A Codimension-Zero Approach to Discretizing and Solving Field Problems

F. Milicchio [a] A. DiCarlo [b] A. Paoluzzi [a] V. Shapiro [c]

[a] *Dip. Informatica e Automazione, Università "Roma Tre", Rome, Italy*

[b] *SMFM@DiS, Università "Roma Tre", Rome, Italy*

[c] *Mechanical Engineering & Computer Sciences, University of Wisconsin, Madison, USA*

Abstract

Computational science and engineering are dominated by field problems. Traditionally, engineering practice involves repeated iterations of shape design (i.e., shaping and modeling of material properties), simulation of the physical field, evaluation of the result, and re-design. In this paper, we propose a specific interpretation of the algebraic topological formulation of field problems, which is conceptually simple, physically sound, computational effective and comprehensive. In the proposed approach, physical information is attached to an adaptive, full-dimensional decomposition of the domain of interest. Giving preeminence to the cells of highest dimension allows us to generate the geometry and to simulate the physics simultaneously. We will also demonstrate that our formulation removes artificial constraints on the shape of discrete elements and unifies commonly unrelated methods in a single computational framework. This framework, by using an efficient graph-representation of the domain of interest, unifies several geometric and physical finite formulations, and supports local progressive refinement (and coarsening) effected only where and when required.

Key words: Solid & physical modeling, finite methods, algebraic topology
PACS: 02.40.Re, 02.70.Bf, 02.70.Dc

Email addresses: milicchio@dia.uniroma3.it (F. Milicchio), adicarlo@mac.com (A. DiCarlo), paoluzzi@dia.uniroma3.it (A. Paoluzzi), vshapiro@engr.wisc.edu (V. Shapiro).

1 Introduction

Many physical phenomena are modeled as field problems, in terms of a system of partial differential equations. With rare exceptions, the construction of explicit exact solutions to such problems is out of question. The mathematical problem is therefore subjected to qualitative analyses on one side, and to computationally tractable quantitative techniques on the other. Over the years, a number of commonly used approximation methods have been developed, namely finite differences, finite elements, and finite volumes (briefly FDM, FEM and FVM), all of which are based on finite approximations of the geometry and the fields involved. All such methods may be viewed as instances of a *single* algebraic-topological approach to the representation and solution of field problems. In all cases, the domain is (approximately) subdivided into full-dimensional finite cells, leading to a difference representation of the differential operators of interest.

In our formulation, the construction of a finite approximation relies on the topological concepts of chains and cochains, and the related linear operations of boundary and coboundary. This approach is inspired by the applications to be developed within the next generation of computational sciences, as briefly summarized in Section 1.1. In particular, it should now be possible to construct simulation models of field problems where geometric and physical properties are generated, detailed, and refined simultaneously and progressively.

1.1 Motivation

Field problems dominate computational science and engineering. Traditionally, engineering practice involves repeated iterations of design (i.e., shaping and modeling of material properties), simulation, evaluation and re-design. Advances in computer technology—both software and hardware—in computational science and in simulation methods have made such iterations more efficient and accurate, enhancing productivity and shortening time-to-market. But the trial-and-error procedure in itself has not changed in a significant way, involving a pipelined sequence of *separate* modeling tasks, computational steps, and conversions between different representations (such as re-meshing).

Novel application areas are characterized by an enormous increase in size of computer models. For example, a typical quantity of elementary data to take into account in biological simulations is easily 10^3 to 10^6 times greater than in conventional engineering design problems. This huge increase in size is due to several factors. First of all, such problems typically involve large cellular decompositions (instead of the more compact boundary representations).

This fact alone accounts for an increase of one to two orders of magnitude of model size. Moreover, database factoring of repeated substructures, implicitly produced by hierarchical graphs, cannot be used when dealing with large deformations. This impossibility implies a further size increase of several orders of magnitude. Finally, the sheer number and complexity of components should be considered: there are several thousand atoms in a protein, several thousand proteins in a cell, and so on (see, e.g. (1; 2)).

Also, very-large-scale visualization problems have been recently approached in computer graphics by hierarchical multiscale representations. In scientific visualization, new progressive methods allow the scientist to get real-time interaction with terascale data sets, making the best use of the available bandwidth between storage, processors and graphics hardware. However, the limiting factor is once more the fact that the sequential operations of modeling, meshing, simulating and visualizing are typically performed by different people, using different computational methods and different data structures.

In contrast, the approach we advocate here aims at seamlessly combining the geometric model of the body under consideration, the description of its physical properties and the simulation of the relevant patterns emerging from geometry and physics. In particular, we rely on well-established concepts from algebraic topology, based on the standard definitions of *chain* and *cochain*, to be summarized in the next section. As we shall see, physical quantities are aptly described by attaching values to point-subsets (cells) from the model decomposition, thus supporting simultaneously and progressively both geometric detailing and physical simulation. As a result, such a capability should provide scientists and engineers with a prompt information on how their models respond to progressive changes in design and/or simulation parameters.

1.2 Related work

The quest for classification and unification in physical field theories dates back at least to the work of Maxwell (3). The computational advantages of this unified view have been realized by Kron (4), who developed analog-computer models to simulate a variety of physical field problems. Roth (5) appears to have been the first to observe that algebraic topological foundations underlie all such models. Branin (6) advocated a unified discrete view of all physical theories using concepts from algebraic topology and the De Rham cohomology. This line of inquiry culminated in a comprehensive *classification* of many diverse physical theories in terms their topological structure (7).

More recently, this early research led to new efforts in developing unified computational models and languages for analysis, simulation, and engineering de-

sign. Notably, Palmer and Shapiro (8) proposed a unified computational model of engineering systems that relies on concepts from algebraic topology. Their idea appears as a natural consequence of the *Stokes theorem*, which relates the integral of a differential form ω on the boundary ∂R of a domain R to the integral of the exterior derivative $d\omega$ of ω over the domain itself:

$$\int_R d\omega = \int_{\partial R} \omega \tag{1}$$

The fact that a cell-by-cell integration of a differential p-form yields a p-cochain may be summarized by the commutative diagram (9):

$$\begin{array}{ccc} p\text{-cochain} & \xrightarrow{\delta} & (p+1)\text{-cochain} \\ \int \uparrow & & \uparrow \int \\ p\text{-form} & \xrightarrow{d} & (p+1)\text{-form} \end{array} \tag{2}$$

This property holds for every form in any dimension, is metric-free and obviously independent of coordinate parametrization. Realizing that cochains are discrete (integrated) analogues of differential form, a number of researchers proposed to build numerical simulation models directly in terms of cochains (or chains, considered as isomorphic to cochains). Palmer (10) proposed to encode conventional FEMs for plane elasticity problems using chains as a basic data type. In (11; 12) this approach was extended in a substantial way, leading to the implementation of a general-purpose language for specifying and computing cell-based models. A discrete vector calculus on regular lattices was proposed and variously exemplified in (13).

A number of researchers went beyond the use of chains and cochains as general-purpose data types, considering that a sound numerical method should reflect the algebraic-topological structure of the underlying physical theory in a faithful way. Notably, Strang (14) observed that the FEM encodes a pervasive equilibrium pattern, which is at the center of the classification in (7). Mattiussi(15) provided interpretations of FEM, FVM, and FDM in terms of the topological properties of the corresponding field theory. Tonti (16) presented his *cell method* as a direct discrete method, bypassing the underlying continuum model. In (17) FDMs that satisfy desired topological properties are discussed.

Two notions introduced in (8) deserve consideration. First, many physical laws may be expressed *combinatorially* (not just discretely) from first principles. This idea was formalized in (18), where the authors proposed combinatorial representations for differential forms, equations, and balance laws, and proved

a dual version of Stokes' theorem in the form of a commutative diagram:

$$\begin{array}{ccc} p\text{-cochain} & \xrightarrow{\delta} & (p+1)\text{-cochain} \\ \lim_{cell \to 0} \downarrow & & \downarrow \lim_{cell \to 0} \\ p\text{-form} & \xrightarrow{d} & (p+1)\text{-form} \end{array} \tag{3}$$

whenever the cell-by-cell limit is well defined. These results provide a basis for developing new languages to describe physical models and systematically transform them into strongly typed numerical simulations. Second, the existence of isomorphisms between chains and cochains, primal and dual decompositions, boundary and coboundary operators, suggests a multitude of alternative but (in some sense) equivalent formulations. For example, in (19) it is shown that a small set of combinatorial operators (namely boundary, coboundary, and dualization) is sufficient to represent a variety of physical laws and invariants in the context of design automation. Specific examples include geometric integrity, balance, and surface smoothing.

In this paper, we advance a specific interpretation of the algebraic-topological formulation, which is conceptually simple, physically sound, computational effective and comprehensive. Our goal is to apply this approach to field modeling onto an already established computational framework for geometric modeling with cell complexes (20). This framework has been recently extended to provide parallel and progressive generation of very large datasets using streams of continuous approximations of the domain with convex cells (21). This approach also supports progressive Boolean operations (22), providing continuous streaming of geometrical features and adaptive refinement of their details.

1.3 Preview

Section 2 provides the reader with a minimal set of basic definitions from algebraic topology. Section 3 introduces our finite representation of field problems, centered on cells of *codimension* [1] zero, and compares it with the more common presentation focused on nodes, i.e., cells of *dimension* zero, and in Section 4 we will introduce the SPLIT algorithm as a means of mesh refining. In Section 5 our algebraic-topological approach is applied to a simple prototype problem involving the Laplacian.

[1] Codimension is a basic topological concept that applies to subspaces of vector spaces, and more generally to submanifolds. It is, by definition, the dimension of any complementary subspace (submanifold).

2 Background

In the following definitions, which are mainly adapted from (23), we refer to *simplicial* complexes only for the sake of simplicity. In fact, all we say can be extended seamlessly to more general cellular complexes, in particular to cell complexes with convex polyhedral cells. In the following, we often refer to simplices and cells as if they were synonyms.

Oriented complexes An oriented *p-simplex* [2] is defined as

$$\sigma_p = [\,\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_p\,] := \mathrm{conv}\,(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_p) \subset \mathbb{E}^d, \qquad p \leq d\,.$$

Its $(p-1)$-*faces* are the $(p-1)$-simplices

$$\sigma_{p-1,\,k} := [\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_p], \qquad 0 \leq k \leq p\,,$$

where each face gets the orientation induced by its ordered vertex subsequence, as extracted from $(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_p)$. The simplex $[\,\mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_2, \ldots, \mathbf{x}_p\,]$ sharing the *support* of σ_p with *opposite* orientation is denoted $-\sigma_p$.

Let K be an oriented simplicial *complex*, defined as a set of oriented simplexes such that: (i) if a simplex is in K, then so are all its faces, and (ii) every two simplexes in K either do not intersect or intersect on their common face. Let $K_p \subset K$ be the set of p-simplices in K. Each p-simplex (and consequently the whole complex) is oriented by a total ordering of the set K_0 (the set of *nodes* of K).

Chain groups Let $(G, +)$ be a free abelian (i.e., commutative) group. A *p-chain* of K with coefficients in G is a mapping $c_p : K_p \to G$ such that

$$c_p(-\sigma) = -c_p(\sigma), \qquad \sigma \in K_p$$

Chain addition is defined by sum of chain values: if c_p, d_p are p-chains, then $(c_p + d_p)(\sigma) = c_p(\sigma) + d_p(\sigma)$, for each $\sigma \in K_p$. The resulting group is denoted $C_p(K; G)$.

Let σ be an oriented simplex in K and $g \in G$. The *elementary chain* whose value is g on σ, $-g$ on $-\sigma$ and 0 on any other simplex in K is denoted $g\sigma$.

[2] The simplex σ_p, based at $\mathbf{x}_0$, is non degenerate if and only if the p vectors $\mathbf{x}_k - \mathbf{x}_0$ $(1 \leq k \leq p)$ are linearly independent. We write $\mathrm{conv}\,(\ldots)$, instead of $\mathrm{conv}\,\{\ldots\}$, to underline that vertex *ordering* matters.

Each chain can then be written in a unique way as a finite sum of elementary chains:

$$c_p = \sum_{\sigma_{p,\,k} \in K_p} g_k \sigma_{p,\,k} \ .$$

Chains attach *multiplicity* to cells. If their coefficients are taken in the smallest nontrivial group, i.e. $G = \{0, 1, -1\}$, cells can only be discarded or selected, possibly inverting their orientation.

Boundary The *boundary operator* $\partial : C_p(K;G) \to C_{p-1}(K;G)$ is first defined on cells, as follows:

$$\partial \sigma_p := \sum_{k=0}^{p} (-1)^k \sigma_{p-1,\,k} \ ,$$

then extended to elementary chains, by taking

$$\partial(g\sigma) := g(\partial\sigma) \, ,$$

and finally to all chains by assuming ∂ to be *additive*.

Cochain groups By definition, the set of *p-cochains* of K, with coefficients in G, is the group of all homomorphisms of $C_p(K;G)$ into G:

$$C^p(K;G) := Hom(C_p(K;G), G),$$

In other words, cochains measure the *content* of G-valued *additive* quantities in chains. If γ^p is a p-cochain, its content in the p-chain c_p is often denoted as a pairing:

$$\langle \gamma^p, c_p \rangle := \gamma^p(c_p) \, .$$

Coboundary The *coboundary operator* δ is defined as the dual of the boundary operator $\partial : C_{p+1}(K;G) \to C_p(K;G)$, so that

$$C^{p+1}(K;G) \ \stackrel{\delta}{\leftarrow} \ C^p(K;G)$$

with

$$\langle \delta\gamma^p, c_{p+1} \rangle = \langle \gamma^p, \partial c_{p+1} \rangle \, .$$

The pairing notation makes transparent that this defining property is a combinatorial prototype of the Stokes theorem (24).

Let σ_α^* denote the *elementary cochain* which takes value 1 on the elementary chain σ_α and value 0 on all other elementary chains. Let us also denote $g\sigma_\alpha^*$ the elementary cochain whose value is g on σ_α and zero on all other elementary chains. It can then be seen (23) that $C^p(K;G)$ is isomorphic to the Cartesian product of n_p copies of G, where n_p is the number of p-simplices in K. Under such isomorphism, each cochain γ^p corresponds to a tuple $(g_\alpha\sigma_\alpha^*)_{\alpha\in\{1,\ldots,n_p\}}$, which is often written as a sum:

$$\gamma^p = \sum_{\alpha=1}^{n_p} g_\alpha \sigma_\alpha^* \,,$$

because of the additivity property:

$$\delta\gamma^p = \sum_{\alpha=1}^{n_p} g_\alpha(\delta\sigma_\alpha^*).$$

Fundamental identities The key property of boundary and coboundary operators is embodied in the two dual identities:

$$\partial^2 := \partial \circ \partial = 0 \,, \qquad \delta^2 := \delta \circ \delta = 0 \,, \tag{4}$$

where 0 denotes the null chain or cochain, respectively. The chains whose boundary is null form a group called the group of *cycles*, and the respective null-coboundary cochains, form the *cocycles* group. The fundamental identities shown above state that every (co)boundary is a (co)cycle.

Incidence A p-incidence matrix $A_p = [a_{ij}]$ describes how p- and $(p-1)$-cells intersect: $a_{ij} = 0$ if $\sigma_{p,i} \cap \sigma_{p-1,j} = \emptyset$; $a_{ij} = \pm 1$ otherwise, with the sign decided by the relative orientation between the simplex $\sigma_{p-1,j}$ and the j-th face of $\sigma_{p,i}$. A small 2-complex and its incidence matrix A_2 are given in Figure 1 and Table 1, respectively. Incidence matrices are practical computational tools. It is easy to see that A_p and its transpose $A_p^\top$ represent through matrix multiplication the action of the boundary operator $\partial : C_p \to C_{p-1}$ and of the coboundary operator $\delta : C^{p-1} \to C^p$, respectively:

$$\begin{aligned} [a_{ij}]\,[\sigma_{p-1,\,j}] &= [\partial\sigma_{p,\,i}] \\ [a_{ji}]\,[\sigma_{p,\,i}] &= [\delta\sigma_{p-1,\,j}] \end{aligned}$$

Table 1
The incidence matrix $A_2^\top$ of the complex K in Figure 1.

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9
σ_1	-1	-1	-1	0	0	-1	0	0	0
σ_2	0	-1	0	1	-1	0	0	0	0
σ_3	0	0	0	0	-1	0	-1	0	-1
σ_4	0	0	0	0	0	-1	-1	1	0

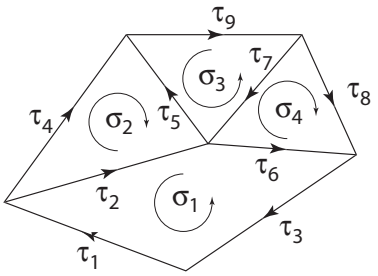

Figure 1. A 2-complex K, with 1- and 2-cells arbitrarily oriented.

3 Discretization of field problems

Commonly, the approximate solution of field problems is approached starting from nodal values as field samples, i.e., considering 0-cochains. Cells of dimension 0, called *nodes*, are connected to each other by 1-dimensional cells obtained from a cellular discretization of the domain, often called *edges*. Edges bound 2-dimensional cells called *faces*, and so on. Together, the collection of cells of all dimensions is an oriented cell complex, usually referred to as a *mesh*. In this paper we offer a different perspective, in which the starting point is provided by the opposite extreme of the hierarchy, i.e., by cells of *codimension* 0 and 1. This approach allows us to formulate field problems with meshes of general type, without the usual constraints imposed on the shape discretizations by standard finite methods.

3.1 Mesh duality

We refer to a given mesh in an n-dimensional space as a *primal* cell complex K. A dual cell complex D can be constructed in many ways [3] (23; 7). In the

[3] Depending on the construction, D may fail to be a honest cell complex, since the boundary of some dual cells is not in D. However, there are several standard ways

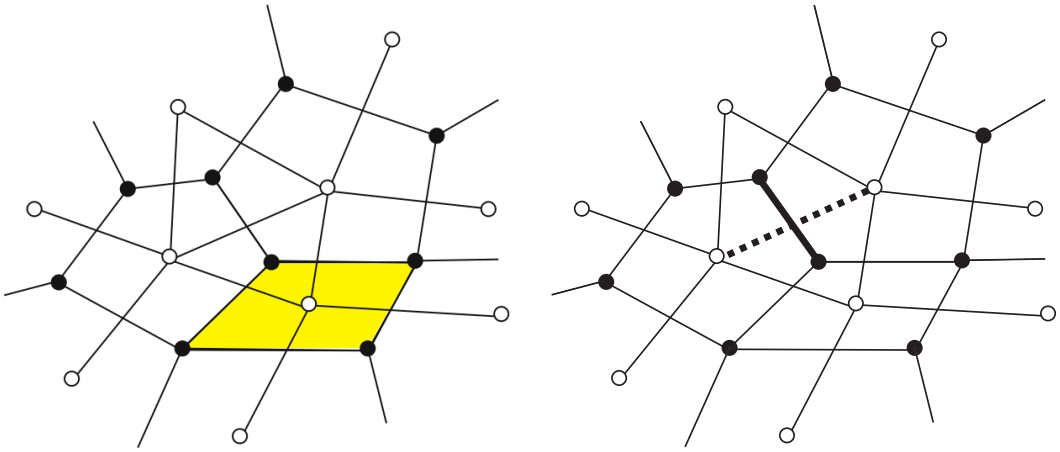

Figure 2. Primal and dual complexes: (a) duality between K_2 and D_0; (b) duality between K_1 and D_1.

$K \leftrightarrow D$ duality, to each p-cell in K (the primal complex) there corresponds a unique $(n-p)$-cell in the dual complex D, and vice versa. Such an association is purely topological, and the detailed geometry of the cells is immaterial in this respect. The duality in 2-space between a 2-cell in K and the corresponding 0-cell in D, as well as the duality linking a 1-cell in K and its dual 1-cell in D, are illustrated in Figure 2.

The duality between K and D induces a hierarchy of isomorphisms, collectively denoted by ϕ, between the *cochain* group $C^p(K)$ and the *chain* group $C_{n-p}(D)$ (see (23) for details). This produces the commutative diagram

$$\begin{array}{ccc} C^{p-1} & \xrightarrow{\phi} & C_{n-p+1} \\ \delta \downarrow & & \downarrow \partial \\ C^{p} & \xrightarrow{\phi} & C_{n-p} \end{array} \tag{5}$$

implying that boundary and coboundary operators are related as follows:

$$\partial = \phi \circ \delta \circ \phi^{-1}, \qquad \delta = \phi^{-1} \circ \partial \circ \phi. \tag{6}$$

Figures 3 and 4 illustrate these relationships. Figures 3 shows how the boundary of a sample chain in $C_2(K, \mathbb{Z})$, namely the elementary chain 2σ, can be computed via the coboundary of dual cochains. Symmetrically, Figure 4 illustrates the computation of the coboundary of a sample cochain in $C^1(D, \mathbb{Z})$, namely $\gamma = 7\tau_1^* - 3\tau_2^*$ (with τ_1^*, τ_2^* elementary cochains), via the boundary of cells in K.

As a result of properties (5) and (6), the combinatorial version of many physi-

to complete D as a cell complex, see for example (6).

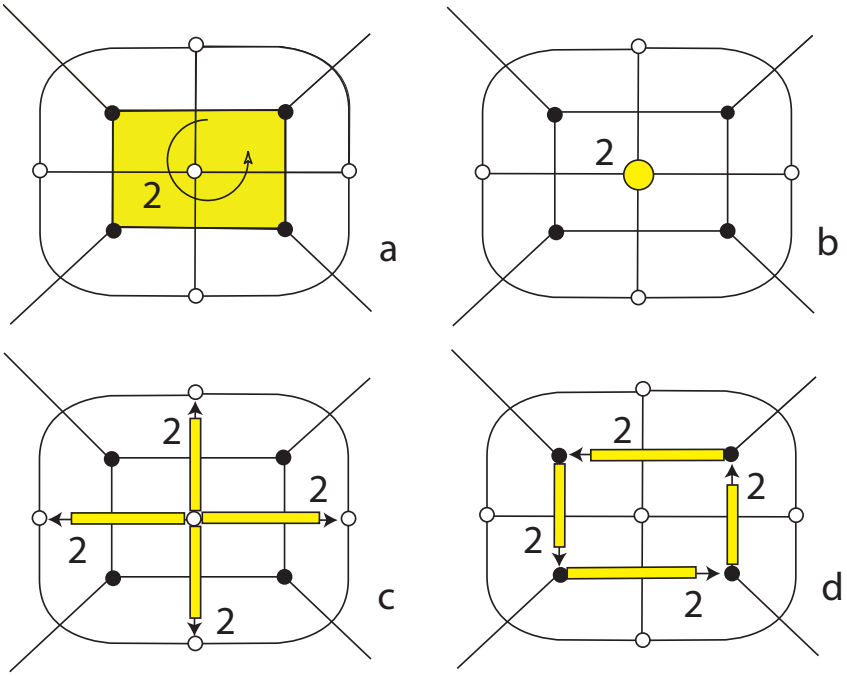

Figure 3. Dual computation of $\partial(2\sigma)$: (a) the elementary chain 2σ; (b) its dual cochain value $\phi^{-1}(2\sigma)$; (c) the coboundary value $\delta(\phi^{-1}(2\sigma))$; (d) back to K: $\partial(2\sigma) = \phi(\delta(\phi^{-1}(2\sigma)))$.

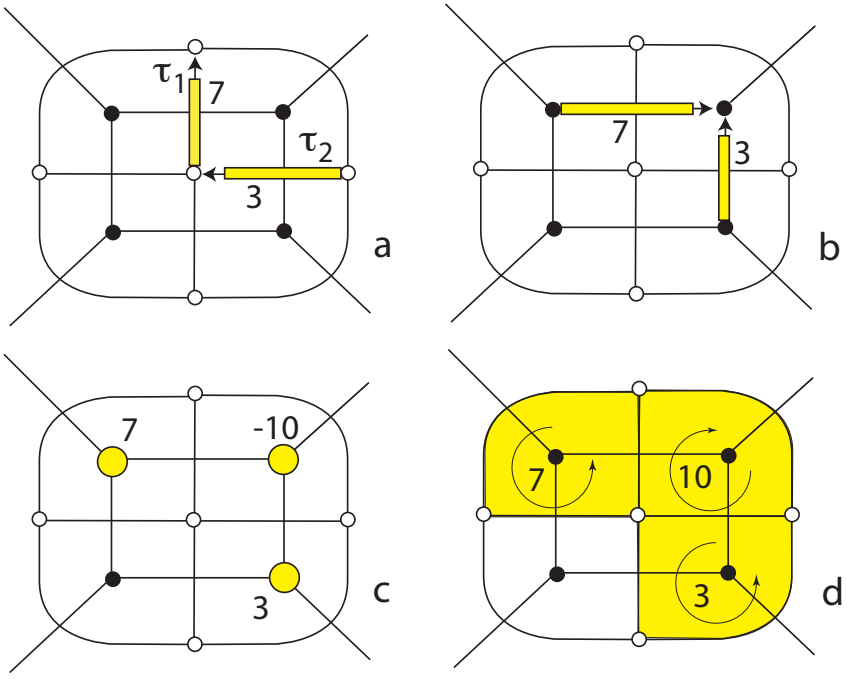

Figure 4. Dual computation of $\delta(7\tau_1^* - 3\tau_2^*)$: (a) the cochain value $\gamma = 7\tau_1^* - 3\tau_2^*$; (b) its dual chain $\phi(\gamma)$; (c) the boundary of the dual chain $\partial(\phi(\gamma))$; (d) back to D: $\delta(\gamma) = \phi^{-1}(\partial(\phi(\gamma)))$.

cal laws may be factorized, as shown in (19), according to the pattern $\partial \circ H \circ \delta$, where chains are considered "*trivial cochains*" on a cell complex and H describes the constitutive relationship between cochains on the dual cell complex.

3.2 Canonical form of finite methods

We are now ready to state our combinatorial approach based on cells of highest dimension. Let us start by discussing the association between mesh nodes and values of a physical quantity. Such an associating could be misinterpreted, implying that the hypothetical physical measurement of that quantity is so accurate to produce the actual value at a single point. However, what is actually measured is an *average* on a small cell "centered" on that point. If field values are regarded as volumetric averages, then the differences between adjacent cells should be associated to their separating surface. Therefore, $(p-1)$-dimensional interfaces between p-cells play the role played by 1-cells in node-based approaches. For instance, in a heat transfer problem the measured temperature is associated to a p-cell, representing the average value in the region, and differences of temperatures are related to the separating surface between two adjacent p-cells, that is a $(p-1)$-cell.

The association of a discrete field with a chain is a natural choice. Chains express values associated to cells, and thus are well suited representations of both domain and codomain values of a physical problem. Cochains on the other side are functions, whose applications on chains (input discrete fields) give other chains (output discrete fields). For example, the heat flux chain is the result of the application of the constitutive relation—represented as a cochain on a cell complex—on spatially-distributed temperatures. This distinction between domain and codomain sets is enforced in our approach, achieving a *strict-typedness* for input and output discrete fields (i.e. chains), the spatial distributions of temperature and heat flux, respectively. Our approach also differs from usual ones in the fact that it does not require two meshes: instead of specifying dual cochains, our approach uses chains and cochains on a single cell complex, and thus is more consistent with common practical numerical methods relying on only one mesh.

3.2.1 Lexicon

A variety of well-known physical theories may be constructed by combining primitive topological and metrical maps. In fact, many authors, including (5; 6; 7; 15; 18) and (19), observed that similar patterns emerge in the factorization of different physical laws.

Domain Let us refer, without loss of generality, to the domain as represented by a cell complex K of codimension 0 and dimension p.

Field Recalling the definition of chains given in Section 2, notice that a field $F : \mathcal{D} \to G$ is piecewise approximated over the representative cell complex K by a chain c_p, i.e. a map $K_p \to G$.

Differentiation The field represented by c_p is transformed by the boundary operator ∂. This operation corresponds to a dual approximation of the exterior differentiation d, which produces a $(p-1)$-chain.

Transfer function A transfer function $\Omega \in Hom(C_{p-1}(K;G), G)$ transforms the input chain c_{p-1} into a cochain γ^{p-1}. In other words, the transfer function operates cell by cell on local approximations.

Integration The last operator applied in Equation (8) is the coboundary δ. This corresponds to an *integration* on the cell decomposition of the domain, producing a p-cochain γ^p.

3.2.2 Existence of canonical form

In the following of this paper we are going to show that a strictly-typed formulation of the combinatorial laws that underlies all finite methods can be summarized as follows:

$$C_p \xrightarrow{\partial} C_{p-1} \xrightarrow{\Omega} C^{p-1} \xrightarrow{\delta} C^p \tag{7}$$

so that the equivalent functional formulation is expressed as

$$\langle \gamma^p, c_p \rangle = (\delta \circ \Omega \circ \partial)(c_p) \tag{8}$$

where c_p is the chain representation of the input field and γ^p the cochain producing the output field.

Traditional finite problem-solving methods include *finite elements*, *finite volumes* and *finite differences*. There are authors that in their work recognized some underlying structure of these methods, notably (7; 15; 18; 12) and (19), and provided a combinatorial interpretation of some numerical methods such as finite volumes and elements.

In this section we demonstrate that all finite methods can be expressed with our framework, separating the topological components from the metrical and physical counterpart. We also provide an interpretation of such methods within the algebraic-topological approach presented in this paper.

Any given finite method supply a way of expressing complex differential equations approximating them with a system of linear equations $M\,\xi = b$, so reducing differential relations to finite differences. Let $A := A_p$ be the incidence matrix of a mesh, and let M be a matrix supplied by any finite method on the same cell complex.

Theorem 3.1 *The matrix M can be reduced in* canonical form

$$M = A^\top \Omega A = \delta\,\Omega\,\partial,$$

if and only if the incidence matrix has trivial kernel, i.e. $Ker(A) = \{0\}$.

Proof The necessity of $Ker(A) = \{0\}$ follows directly from the fact that M, provided by a finite method, should invertible, and consequently it must result $Ker(M) = \{0\}$. Let us suppose that the incidence matrix has a null-dimensional kernel, that is $Ker(A) = \{0\}$. This condition is sufficient to find a matrix C such that is a *left inverse* of A, so $CA = I$, with I being the identity matrix. Then we can construct the matrix Ω that expresses the system of linear equations under our framework, by taking $\Omega = C^\top MC$:

$$\begin{aligned} M &= A^\top \Omega A \\ &= A^\top (C^\top MC) A \\ &= (CA)^\top M (CA) = M \end{aligned}$$

□

3.3 Finite methods

The previous result makes it possible to express any linear problem $M\xi = b$ given by any finite method, with an equivalent one—and vice versa—which separates the metrical and physical components from the topological relations:

$$M\,\xi = \delta\,\Omega\,\partial\,\xi = b.$$

In other words, ∂ and δ are determined only by the topology of the mesh, while Ω concentrates all metrical and physical relationships between mesh and field elements. Moreover, the canonical form (3.1) provides a simple way

of establishing *a priori* if a given mesh may lead to a solvable system of linear equations, since $Ker(A) = \{0\}$ implies that A has full rank.

Expressing a problem in the canonical form not only gives a direct insight on the topological and constitutive parts of a given finite method, but allows us to formulate it without being limited by the usual restrictions on cell shapes, provided that the resulting incidence matrix has full rank.

3.3.1 Finite differences

Finite differece methods approximate field integrals by sums and field derivatives by differences. The finite differential and integral operators are expressed using our algebraic-topological approach with boundary and coboundary respectively. The geometric information about cells is made explicit in Ω by the fact that derivatives and integrals depend on the metric established on the domain by its decomposition.

A generic coefficient $\omega_{i,j}$ of Ω expresses the effect of the thermal tensions between the i-th and j-th faces in the cell complex, given by the application of the boundary ∂. The extent of such an influence, derived from metrical informations and not only on a pure-topological basis, produces Ω matrices with different shapes; for instance a five point FD method relates faces whose distance is within a fixed range ϑ dependent on the minimum volume of $(p-1)$-cells in the p-complex: $2\vartheta = \min\{\mathrm{Vol}(F) | F \in K_{p-1}\}$.

The explicit metric of FD could be misleading comparing finite differences with finite elements or volumes methods. While finite differences express distances, areas and volumes directly in the problem formulation, finite elements and volumes simply hide these measures within integral forms. An example of computation of the thermal field with the five-point finite difference method is given in detail in Section 5.

3.3.2 Finite elements

Classical finite elements replace the original problem, expressed usually under a differential formulation, with an equivalent easier version on the problem mesh, i.e. a domain discretization with a p-complex. The field, computed exactly on the nodes by solving a system of linear equation, is approximated elsewhere by interpolation of the nodal values, using a series of functions called *shape* functions (or *basis* functions), as in Figure 5.

Conversely, in our approach nodal values are replaced by a chain c_p on the complex K of dimension p (and codimension 0) that is a *partition* of the problem domain $\mathcal{D}$. Differential and integral operations are represented by

boundary and coboundary operators, respectively. The interpolating step and the subsequent integration are based on the domain decomposition and affects only the transformation Ω, which links the input field to the output. Notice that the canonical form leaves field interpolators unaffected by any variation, and every change is strictly bounded to the Ω matrix.

In other words, FE methods focus on the problem formulation in terms of local approximations, reflecting this approach on the canonical form $\delta\Omega\partial$ by concentrating on the $\Omega\partial$ part expressing the local formulation: this mimics the variational method solving differential equations like $u'' = f$ by expressing its variational form $u'' \rightarrow -u'v'$, where u and v are the field and variation functions, respectively.

To achieve better numerical approximations FEMs usually make use of higher order interpolating functions, influencing the $\omega_{i,j}$ coefficients that establish constitutive relations between faces in the complex: symmetrically to the number of nodal values involved in a higher order interpolation, Ω will correlate a higher number of faces in the output cochain. Another way to achieve an increase in numerical accuracy is refining the mesh, or with combinations of both techniques.

3.3.3 Finite volumes

Similarly to the finite difference method, the finite volume method is a method for representing and evaluating partial differential equations as algebraic equations, with values calculated at discrete places on a meshed geometry. A problem is described using some *balance law* over a finite portion of space, as the name itself suggests, for instance comparing the inner heat generation of a volume with the total heat flux across its boundary. The equations are solved using the conservation principle across each given volume. In particular, volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem. These terms are then evaluated as fluxes at the boundaries of each finite volume. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative.

In our approach, the field is represented as a c_p chain. Separating the differential operations implied by the divergence, represented by the boundary operator, from the flux calculus operated by the transformation Ω, which relies on the result of the boundary operator, we calculate the flux per surface on the boundary of a given volume. The coboundary operator, representing an integral, finally sums the contribution of each surface. Conversely to finite element methods which concentrate more on a local field approximation, finite volumes center on the balance of global quantities in a finite region of space,

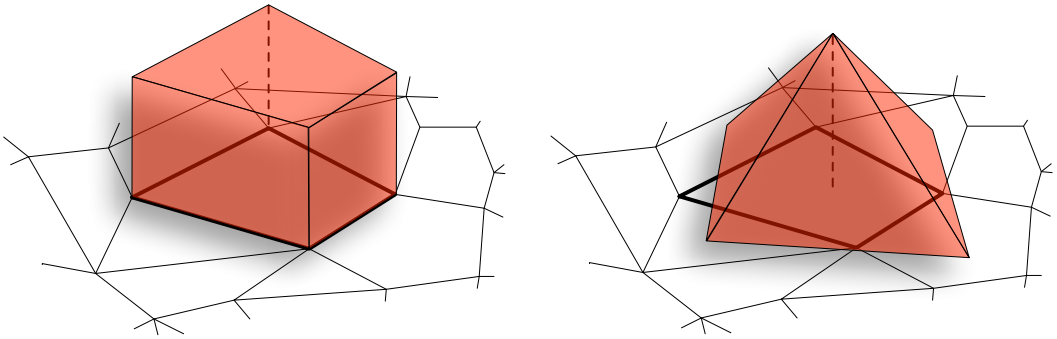

Figure 5. Different shape functions on the same mesh.

in other words FV methods focus on the δ side of the canonical form $\delta\Omega\partial$.

For instance, let us consider a thermal conductivity problem, and let us denote T_p the temperature chain over a cell complex. FV methods compare the total heat flux on the boundary of a volume to the inner heat generation. The temperature gradient in a heat transfer problem can be obtained applying the boundary on T_p, producing the gradient chain $G_{p-1} = \partial T_p$. Applying the transformation Ω on this chain we have the heat flux per surface expressed as a cochain $Q^{p-1} = \Omega G_{p-1}$. The balance is finally enforced by applying the coboundary on the flux chain and equating the result to the heat generation. The final result directly connects the temperature, a chain over the mesh, to the heat flux cochain, a function over the temperature chain.

4 Adaptive graph representation

In the following sections we will introduce a data structure capable of handling both topological and metrical informations, and we will show how to extend it to be suitable for field problems expressed in the canonical form. The SPLIT algorithm will be presented as an efficient mesh refining tool, and analyzed from the viewpoint of our proposed method.

4.1 The Hasse diagram

In order theory, a *Hasse diagram* is a graph $\mathcal{H} = (N, E)$, whose nodes form a finite partially ordered set, and where there exists an arc from x to y if and only if: (a) $x < y$ and (b) there is no z such that $x < z < y$. In this case, we say y *covers* x, or y is an immediate successor of x, as described in (25). Hasse diagrams can be used to give a complete representation of the inclusion between k-faces, $0 \leq k \leq p$, in a p-complex. This structure was introduced for solid modeling in (26), together with an efficient SPLIT algorithm for splitting a convex cell (and its boundary faces) with an hyperplane, as shown in Figures 6

and 7. Such a spitting is the very basic operation when building a progressive BSP-tree as described in (22).

It is easy to see that the incidence matrices A_k of a p-complex K, $1 \leq k \leq p$ directly correspond to arc subsets $E_k \subset E$, such that $\cup_k E_k = E$ and $E_i \cap E_j = \emptyset$, $i \neq j$. This representation is very useful, because for the chain group $C_k(K, G)$ we have

$$C_k(K, G) = (N_k \times G, +), \tag{9}$$

where $N_k \subset N$ are the nodes of $\mathcal{H}$ that correspond to k-cells. Notice that a label from the set $\{-1, 1\}$, and associated to the arc (n_i, n_j) is sufficient to specify the relative orientation between n_i and n_j. For each node $n \in N_k$, let us define $E_n \coloneqq \{(n, n_j) \in E | n_j \in N_{k-1}\}$, and $N_n \coloneqq \{n_j \in N_{k-1} | (n, n_j) \in E\}$. With a tolerable abuse of notation which identifies nodes with cells, and denoting with $g \in G$ the coefficient associated with $n \in N$, the boundary operation may be computed as:

$$\partial(gn) = g \sum_{h \in N_n} sign(n, h)h.$$

In other words, the boundary of the elementary chain gn is obtained simply by summing the (properly signed) coefficient transfered from n to its children in $\mathcal{H}$. The *dual graph* $\mathcal{H}^* = (N^*, E^*)$, with

$$\begin{aligned} N_k^* &= N_{p-k}\,, \\ E^* &= \{(n_j, n_i) | (n_i, n_j) \in E\} \end{aligned}$$

is clearly the Hasse representation of the dual complex $D(K)$. If the nodes in N^* are labeled from G, and the arcs in E^* are labeled with the relative sign of node pairs, then for cochain groups we have

$$C^k(K, G) = (N_k^* \times G, +). \tag{10}$$

The coboundary on $\mathcal{H}$ is the boundary on $\mathcal{H}^*$, as expected. From a practical viewpoint, the same graph, using double links for implementing arcs, may be used for all topological computations.

This graph structure is then a 0-codimension representation of a cell complex, with all the topological and metrical informations needed to express a physical problem in the canonical form. Explicitly representing the cells of any dimension allows for a straightforward retrieval of metrical measures, fundamental to generate physical coefficients, for instance the thermal conductivity,

which is dependent on material properties and on the volume of the cell under analysis.

The Hasse diagram for a cellular decomposition of a field domain can be employed with all numerical methods if using the canonical form $\delta\,\Omega\,\partial$. Such relationship separates the topological aspects of the domain decomposition from the metrical and physical properties of the transfer function of a computation. As a consequence, the Hasse representation allows us not only to describe numerical methods, but also to locally refine the cell complex by conveniently updating the topological and metrical data, as well as the physical quantities represented in the cells, such as the temperatures or the internal heat sources. This approach can be efficiently implemented with the SPLIT algoritm as seen in (26) and will be described in Section 4.2.

4.2 Mesh refining

The framework presented in this paper applies to general meshes, and does not require any particular cell shape. Representing a finite method in its canonical form allows us to use the Hasse diagram for both topological and geometrical information storage, granting the ability to efficiently refine the data structure which characterize the mesh, while avoiding *ad hoc* modifications of hard-wired codes. Recently the same data structure has been shown to support progressive Boolean operations (22), providing an effective method of adaptive geometrical refinement of complex shapes.

In the present section we extend the SPLIT algorithm (26) to update geometry and topology along with the physical and metrical details needed to solve a field problem. We will refer for clarity sake to a two-dimensional cell complex used to formulate a steady-state heat transfer problem, but the same approach applies to cell complexes in any dimensions.

Let us focus on a small region of the mesh $K = K_0 \cup K_1 \cup K_2$ where K_i is the set of all i-cells in the complex, as shown in in Figure 6, where both the cell complex and the relative Hasse diagram are depicted. The thermal field in our example is then represented as a 2-chain $c_2 = t_1 C_1 + t_2 C_2 + \ldots$ which can be expressed with a vector ξ. With a small abuse of notation we will identify the temperatures with their relative cell names, so that the 2-chain may be represented as

$$c_2 = t_1 C_1 + t_2 C_2 + \ldots \longrightarrow \xi = [C_1, C_2, \ldots]^\top$$

Boundary and coboundary operators on 2-cells are expressed with the incidence matrix A_2, so that the canonical form $(\delta\,\Omega\,\partial)(c_2)$ may be represented

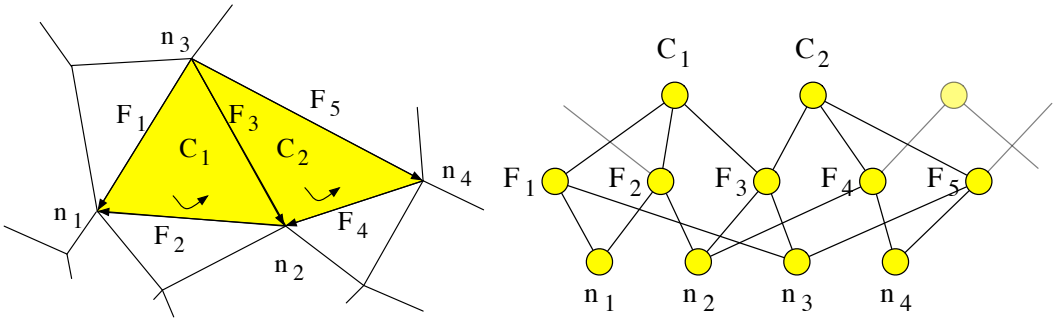

Figure 6. 2-cell complex with the corresponding Hasse diagram, focusing on two cells, C_1 and C_2.

with a product of matrices as $A_2^\top \, \Omega \, A_2 \, \xi$. The thermal tension between adjacent cells is then given by the 1-chain f_1:

$$\partial(c_2) = A_2 \, \xi = \begin{bmatrix} +1 & 0 \ldots \\ -1 & 0 \\ -1 & +1 \\ 0 & -1 \\ 0 & -1 \\ & \vdots \ddots \end{bmatrix}_{|K_1| \times |K_2|} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \end{bmatrix}_{|K_2| \times 1} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ \vdots \end{bmatrix}_{|K_1| \times 1} = f_1$$

where the same notation has been applied to the face vector.

The Ω matrix relates faces to faces, expressing the influence of thermal tensions in the cell complex represented by the 1-chain f_1, and producing the output 1-cochain of heat flux functions. Each coefficient ω_{F_i,F_j} of Ω will be the result of constitutive relations—involving metrical and physical quantities—and measures the heat flux between 1-faces F_i and F_j:

$$\Omega = \begin{bmatrix} \omega_{F_1,F_1} & \omega_{F_1,F_2} & \omega_{F_1,F_3} & \omega_{F_1,F_4} & \omega_{F_1,F_5} & \ldots \\ \omega_{F_2,F_1} & \omega_{F_2,F_2} & \omega_{F_2,F_3} & \omega_{F_2,F_4} & \omega_{F_2,F_5} & \ldots \\ \vdots & & \ddots & & & \end{bmatrix}_{|K_1| \times |K_1|}$$

The SPLIT algoritm used to refine a mesh cuts a cell with an hyperplane and updates the Hasse diagram: such an operation may lead to a complex constituted by homogeneous cells, in our example with reference to Figure 7, a simplex C_1 may be split into a simplex and a quadrilateral cell. While classical methods would enforce the last to be a triangle, adding new cuts in the mesh, our algebraic-topological approach allows the use of cells of different shapes

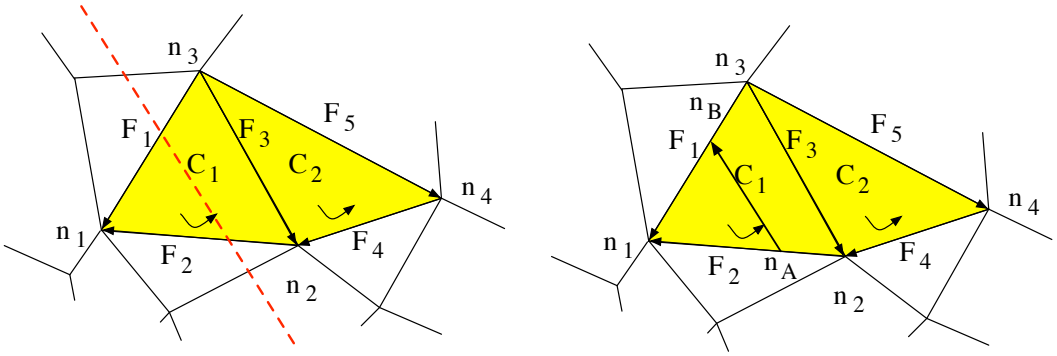

Figure 7. (a) Splitting the C_1 simplex with an hyperplane; (b) non-simplicial cells are generated.

producing self-similar matrices as the algorithm spits or collapses cells in the complex.

Updating the Hasse diagram starts from cells of lower dimension, creating two new 0-cells n_A and n_B, as we can see in Figures 7 and 8. This information is propagated to higher-dimensional cells, creating new faces F_A, F_B, F_C, F_D, F_E that substitute 1-cells F_1 and F_2.

Such an update involves not only the topological structure, but also impacts the Ω matrix by updating its elements on a metrical basis as opposed to a pure topological one. Each element ω_{F_i,F_j} represents the influence of thermal tensions on the faces F_i and F_j, strongly related to metrical properties such as faces volumes. The updated matrix Ω', of dimension $|K_1'| \times |K_1'|$, where $|K_1'| = |K_1| + 3$, will then reflect the newly created cells that replace F_1 and F_2:

$$\Omega' = \begin{bmatrix} \omega_{F_A,F_A} & \omega_{F_A,F_B} & \dots & \omega_{F_A,F_3} & \omega_{F_A,F_4} & \omega_{F_A,F_5} & \dots \\ \vdots & & & & & & \\ \omega_{F_E,F_A} & \omega_{F_E,F_B} & \dots & \omega_{F_E,F_3} & \omega_{F_E,F_4} & \omega_{F_E,F_5} & \dots \\ \omega_{F_3,F_A} & \omega_{F_3,F_B} & \dots & \omega_{F_3,F_3} & \omega_{F_3,F_4} & \omega_{F_3,F_5} & \dots \\ \vdots & & \ddots & & & & \end{bmatrix}_{|K_1'| \times |K_1'|}$$

The SPLIT algorithm proceeds propagating the informations upwards to the 0-codimension cells, and reconstructing at each step both the topological and the metrical-physical properties. The last step requires the update of the properties relative to 2-cells. This procedure will update the incidence matrix A_2 accordingly to the relative orientation of the new 2-cells C_A and C_B—that replace the split cell C_1—with respect to their faces, so giving the new matrix A_2' with dimensions $|K_1'| \times |K_2'|$, where $|K_2'| = |K_2| + 1$. With reference to Figure 9, the updated boundary chain f_1' can be expressed with the following

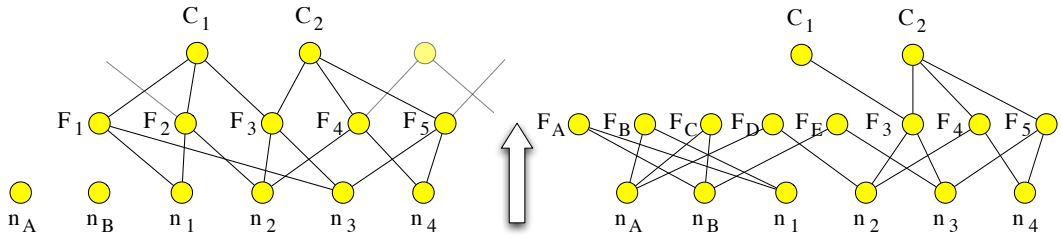

Figure 8. (a) Creation of two new 0-cells n_A and n_B; (b) the subsequent creation of 1-cells F_A, F_B, F_C, F_D, F_E, replacing F_1 and F_2.

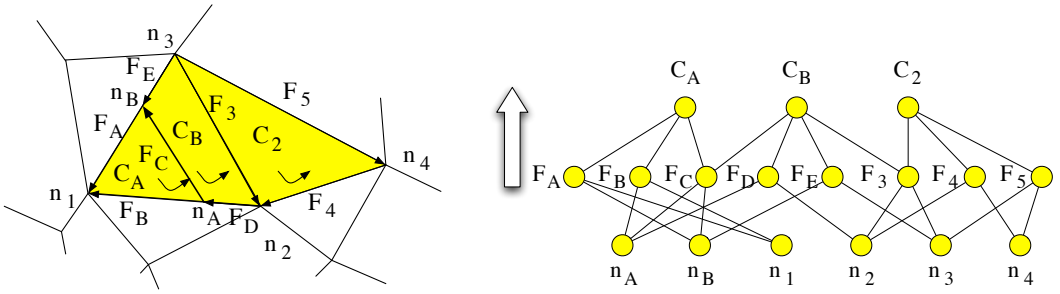

Figure 9. The final cell complex and its relative Hasse diagram.

product of matrices:

$$\partial(c'_p) = A'_2\,\xi' = \begin{bmatrix} +1 & 0 & 0 \dots \\ -1 & 0 & 0 \\ +1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & +1 & 0 \\ 0 & -1 & +1 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \\ & & \vdots \ddots \end{bmatrix}_{|K'_1|\times|K'_2|} \begin{bmatrix} C_A \\ C_B \\ C_2 \\ \vdots \end{bmatrix}_{|K'_2|\times 1} = \begin{bmatrix} F_A \\ F_B \\ F_C \\ F_D \\ F_E \\ F_3 \\ F_4 \\ F_5 \\ \vdots \end{bmatrix}_{|K'_1|\times 1} = f'_1$$

5 A sample finite difference construction

In this section we recover the standard finite-difference approximation of a problem in linear heat conduction through the algebraic-topological approach presented in the previous sections. To compare, we first present the conventional finite-difference construction of the same approximation.

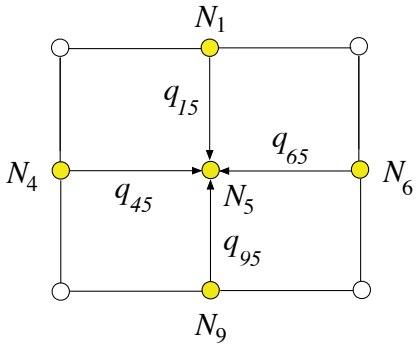

Figure 10. Five-point finite-difference scheme.

Let us approximate the two-dimensional domain with a uniform Cartesian mesh, and let T_i denote the value of the temperature at the node N_i. Let h_x and h_y be the mesh sizes in the x and y direction of the rectangles, respectively, and let us use a five-point difference stencil. With reference to Figure 10, let us focus on node N_5 and its adjacent nodes N_1, N_4, N_6 and N_9. The values of the temperature at the five nodes will be denoted T_5, T_1, T_4, T_6 and T_9, respectively.

Heat flux components, identified with the partial derivatives of the temperature, are approximated with the following divided differences (see Figure 10):

$$\begin{cases} q_{15} = (\lambda_{15}/h_y)(T_1 - T_5)(-1) \\ q_{65} = (\lambda_{65}/h_x)(T_6 - T_5)(-1) \\ q_{95} = (\lambda_{95}/h_y)(T_5 - T_9) \\ q_{45} = (\lambda_{45}/h_x)(T_5 - T_4) \end{cases} \tag{11}$$

where λ_{ij} is the material thermal conductivity and q_{ij} is the heat flux, both related to the nodes N_i and N_j: note that q_{ij} can be interpreted as the coefficients of a 1-cochain q^1). A balance equation is then associated to node N_5, stating that the contributions from all the adjacent 0-cells sum up to a given quantity, say zero:

$$\sum q_{ij} = q_{15} + q_{65} + q_{95} + q_{45} = 0\,. \tag{12}$$

Let us now restate the same procedure in our own algebraic-topological terms. As previously said, each node N_i will be represented by a 2-cell C_i, to which temperature values T_i will be attached. Let σ_{ij} denote the heat conductance between two adjacent cells C_i and C_j. Figure 11 shows the oriented cell complex corresponding to the FD mesh in Figure 10.

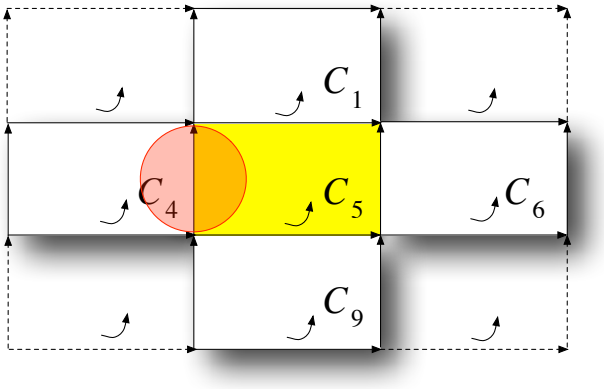

Figure 11. Cell-based finite difference scheme with its metrics.

We start from a 2-chain representing the temperature distribution over the cells:

$$T = \sum_i T_i \, C_i \tag{13}$$

Let us go to compute the canonical form $\delta \, \Omega \, \partial \, (T)$. The first operator to be applied is the boundary ∂, which will produce a 1-chain F of temperature differences. Writing down the components for the only 2-cell C_5, we obtain

$$\begin{cases} F_{15} = T_1 - T_5 \\ F_{65} = T_5 - T_6 \\ F_{95} = T_5 - T_9 \\ F_{45} = T_4 - T_5 \end{cases} \tag{14}$$

As expected, these coefficients are attached to the 1-faces of cell C_5.

The next step is the application of the mapping Ω, which results in the 1-cochain of heat fluxes through the 1-faces. This map relies on the metrics underlying our cell complex, and as shown in Figure 11, our choice is to map each 1-face with all the 1-cells within an euclidean distance of $2\vartheta = \min\{\mathrm{Vol}(F) | F \in K_1\}$ from the center of mass of each face, where K_1 is the 1-complex extracted from the mesh. Representing Ω with a matrix, its general element ω_{ij} will be null unless cells C_i and C_j are adjacent, in which case it will equal σ_{ij}:

$$\Omega = [\omega_{i,j}] \, , \quad \omega_{i,j} = \begin{cases} 0, & C_i \bigcap C_j = \emptyset \\ \sigma_{ij}, & C_i \bigcap C_j \neq \emptyset \end{cases} \tag{15}$$

This gives us a 1-cochain q. Writing down the components for the 1-faces of the 2-cell C_5, we have:

$$\begin{cases} q_{15} = \sigma_{15}(T_1 - T_5) \\ q_{65} = \sigma_{65}(T_5 - T_6) \\ q_{95} = \sigma_{95}(T_5 - T_9) \\ q_{45} = \sigma_{45}(T_4 - T_5) \end{cases} \tag{16}$$

Finally, we have to apply the coboundary operator to the 1-cochain q. This operation sums up the fluxes attached to all the faces of each cell, producing a 2-cochain Q'^2 which provides the total heat flux entering each cell. Focussing as before on cell C_5, this gives the equation:

$$\delta\, q = \sum q'_{ij} = -q'_{15} + q'_{65} + q'_{95} - q'_{45} = 0 \tag{17}$$

which coincides with Equation (12), provided that

$$\begin{cases} \sigma_{15} = \lambda_{15}/h_y \\ \sigma_{65} = \lambda_{65}/h_x \\ \sigma_{95} = \lambda_{95}/h_y \\ \sigma_{45} = \lambda_{45}/h_x \end{cases} \tag{18}$$

We may notice as h_x and h_y both depend on the metrics as well as λ_{ij}: the coefficient σ depends on the material property λ which may be related to face volumes, for instance $\lambda_{ij} \propto \mathrm{Vol}(F_{ij})$.

In case of a SPLIT, the structure of our problem remains the same. With reference to Figure 12, splitting the 2-cell C_5 results in a change in the ϑ distance used to formulate the problem, changing the minimum volume among faces in the cell complex. This fact is reflected on the Ω matrix relating each face to its immediate neighbor preserving the five-point finite difference structure. As presented in Section 3, these metrical informations are stored and comfortably updated in the Hasse diagram as the SPLIT algorithm proceeds.

6 Prospective developments

This paper has shown a physically valid and theoretically straightforward interpretation of finite methods based on an algebraic-topological formulation

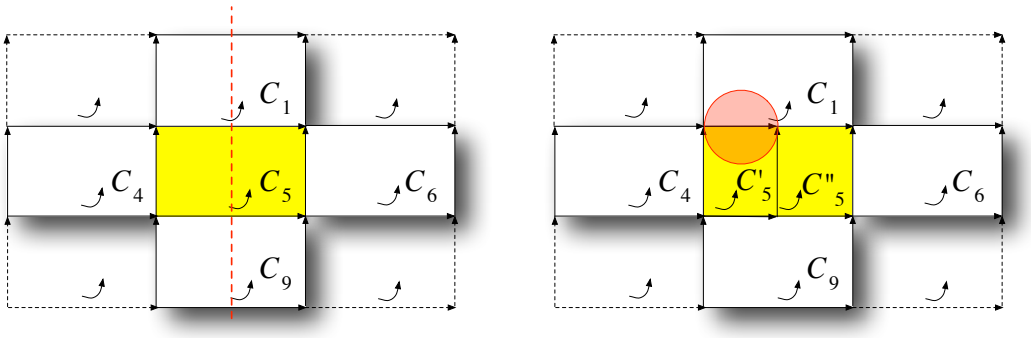

Figure 12. Splitting the cell C_5 yields a change in the distance used to describe face relationships, affecting the $\sigma_{ij} = \sigma(\text{Vol}(F_{ij}))$ parameters.

of field problems. Focusing on the cells of 0-codimension allows us to design the geometry and simulate the physics simultaneously, removing most artificial constraints on the shape of finite elements. Moreover, the separation of metrical and physical properties from the purely-topological ones unifies all the finite methods within a single algebraic-topological framework. In the proposed approach, we make use of an efficient full-dimensional graph-based decomposition of the domain of interest, supporting a streaming-like progressive refinement in case it is needed, and only where it is required.

It should be clear that separating topology from other factors eases the process of mixing different finite methods on different domain regions, since each method influences only the Ω matrix, as we have seen in Section 3. When describing multiple finite approaches the Ω matrix will become a block-structured matrix, where each block $\Omega_{i,i}$ implements the chosen finite method for the given subdomain, and $\Omega_{i,j}$, with $i \neq j$, describes the interactions between different regions:

$$\Omega = \begin{bmatrix} \Omega_{1,1} & \Omega_{1,2} & \dots \\ \vdots & & \ddots \end{bmatrix}.$$

The same consideration applies when each subdomain involves different physical phenomena, since all the field knowledge is limited to the Ω matrix. We may also notice as the blocks are not to be considered "*sharply defined*" but "*fuzzy bounded*". At the frontier of each $\Omega_{i,j}$ subdomain there will be a "*mutual interaction*" zone whose extent is defined by the physical (and metrical) description of the given problem, and of course is not cleanly confined by subdomain boundaries.

Another promising research direction is the application of the framework presented in this paper in a mixed multi-grid/multi-physics environment. The current multi-grid approach approximate solutions on multiple mesh at different granularity, interchanging results with relaxation $\mathcal{R}$ (fine-to-coarse grid)

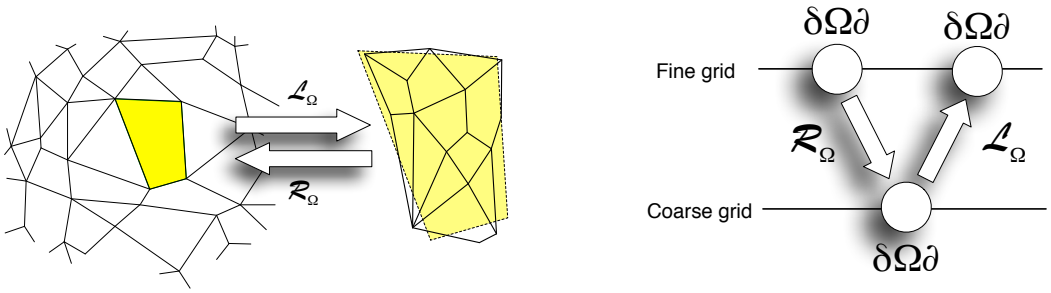

Figure 13. A mixed multi-grid/multi-physics approach mesh (a) and V-Cycle (b).

and interpolator $\mathcal{L}$ operators (coarse-to-fine grid), see e.g. (27) and (28). More than merely coarse-refine tools, these operators can carry physical informations between layers of meshes, describing the mutual influence joining multiple cell complexes. These operators $\mathcal{L}_\Omega$ and $\mathcal{R}_\Omega$ will modify the Ω matrix at each stage of computation as the subscript suggests. Then a typical multi-grid cycle, which solves a problem on the fine grid, pushes the solution to the coarser one to obtain a new initial result for the finer mesh, called a V-Cycle, would not only transfer values between different level of details, but in addition influence the physical description:

$$\delta\Omega_\xi \partial\, \xi \xrightarrow{\mathcal{R}_\Omega} \delta\Omega_\eta \partial\, \eta \xrightarrow{\mathcal{L}_\Omega} \delta\Omega'_\xi \partial\, \xi' \xrightarrow{\mathcal{R}_\Omega} \delta\Omega'_\eta \partial\, \eta' \xrightarrow{\mathcal{L}_\Omega} \dots$$

As an example of this possibility, we may analyze a heat transfer problem at the coarse level, and an electrical charge flow at the finer grid. These two layers may interact by changing the impedance—that is implemented by the Ω_ξ matrix—according to the temperature, as well as the thermal conductivity expressed in Ω_η.

Acknowledgments

The authors are grateful to Dr. Igor Tsukanov for several helpful discussions. The work of the last author is supported in part by the National Science Foundation grants DMI-0500380, DMI-0323514, DMI-0621116 and OCI-0537370.

References

[1] H. Lodish, A. Berk, L. S. Zipursky, P. Matsudaira, D. Baltimore, J. Darnell, Molecular Cell Biology, W. H. Freeman, New York, NY, USA, 2000, fourth edition.

[2] D. Boal, Mechanics of the Cell, Cambridge University Press, Cambridge, UK, 2005, third edition.

[3] J. C. Maxwell, On the mathematical classification of physical quantities, in: W. Niven (Ed.), The Scientific Papers of James Clerk Maxwell, Cambridge University Press, 1890, pp. 257 – 266.
[4] G. Kron, Numerical solution of ordinary and partial differential equations by means of equivalent circuits, Journal of Applied Physics 126 (1945) 172–186.
[5] J. P. Roth, An application of algebraic topology to numerical analysis: On the existence of a solution to the network problem, Proc. Nat. Acad. Sci 41 (1955) 518–521.
[6] F. H. Branin, The algebraic-topological basis for network analogies and the vector calculus, in: Proceeding of the Symposium on Generalized Networks, Vol. 16, Polytechnic Institute of Brooklyn, 1966, pp. 453–491.
[7] E. Tonti, On the formal structure of physical theories, Tech. rep., Istituto di Matematica del Politecnico di Milano (1975).
[8] R. Palmer, V. Shapiro, Chain models of physical behavior for engineering analysis and design, Research in Engineering Design 5 (3) (1993) 161–184.
[9] P. Bamberg, S. Sternberg, A Course in Mathematics for Students of Physics, Cambridge University Press, Cambridge, England, 1988,1990.
[10] R. S. Palmer, Chain models and finite elements analysis: an executable chains formulation of plane stress, Computer Aided Geometric Desisgn 12 (7) (1995) 733–770.
[11] R. Egli, N. F. Stewart, A framework for system specification using chains on cell complexes, Computer Aided Design 32 (9) (2000) 447–459.
[12] R. Egli, N. F. Stewart, A framework for system specification using chains on cell complexes, Mathematics and Computers in Simulations 66 (2004) 449–468.
[13] W. Schwalm, B. Moritz, M. Giona, Vector difference calculus for physical lattice models, Physical Review E 59 (1).
[14] G. Strang, A framework for equilibrium equations, SIAM Review (30) (1988) 283–297.
[15] C. Mattiussi, An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology, Journal of Computational Physics 133 (1997) 289–309.
[16] E. Tonti, A direct discrete formulation of field laws: The cell method, Computer Modeling in Engineering & Sciences 2 (2) (2001) 237–258.
[17] J. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient and curl on logically rectangular grids, International Journal of Computers and Mathematics with Applications 33 (4) (1997) 81–104.
[18] J. A. Chard, V. Shapiro, A multivector data structure for differential forms and equations, IMACS Transactions Journal, Mathematics and Computer in Simulation 54 (1) (2000) 33–64.
[19] V. Ramaswamy, V. Shapiro, Combinatorial laws for physically meaningful design, Journal of Computing and Information Science in Engineering 4 (3) (2004) 3–10.
[20] A. Paoluzzi, V. Pascucci, M. Vicentino, Geometric programming: a pro-

gramming approach to geometric design, ACM Trans. Graph. 14 (3) (1995) 266–306.
[21] G. Scorzelli, A. Paoluzzi, V. Pascucci, Parallel solid modeling using BSP dataflow, Journal of Computational Geometry and Applications 16, accepted for publication.
[22] A. Paoluzzi, V. Pascucci, G. Scorzelli, Progressive dimension-independent boolean operations, in: G. Elber, N. Patrikalakis, P. Brunet (Eds.), Proceeding of the 9-th ACM Symposium on Solid Modeling and Applications, ACM, 2004, pp. 203–212.
[23] J. R. Munkres, Elements Of Algebraic Topology, Addison Wesley, Reading MA, 1984.
[24] W. Franz, Algebraic Topology, Frederick Ungar Publishing Co., New York, NY, 1968.
[25] S. Skiena, Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Addison-Wesley, Reading, MA, 1990.
[26] C. L. Bajaj, V. Pascucci, Splitting a complex of convex polytopes in any dimension, in: SCG '96: Proceedings of the twelfth annual symposium on Computational geometry, ACM Press, Philadelphia, PA, 1996, pp. 88–97.
[27] Multigrid Methods, Lecture Notes in Pure and Applied Mathematics, Dekker, 1988.
[28] Matrix-Based Multigrid: Theory and Applications, Numerical Methods and Algorithms, Kluwer Academic Publishers, 2003.