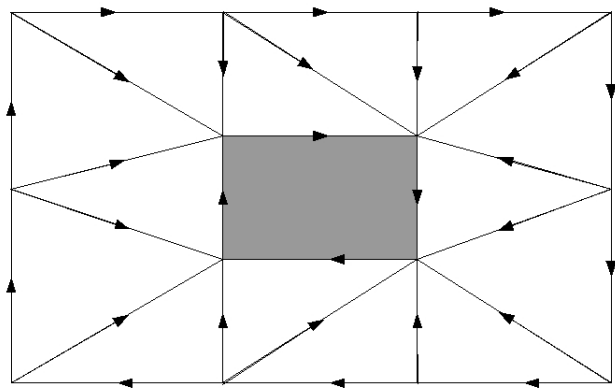
GSTT, the standard algorithm used by the engineers to construct *thick cut*

- ▶ To obtain *thick cut* it is enough to enforce the evaluation of the cochain on all the first homology generators, and enforce zero evaluation on all boundaries (once this is done, all other cycles are spanned "automatically").
- ▶ There exists a standard algorithm to compute the *thick cut* (homology generators needed at the input).
- ▶ So called *belted tree* is constructed at the beginning by the algorithm.
- ▶ $h_1, \dots, h_{\beta_1(D_a)}$ – representatives of the $H_1(D_a)$ basis.
- ▶ A belted tree \mathcal{T} in D_a – a maximal spanning tree T' with the set of edges $\{E_i\}_{i=1}^{\beta_1(D_a)}$ such that only cycle in $T' \cup E_i$ is a support of h_i .

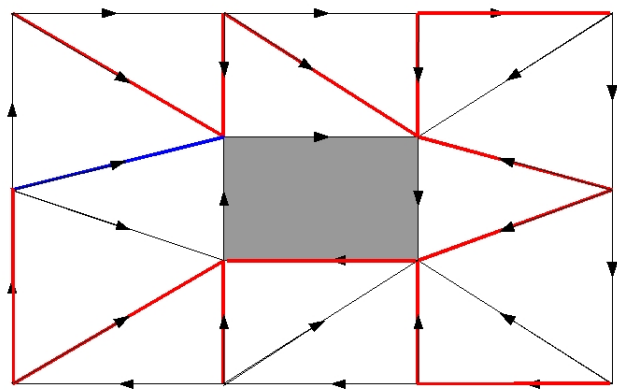
Generalized spanning tree technique (GSTT)

- ▶ When a belted tree $\mathcal{T} = T' \cup \{E_i\}_{i=1}^{\beta_1(D_a)}$ is given the following algorithm is called for each $i \in \{1, \dots, \beta_1(D_a)\}$.
- ▶ For every edge $E \in T'$ put $\langle d_i, E \rangle = 0$ and $\langle d_i, E_j \rangle = \delta_{ij}$ for $j \in \{1, \dots, \beta_1(D_a)\}$.
- ▶ Let L be the list of all 2-simplexes in D_a .
- ▶ *while* L is not empty
 - ▶ Take $T \in L$
 - ▶ *if* the cochain d_i is defined for all edges in ∂T , *then if* $\langle d_i, \partial T \rangle \neq 0$, *return* FAILURE, *else* $L := L \setminus T$.
 - ▶ *if* the cochain d_i is not defined for a single edge E in the boundary of T , then define the value of $\langle d_i, E \rangle$ to have $\langle d_i, \partial T \rangle = 0$ and $L := L \setminus T$.

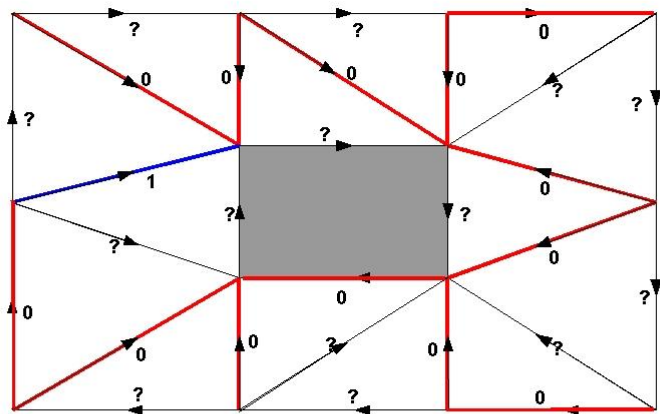
How does the *GSTT* work?



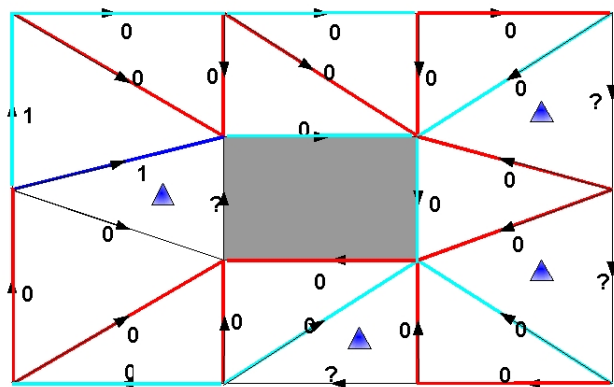
The belted tree



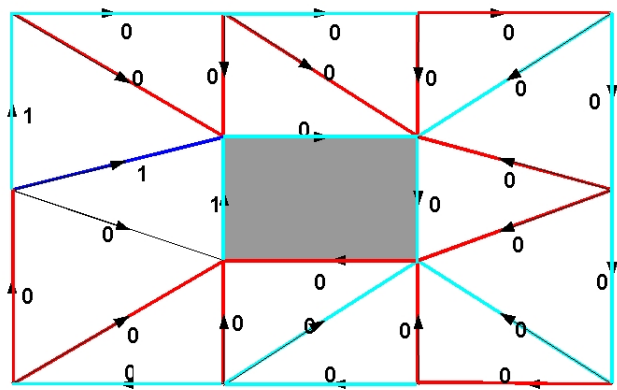
The initial value of a cochain



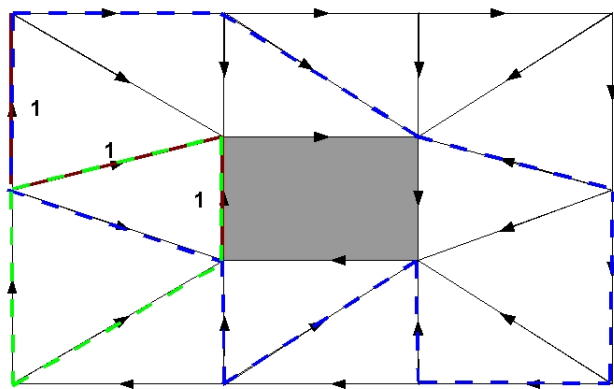
Which 2-simplexes can be removed from the list L in the second step of the algorithm?



Output











Output...



Combining of the GSTT algorithm with the computational homology.

- ▶ The basis of $H_1(D_a)$ is needed to run the presented algorithm.
- ▶ So far – computed by pure Smith Normal Form or chosen "by hand" (only toy examples could have been computed).
- ▶ *GSTT* algorithm using the homology generators provided by the CAPD software was implemented by P.D., Ruben Specogna and Francesco Trevisan in [4].
- ▶ Fast reduction methods from CAPD open the field of computations for far bigger meshes than it was possible before.
- ▶ However the algorithm leaves some open questions.
 - ▶ Does the algorithm always converge?
 - ▶ Can it return FAILURE for some input?
 - ▶ What is the output from the mathematical point?

The bibliography.

-  A. Bossavit, Computational Electromagnetism.
-  T. KACZYNSKI, K. MISCHAIKOW, M. MROZEK, *Computational Homology*.
-  A. HATCHER, *Algebraic topology*.
-  P. D., R. SPECOGNA, F. TREVISAN, *Automatic generation of cuts suitable for the $T - \Omega$ geometric eddy-current formulation*, accepted, CAME.
-  E. Tonti, *On the formal structure of physical theories*.
-  M. Mrozek, P. Pilarczyk, N. Żelazna, Homology Algorithm Based on Acyclic Subspace.
-  Computer Assisted Proofs in Dynamics, <http://capd.wsb-nlu.edu.pl>.
-  Computational Homology Project, <http://chomp.rutgers.edu>.

How the STT \ GSTT is related to Coreduction?

- ▶ Let us stick to the homologically trivial complexes (there exists a version of GSTT designed for this purpose referred to as STT).
- ▶ In the STT the 1-simplex E can be imposed:
 - ▶ as a part of belted tree
 - ▶ when there exist a 2-simplex T having E as the only non-imposed face.
- ▶ In the Coreduction algorithm the 1-simplex can be removed:
 - ▶ In a coreduction with a 0-simplex.
 - ▶ In a coreduction with a 2-simplex.
- ▶ In STT a maximal spanning tree is imposed at first.
- ▶ It can be demonstrated, that the coreduction process can be reorganized in the way, that the maximal spanning tree is removed from a complex first.

How the STT \ GSTT is related to Coreduction?

- ▶ When the maximal spanning tree is removed, no other reductions of edge with 0-simplexes are possible (all the 0-simplexes are already gone).
- ▶ In the Coreduction algorithm (MST is already removed!) the 1-simplex E can be removed if E is the only non-removed face of some 2-simplex T .
- ▶ In the STT the 1-simplex E is imposed, if it is the only non-imposed face of some 2-simplex T .
- ▶ Conclusion– the STT converge iff some sequence of coreductions can remove all the 1-simplexes from the complex.
- ▶ In case of homologically non-trivial domains the GSTT corresponds to so called extended coreduction algorithm (under construction).

The end.

Thank you for your attention!