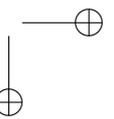
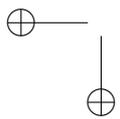
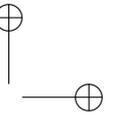
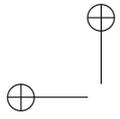




Roma Tre University
Ph.D. in Computer Science and Engineering

Towards topological unification of finite computational methods

Franco Milicchio



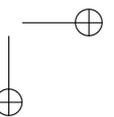
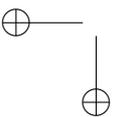
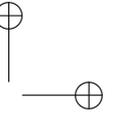
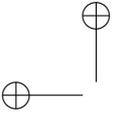
Towards topological unification of finite computational methods

A thesis presented by
Franco Milicchio
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Engineering
Roma Tre University
Dept. of Informatics and Automation
March 2007

COMMITTEE:
Prof. Alberto Paoluzzi

REVIEWERS:
Prof. Chandrajit Bajaj
Prof. Vadim Shapiro

Per i miei Genitori



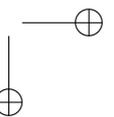
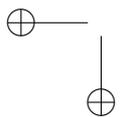
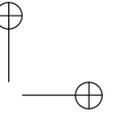
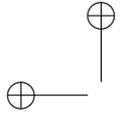
Abstract

Traditional engineering practice requires the iteration of shaping and modeling geometric and material properties, simulating and evaluating the results, eventually redesigning. This trial-and-error course of action has not considerably changed during years, and to simplify approaches, boundary representations became the most prominent data structures in almost all academic and industry-level commercial solid modeling systems.

In this thesis, we advocate the cochain complex represented by a sparse block-bidiagonal matrix that we call the Hasse matrix, which captures all combinatorial relationships present in geometric data structures for both primal and dual complexes, without consideration of the representation of a cell complex, such as and manifold or and simplicial decompositions.

Additionally we also show that refinements that preserve topological features of such cell complexes correspond to simple Euler operators, which can be simply formulated as multi-linear transformations acting on the Hasse matrix. An additional well defined Euler operator will be shown, expressing the splitting operator in algebraic topological terms.

We also claim that all cell complexes representations are accurately represented by a cochain complex, encapsulating all combinatorial relationships of interest in solid and physical modeling, formally and unambiguously, with the use of standard operators of algebraic topology such as boundary and coboundary. This approach combines the geometric model, the description of its physical properties, and the simulation of the relevant patterns emerging from both geometry and physics.



Acknowledgments

Mathematics is the language with which God has written the universe.
Galileo Galilei, 1564–1642

First of all I have to thank my family for supporting me on this “useless” journey. An educational tour that I’d define useless in the sense that the more you know, the more you are aware that there are so many other things to study. In this view, I appreciate Socrates and his view of knowledge, since I completely agree that “I know that I don’t know”.

Next, I want to thank my advisor Prof. Alberto Paoluzzi for his belief in this work, and more than that. Moreover, I have to thank him for introducing me in the field of algebraic topology by letting me visit the University of Wisconsin-Madison. There I found another inspiration in Prof. Vadim Shapiro, and my gratefulness goes to him without any doubt. Finally I must thank Prof. Antonio DiCarlo for his advices on matters that not many can handle with such a flawless confidence.

I had two different lives during this voyage, and my advisor well knows all the difficulties in confronting two different worlds, engineering and computer science. For my second life, I’d like to thank Dr. Wolfgang Alexander Gehrke for the work we did together, producing something that we both hope to be appreciated. Last in this parade, the gratitude surely goes to Prof. Carla Limongelli, since she must have been a little uncomfortable during the writing of our book.

My indebtedness goes to all my friends, Cristiano Iacoangeli, Damiano Mazza, Mattia Prospero, Fabrizio Pecoraro, Marco Donati, Laura Fiorini and Fabrizia Zacchia. No words can express the friendship, love and gratitude. Last

x

but not least I must thank Cinzia Costa, for letting me see the truth coming from years of blindfolded wandering.

To all others, you will always have my gratitude with spoken and even unspoken words. Life's path is full of people we shall thank, persons who leave a mark as soon as we accept what life prepares to us, and start listening:

It is an ancient Marinere,
And he stoppeth one of three:
"By thy long grey beard and thy glittering eye
"Now wherefore stoppest me?"

The Bridegroom's doors are open'd wide
"And I am next of kin;
"The Guests are met, the Feast is set, —
"May'st hear the merry din. —

But still he holds the wedding-guest —
There was a Ship, quoth he —
"Nay, if thou'st got a laughsome tale,
"Marinere! come with me."

He holds him with his skinny hand,
Quoth he, there was a Ship —
"Now get thee hence, thou grey-beard Loon!
"Or my Staff shall make thee skip.

He holds him with his glittering eye —
The wedding guest stood still
And listens like a three year's child;
The Marinere hath his will.

The Rime of the Ancient Marinere, in seven parts
Samuel Taylor Coleridge, 1798

Contents

Contents	xi
List of Figures	xiii
1 Introduction	1
1.1 Preview	3
2 Mathematical Background	5
2.1 Homology and Cohomology	7
2.2 Differential Geometry	14
3 Finite Methods Background	21
3.1 Finite Differences	21
3.2 Finite Elements	25
3.3 Finite Volumes	32
4 (Co)Chain Complex Modeling	39
4.1 Representing (co)chains	39
4.2 Matrix representation of a Complex	43
4.3 Euler operators	45
4.4 Hasse transformations	52
4.5 Hyperplane splitting	54
4.6 Geometry & physics modeling	59
5 Canonical Form of Finite Methods	61
5.1 Background	61
5.2 Discretization of Field Problems	63
5.3 Adaptive Graph Representation	72

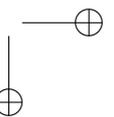
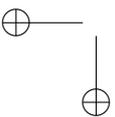
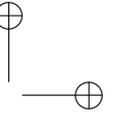
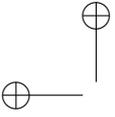
5.4	A Sample Finite Difference Construction	78
6	A Metrical Example	83
6.1	The Normal Gradient	83
6.2	A First Approximation	85
6.3	A Finer Estimate	87
	Conclusions	97
	Final Remarks	97
	Future Work	98
	Appendix	103
	Bibliography	107

List of Figures

2.1	Two homeomorphic spaces and two non-homeomorphic ones.	5
2.2	An example of 1-, 2- and a 3-simplexes.	6
2.3	Simplicial complexes: only K_1 and K_2 are valid complexes.	7
2.4	The simplexes of Figure 2.2 with an orientation.	8
2.5	The unit chain on a coherently oriented 2-complex.	10
2.6	The boundary of the chain in Figure 2.5.	10
2.7	A 1-cochain γ^1 on a cell complex.	13
2.8	The coboundary $\gamma^2 = \delta^1(\gamma^1)$ of the 1-cochain in Figure 2.7.	14
2.9	A manifold with two atlases ψ_α and ψ_β	15
2.10	A manifold obtained by the bundle X and fibre Y	17
2.11	A manifold X with a trajectory g and a scalar field f	17
2.12	(a) Two tangent trajectories, and (b) two tangent scalar fields on a manifold X	18
3.1	An example of finite difference approximation.	24
3.2	An example of an elliptic problem.	29
3.3	The piecewise linear finite element basis functions.	30
3.4	A piecewise linear FE basis for a 2-dimensional problem.	31
3.5	Example of a quadratic base function on a triangle.	32
3.6	A finite volume <i>three point stencil</i> schema.	34
3.7	The Gudnov algorithm constraints for the CFL condition.	36
3.8	The discrete approximation of the domain \mathcal{D}	37
3.9	The corner transport upwind updating schema.	38
4.1	A 2-complex K , whose 2-cells are coherently oriented.	41
4.2	The whole scheme holds for d odd; for d even, the last block-row should be discarded.	44

4.3	The Hasse diagram of the chain complex representing the topology of a 3-cube.	44
4.4	Coarse complex $K = (K_0, K_1, K_2)$	49
4.5	First refinement step: $\tilde{K} = \beta^0(K) = (K_0 \cup \{\sigma_0^4\}, K_1 \cup \{\sigma_1^4\}, K_2)$	49
4.6	Second refinement step: $\tilde{\tilde{K}} = \beta^1(\tilde{K}) = (\tilde{K}_0, \tilde{K}_1 \cup \{\sigma_1^5\}, \tilde{K}_2 \cup \{\sigma_2^2\})$	50
4.7	The SPLIT algorithm, implemented by using a classification chain and the coboundary operator.	56
4.8	(a) The splitting hyperplane h , and (b) the classification of vertices.	56
4.9	The 1-chains c_1 and f_1 used to detect the 1-cells that intersect the splitting hyperplane.	57
4.10	The updated cell complex, with 1-cells reclassified.	57
4.11	(a) Classification of the 2-cells, (b) the classification 1-chain on the refined 1-skeleton, and (c) the refined 2-skeleton with the classification 2-chain.	58
5.1	Primal and dual complexes: (a) duality between K_2 and D_0 ; (b) duality between K_1 and D_1	64
5.2	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 : $\partial(2\sigma) = \phi(\delta(\phi^{-1}(2\sigma)))$	65
5.3	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 : $\delta(\gamma) = \phi^{-1}(\partial(\phi(\gamma)))$	66
5.4	Different shape functions on the same mesh.	72
5.5	2-cell complex with the corresponding Hasse diagram, focusing on two cells, C_1 and C_2	75
5.6	(a) Splitting the C_1 simplex with an hyperplane; (b) non-simplicial cells are generated.	76
5.7	(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	77
5.8	The final cell complex and its relative Hasse diagram.	78
5.9	Five-point finite-difference scheme.	79
5.10	Cell-based finite difference scheme with its metrics.	79
5.11	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.	81
6.1	The domain \mathcal{D} of equation (6.1).	85
6.2	The mesh corresponding to the domain \mathcal{D}	86

<i>List of Figures</i>	xv
6.3 A cell in the discretized version of the domain \mathcal{D}	86
6.4 Image of the function f in equation 6.2, with $\ell = 50$ and $m = 25$. . .	87
6.5 Samples of the field in Figure 6.4.	88
6.6 The gradient function of Equation (6.2).	89
6.7 The normal gradient function f with $\ell = 50$ and $m = 25$	90
6.8 Normal gradient with a naïve metrical approach.	91
6.9 Image of the function f , with $\ell = 50$ and $m = 250$	92
6.10 Samples of the function pictured in Figure 6.9.	93
6.11 The metrical relation of equation (6.3).	93
6.12 Numbering of 1-cells of the mesh.	94
6.13 Normal gradient $\nabla f \cdot n$, with $\ell = 50$ and $m = 250$	94
6.14 The resulting normal gradient.	95
1 A mixed multi-grid/multi-physics approach mesh (a) and V-Cycle (b).	99
2 A 3-complex $K := (K_0, K_1, K_2, K_3)$	104



Chapter 1

Introduction

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).

The boundary representation has historically become the representation of choice in many academic and virtually all commercial solid modeling systems. As a consequence, most geometric, scientific and engineering applications have to be formulated in terms of boundary representations, often leading to non-trivial representation conversion problems. Well known examples of such problems include Boolean set operations, finite element meshing, and subdivision algorithms.

Formally, all boundary representations are widely recognized as graph-based data structures [Bau72, GS85, Män88, Bri93] representing one of several possible cells complexes [Req77, Req80, Sil81, OR90]. Space requirements and computational efficiency of such data structures have been studied in the literature (see, e.g., [Woo85]). Historically, such cell complexes have been restricted to (unions of) two-dimensional orientable manifolds, but a number of extensions to more general orientable cellular spaces have been proposed (see, e.g., [Mas93, YK95, OR90]). Depending on a particular choice of data struc-

tures, boundary representations are constructed, edited, and updated using a small set of basic operators on the graph representation, while preserving and/or updating the basic topological invariants of the cell complex. Such operators are commonly called Euler operators [EW79, Män88, Mas93], because they enforce the Euler-Poincaré formula. All higher-level algorithms and applications of boundary representations are implemented in terms of such operators.

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. [LBZ⁺00, Boa05]).

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.

A mere restriction to boundary representations obviously decreases the size of a model, but the development of such representations led to several fundamental difficulties:

- Variety of assumptions about the cell complexes and graph representations make standardization difficult. This in turns complicates the issues of data exchange and transfer, and leads to proliferation of incompatible algorithms.
- Boundary representation algorithms are dominated by graph searching algorithms (boundary traversals) that tend to force serial processing. Nor

1.1. PREVIEW

3

is it clear how to combine such graph representations with multi-resolution representations and algorithms.

- Extending boundary representations to more general cellular spaces has proved challenging. Despite many proposals, most commercial systems are still restricted to two-dimensional orientable surfaces.
- Last, but not least, solid modeling has developed into a highly specialized discipline that is largely disconnected from many standard computational techniques. In particular, boundary representations do not appear to be directly related to the methods for physical analysis and simulation such as finite differences, finite elements, and finite volumes.

In this thesis, we show that the (co)chain complex can be represented by a sparse block-bidiagonal matrix that we call the Hasse matrix. The combinatorial relationships present in geometric data structures can be represented by the Hasse matrix which captures combinatorial relationships for both primal and dual (co)chain complexes, regardless of the representation of a cell complex, such as triangulations and manifold decomposition.

We also show that topology-preserving refinements of such cell complexes correspond to simple Euler operators and are easily formulated as multi-linear transformations of the Hasse matrix. The splitting operator will be shown to be expressible in algebraic topological terms, promoting it as a well defined Euler operator.

Moreover, we claim that all representations of cell complexes are properly represented by a (co)chain complex [Mun84, Hat02]. It captures all combinatorial relationship of interest in solid and physical modeling formally and unambiguously, using standard algebraic topological operators of boundary ∂ and coboundary δ . Such an approach 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.

1.1 Preview

Chapter 2 will introduce some basic concepts from algebraic topology, and in the following of the same chapter a brief introduction to differential geometry and forms will be given. Chapter 3 will outline the three major methods for solving field problems, namely the finite differences, finite elements and finite volumes. In chapter 4 we will introduce a matrix-based representation for the

chain (cochain) complex derived from the Hasse diagram, giving an insight on the transformations of a complex preserving the Euler characteristic. Chapter 5 will formulate an algebraic topological approach to field problems, expressing a canonical form that separates metrical and physical information from topological details, proving that all finite methods are essentially equivalent (under the canonical form). In chapter 6 we will sketch a numerical example involving the approximation of the normal gradient of a scalar field over a toroidal domain.

Chapter 2

Mathematical Background

Before proceeding with the introduction of basic concepts of algebraic topology and differential geometry, we will introduce common notions that will be the fundamentals of the following sections.

One of the underlying concepts of algebraic topology is the classification of spaces having “*the same shape*”, thus considered equivalent, or more precisely *homeomorphic spaces*:

Definition 1 (Homeomorphism). *Two spaces X and Y are homeomorphic if there exists a continuous map $f : X \rightarrow Y$, with continuous inverse.*

Such a definition induces the concept of deformation of a space into another, considering two objects similar and creating equivalence classes. For instance there exists a continuous map from a five-pointed star into a circle in \mathbb{R}^2 , as pictured in Figure 2.1, but the same star with a hole is not homeomorphic to the same circle—though it is clearly homeomorphic to a 2-dimensional torus. A famous result of this property is that, in general, \mathbb{R}^n and \mathbb{R}^m are not homeomorphic with $n \neq m$. Introducing the concept of *d-Ball* as the subset $\{x \in \mathbb{R}^d : \|x\| \leq 1\}$, we can define the basic object called *cell*, as a subset $c \subset \mathbb{R}^d$ homeomorphic to an *d-Ball*.



Figure 2.1: Two homeomorphic spaces and two non-homeomorphic ones.

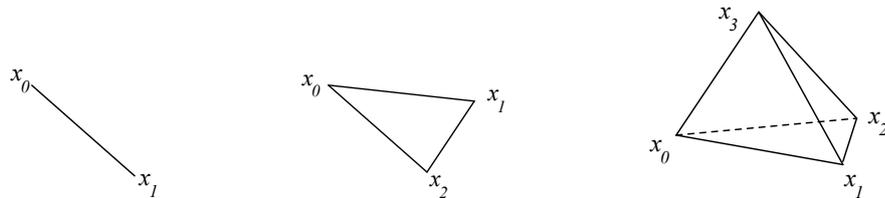


Figure 2.2: An example of 1-, 2- and a 3-simplices.

Cells are the “building blocks” of all the following definitions, and basic concepts for homology and cohomology. Let us start with defining a set of points $\{x_0, \dots, x_p\}$ as *geometrically independent* if $\forall t_0, \dots, t_p \in \mathbb{R}$, with

$$\sum_{i=0}^p t_i = 0, \quad \text{and} \quad \sum_{i=0}^p t_i x_i = 0,$$

results $t_i = 0, \forall i = 1, \dots, p$. The geometric independence may be expressed also with notions from the standard linear algebra, having that $\{x_0, \dots, x_p\}$ are geometrically independent if and only if $x_i - x_{i+1}$ are linearly independent, with $0 \leq i < p$. It follows the standard definition

Definition 2 (Simplex). *Let $\{x_0, \dots, x_p\}$ be a geometrically independent set in \mathbb{R}^d . We define p-simplex σ as the set of all points $x \in \mathbb{R}^d$ such that*

$$x = \sum_{i=0}^p t_i x_i \quad \text{with} \quad \sum_{i=0}^p t_i = 1, \quad t_i \geq 0 \quad \forall i.$$

We define also the coefficients t_i the barycentric coordinates of $x \in \sigma$ with respect to $\{x_0, \dots, x_p\}$, and the set $\{x_i\}$ the set of vertices of σ .

It is obvious that a simplex is a cell, being homeomorphic to a ball. Such a fundamental object is then a segment when defining a 1-simplex, a triangle and a tetrahedron for a 2- and 3-simplex, respectively, as pictured in Figure 2.2. A 0-simplex is, by definition, a single point $\sigma \equiv \{x_0\}$. We may recall the definition of *convex set* as a set $A \subseteq \mathbb{R}^d$ where for each pair $x, y \in A$, the segment connecting the two points is entirely contained in A . Then a simplex is the union of all the segments connecting one of the vertices to all the others, known also as *convex combination* of the vertices. A simplex composed by a subset of the vertices of a simplex σ is called *face* of σ , in particular any face $\tilde{\sigma} \neq \sigma$ is a *proper face*.

2.1. HOMOLOGY AND COHOMOLOGY

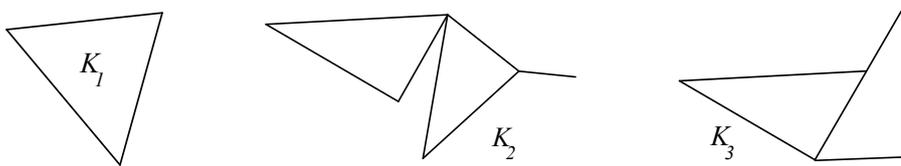


Figure 2.3: Simplicial complexes: only K_1 and K_2 are valid complexes.

Definition 3 (Simplicial Complex). *A set of simplexes $K = \{\sigma_i\} \subseteq \mathbb{R}^d$ is called simplicial complex if*

1. *Every face of any simplex $\sigma \in K$ is in K ;*
2. *The intersection of two simplexes of K is a face of both of them.*

For instance, given the set of simplexes in Figure 2.3, we notice that K_1 , a single simplex, is a valid simplicial complex; K_2 is also a legal complex since the intersection of any of the simplexes are faces for both of them, sharing a common vertex—a 0-simplex. The last of the series, K_3 , cannot be classified as a simplicial complex since the intersection of the two simplexes is a face to just one of them.

Let us define some common objects in a cell complex. A subset $L \subseteq K$ containing all faces of its elements, is called *subcomplex* and is, by definition, a simplicial complex by itself. The collection of all the simplexes of dimension p in a complex K is called *p-skeleton* of K and denoted as K_p . In particular, K_0 is the set of all the vertices of K . The set $[K] \subseteq \mathbb{R}^d$ union of all the simplexes of K is called *underlying space*, or *support* of the simplicial complex K . The *dimension* of a complex is defined as the maximum dimension of all its simplexes, and the complex is usually referred as a p -complex expressing its dimension.

2.1 Homology and Cohomology

Before defining homological entities and their cohomological counterparts we have to briefly reformulate the notions of simplex and complex adding an *orientation* to them. Until now cells were defined as purely geometric objects without any particular order of their vertices. Let us define two *ordered sets* as *equivalent* if one can be obtained from the other with an even number of

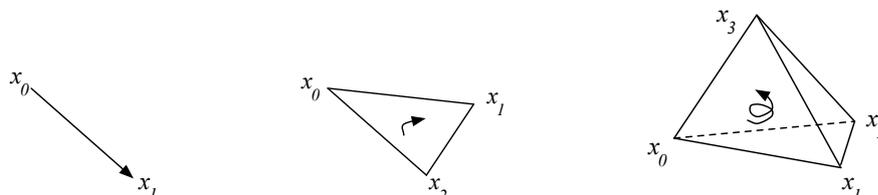


Figure 2.4: The simplices of Figure 2.2 with an orientation.

permutations of its elements. With this equivalence between sets we can add an orientation to simplices as follows:

Definition 4 (Oriented Simplex). *Given the set $\{x_0, \dots, x_p\}$ of $p+1$ independent points, an oriented p -simplex is defined as*

$$\sigma_p := [x_0, \dots, x_p] \subset \mathbb{R}^d, \quad p \leq d.$$

The notation $[x_0, \dots, x_p]$ indicates the ordered set of cells, considering the equivalence classes induced by the odd/even permutation of elements. As an example, in Figure 2.4 are depicted three oriented simplices. The 2-simplex $[x_0, x_1, x_2]$ falls in the same equivalence class of $[x_1, x_2, x_0]$, while $[x_0, x_2, x_1]$ would have opposite orientation. The same considerations apply to the 3-simplex $[x_0, x_1, x_2, x_3]$.

The simplex $[x_1, x_0, x_2, \dots, x_p]$, or any simplex in the same equivalence class, 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 simplices such that: (i) if a simplex is in K , then so are all its faces, and (ii) every two simplices 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).

Homology

In the following we are going to introduce the basic concepts of homology. We have already presented oriented simplicial complexes, indicating different orientations of a simplex with a minus sign, as in $-\sigma$. The sign can be regarded as the *coefficient* -1 “attached” to the simplex σ : -1σ . Such a concept is a function called *chain*, which is formally described as:

2.1. HOMOLOGY AND COHOMOLOGY

9

Definition 5 (Chain). *Let $(G, +)$ be a free abelian (i.e., commutative) group. A p -chain of K with coefficients in G is a mapping $c_p : K_p \rightarrow G$ such that*

$$c_p(-\sigma) = -c_p(\sigma), \quad \sigma \in K_p.$$

Chain addition is defined by sum of chain values: if c_p and 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 $C_p(K; G)$, the *chain group*.

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 $g\sigma$. Each chain can then be written in a unique way as a finite sum of elementary chains:

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

Let us define the basic homomorphism on chain groups:

Definition 6 (Boundary). *The boundary operator is an homomorphism on chains $\partial_p : C_p(K; G) \rightarrow C_{p-1}(K; G)$ defined as follows:*

$$\partial_p \sigma_p := \sum_{k=0}^p (-1)^k \sigma_{p-1,k}, \tag{2.1}$$

where $\sigma_{p-1,k}$ denotes the k -th face of σ_p , and then extended to elementary chains, by taking

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

The boundary operator is applied to all chains by assuming the additivity of ∂_p . The definitions of chain and boundary may be easily extended to *cell complexes*, by partitioning them into simplexes and applying the additivity of ∂_p .

The operator name *boundary* becomes evident when the group on which the chain is defined is the smallest non-trivial group $G = \{1, 0, -1\}$. The coefficients assigned by chains to simplexes on a complex select, discard or select a simplex inverting its orientation. If we apply the boundary on a unit chain (i.e. a chain whose coefficients are 1 for all the cells in the complex) defined on a coherently oriented cell complex of dimension p , the obtained $(p-1)$ -chain is constituted only by the “boundary” cells. For instance, let us define a unit chain $c_2 = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4$ on the (non simplicial) 2-complex pictured in Figure 2.5. The boundary of such a chain is a $(p-1)$ -chain $c_1 := \partial_2 c_2 = \tau_1 + \tau_3 + \tau_4 + \tau_8 + \tau_9$,

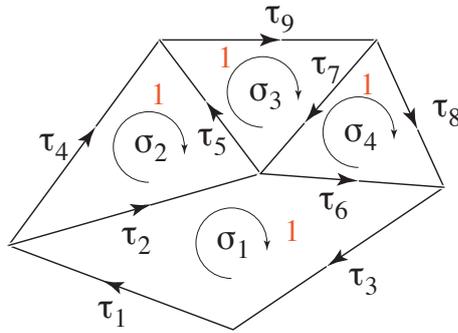


Figure 2.5: The unit chain on a coherently oriented 2-complex.

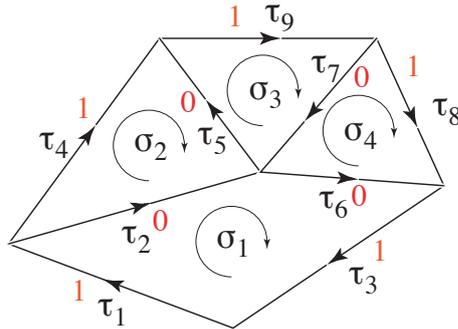


Figure 2.6: The boundary of the chain in Figure 2.5.

as in Figure 2.6, with all the internal $(p - 1)$ -faces τ_2 , τ_5 , τ_6 , and τ_7 discarded by the application.

We can define then the group of *cycles* as the null space of the boundary operator, denoted as $Z_p(K; G) := \text{Ker}(\partial_p)$ (from the german word *Zyklus*), and the group of *boudaries* as the image of the operator: $B_p(K; G) := \text{Im}(\partial_p)$. We define the quotient space

$$H_p(K; G) := Z_p(K; G) / B_p(K; G),$$

as the *homology group* of K with coefficients in G .

2.1. HOMOLOGY AND COHOMOLOGY

11

The boundary operator has the fundamental property:

$$\partial_p \circ \partial_{p+1} \equiv 0, \tag{2.2}$$

where 0 denotes the null chain. It follows directly from this property that $B_p(K; G) \subset Z_p(K; G)$: every boundary of a $(p + 1)$ -chain is a p -cycle.

A core concept in algebraic topology is the pairing between a complex and the complex constituted by chains. Such an object is formally defined as follows:

Definition 7 (Chain Complex). *A chain complex $\mathcal{C} \{C_p, \partial_p\}$ is a sequence*

$$\dots \longrightarrow C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \longrightarrow \dots$$

of abelian groups C_i , paired with homomorphisms ∂_i , with $i \in \mathbb{N}$, satisfying $\partial_p \circ \partial_{p+1} = 0, \forall p$.

If $C_p = 0$ for $p < 0$ then we define the chain complex $\mathcal{C} \{C_p, \partial_p\}$ as *non-negative*; if C_p is a free abelian group (i.e. it has a basis) then the chain complex is called *free*. A special mapping between chain complexes may be established as the following:

Definition 8 (Chain Map). *Let $\mathcal{C} \{C_p, \partial_p\}$ and $\tilde{\mathcal{C}} \{\tilde{C}_p, \tilde{\partial}_p\}$ be two chain complexes, then we define a chain map $\phi : \mathcal{C} \rightarrow \tilde{\mathcal{C}}$ as the family of homomorphisms*

$$\phi_p : C_p \longrightarrow \tilde{C}_p$$

such that $\tilde{\partial}_p \circ \phi_p = \phi_{p-1} \circ \partial_p, \forall p$.

Cohomology

In the preceding definitions we have introduced a series of abelian groups, and in the following section our objective is to define their “dual” counterparts. With any pair of abelian groups there is a third one, $\text{Hom}(A, B)$, the group of all homomorphisms between of A into B . We may mention as a well known example, the abelian group of linear forms on a vector space.

The group of homomorphisms $\text{Hom}(A, B)$ may be risen to the rank of abelian group by defining $\forall a \in A$:

$$(\alpha + \beta)(a) = \alpha(a) + \beta(a).$$

We may proceed then defining a special group of homomorphisms on chains, basic concept of cohomology. As usual, we will refer to simplicial complexes for clarity, but the same discussion may be extended to cell complexes in general.

Definition 9 (Cochain). *Let K be a simplicial complex, and G be an abelian group. Then we will define the group*

$$C^p(K; G) := \text{Hom}(C_p(K; G), G)$$

as the group of p -cochains of K with coefficient in G .

Definition 10 (Coboundary). *The coboundary operator δ^p is the dual to the boundary operator, so is defined as*

$$C^p(K; G) \xrightarrow{\delta^p} C^{p+1}(K; G).$$

Symmetrically to the homological groups of cycles and boundaries, we define the group of *cocycles* as the null space of the coboundary, denoted as $Z^p(K; G) := \text{Ker}(\delta^p)$, and the group of *coboundaries* as $B^p(K; G) := \text{Im}(\delta^p)$. The quotient space

$$H^p(K; G) := Z^p(K; G) / B^p(K; G)$$

is the *cohomology group* of K with coefficients in G .

Let $\gamma^p \in C^p(K; G)$ be a p -cochain. The value of γ^p on c_p is commonly denoted with the pairing

$$\langle \gamma^p, c_p \rangle := \gamma^p(c_p). \tag{2.3}$$

In other words, cochains measure the *content* of G -valued *additive* quantities in chains. The coboundary operator may be defined with the following notation, known as the *discrete Stokes theorem*:

$$\langle \delta^p \gamma^p, c_{p+1} \rangle = \langle \gamma^p, \partial_{p+1} c_{p+1} \rangle. \tag{2.4}$$

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 [Mun84] 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 $(g_\alpha \sigma_\alpha^*)_{\alpha \in \{1, \dots, n_p\}}$, which is often written as a sum:

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

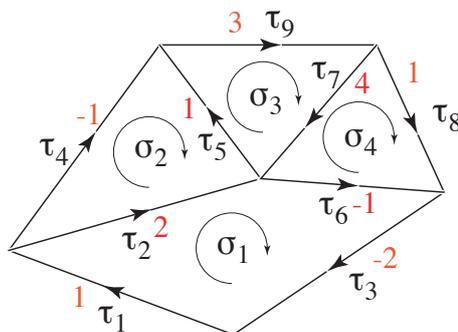


Figure 2.7: A 1-cochain γ^1 on a cell complex.

because of the additivity property:

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

For instance, let us consider the 1-cochain γ^1 pictured in Figure 2.7 on a (non simplicial) cell complex:

$$\gamma^1 = +1\tau_1 + 2\tau_2 - 2\tau_3 - 1\tau_4 + 1\tau_5 - 1\tau_6 + 4\tau_7 + 1\tau_8 + 3\tau_9,$$

and let us calculate its coboundary $\gamma^2 = \delta^1(\gamma^1)$. The operation will result in the 2-cochain

$$\gamma^2 = \delta^1(\gamma^1) = -4\sigma_2 + 8\sigma_3 - 2\sigma_4,$$

since expressing each individual coefficient resulting from the application of δ^1 we have $(+1 + 2 - 1 - 2)\sigma_1$, $(-2 - 1 - 1)\sigma_2$, $(+1 + 3 + 4)\sigma_3$, and $(+1 - 4 + 1)\sigma_4$.

As a dual property of the boundary operator expressed in (2.2), we have that

$$\delta^{p+1} \circ \delta^p \equiv 0, \tag{2.5}$$

where 0 denotes the null $(p + 2)$ -cochain. Duality between homology and cohomology may be carried on by analyzing the concepts of chain complexes and chain maps, with their respective cohomological equivalents:

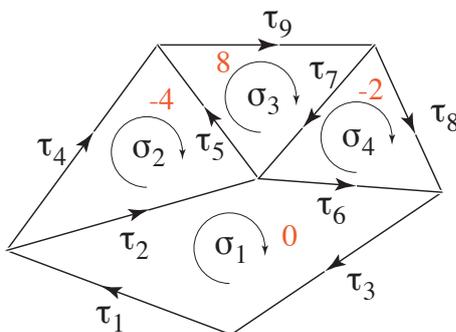


Figure 2.8: The coboundary $\gamma^2 = \delta^1(\gamma^1)$ of the 1-cochain in Figure 2.7.

Definition 11 (Cochain Complex). *Let $\mathcal{C} \{C_p, \partial_p\}$ be a chain complex, then the family of groups paired with homomorphisms $\{C^p(\mathcal{C}; G), \delta^p\}$ is called cochain complex, also indicated by the sequence*

$$\dots \longrightarrow C^{p-1} \xrightarrow{\delta^{p-1}} C^p \xrightarrow{\delta^p} C^{p+1} \longrightarrow \dots$$

Definition 12 (Cochain Map). *Let $\mathcal{C} \{C_p, \partial_p\}$ and $\tilde{\mathcal{C}} \{\tilde{C}_p, \tilde{\partial}_p\}$ be two chain complexes. Let $\phi : \mathcal{C} \rightarrow \tilde{\mathcal{C}}$ be a chain map, then we define a cochain map the dual homomorphism $\tilde{\phi}$:*

$$\tilde{\phi} : C^p(\tilde{\mathcal{C}}; G) \longrightarrow C^p(\mathcal{C}; G)$$

such that it commutes with the coboundary operator.

2.2 Differential Geometry

Homology and cohomology have always been coupled with the ideas behind the world of differential geometry. In the following section we will review the basic definition of differential geometry, partly adapted from [Kre91].

The basic concept of differential geometry is the *manifold*. Such a notion pervades all the ideas behind geometric objects and physical environments, for example the earth surface, is a manifold, embedding an idea of “continuum” space. The actual definition of a manifold is a little more abstract than we would perceive from the reality, as is as follows:

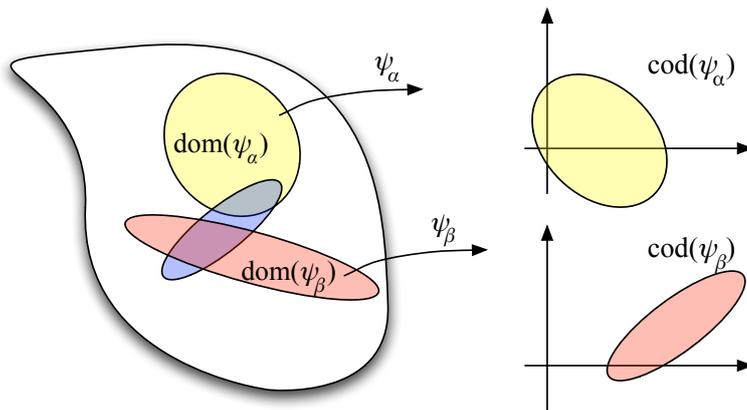


Figure 2.9: A manifold with two atlases ψ_α and ψ_β .

Definition 13 (Manifold). A p -manifold is the assembly of a set X and a set of injective functions $\mathcal{A} = \{\psi_\alpha : X \rightarrow \mathbb{R}^p\}$ with the following properties:

1. The codomain of each ψ_α is a connected open subset of \mathbb{R}^p ;
2. $X = \bigcup_i \text{dom}(\psi_i)$;
3. All the functions ψ_α are compatible.

The functions ψ_α are called *charts*, while the set of all charts \mathcal{A} is named *atlas*. A practical example of a manifold definition is a geographical atlas of the world. Each nation is the domain of a chart, where charts are the geographical coordinates of a location of a country: it is also possible that each country has a different (local) system of coordinates. The entire world is *covered* by the domain of our *charts*, whose codomain (the coordinates in the country’s system) is connected. An example of a manifold and its charts is given in Figure 2.9.

The third property of a manifold has yet to be explained. Let us consider two different charts ψ_α and ψ_β , such that $\text{dom}(\psi_\alpha) \cap \text{dom}(\psi_\beta) \neq \emptyset$, and let us define the *restriction* of a chart with respect to another:

$$\begin{aligned} \psi_{\alpha\beta} &:= \psi_\alpha|_{\text{dom}(\psi_\beta)} \\ \psi_{\beta\alpha} &:= \psi_\beta|_{\text{dom}(\psi_\alpha)}, \end{aligned}$$

so that we may define the *transition function* $\gamma_{\alpha\beta} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ as

$$\gamma_{\alpha\beta} := \psi_{\beta\alpha} \circ \psi_{\alpha\beta}^{-1}.$$

We may then define the *compatibility between charts*, stating that the functions $\psi_\alpha \in \mathcal{A}$ are C^k -compatible if $\exists k \geq 0$ such that $\gamma_{\alpha\beta} \in C^k$.

The transition functions are a change in the coordinate system between two charts, and for example in the real-world example stated above, we require such functions to be of class C^0 , called a *topological manifold*. If $\gamma_{\alpha\beta} \in C^k$, with $k \geq 1$, then the manifold is called *differentiable*.

Two differentiable manifolds X and Y are *diffeomorphic* if there exists an invertible bijection $f : X \rightarrow Y$. If two atlases are defined on the same set X , both of them appropriate for a manifold definition, then the two manifolds are *equivalent*. As a side-note, we stress that it may be easy to mix the concepts of *diffeomorphism* with the previously-defined *homeomorphism*, as for dimensions up to \mathbb{R}^3 the two coincide, or better each pair of diffeomorphic manifolds are homeomorphic and vice versa. In \mathbb{R}^4 and higher dimensions, there are examples of diffeomorphic manifolds that are not homeomorphic and homeomorphic spaces that are not diffeomorphic. Historically, the Fields Medal awarded John Willard Milnor was the first to show a 7-manifold homeomorphic and not diffeomorphic to a 7-sphere.

We can define the *cartesian product* of manifold as the following:

$$\{x, y\} \in X \times Y \rightarrow \{\phi_\alpha, \psi_\beta\} \in \mathbb{R}^p \times \mathbb{R}^q,$$

where $\{X, \mathcal{A}\}$ and $\{Y, \mathcal{B}\}$ are two manifolds. Let F and B be two manifolds called *fibre* and *bundle* respectively, with F defined with a single chart for simplicity. For each chart $\psi_\alpha \in B \rightarrow \mathbb{R}^p$, we consider the product of manifolds $\text{cod}(\psi_\alpha) \times F$, and for all the pairs of overlapping charts ψ_α and ψ_β we identify the pairs

$$\{\xi_\alpha, f\} = \{\xi_\beta, f\} \iff \psi^{-1}(\xi_\alpha) = \psi^{-1}(\xi_\beta)$$

“gluing” together the manifolds $\text{cod}(\psi_\alpha) \times F$ with $\text{cod}(\psi_\beta) \times F$, and creating a new manifold $V = B \times F$. An example is given in Figure 2.10. For all $v \in V$ there exists a point $x \in B$ such that $\psi_\alpha(x) = \xi_\alpha$ for all charts about x : the point is called a *projection* of v , $x = p(v)$. and $F_x = p^{-1}(x)$ is called *fibre above the point x* . A function $s \in B \rightarrow V$ is called *section* of a bundle V of base B if $p(s(x)) = x$ with $x \in \text{dom}(s)$. A section basically assigns to each point x in the base a fibre F_x above x .

Let us now expand the notion of manifold introducing an external embedding space. Let us define a *trajectory* in a manifold as a smooth function

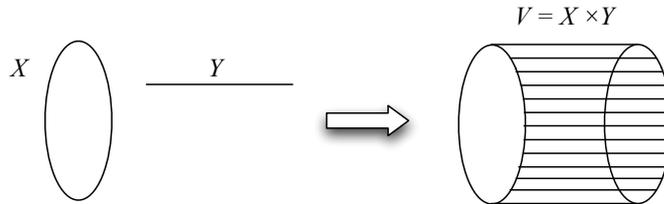


Figure 2.10: A manifold obtained by the bundle X and fibre Y .

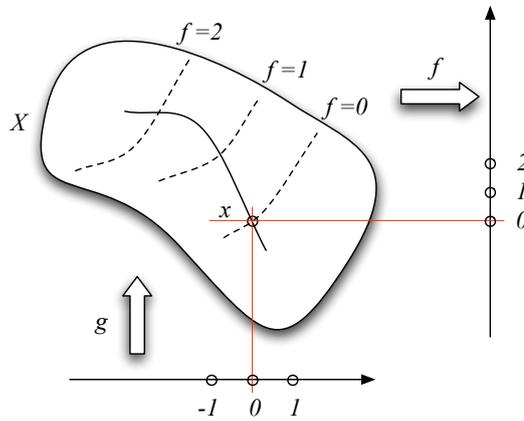


Figure 2.11: A manifold X with a trajectory g and a scalar field f .

$g : \mathbb{R} \rightarrow X$ with connected domain. A *scalar field* over a manifold is a smooth function $f : X \rightarrow \mathbb{R}$ whose codomain, because of the connectivity of X , is connected. A *trajectory over a point* $x \in X$ is a trajectory g with $0 \in \text{dom}(g)$ and $g(x) = 0$, symmetrically we can define a *scalar field vanishing at a point* x a scalar field f with $f(x) = 0$. As we can see from Figure 2.11, a trajectory is a graded oriented curve on X , while a scalar field “grades” the manifold with *isosurfaces*

$$X_a = \{x \in X : f(x) = a\}.$$

Two trajectories g and \tilde{g} are *tangent trajectories at a point* x if for all the

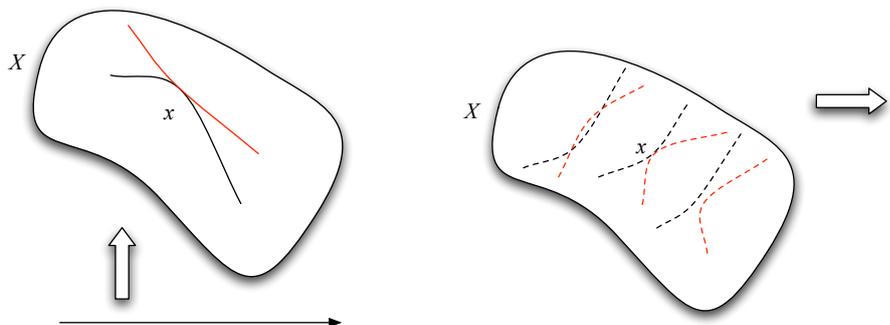


Figure 2.12: (a) Two tangent trajectories, and (b) two tangent scalar fields on a manifold X .

charts $\psi \in \mathcal{A}$ of a neighborhood of x we have

$$\frac{1}{t} |\psi(g(t)) - \psi(\tilde{g}(t))| = o(t),$$

with $t \rightarrow 0$. A similar definition may be given for scalar fields, defining f and \tilde{f} as *tangent scalar fields at a point* $x \in X$ if $\forall y \in \text{dom}(\psi)$ we have

$$f(x) - \tilde{f}(x) = o(|\psi(y) - \psi(x)|),$$

with $y \rightarrow x$. An example of tangent trajectories and scalar fields are represented in Figure 2.12.

We may define then the following well-known equivalence classes:

Definition 14 (Tangent Vector). *The equivalence class of smooth trajectories tangent at a point x on a manifold X is called tangent vector, denoted g_* .*

Definition 15 (Covector). *The equivalence class of all smooth functions vanishing at a point x on a manifold X is called covector and denoted f^* .*

Let us now associate the components of a chart $\psi(y)$ with the components of $\psi_*(x)g_*(y)$, a vector in $\psi(y)$. The manifold constituted by pairs $\{y, g_*(y)\}$, is the manifold of tangent vectors denoted TX . Sections of such a manifold are called *vector fields*. Symmetrically, we can associate $\psi(y)$ with $(\psi^{-1})^*(y)f^*(y)$, a covector at $\psi(y)$. The manifold of tangent covectors $\{y, f^*(y)\}$ is denoted T^*X , and its sections are the *covector fields*, or *1-forms*.

2.2. DIFFERENTIAL GEOMETRY

19

The above definitions establish a *duality* between vectors and covectors, between the tangent spaces TX and T^*X . Let us consider a trajectory $g \in \mathbb{R} \rightarrow X$ through $x \in X$, and a function $f : X \rightarrow \mathbb{R}$ vanishing at x , with f and g both smooth. The function $f \circ g \in \mathbb{R} \rightarrow \mathbb{R}$ is differentiable, and let us consider

$$\langle g_*, f^* \rangle = \frac{d}{dt}(f \circ g)_{t=0}$$

then we will *chain* the differentiation by the following

$$\langle g_*, f^* \rangle \equiv \frac{\partial f}{\partial x} \frac{dg}{dt} \Big|_{t=0}$$

that is the differentiation of f in the direction of g : the *gradient* of f at x is then the covector f^* . This interpretation leads to the consequence that a vector field, a section of TX , can be seen as a differential operator: vector fields are differential operators on manifolds [Wil82]. Moreover, it can be proven that if ∂ is a (first order) differential operator—not to be confused with the topological operator *boundary*—then there exists a unique vector field v such that

$$\partial f = x \rightarrow \langle v, f^* \rangle.$$

Let us consider then a vector field v with components in the basis v^i , and the differential operator $\partial f = x \rightarrow \langle v, f^* \rangle$. Then we define

$$df(v) := \partial f = \sum_i v^i \frac{\partial}{\partial x_i} f \tag{2.6}$$

as the *exterior derivative* of f , where $\partial/\partial x_i$ are the basis vectors relative to chart about x .

The notion of 1-form may be generalized as we will see in the following. Let us define first a 2-covector as a bilinear map $\omega(v_1, v_2) \in T_x X \times T_x X \rightarrow \mathbb{R}$, with the property of skew-symmetry, that is $\omega(v_1, v_2) = -\omega(v_2, v_1)$. Let $\partial_i := \partial/\partial x_i$ be a basis for $T_x X$, we have so

$$\omega_x(v_1, v_2) = \omega_x \left(\sum_i v_1^i \partial_i, \sum_j v_2^j \partial_j \right) = \sum_{i,j} \omega_x(\partial_i, \partial_j) v_1^i v_2^j$$

The above can be reduced to

$$\omega_x(v_1, v_2) = \sum_{i < j} \omega_x(\partial_i, \partial_j) (v_1^i v_2^j - v_2^i v_1^j)$$

By defining $\omega_{ij}(x) := \omega_x(\partial_i, \partial_j)$ and $(d^i \wedge d^j) := v_1^i v_2^j - v_2^i v_1^j$, we may write the 2-form at a point x as

$$\sum_{i < j} \omega_{ij}(x) d^i \wedge d^j$$

Generalizing the dimension of forms, retaining skew-symmetry and linearity, we may define the following:

Definition 16 (Form). *A p -form is defined as the field of multilinear alternating p -covectors at a point x .*

The operator \wedge introduced for 2-forms is called *exterior product* of forms. Let ω and η be a p -covector and a q -covector respectively, then we define the exterior product of covectors the $(p + q)$ -covector

$$(\omega \wedge \eta)(v_1, \dots, v_{p+q}) = \sum_{\sigma \in C(p, p+q)} \text{sgn}(\sigma, \varsigma) \omega(v_{\sigma_1}, \dots, v_{\sigma_p}) \eta(v_{\varsigma_1}, \dots, v_{\varsigma_q}) \quad (2.7)$$

It is easy to see that the product defined in Equation (2.7) defines an alternating form satisfying the requirements expressed in the definition of a p -form (p -covector), since $(\omega \wedge \eta) = (-1)^{pq}(\eta \wedge \omega)$. Moreover, we may note that $\omega \wedge \omega = 0$ if ω is a $(2k + 1)$ -form.

Chapter 3

Finite Methods Background

In this chapter we introduce three major methods for solving partial differential equations that emerge from a wide variety of physical phenomena, from fluid dynamics, to elasticity, from biomechanics to relativity.

In particular, each problem-solving method deals with usually complex differential equations modeling *non-linear* systems, and tries to reduce the complexity of such equations avoiding computational and conceptual challenges.

In the following sections we will briefly recall the three most common methods in engineering. In Section 3.1 we will outline the *finite difference* method, the oldest of the three techniques. Section 3.2 will sketch the *finite elements* method, a widely applied method for mechanical problems, and in Section 3.3 the *finite volumes* method will be introduced, an approach that gained momentum in the fluid dynamics community.

3.1 Finite Differences

The calculus of finite differences, briefly FD, deals with discontinue processes the same way as differential calculus cope with continuous ones. The field of finite difference calculus was unquestionably pioneered by George Boole (Lincoln, England 1815–Ballintemple, Ireland 1864) with his celebrated *A treatise on the calculus of finite differences* [Boo03], which follows the masterpiece *Treatise on Differential Equations*.

In the introduction of [Boo03], Boole states a canny definition of the nature of this calculus: “*the calculus of finite differences may be strictly defined as the science which is occupied about the ratios of the simultaneous increment of*

quantities [while] the differential calculus is occupied about the limits to which such ratios approach as the increments are indefinitely diminished”. In fact, difference equations are widely used as the finite approximation of differential problems.

Difference Equations

The following definitions are partly adapted from [LL92], but of course they may be traced back to [Boo03]. An *ordinary difference equation* of the r -th order is a relation between a sequence of values y_i in the form

$$y_{n+r} = F(n, y_n, y_{n+1}, \dots, y_{n+r-1}), \quad (3.1)$$

uniquely determined by r successive values y_1, y_2, \dots, y_r called *boundary conditions*. The concept of a difference equation naturally arises when introducing a function $y_n = f(a + nh)$, with a and h constants. Then the preceding expression corresponds to $f(a), f(a + h), f(a + 2h), \dots, f(a + nh)$, where h is called *step* or *interval*. It is evident that the variable of the function f increases with regular intervals of h .

The expression $y_{n+1} - y_n$ is called *difference* of y_n . Introducing an operator called *difference operator* Δ , then we may write the following

$$\Delta y_n := y_{n+1} - y_n.$$

It is straightforward to see the Δ operator as the finite analogous of the differential operator, since by the definition of derivative we have that taking $\lim_{h \rightarrow 0} \Delta y := y'$. As its continuous counterpart, Δ follows distributive, commutative and index laws, namely:

$$\Delta(y_n + z_n) = \Delta y_n + \Delta z_n,$$

$$\Delta(ky_n) = k\Delta y_n,$$

$$\Delta^r \Delta^s y_n = \Delta^{(r+s)} y_n.$$

As an example, the second order difference, approximating a second-order derivative, is

$$\Delta^2 y_n = \Delta(\Delta y_n) = y_{n+2} - 2y_{n+1} + y_n.$$

Without loss of generality, we may define the difference operator on a function of a real variable $y(x)$, existing in a domain $\text{dom}(y) := \mathcal{D} \subseteq \mathbb{R}$, as

$$\Delta y(x) := y(x + 1) - y(x). \quad (3.2)$$

3.1. FINITE DIFFERENCES

23

Finite differences may be then employed in the approximation of functions, directly applying the above definition. As an example, let $f(a + hx)$ be a polynomial function of x of the r -th order, and let us write the polynomial as

$$f(a + hx) = A_0 + A_1(x)_1 + \dots + A_r(x)_r,$$

with A_i constants $\forall i = 0, 1, \dots, r$. By applying successive computations, we can find the constants A_i : with $x = 0$ we have $A_0 = f(a)$; by considering the first-order difference, we have $A_1 = \Delta f(a)$, the second-order derivative gives us $A_2 = \Delta^2 f(a)/2!$ and so on. The polynomial above may be then written with the following expression:

$$f(a + hx) = f(a) + (x)_1 \frac{\Delta f(a)}{1!} + (x)_2 \frac{\Delta^2 f(a)}{2!} + \dots + (x)_r \frac{\Delta^r f(a)}{r!},$$

which is obviously the finite-difference analog of the Taylor series approximation $\sum_i (x - a)^i D^i f(a)/i!$.

Euler Method

The finite difference method is one of the easiest techniques to approximate differential equations, making extensive use, in the most naïve implementation, of the *Euler method*.

The Euler approximation of the differential operator substitutes a finite difference to a differential expression, for instance

$$u'(x) \approx \frac{u(x + h) - u(x)}{h} = \frac{\Delta u(x)}{h}.$$

As an example, the following differential equation

$$u''(x) + 4u'(x) - 7 = 0,$$

is substituted with the following:

$$\Delta^2 u(x) + 4h\Delta u(x) - 7h^2 = 0,$$

which of course gives an equation approximating the original differential relation. It is noticeable that the *approximation error* is dependent on the parameter h , and in the limit it converges to the exact result.

Recalling equations (3.1) and (3.2), it is evident that a discretization of a domain of interest leads to the transformation of the differential equation into a series of difference equations, or in other words, a system of linear equations.

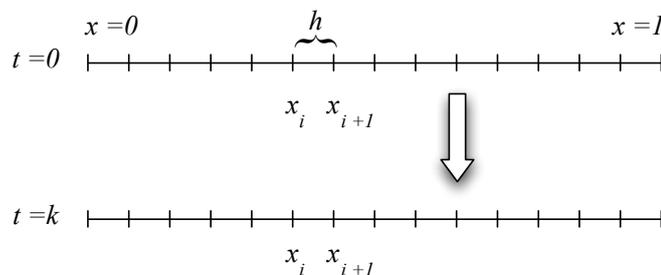


Figure 3.1: An example of finite difference approximation.

We may notice as the transition from differential to difference equations strongly rely on the increment of the variables in their finite form. This fact leads to a straightforward translation using uniform grids where increments are constants, but provide a non-naïve conversion on general discrete domains. An example of finite difference formulation on non uniform grids may be found in [CHSS95].

Example of Computation

In the following we will explain the finite difference approach with a concrete example involving the Laplacian. Let us approximate the following time-dependent heat transfer problem

$$\lambda \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = 0, \tag{3.3}$$

where λ is the thermal conductivity and u is the temperature, and let the domain of interest be $\mathcal{D} = [0, 1]$, with known boundary conditions $u(x, t)|_{t=0}$. Let us discretize the domain in $n - 1$ intervals, obtaining a sequence of variables x_1, x_2, \dots, x_n , as in Figure 3.1, with $x_{i+1} - x_i = h$, for all $i = 1, \dots, n - 1$. The Euler method may be then applied on (3.3) by discretizing the time variable t in intervals of k time units, obtaining the following difference equation:

$$\lambda \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} - \frac{u_i^{j+1} - u_i^j}{k} = 0$$

where the notation u_i^j stands for the temperature at the node x_i at a time-step t^j . Assembling all the addends related to time t^{j+1} , and supposing $\lambda k/h^2 = 1$,

we obtain a system of linear equations $A\xi = b$ with

$$A = \begin{bmatrix} 3 & -1 & 0 & \dots & 0 \\ -1 & 3 & -1 & & \\ 0 & -1 & 3 & & \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 3 \end{bmatrix}.$$

3.2 Finite Elements

The *finite element analysis* is a well known method for solving (partial) differential equations (PDE) developed mainly in the field of civil, aeronautical engineering and structural mechanics. Finite element methods, or briefly FEMs, rely strongly on previous results in the field of solving differential equations by Galerkin and others, notably Rayleigh and Ritz. A rigorous mathematical description of FEMs was lately given by Strang in [SF73].

The basic idea behind finite elements is a restatement of a given problem in a different formulation, and the approximation of the solution interpolating the results. The following sections will briefly introduce the notation and lexicon used in FE, and finally a practical example of its use.

The Weak Form

As we already introduced, finite methods will, in a first step, focus on the formulation of the problem. In the following we will use a practical example to elucidate the FE approach, so let us consider the following differential equation:

$$u'' + f(x) = 0$$

The *strong* (or *classical*) form of a problem is given by the (partial) differential equation and the necessary boundary conditions on the domain \mathcal{D} , for instance

$$\begin{cases} u'' + f(x) & = 0 \\ u(1) & = \alpha \\ u'(0) & = \beta \end{cases} \quad (3.4)$$

where our domain is $\mathcal{D} = [0, 1] \subset \mathbb{R}$. Usually the strong formulation of a problem is difficult to solve, so the problem must be rephrased using a different approach.

Let us introduce a class of functions called *trial functions*, which are required to satisfy one of the boundary conditions, for instance $u(1) = \alpha$. Such functions will be bound to belong to the *Sobolev space of functions* [Ada75]. This space is a notable example of a *Hilbert space*, and is defined as

$$H^p = \left\{ \varphi : \varphi \in L_2, \frac{\partial \varphi}{\partial x} \in L_2, \frac{\partial^2 \varphi}{\partial x^2} \in L_2, \dots, \frac{\partial^p \varphi}{\partial x^p} \in L_2 \right\}$$

where the L_p is defined as

$$L_p = \left\{ \varphi : \int_{\mathcal{D}} \varphi^p dx < \infty \right\}$$

In our case we will require that the functions have square-integrable derivatives:

$$\int_0^1 (u')^2 d\xi < \infty$$

and thus our trial functions will belong to the $H^1(\mathcal{D})$ space. The set of trial solutions will then be the following

$$\mathcal{S} = \{ u \in H^1(\mathcal{D}) : u(1) = \alpha \}$$

A second class of functions called *weighting functions*, consists of maps that satisfy the homogeneous counterpart of the trial functions:

$$\mathcal{V} = \{ w \in H^1(\mathcal{D}) : w(1) = 0 \}$$

The sets \mathcal{S} and \mathcal{V} are broadly known as the sets of *solutions* and *variations*, respectively. Then it is possible to show that the previous problem (3.4) is equivalent to the following form, called *weak form*:

$$\int_0^1 w' u' dx = \beta w(0) + \int_0^1 w f dx \tag{3.5}$$

It is a notable result that the weak form, also known as *variational equation*, has a unique solution identical to the strong formulation. The equivalence between the two formulations is a non-trivial proof, and uses the Riesz representation theorem (see [Rie07] and [Rie09]) on Hilbert spaces; a more recent revision can be found in [Har83].

Bubnov-Galerkin Method

The last step in FEMs is to provide an approximate solution of the weak formulation, and this is usually achieved by the Galerkin approximation method, or more correctly Bubnov-Galerkin method, which constructs a finite approximations of the solution and variations sets.

Let us discretize the domain of interest \mathcal{D} with a finite discrete version $\tilde{\mathcal{D}}$, called *mesh*: a cell complex. The cells of highest dimension belonging to the complex are called *elements*, and the 0-cells *nodes*. Both the set of function \mathcal{S} and \mathcal{V} will be approximated with the subsets $\tilde{\mathcal{S}} \subset \mathcal{S}$ and $\tilde{\mathcal{V}} \subset \mathcal{V}$. As a consequence, we have that the trial and the weighting functions will satisfy the boundary conditions in their discrete formulation, so $\tilde{u}(1) = \alpha$ and $\tilde{w}(1) = 0$.

For each element $\tilde{v} \in \tilde{\mathcal{V}}$, we consider trial functions $\tilde{u} \in \tilde{\mathcal{S}}$ such that

$$\tilde{u} = \tilde{v} + \tilde{\alpha}$$

where $\tilde{\alpha}$ is a known function that satisfies the boundary condition $\tilde{\alpha}(1) = \alpha$. The variational problem may be expressed as follows

$$\int_0^1 \tilde{w}' \tilde{u}' dx = \beta \tilde{w}(0) + \int_0^1 \tilde{w} f(x) dx$$

Finally, by linearity, and substituting $\tilde{u} = \tilde{v} + \tilde{\alpha}$ we obtain the final Bubnov-Galerkin variational formulation

$$\int_0^1 \tilde{w}' \tilde{v}' dx = \beta \tilde{w}(0) + \int_0^1 \tilde{w} f(x) dx - \int_0^1 \tilde{w}' \tilde{\alpha}' dx \quad (3.6)$$

The key feature in FEMs is that all the functional spaces, $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{V}}$, are constituted by the same set of functions. Let us define the functional space $\tilde{\mathcal{V}}$ as the finite linear combination of *basis* functions $B_i : \mathcal{D} \rightarrow \mathbb{R}$, such that $\forall \tilde{w} \in \tilde{\mathcal{V}}$, we can express the function as the linear combination

$$\tilde{w} = \sum_{i=1}^n c_i B_i$$

requiring that the basis functions satisfy $B_i(1) = 0, \forall i$; it follows then $\tilde{w}(1) = 0$. The B_i functions are also known as *shape* or *interpolating* functions. Introducing an additional function $B_{n+1} : \mathcal{D} \rightarrow \mathbb{R}$, such that $B_{n+1}(1) = 1$ we define $\tilde{\alpha} = \alpha B_{n+1}$, we can define the members of the trial functions $\tilde{u} \in \tilde{\mathcal{S}}$ by linear

combination of the chosen basis functions:

$$\tilde{u} = \sum_{i=1}^n d_i B_i + \alpha B_{n+1}$$

substituting these function in the Bubnov-Galerkin weak formulation (3.6), we obtain the following

$$\sum_{j=1}^n \int_0^1 d_j B'_i B'_j dx = \beta B_i(0) + \int_0^1 B_i f(x) dx - \int_0^1 \alpha B'_i B'_{n+1} dx$$

Let us define the matrix

$$A = [a_{ij}] := \int_0^1 B'_i B'_j dx$$

and the vector

$$b = [b_i] := \beta B_i(0) + \int_0^1 B_i f(x) dx - \int_0^1 \alpha B'_i B'_{n+1} dx$$

The variational formulation is then rephrased as a system of linear equations

$$A\xi = b$$

where $\xi := [d_j]$. The vectors and the matrix have different names depending on the physical phenomenon they describe, for instance in mechanics A is the *stiffness matrix*, ξ the *displacement vector*, and b the *force vector*.

The shape functions are usually wisely chosen having *compact support*, so being null everywhere except in a neighborhood of a node. This property influences the shape of the A matrix, since the integrands will be null if the basis functions are not “sufficiently near”: the matrix will result then sparse with the proper choice of basis functions.

In the following section we will show a one dimensional elliptic example to elucidate the process of producing a system of linear equations with the Bubnov-Galerkin method.

Example of Computation

Let us introduce an example of a classical FE computation [Joh88] as pictured in Figure 3.2. The problem is an elastic (elliptic) one, governed by the following differential (strong) formulation:

$$u''(x) + f(x) = 0$$

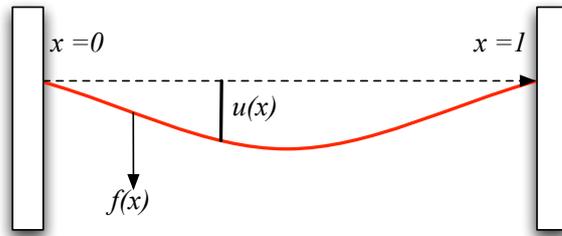


Figure 3.2: An example of an elliptic problem.

where $u(x)$ indicates the distance from the equilibrium, and $f(x)$ is the force applied to the string. The boundary conditions will impose that

$$u(0) = u(1) = 0$$

Let us formulate the same problem with the variational form, discretizing the domain $\mathcal{D} = [0, 1]$ with a finite number n of elements $e_i = [x_i, x_{i+1}]$, where the mesh is constituted by a 1-complex. For simplicity reasons, the mesh will contain elements of same size, so $x_{i+1} - x_i = h, \forall i = 1, \dots, n + 1$. The variational formulation will be as follows

$$\int_0^1 u'v' dx = \int_0^1 f v dx, \quad \forall v \in \tilde{\mathcal{V}} \tag{3.7}$$

We shall now choose the basis functions for the functional spaces $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{V}}$. Let us use a *piecewise linear approximation* for our problem, using the basis functions $B_i(x_j)$ defined by

$$B_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

that is, our shape functions $B_i(x)$ will be 1 at each node x_i and 0 on all the other nodes, as depicted in Figure 3.3. Substituting the shape functions in the variational equation (3.7) we obtain the following equation

$$\int_0^1 \sum_j x_j B_j' \sum_i B_i' dx = \int_0^1 f \sum_i B_i dx$$

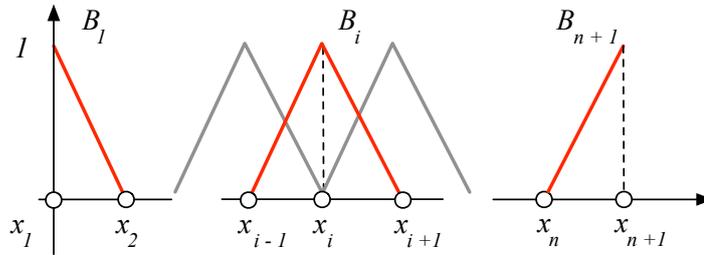


Figure 3.3: The piecewise linear finite element basis functions.

The system of linear equations $A\xi = b$ for this problem will be defined by the vector of unknowns $\xi = [x_i]$, the vector of *known* values $b = [b_i]$, where

$$b_i = \int_0^1 f B_i dx$$

and the matrix $A = [a_{ij}]$, with coefficients

$$a_{ij} = \int_0^1 B_i' B_j' dx$$

We may notice as the matrix A sparse, and more precisely tri-diagonal, since the integrands B_i and B_j will have empty intersection unless $j \in \{i - 1, i, i + 1\}$; in our particular case where the mesh is a regular partition of the domain \mathcal{D} , with $x_{i+1} - x_i = 1 \forall i$, we have

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & & \\ 0 & -1 & 2 & & \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix}$$

The solutions of this system of linear equations are the *nodal values* of the field, an approximated sampling on the nodes. The actual values in the domain is obtained interpolating these results.

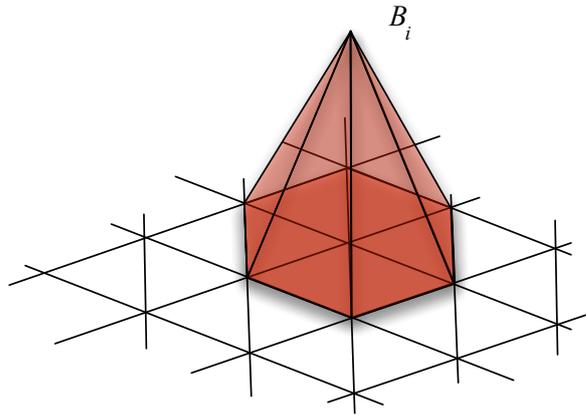


Figure 3.4: A piecewise linear FE basis for a 2-dimensional problem.

Finite Element Spaces

The spaces of functions $\tilde{\mathcal{V}}$ and $\tilde{\mathcal{S}}$ belong, as we previously said, to the Sobolev space $H^1(\mathcal{D})$. The basis functions that form the functional spaces should be then specified by

1. The discretization $\tilde{\mathcal{D}}$ of the domain \mathcal{D} ;
2. The functions themselves $\tilde{v} \in \tilde{\mathcal{V}}$;
3. Additional parameters describing such functions.

Let us consider some examples of such spaces. The *piecewise linear* approximation described in the above example may be easily generalized to a d -dimensional case, where the basis functions B_i are defined so that they assume the value 1 on the i -th node, and 0 on others. An example of a two-dimensional case is pictured in Figure 3.4. We notice as the support of such functions is constituted by all the triangles that share the node regarding the basis function.

The generalization of a linear approximation is straightforward. We may opt for *polynomial* functions, which of course increase the number of nodes based on the polynomial degree. For instance a second order polynomial on a triangular element requires the addition of three nodes in the element, as we can see in Figure 3.5. A cubic basis function requires nine points, and so

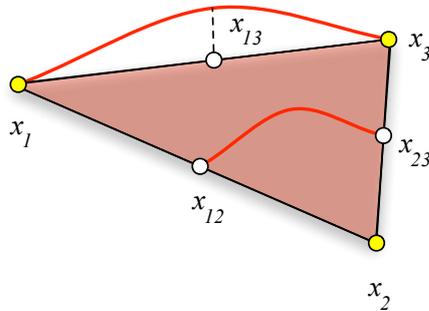


Figure 3.5: Example of a quadratic base function on a triangle.

on. Functions may also require additional parameters that are not necessarily nodes: for instance the value of the derivative of functions may be required as a known parameter. All of these variables are collectively known as *degrees of freedom*.

In general then a finite element is a triple (K, P_K, Σ) where K is a geometric entity, P_K is a finite dimensional space of functions, and Σ is the set of degrees of freedom. In the example pictured in Figure 3.5, K is the simplex $[x_1, x_2, x_3]$, P_K is the set of quadratic polynomials, and Σ is the set of nodal values $\{x_1, x_2, x_3, x_{12}, x_{13}, x_{23}\}$.

3.3 Finite Volumes

Finite volumes are the fluid dynamics counterpart of finite elements, born in the aerodynamics community for solving the Euler equations for compressible gas dynamics. In this approach differential equations expressing a conservation law, are approximated on a domain by taking into account the *average* of a quantity in a limited subspace of the domain of interest, called *grid cell* or *finite volume*. This procedure usually involves the use of the *divergence theorem*, transforming a volume integral in a surface one.

The fundamental difference between finite difference and finite volume methods is that a FV is based on the *integral form* of a conservation law, while FD approximate the differential relation itself. However, finite volumes are strongly related to finite differences and may be regarded as a particular FD approach [Lev02].

Conservation Laws

A finite volume method subdivides the domain into finite volumes, approximating the integral value of a field on each volume. In the following we will illustrate a one-dimensional time-dependent hyperbolic problem, for instance a transient heat flux problem.

Let us consider the time-dependent heat problem described by the differential equation

$$\frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} = 0,$$

and let the domain be discretized into a finite set of volumes $[x_{i-1/2}, x_{i+1/2}]$, approximating the heat flux Q_i^j with the following

$$Q_i^j \approx \frac{1}{h} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t^j) dx, \quad (3.8)$$

where the notation Q_i^j stands for the flux for the volume $c_i := [x_{i-1/2}, x_{i+1/2}]$ at a time-step t^j . The conservation laws are directly applied to the cells at each time-step, providing a tool that approximates Q_i^{j+1} from the preceding timeframe. The integral form for our problem is then the following differential relation

$$\frac{d}{dt} \int_{c_i} q(x, t) dx = f\left(q(x_{i-1/2}, t), q(x_{i+1/2}, t)\right). \quad (3.9)$$

Integrating equation (3.9) from t^j to t^{j+1} , with $t^{j+1} - t^j =: k$, we obtain

$$\int_{c_i} q(x, t^{j+1}) dx - \int_{c_i} q(x, t^j) dx$$

for the left hand side, and

$$\int_{t^j}^{t^{j+1}} f\left(q(x_{i-1/2}, t), q(x_{i+1/2}, t)\right) dt$$

for the right hand side of the equation. Rearranging the summands at the time-step t^{j+1} and introducing the cell volume h and time-frame k , we consider the approximation

$$Q_i^{j+1} = Q_i^j - \frac{k}{h} \left(F_{i+1/2}^j - F_{i-1/2}^j \right).$$

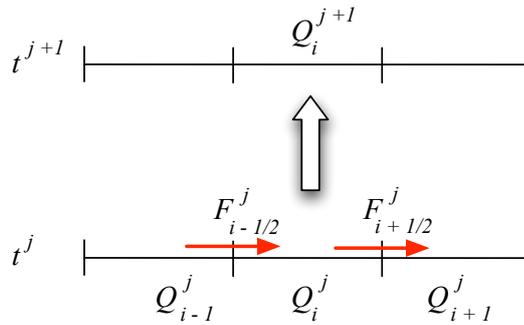


Figure 3.6: A finite volume *three point stencil* schema.

We may notice as the above equation may be also interpreted as a finite difference approximation, as in Section 3.1. The functions $F_{i\pm 1/2}^j$ are approximations of the average heat flux along the boundary $x = x_{i\pm 1/2}$, and can be well estimated by

$$F_{i\pm \frac{1}{2}}^j \approx \frac{1}{k} \int_{t^j}^{t^{j+1}} f(q(x_{i\pm \frac{1}{2}}, t)) dt. \quad (3.10)$$

Without any loss of generality we may suppose that heat propagates with at a finite speed, and its approximation at a time-step t^{j+1} may depend on the values Q_{i-1}^j , Q_i^j , and Q_{i+1}^j , at t^j , called a *three point stencil*, and presented in Figure 3.6. The conservation law then implies that

$$h \sum_i Q_i^{j+1} = h \sum_i Q_i^j - \frac{k}{h} (F_{N+\frac{1}{2}}^j - F_{1-\frac{1}{2}}^j),$$

where the domain has been divided into N finite volumes, and consequently the fluxes $F_{i\pm 1/2}^j$ cancel out except for the extreme cells.

The approximating power of FV depend, as all the previous methods, on the so-called *CFL necessary condition*, that states as follows: “a numerical method can be convergent only if its discrete domain contains the mathematical domain of the partial differential equations describing a phenomenon, at least in the limit as time-steps (k) and space-steps (h) go to 0”, named after Courant, Friedrichs, and Lewy [CFL28].

As a side-note, we may recall that the approximating equations specified above are a direct statement of the *divergence theorem*, known as Gauss, Green,

3.3. FINITE VOLUMES

35

or Gauss-Ostrogradsky theorem

$$\int_{\mathcal{D}} \nabla \cdot \phi = \int_{\partial \mathcal{D}} \phi,$$

where ϕ is a differentiable vector field: an obvious special instance of the widely applied Stokes theorem on differential forms.

Godunov Method

The description of conservation laws previously illustrated may be enhanced by considering that not only informations (e.g. waves, heat) propagate at a finite speed, but they also move in certain directions.

This knowledge is then employed in the solution of problems by taking into account the information in the direction where they should come from. Such an approach is called *upwind* method, and uses one-sided relations between solutions at different time-steps. For instance in the example depicted in Figure 3.6, Q_i^{j+1} may be determined by the values Q_ℓ^j , with $\ell \leq i$.

Godunov [God59] proposed his approach to upwind methods for gas dynamics, improving the existing approaches for compressible flows. The proposed algorithm is known also as *REA algorithm*, standing for reconstruct, evolve, and average, and is as follows:

1. Reconstruct a piecewise polynomial approximation $\tilde{q}^j(x, t^j)$ from cell averages Q_i^j , with the simple case of a piecewise constant approximation;
2. Evolve the hyperbolic equation to the next time-step $\tilde{q}^j(x, t^{j+1})$ (with time-frame k);
3. Average the result on each finite volume obtaining new initial values

$$Q_i^{j+1} = \frac{1}{h} \int_{c_i} \tilde{q}^j(x, t^{j+1}) dx.$$

The second step of the algorithm requires the formulation of a (numerical) approximation of the solution of a (set of) partial differential equation(s). For example, if we solve a simple *advection* problem governed by the following differential relation

$$\frac{\partial q}{\partial t} + \lambda \frac{\partial q}{\partial x} = 0,$$

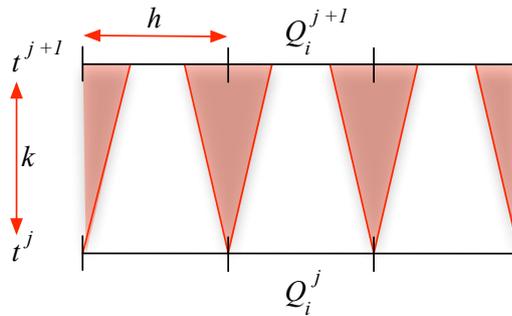


Figure 3.7: The Gudnov algorithm constraints for the CFL condition.

following the method proposed in the previous section and the upwind schema illustrated above, we would approximate the numerical flux as

$$F_{i-\frac{1}{2}}^j = \lambda Q_{i-1}^j.$$

As a result, a first order upwind method for the advection problem can be stated as follows:

$$Q_{i-1}^{j+1} = Q_i^j - \lambda \frac{k}{h} (Q_i^j - Q_{i-1}^j). \quad (3.11)$$

The solution of a problem at a time-step t^{j+1} from the preceding t^j imposes a constraint on the choice about time-steps and finite volume measures, outside of the obvious numerical accuracy.

An information, being it an acoustic wave or heat, move at a finite speed c . This fact forces the time-step k and the volume h to satisfy the relation $ck \leq h/2$, or rearranging the terms, to meet the limitation

$$\nu := c \frac{k}{h} \leq \frac{1}{2}. \quad (3.12)$$

The restriction in equation (3.12), pictured in Figure 3.7, is part of the CFL condition, and ν is called *Courant number*. Less naïve considerations on the Gudnov approach extend the CFL condition to $\nu \leq 1$.

Example of Computation

As we outlined a FD example in Section 3.1 and FE example in Section 3.2, in the following we will illustrate a basic finite volume approach. Let us introduce

3.3. FINITE VOLUMES

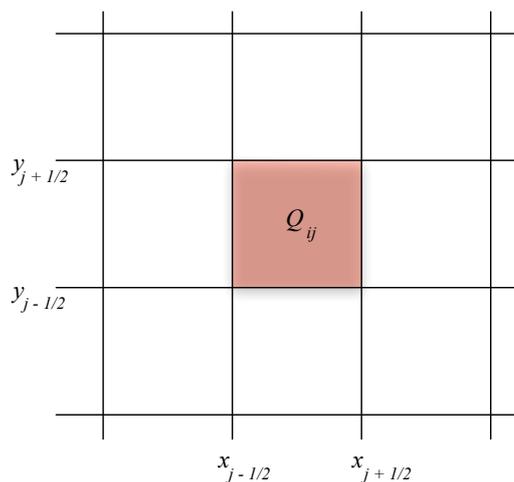


Figure 3.8: The discrete approximation of the domain \mathcal{D} .

the following differential equation on a two dimensional domain $\mathcal{D} = [0, 1] \times [0, 1]$:

$$\frac{\partial q}{\partial t} + \frac{\partial q}{\partial x} + \frac{\partial q}{\partial y} = 0,$$

and let the domain be discretized in n finite volumes $c_i = [x_i, x_{i+1}] \times [y_i, y_{i+1}]$ so that $\mathcal{D} = \cup_i c_i$, and $x_{i+1} - x_i = y_{i+1} - y_i = h$, for all i , as pictured in Figure 3.8.

The chosen polynomial estimation for Q_{ij}^p is a piecewise constant, with time-steps of k time units. The solution Q_{ij}^{p+1} at a time t^{p+1} depends on the previous solutions as we already described in the previous sections:

$$Q_{ij}^{p+1} = \frac{1}{h^2} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \tilde{q}^p(x - k, y - k, t^p) dx dy,$$

where k is the timeframe between successive steps $k = t^{p+1} - t^p$ for all p . Since \tilde{q} has been chosen constant on each finite volume, the equation above simplifies in the following

$$Q_{ij}^{p+1} = \frac{1}{h^2} [(h - k)^2 Q_{ij}^p + k(h - k)(Q_{i,j-1}^p + Q_{i-1,j}^p) + (hk)^2 Q_{i-1,j-1}^p],$$

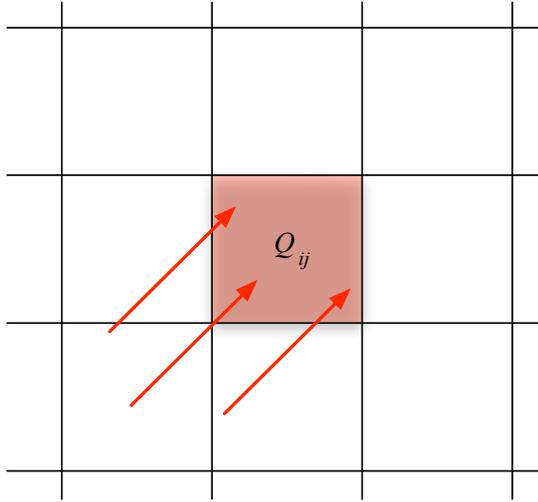


Figure 3.9: The corner transport upwind updating schema.

which yields a system of linear equations providing the solution for the given problem at each time-step.

We may notice that the obtained equation is again a particular finite difference equation. Moreover all the terms may be reorganized in order to obtain the Courant numbers for this instance of an upwind method, called, because of the two dimensions, *corner transport upwind*: the values in each finite volume Q_{ij}^{p+1} are obtained considering the informations propagating from the “lower” cells Q_{ij}^p , $Q_{i-1,j}^p$, $Q_{i,j-1}^p$, and $Q_{i-1,j-1}^p$, as pictured in Figure 3.9.

Chapter 4

(Co)Chain Complex Modeling

In this chapter we will analyze the mesh, defined in the previous chapter, from an algebraic topological point of view, associating its description with a block-bidiagonal matrix. We will present a representation of the (co)chain complex related with a decomposition of a computational domain, allowing the reduction of topology-preserving mesh refinements to the simplest set of Euler operators.

4.1 Representing (co)chains

In Section 2 we have introduced the concepts of chains and cochains, and underlined their characteristics. A very simple and powerful abstraction of such concepts consists in representing p -chains and p -cochains as matrices *indexed* on the cells of K and *parameterized* in the underlying G group.

Let K be a d -complex, with $k_p = |K_p|$, $0 \leq p \leq d$, where we recall that K_p represents the set of p -cells in the complex, known as p -skeleton. We may conveniently represent a p -chain $c_p \in C_p(K)$ as a *column matrix* $x_p \in G^{k_p}$, and we write $x_p = [c_p]$, or $x_p^i = [c_p]^i$. Analogously, we may represent a p -cochain $\gamma^p \in C^p(K)$ as a *row matrix* $y^p \in G^{k_p}$, and we write $y^p = [\gamma^p]^\top$, or $y_i^p = [\gamma^p]_i$.

The *content* of the p -cochain γ^p in the p -chain c_p is given by the *matrix product*

$$y^p x_p = \langle \gamma^p, c_p \rangle,$$

with the pairing notation already introduced in equation 2.3.

Incidence matrices

The representation of chains and cochains with matrices allow us to introduce the boundary and coboundary operators, describing them in terms of linear algebra. Let us first introduce the following concept regarding the relation between cells of different order.

The intersection between p -cells and $(p + 1)$ -cells may be characterized by the p -incidence matrix (a_p^{ij}) defined by:

$$a_p^{ij} = 0 \text{ if } \sigma_p^i \cap \bar{\sigma}_{p+1}^j = \emptyset \text{ (}\bar{\sigma} \text{ being the closure of } \sigma\text{);}$$

$$a_p^{ij} = \pm 1 \text{ otherwise, with } +1 \text{ (}-1\text{) if the orientation of } \sigma_p^i \text{ is equal (opposite) to that of the corresponding face of } \sigma_{p+1}^j.$$

Of course, the transpose of (a_p^{ij}) describes how $(p + 1)$ -cells intersect with p -cells.

It is easy to check that (a_p^{ij}) represents through matrix multiplication the action of the boundary operator $\partial_{p+1} : C_{p+1} \rightarrow C_p$, while its transpose represents the action of the coboundary operator $\delta^p : C^p \rightarrow C^{p+1}$:

$$\sum_{j=1}^{k_{p+1}} a_p^{ij} [c_{p+1}]^j = [\partial_{p+1} c_{p+1}]^i, \tag{4.1}$$

$$\sum_{i=1}^{k_p} a_p^{ij} [\gamma^p]_i = [\delta^p \gamma^p]_j. \tag{4.2}$$

Example 1 (Boundary and Coboundary). *Let the 2-chain $c \in C_2(K)$ be defined by*

$$c(\sigma_1) = 1, \quad c(\sigma_2) = 1, \quad c(\sigma_3) = 1, \quad c(\sigma_4) = 1,$$

where K is the 2-complex given in Figure 4.1. The boundary 1-chain

$$\partial_2 c = \partial_2(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4) = \tau_1 + \tau_3 + \tau_4 + \tau_8 + \tau_9$$

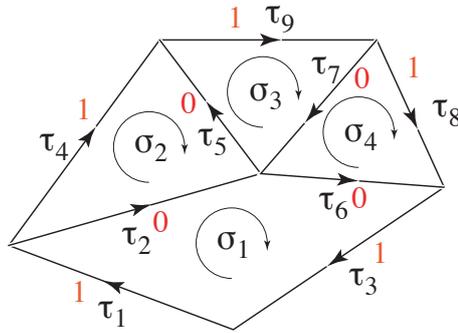


Figure 4.1: A 2-complex K , whose 2-cells are coherently oriented.

is represented by

$$[\partial_2] \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix},$$

where the incidence matrix $[\partial_2] = [\delta^1]^\top$, and

$$[\delta^1] = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 \end{bmatrix}$$

Hasse Diagram of a Chain Complex

A Hasse diagram, named after the German mathematician Helmut Hasse (August 1898–December 1979), illustrates the cover relation of a partial order and is commonly used for representing lattices.

In order theory, a *Hasse diagram* is a graph $\mathcal{H} = (N, E)$, where N is a finite poset, such that for any $x, y \in N$, there exists $(x, y) \in E$ if and only if $x < y$, and there is no $z \in N$ such that $x < z < y$.

If, given a d -complex K , the sets N and E are defined as follows, then the graph $\mathcal{H}(K) = (N, E)$ provides a complete representation of the topology of K :

1. $N := K_0 \cup K_1 \cup \dots \cup K_d$,
2. $E := E_1 \cup \dots \cup E_d$, with
3. $E_p := \{(\sigma_p, \sigma_{p-1}) \mid \sigma_{p-1} \in \partial\sigma_p\}$, $1 \leq p \leq d$.

Attaching a label from $\{-1, 1\}$ to the arc $(x, y) \in E_p$, denoted $\text{sgn}(x, y)$, suffices to specify the relative orientation between the p -cell represented by the node x and the $(p-1)$ -cell represented by the node y .

Given a Hasse graph $\mathcal{H}(K) = (N, E)$, with $N = \cup_p K_p$, for each node $x \in N$ we can define the following sets:

1. $E^x := \{(x, y) \mid y \in N, (x, y) \in E\}$,
2. $N^x := \{y \mid y \in N, (x, y) \in E^x\}$.

Let $\sigma \in K$ be the cell represented by the node x . Then, the boundary of the elementary chain $g\sigma$ is obtained by transferring the (properly signed) coefficient from the node x to its “children” in $\mathcal{H}(K)$:

$$\partial(g\sigma) = g(\partial\sigma) = g \sum_{y \in N^x} \text{sgn}(x, y) \tau(y) = \sum_{y \in N^x} \text{sgn}(x, y) g\tau(y),$$

where $\tau(y)$ denotes the cell represented by the node y . The computation of the boundary ∂c of a general chain c follows by linearity.

Chain and cochain complexes

A Hasse diagram, together with the above representation of the boundary operator ∂ , is a convenient representation of a *chain complex*, whose formal definition was given earlier.

Let us recall that a *chain complex* $\mathcal{C} = (C_p, \partial_p)$ is a sequence

$$\dots \longrightarrow C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \longrightarrow \dots \longrightarrow C_1 \xrightarrow{\partial_1} C_0$$

4.2. MATRIX REPRESENTATION OF A COMPLEX

43

of abelian groups C_p , paired with homomorphisms ∂_p , $p \geq 1$, that satisfies the relation $\partial_p \circ \partial_{p+1} = 0$, for each $p \geq 1$.

The dual *cochain complex* $\tilde{\mathcal{C}} = (C^p, \delta^p)$ is the sequence

$$\dots \longleftarrow C^{p+1} \xleftarrow{\delta^p} C^p \xleftarrow{\delta^{p-1}} C^{p-1} \longleftarrow \dots \longleftarrow C^1 \xleftarrow{\delta^0} C_0$$

The relations $\delta^p \circ \delta^{p-1} = 0$ ($p \geq 1$) are satisfied by duality.

Chain maps

Let $\mathcal{C}(C_p, \partial_p)$ and $\tilde{\mathcal{C}}(\tilde{C}_p, \tilde{\partial}_p)$ be two chain complexes. A *chain map* $\phi : \mathcal{C} \rightarrow \tilde{\mathcal{C}}$ is a p -family of homomorphisms

$$\phi_p : C_p \longrightarrow \tilde{C}_p$$

such that $\tilde{\partial}_p \circ \phi_p = \phi_{p-1} \circ \partial_p$, i.e., the following diagram is *commutative*:

$$\begin{array}{ccc} C_p & \xrightarrow{\phi_p} & \tilde{C}_p \\ \partial_p \downarrow & & \downarrow \tilde{\partial}_p \\ C_{p-1} & \xrightarrow{\phi_{p-1}} & \tilde{C}_{p-1} \end{array}$$

4.2 Matrix representation of a Complex

In this section we introduce a block-matrix representation of the topology of the chain complex associated to a decomposition of the computational domain, and call it *Hasse matrix*. Later we show that, since all blocks transform by the one and the same pattern of transformations, so also the Hasse matrix transforms by the same pattern.

Block-Matrix Decomposition

A chain complex $\mathcal{C}(C_p, \partial_p)$ and its dual $\tilde{\mathcal{C}}(C^p, \partial^p)$ can be represented by a block-bidiagonal matrix. Since the boundary operators ∂_p ($p \geq 1$) are well represented by incidence matrices and the coboundary operators δ^{p-1} by their transposes, we may represent the p -families of homomorphisms ∂_p, δ^{p-1} ($p \geq 1$) by a block-structured matrix. Notice that, from now on, we shall often write δ_p instead of δ^p .

Let K be a d -complex and $\mathcal{H}(K)$ its Hasse graph. The Hasse graph may be suitably represented by a matrix, which we will call *Hasse matrix* to underline its ancestry. The matrix will have the block structure shown in Figure 4.2:

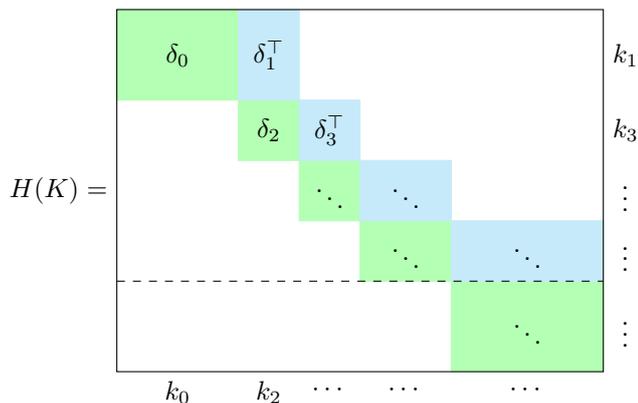


Figure 4.2: The whole scheme holds for d odd; for d even, the last block-row should be discarded.

The transposed Hasse matrix $H^T(K)$ represents the dual complex K^* , whose Hasse graph $\mathcal{H}(K^*)$ is isomorphic to $\mathcal{H}(K)$, with $K_p^* \cong K_{d-p}$ ($0 \leq p \leq d$), where the boundary and coboundary operators are swapped by duality. The dual cell complex and its importance will be described later in Section 5.2.

Example 2 (Hasse graph (3D)). *Below we give a picture of the graph $\mathcal{H}(K)$ of a 3-complex K (a cube), representing its 6 boundary and coboundary operators as topological mappings between its sub-complexes K_p .*

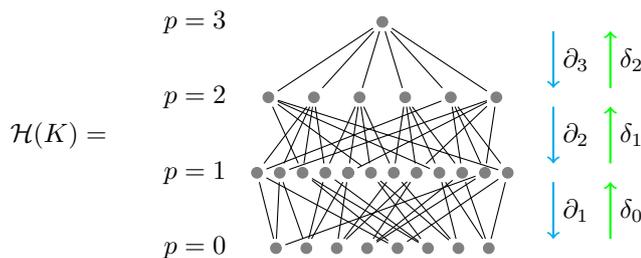


Figure 4.3: The Hasse diagram of the chain complex representing the topology of a 3-cube.

4.3. EULER OPERATORS

Example 3 (Hasse matrix (3D)). Operators $\delta_0, \delta_1^\top = \partial_2$, and δ_2 may be represented as a single block-matrix (the Hasse matrix):

$$H \in \mathcal{M}_{k_0+k_2}^{k_1+k_3}(G), \quad G = \{-1, 0, 1\},$$

defined as below:

$$H = \begin{array}{|c|c|} \hline \delta_0 & \delta_1^\top & k_1 \\ \hline 0 & \delta_2 & k_3 \\ \hline k_0 & k_2 & \end{array}$$

According to their definition, the operators $\partial_1, \partial_2^\top = \delta_1$, and ∂_3 are collected in the transpose matrix H^\top :

$$H^\top = \begin{array}{|c|c|} \hline \partial_1 & 0 & k_0 \\ \hline \partial_2^\top & \partial_3 & k_2 \\ \hline k_1 & k_3 & \end{array}$$

Example 4 (Linear graph). If K is a 1-complex, i.e. a linear graph, then $H(K)$ and the incidence matrix of vertices on edges coincide. $H(K)$ and its transpose represent the two topological operators available, i.e., δ_0 and ∂_1 .

4.3 Euler operators

In solid modeling it is common to refer to Euler operators as an independent set of operators [Män88, Hof89] that transform a boundary representation of a solid into a different one, satisfying the Euler-Poincaré formula.

Euler characteristic

A well-known invariant of a finite d -dimensional cell complex K is its *Euler characteristic*, that can be defined as the alternating sum

$$\chi = k_0 - k_1 + k_2 - k_3 + \cdots + (-1)^d k_d.$$

For polyhedra homeomorphic to a 2-sphere, the Euler characteristic is $\chi = V - E + F = 2$, where V , E , and F are the number of vertices, edges and faces, respectively. This fact leads to the conclusion that any 2-complex K with $|K|$ topologically equivalent to a sphere, has $\chi(K) = 2$.

Euler-Poincaré formula

The above formulation may be specialized easily to complexes on an orientable compact surface, with the following:

$$V - E + F = 2(S - H) - B + R,$$

where S , H , B , and R are the number of shells (i.e. connected components), holes, border components and rings, respectively. The general equation that links the Euler characteristic with homology is given in [Kin93], and is as follows:

$$\chi(K) = \sum_{i=0}^n (-1)^i \beta_i,$$

with β_i indicating the *Betti numbers* [Mun84] of the n -complex K , which correspond to the Betti numbers of homology groups of K introduced in Section 2.1

According to the above, the *simplest* set of independent refining (coarsening) operators for a d -space that do not change its Euler characteristic has to increase (decrease) by one both k_{p-1} and k_p , for $p \in \{1, \dots, d\}$. There are therefore d elementary refining operators and the same number of elementary coarsening operators. A complete list and description of Euler operators (e.g. MEV “Make an Edge and a Vertex”, and KFE “Kill a Face and an Edge”) can be found in [Män88].

In order to change the Euler characteristic, i.e. to change the shape of a space, it is appropriate to use some Boolean operator, according to the properties [Ale98, Bae03] recalled below.

4.3. EULER OPERATORS

47

Properties of the Euler characteristic

Let $\chi(M)$ and $\chi(N)$ be the Euler characteristics of any two topological spaces M and N . Then, their sum is the Euler characteristic of the disjoint union of M and N :

$$\chi(M \sqcup N) = \chi(M) + \chi(N).$$

More generally, if M and N are subspaces of a larger space X , then so are their union and intersection, and the Euler characteristic obeys the rule:

$$\chi(M \cup N) = \chi(M) + \chi(N) - \chi(M \cap N).$$

Moreover, the Euler characteristic of any product space is

$$\chi(M \times N) = \chi(M) \chi(N).$$

Make and Kill operations

The simplest Euler operators that transform a cell complex K into a new complex \tilde{K} without changing its Euler characteristic χ , add (remove) just *two* cells to (from) the complex, with dimensions p and $(p + 1)$. They will be denoted as β and κ , from the Greek words “blastos” and “klastos”, referring to construction and destruction, respectively. Note that the β operator has not to be confused with a Betti number.

By definition, the operator β^p adds a p -cell and a $(p + 1)$ -cell to K , thus transforming it into \tilde{K} . The reverse operator κ^p deletes a p -cell and a $(p - 1)$ -cell.

In this section we discuss how the coboundary operators transform under the action of a refinement operation β^q :

$$\delta_p \circ \beta^q : \delta_p(K) \mapsto \delta_p(\beta^q(K)), \quad p = 0, \dots, n - 1.$$

It is easily seen that β^q affects in a nontrivial way only the coboundary operators whose domain and/or codomain change under its action, namely:

1. $\delta_{q+1} \mapsto \tilde{\delta}_{q+1}$,
2. $\delta_{q-1} \mapsto \tilde{\delta}_{q-1}$,
3. $\delta_q \mapsto \tilde{\delta}_q$.

as shown by the commutative diagram:

$$\begin{array}{ccccccc}
 \tilde{C}^{q+2} = C^{q+2} & \xleftarrow{\delta_{q+1}} & C^{q+1} & \xleftarrow{\delta_q} & C^q & \xleftarrow{\delta_{q-1}} & C^{q-1} = \tilde{C}^{q-1} \\
 & \searrow & \downarrow \beta^q & & \downarrow \beta^q & \swarrow & \\
 & \tilde{\delta}_{q+1} & \tilde{C}^{q+1} & \xleftarrow{\tilde{\delta}_q} & \tilde{C}^q & \xleftarrow{\tilde{\delta}_{q-1}} & \\
 & & & & & &
 \end{array}$$

Three different computations have to be performed, depending on whether only the domain changes (case 1), or only the codomain (case 2), or both change (case 3).

Addition of a column ($\delta_{q+1} \mapsto \tilde{\delta}_{q+1}$)

Let the matrix $[\delta_{q+1}]$ be $m \times n$; then, the matrix $[\tilde{\delta}_{q+1}]$ will be $m \times (n + 1)$. The column to be added to $[\delta_{q+1}]$ represents the cochain in $\beta^q(C^{q+2})$ incident on the new cell $\tilde{\sigma}_{q+1}$. It is a linear combination of the columns of $[\delta_{q+1}]$, i.e., of the preexistent cochains in C^{q+2} . We have:

$$[\tilde{\delta}]_{m \times (n+1)} = [\delta]_{m \times n} \left[\begin{array}{c|c} I_{n \times n} & \begin{matrix} c_1 \\ \vdots \\ c_n \end{matrix} \end{array} \right] = [\delta]_{m \times n} C$$

Addition of a row ($\delta_{q-1} \mapsto \tilde{\delta}_{q-1}$)

The row to be added to $[\delta_{q-1}]$ represents the chain of $\beta^q(C_{q-1})$ incident on the new cell $\tilde{\sigma}_q$. It is a linear combination of the rows of $[\delta_{q-1}]$. We have:

$$[\tilde{\delta}]_{(m+1) \times n} = \left[\begin{array}{c} I_{m \times m} \\ r_1 \ \cdots \ r_m \end{array} \right] [\delta]_{m \times n} = R [\delta]_{m \times n}$$

Addition of a column and a row ($\delta_q \mapsto \tilde{\delta}_q$)

One of the rows of $[\delta_q]$ (one chain in C_q) is substituted by *two* rows (two chains in $\beta^q(C_q)$), whose components on the added cell $\tilde{\sigma}_q$ sum up to zero. The matrix $[\tilde{\delta}_q]$ is obtained as the sum

$$[\tilde{\delta}_q]_{(m+1) \times (n+1)} = \sum_{i=1}^3 S_i [\delta_q]_{m \times n} T_i,$$

4.3. EULER OPERATORS

where the first term ($i = 1$) provides the contribution of the split cell σ_{q+1} , the second one ($i = 2$) the contribution of the added cell $\tilde{\sigma}_{q+1}$, and the third one ($i = 3$) the contribution of all of the other cells in K_{q+1} .

Examples

In Figures 4.4–4.6 we show a very simple 2-complex K and its refinement \tilde{K} , obtained by applying first the operator β^0 to split the 1-cell σ_1^1 , then the operator β^1 to split the 2-cell σ_2^1 .

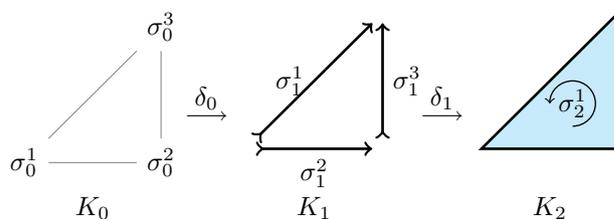


Figure 4.4: Coarse complex $K = (K_0, K_1, K_2)$.

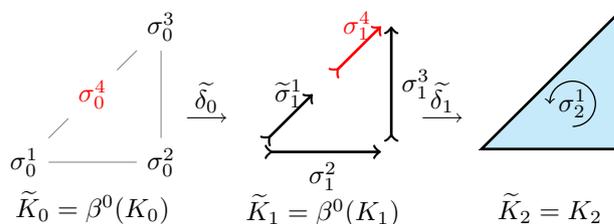


Figure 4.5: First refinement step: $\tilde{K} = \beta^0(K) = (K_0 \cup \{\sigma_0^4\}, K_1 \cup \{\sigma_1^4\}, K_2)$

Let us compute the matrix representation of the coboundary operators δ_0, δ_1 , on K and on their refinements $\tilde{K} = \beta^0(K)$ and $\tilde{\tilde{K}} = \beta^1(\tilde{K})$. The boundary operators ∂_1, ∂_2 , as well as their refinements, are obtained by transposition.

Example 5 (Coboundary $\delta_0 : C^0(K) \rightarrow C^1(K)$). Both domain and codomain have dimension 3. From Figure 4.4 it is seen that the matrix representation of

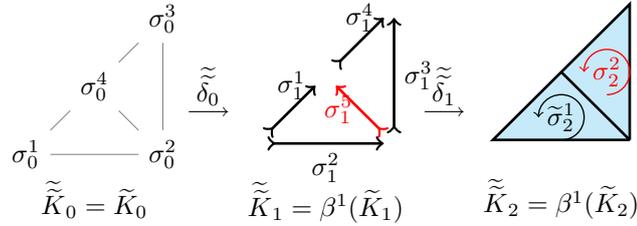


Figure 4.6: Second refinement step: $\tilde{K} = \beta^1(\tilde{K}) = (\tilde{K}_0, \tilde{K}_1 \cup \{\sigma_1^3\}, \tilde{K}_2 \cup \{\sigma_2^2\})$

$\tilde{\delta}_0$ is

$$[\tilde{\delta}_0] = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}.$$

Example 6 (Coboundary $\delta_1 : C^1(K) \rightarrow C^2(K)$). In this case we have $k_1 = 3$ and $k_2 = 1$, so that

$$[\delta_1] = \begin{pmatrix} -1 & 1 & 1 \end{pmatrix}.$$

Example 7 (Coboundary $\tilde{\delta}_0 : C^0(\tilde{K}) \rightarrow C^1(\tilde{K})$). We have $k_0 = k_1 = 3 + 1$. In Figure 4.5 the new 0-cell and 1-cell are displayed in red. Since both domain and codomain dimensions increase, the new operator has to be computed as the sum of three contributions (see Section 4.3).

$$[\tilde{\delta}_0] = \begin{pmatrix} S_1 & S_2 & S_3 \end{pmatrix} [\delta_0] \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}$$

where $\begin{pmatrix} S_1 & S_2 & S_3 \end{pmatrix}$ and $\begin{pmatrix} T_1 & T_2 & T_3 \end{pmatrix}^\top$ are block-matrices, and

$$S_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

$$S_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

4.3. EULER OPERATORS

51

Matrices S_1, S_2 extract the row of $[\delta_0]$ that corresponds to the 1-cell σ_1^1 to be split (recall that a row of $[\delta_0]$ equals a column of $[\partial_1]$); S_1 associates that row to $\tilde{\sigma}_1^1$, while S_2 associates it to the added cell $\tilde{\sigma}_1^4$; matrix S_3 keeps all other rows of $[\delta_0]$ unchanged. The actions of S_1, S_2 , and S_3 on $[\delta_0]$ are explicitly given below:

$$S_1[\delta_0] = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad S_2[\delta_0] = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix},$$

$$S_3[\delta_0] = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Each column of matrix T_i ($i = 1, 2, 3$) corresponds to a 1-cell in \tilde{K}_1 . Each T_i matrix represents the linear transformation that maps one or more chains of K_0 elements into the corresponding chains of \tilde{K}_0 elements:

$$S_1[\delta_0]T_1 = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$S_2[\delta_0]T_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix},$$

$$S_3[\delta_0]T_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

In conclusion, we get:

$$[\tilde{\delta}_0] = \begin{pmatrix} -1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}.$$

The reader may check this result looking at Figure 4.5.

Example 8 (Coboundary $\tilde{\delta}_1 : C^1(\tilde{K}) \rightarrow C^2(\tilde{K})$).

In this case, $\tilde{k}_1 = 3 + 1$ and $\tilde{k}_2 = 1$; one gets:

$$[\tilde{\delta}_1] = [\delta_1]C = [\delta_1] \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 1 & -1 \end{pmatrix}.$$

Example 9 (Coboundary $\tilde{\delta}_0 : C^0(\tilde{K}) \rightarrow C^1(\tilde{K})$).

We have: $\tilde{k}_0 = \tilde{k}_0 = 4$, $\tilde{k}_1 = \tilde{k}_1 + 1 = 5$, and we get:

$$\begin{aligned} [\tilde{\delta}_0] &= R[\tilde{\delta}_0] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} -1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Example 10 (Coboundary $\tilde{\delta}_1 : C^1(\tilde{K}) \rightarrow C^2(\tilde{K})$).

Now we have $\tilde{k}_1 = \tilde{k}_1 + 1 = 5$ and $\tilde{k}_2 = \tilde{k}_2 + 1 = 2$. Since both domain and codomain dimensions increase, by performing the same operations as in Example 7, we get:

$$\begin{aligned} [\tilde{\delta}_1] &= \begin{pmatrix} S_1 & S_2 & S_3 \end{pmatrix} [\tilde{\delta}_1] \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \\ &= \begin{pmatrix} -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 \end{pmatrix}. \end{aligned}$$

4.4 Hasse transformations

Let K be a d -complex and $H(K)$ be its $n \times m$ Hasse matrix, where $\chi(K) = m - n$. In this section we introduce the Hasse transformations

$$\eta_p^h(K) : \mathcal{M}_m^n \rightarrow \mathcal{M}_{m+1}^{n+1},$$

4.4. HASSE TRANSFORMATIONS

53

such that

$$H(K) \mapsto H(\tilde{K}),$$

where the $(p+1)$ -cell σ_{p+1}^h is split by the *blastos* (or “make”) β^p operator into two cells:

$$\tilde{\sigma}_{p+1}^h \quad \text{and} \quad \tilde{\sigma}_{p+1}^{(k_{p+1})+1},$$

and a new p -cell $\tilde{\sigma}_p^{k_p+1}$ is added to the complex. Notice that, while m and n increase under topology-preserving refinements, their difference does not. Let us distinguish between even and odd values of d , and assume, without loss of generality, that $d = 3$. In this case there are two diagonal blocks $[\delta_0]$ and $[\delta_2]$, and one upper-diagonal block $[\delta_1]^\top$ in H (see Section 3).

Make operators β^0, β^1 and β^2 : Different but similar computational patterns arise, depending on the order of the *make* operator:

$$\beta^0(H) = \left(\begin{array}{c|c} (S_1 \ S_2 \ S_3)[\delta_0] \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} & R[\delta_1]^\top \\ \hline 0 & [\delta_2] \end{array} \right),$$

$$\beta^1(H) = \left(\begin{array}{c|c} R[\delta_0] & (S_1 \ S_2 \ S_3)[\delta_1]^\top \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \\ \hline 0 & [\delta_2]C \end{array} \right),$$

$$\beta^2(H) = \left(\begin{array}{c|c} [\delta_0] & [\delta_1]^\top C \\ \hline 0 & (S_1 \ S_2 \ S_3)[\delta_2] \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \end{array} \right).$$

In 3D the only *make* operators are $\beta^0, \beta^1, \beta^2$. Each β^p inserts two new cells $\tilde{\sigma}_p$ and $\tilde{\sigma}_{p+1}$ into \tilde{K} . In order to specify the corresponding Hasse transformation, we need to extract the diagonal and upper-diagonal blocks of H :

$$H = \begin{pmatrix} [\delta_0] & [\delta_1]^\top \\ 0 & [\delta_2] \end{pmatrix} = \underbrace{\begin{pmatrix} [\delta_0] & 0 \\ 0 & [\delta_2] \end{pmatrix}}_{H_1} + \underbrace{\begin{pmatrix} 0 & [\delta_1]^\top \\ 0 & 0 \end{pmatrix}}_{H_2}$$

Then, we need only to apply the elementary transformations already given for a single operator, and to add the resulting matrices:

$$\beta^p(H) = \beta^p(H_1) + \beta^p(H_2).$$

4.5 Hyperplane splitting

In this section we discuss a subdivision algorithm (SPLIT) developed by Bajaj and Pascucci in [BP96], rephrasing it in terms of the algebraic machinery developed in the previous sections. This algorithm works efficiently on a single d -cell of a d -complex. Our algebraic formulation is general and easy to implement using standard packages for sparse-matrix computation [Dav06].

The SPLIT algorithm is a useful tool for refining cell complexes, providing the ability to compute Boolean operations when combined with BSP trees in a progressive way [PPS04]. The SPLIT algorithm is also useful to approximate continuous maps between cell complexes. A formal definition of *subdivision* of a complex is given in [Mun84]:

Definition 17. *Let K be a cell complex. Then, a complex \tilde{K} is a subdivision of K if:*

1. *for each $\tilde{\sigma} \in \tilde{K}$ there exists $\sigma \in K$ such that $\tilde{\sigma} \subseteq \sigma$;*
2. *for each $\sigma \in K$, there exists a finite subset $\{\tilde{\sigma}_i\} \subseteq \tilde{K}$, such that $\sigma = \cup_i \tilde{\sigma}_i$.*

The SPLIT algorithm—to be detailed in the following—generates a subdivision, since for every cell $\tilde{\sigma} \in \tilde{K}$ we have by construction $\tilde{\sigma} \subseteq \sigma \in K$. Property 2 is also satisfied, since every cell in K is mapped into the union of at most two halves $\tilde{\sigma}^-$ and $\tilde{\sigma}^+$, produced by the operation $\text{split}K$.

The split algorithm

Let us first introduce two auxiliary operators, to be used for the matrix formulation of the SPLIT algorithm.

Definition 18 (Sign function).

The operator $\text{sgn}_\varepsilon : \mathbb{R}^d \rightarrow \{-1, 0, 1\}^d$ returns the matrix listing the signs of the elements v_i of a d -tuple $v = (v_i)$, taking into consideration the numerical tolerance $\varepsilon > 0$:

$$(\text{sgn}_\varepsilon v)_j = \begin{cases} -1, & v_j < -\varepsilon \\ 0, & -\varepsilon \leq v_j \leq \varepsilon \\ 1, & v_j > \varepsilon \end{cases}$$

4.5. HYPERPLANE SPLITTING

Definition 19 (Absolute value function).

The function abs operates on a matrix $M = (m_{ij})$ returning the matrix of the absolute values of its elements:

$$\text{abs}M = (|m_{ij}|)$$

Consider the splitting hyperplane h defined by the equation $\sum_p h_p x_p = b$ as a linear (affine homogeneous) form $\mathcal{E}^{d+1} \rightarrow \mathbb{R}$, represented by the row-matrix

$$h = (h_1 \quad h_2 \quad \dots \quad h_d \quad -b).$$

Let v be the column-matrix representation formed by the homogeneous coordinates of the 0-cell σ_0 :

$$v = (x_1 \quad x_2 \quad \dots \quad x_d \quad 1)^\top$$

Clearly, σ_0 belongs to the *above* subspace h^+ if and only if $h(\sigma_0) > 0$, while it belongs to the *below* subspace h^- if and only if $h(\sigma_0) < 0$. The sign of the scalar product $h v$ solves the *point location problem*.

Introducing the matrix

$$V = (v_1 \quad v_2 \quad \dots \quad v_{k_0})$$

that collects the homogeneous coordinates of all the 0-cells in K_0 , their classification with respect to the h splitting hyperplane is codified by the 0-chain $c : K_0 \rightarrow \{-1, 0, 1\}$, represented by the matrix:

$$c_0 = \text{sgn}_\varepsilon(h V).$$

The SPLIT algorithm proceeds hierarchically from 0-cells up to d -cells by (a) classifying the cells with respect to the splitting hyperplane, and (b) updating the cell complex accordingly, including the new elements in the skeletons of all orders. The algorithm is sketched in Figure 4.7.

For each dimension p , the absolute value $|c_p^i|$ of $c_p(\sigma_p^i)$ is compared with the value $f_p^i = f_p(\sigma_p^i)$ (step 5). In fact, the only p -cells that intersect the splitting hyperplane h are characterized by the inequality $|c_p^i| \neq f_p^i$.

Split example

Let us go back to the splitting example already discussed in Section 4.3 and refine the 2-complex with the hyperplane specified in Figure 4.8a. The reader

algorithm SPLIT (input: K, V, h ; output: \tilde{K}, \tilde{V});

1. $p := 0$
2. Classify the 0-cells:

$$c_0 := \text{sgn}_\varepsilon(hV)$$
3. $p := p + 1$
4. Classify the p -cells and find their “face” class:

$$c_p := (\text{abs}[\delta_{p-1}]) c_{p-1}$$

$$f_p := (\text{abs}[\delta_{p-1}]) (\text{abs}c_{p-1})$$
5. *foreach* $|c_p^i| \neq f_p^i$ *do*: Update the cell complex:

Split the i -th p -cell: $K := \beta^{p-1}(K)$;

Set the new element value: $c_{p-1}^{k_{p-1}} := 0$
6. Re-classify the p -cells of the updated cell complex:

$$c_p := \text{sgn}_\varepsilon((\text{abs}[\delta_{p-1}]) c_{p-1})$$
7. if $p < d$ then GOTO step 3, else STOP.

Figure 4.7: The SPLIT algorithm, implemented by using a classification chain and the coboundary operator.



Figure 4.8: (a) The splitting hyperplane h , and (b) the classification of vertices.

4.5. HYPERPLANE SPLITTING

57

should recall Figs. 4.4, 4.5, and 4.6 and refer to them to locate by name the cells of the complex.

The SPLIT algorithm is initialized by setting $p = 0$ and by classifying the vertices through the 0-chain

$$c_0 = \text{sgn}_\varepsilon (h (v_1 \ v_2 \ v_3)) = (-1 \ 0 \ 1),$$

as shown in Figure 4.8b. Then p is increased to 1 and 1-cells are classified by computing the 1-chains:

$$\begin{aligned} c_1 &= (\text{abs}[\delta_0]) c_0 = (0 \ -1 \ 1), \\ f_1 &= (\text{abs}[\delta_0]) (\text{absc}_0) = (2 \ 1 \ 1). \end{aligned}$$

Results are illustrated in Figure 4.9: we see that σ_1^1 should be split, since $|c_1^1| \neq f_1^1$.

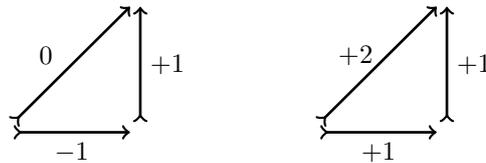


Figure 4.9: The 1-chains c_1 and f_1 used to detect the 1-cells that intersect the splitting hyperplane.

The application of the β^0 operator adds a new 0-cell (classified to 0) and a new 1-cell (see Figs. 4.10). The two 1-cells resulting from the split one, as shown in Figure 4.10b, are reclassified.

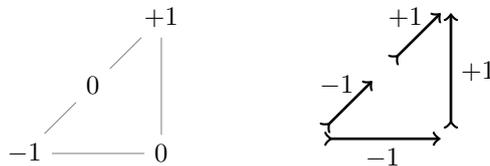


Figure 4.10: The updated cell complex, with 1-cells reclassified.

Then, p is increased to 2 and 2-cells are classified:

$$c_2 = \text{abs}[\delta_1] c_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & 1 & 1 \end{pmatrix}^\top = 0$$

$$f_2 = \text{abs}[\delta_1] \text{absc}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^\top = 4$$

(see Figure 4.11a). Hence, the σ_2^1 cell gets split, the splitting being executed by the β^1 operator that creates one 1-cell and one 2-cell, as shown in Figs 4.11. Finally the algorithm re-classifies the 2-cells and terminates, since $p = d$. The result is illustrated in Figure 4.11c, where the 2-chain generated on the refined complex \tilde{K} is illustrated.

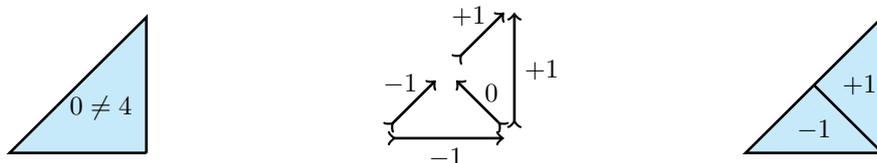


Figure 4.11: (a) Classification of the 2-cells, (b) the classification 1-chain on the refined 1-skeleton, and (c) the refined 2-skeleton with the classification 2-chain.

Subdivision of a complex

Let us denote the support space of the complex K as $\llbracket K \rrbracket$. Since SPLIT is a subdivision generator, the process can be iterated by performing a second split, i.e. $\text{split}_2(\text{split}_1 K)$, and, more in general $\text{split}^N K$.

From the *finite approximation theorem* [Mun84], we have that for any continuous map $\phi : \llbracket K \rrbracket \rightarrow \llbracket L \rrbracket$ between two cell complexes K and L , with K finite, there exists $N \in \mathbb{N}$ such that ϕ may be approximated by a map $\psi : \text{split}^N K \rightarrow L$.

Another property of the SPLIT subdivision is guaranteed by the *algebraic subdivision theorem* [Mun84]. The splitting induces a unique *chain map* ζ :

$$\zeta : \mathcal{C}(K) \longrightarrow \mathcal{C}(\tilde{K}),$$

such that the following diagram is *commutative*:

$$\begin{array}{ccccccc} \cdots & \longrightarrow & C_p & \xrightarrow{\partial} & C_{p-1} & \longrightarrow & \cdots \\ & & \zeta \downarrow & & \downarrow \zeta & & \\ \cdots & \longrightarrow & \tilde{C}_p & \xrightarrow{\partial} & \tilde{C}_{p-1} & \longrightarrow & \cdots \end{array}$$

Therefore, the induced chain map can be applied either to the subdivided chain or to the original chain complex, since

$$\partial \circ \zeta = \zeta \circ \partial.$$

As a result, boundaries in the *refined* cell complex \tilde{K} may be computed by applying the chain map ζ to boundaries evaluated in the *coarse* cell complex K .

4.6 Geometry & physics modeling

The (co)chain-complex formalism and the Hasse-matrix representation applies in a natural and straightforward way to physical modeling. Chains assign *measures* to cells, measures that may be tuned to represent the physical properties of interest (mass, charge, conductivity, stiffness, and so on). Cochains, on the other side, may be used to represent all physical quantities associated to cells through *integration* with respect to a measure. The coboundary operator stays behind the basic structural laws (balance and compatibility) involving physically meaningful cochains [Ton75, PS93, RS04]. It is also well known that k -cochains are the coarse-grained analogue of differential k -forms [Bos88, CS00]. Correspondingly, the cochain complex is a discrete version of the De Rham complex [BT82, Nak90, AFW06], naturally represented by the Hasse matrix (or its transpose).

This view on physical modeling has been increasingly advocated [Bos88, AFW06, HS97] as a way to increase numerical stability and accuracy of various numerical methods. Even more important is the that a proper use of the Hasse matrix has the potential to bring both geometric and physical modeling within a unified computational framework. According to its definition (see Section 4.2), $H(K)$ provides a compact representation of purely *topological* operators, boundary ∂ and coboundary δ , acting on chains or cochains defined on K . Such a representation is mediated by a *metric* structure which embodies far more information than the topology of the cell complex K plus the measure-like properties imparted to it by the introduction of chains. This additional structure is brought in by the seemingly innocuous identification between elementary chains and elementary cochains. However, the “obvious” cell-wise identification is associated with a conventional metric structure, easy to use on K , but totally unrelated—in general—to the metric properties relevant to the physics under consideration. Of course, the underlying topology stays untouched. Therefore, as long as one is only interested in having an easy-

to-use *metrical representation* of topological operators, the metric involved is instrumental and one is allowed to use whichever is found convenient. Nevertheless, when the object itself, not only its representation, *does* depend on the metric—as when introducing the notion of *adjacency* between cells and the related notion of *Laplacian* (see Section 6.3)—then it is essential to import into the model the relevant, physics-based metric structure, through a well-tuned identification of chains with cochains. As a consequence, the elementary chain 1σ will *not* be identified—in general—with the elementary cochain 1σ . Approaching these issues is basic to gain the possibility of transferring information from K to its refinement \tilde{K} .

A deeper discussion on metric issues is out of scope. However, we stress here that the same data structures and algorithms may be used both for solid modeling and physics-based simulations. From our vantage point, boundary representations and finite element meshes appear as two different aspects of the same Hasse representation. Furthermore, there is no fundamental distinction between different types of approximation methods: in the next chapter, by distinguishing the metrical and topological properties embodied in the Hasse representation, we will show that all linear problems formulated by all finite methods are equivalent. One consequence of this view is that the split algorithm described in Section 4.5 becomes a powerful method for progressive refinement not only of *shapes*, but also of the representation of *fields* living on those shapes.

Chapter 5

Canonical Form of Finite Methods

This chapter introduces a specific interpretation of the algebraic topological formulation of field calculus, which is conceptually simple, physically sound, computational effective and comprehensive. Giving preeminence to the cells of highest dimension allows us to generate the geometry and to simulate the physics simultaneously. Section 5.2 introduces our finite representation of field problems, centered on cells of *codimension* zero, and compares it with the more common presentation focused on nodes, i.e., cells of *dimension* zero, and in Section 5.3 we will emphasize the SPLIT algorithm as a means of mesh and field refining. In Section 5.4 our algebraic-topological approach is applied to a simple prototype problem involving the Laplacian.

5.1 Background

The quest for classification and unification in physical field theories dates back at least to the work of Maxwell [Max90]. The computational advantages of this unified view have been realized by Kron [Kro45], who developed analog-computer models to simulate a variety of physical field problems. Roth [Rot55] appears to have been the first to observe that algebraic topological foundations underlie all such models. Branin [Bra66] 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 [Ton75].

More recently, this early research led to new efforts in developing unified computational models and languages for analysis, simulation, and engineering design. Notably, Palmer and Shapiro [PS93] 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:

$$\int_R d\omega = \int_{\partial R} \omega \tag{5.1}$$

The fact that a cell-by-cell integration of a differential p -form yields a p -cochain may be summarized by the commutative diagram [BS90]:

$$\begin{array}{ccc} p\text{-cochain} & \xrightarrow{\delta} & (p+1)\text{-cochain} \\ \int \uparrow & & \uparrow \int \\ p\text{-form} & \xrightarrow{d} & (p+1)\text{-form} \end{array} \tag{5.2}$$

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 [Pal95] proposed to encode conventional FEMs for plane elasticity problems using chains as a basic data type. In [ES00, ES04] 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 [SMG99].

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 [Str88] observed that the FEM encodes a pervasive equilibrium pattern, which is at the center of the classification in [Ton75]. Mattiussi [Mat97] provided interpretations of FEM, FVM, and FDM in terms of the topological properties of the corresponding field theory. Tonti [Ton01] presented his *cell method* as a direct discrete method, bypassing the underlying continuum model. In [HS97] FDMs that satisfy desired topological properties are discussed.

Two notions introduced in [PS93] deserve consideration. First, many physical laws may be expressed *combinatorially* (not just discretely) from first prin-

5.2. DISCRETIZATION OF FIELD PROBLEMS

ciples. This idea was formalized in [CS00], 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:

$$\begin{array}{ccc}
 p\text{-cochain} & \xrightarrow{\delta} & (p+1)\text{-cochain} \\
 \lim_{cell \rightarrow 0} \downarrow & & \downarrow \lim_{cell \rightarrow 0} \\
 p\text{-form} & \xrightarrow{d} & (p+1)\text{-form}
 \end{array} \tag{5.3}$$

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 [RS04] 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.

5.2 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 chapter 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.

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[Mun84, Ton75]. Depending on the construction, D may fail to be a honest cell complex, since

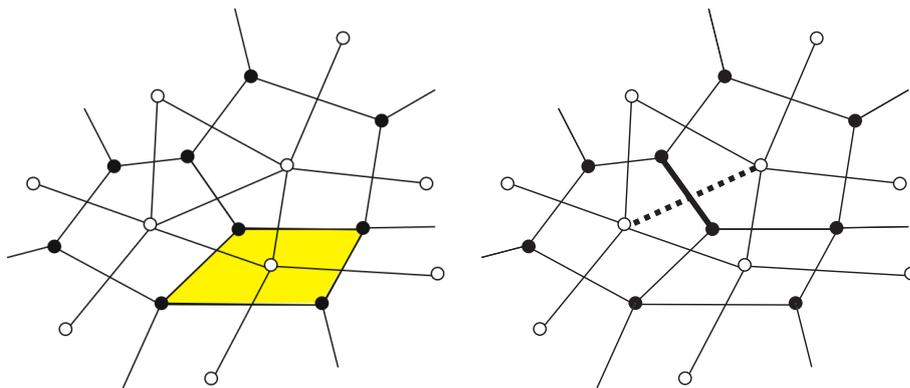


Figure 5.1: Primal and dual complexes: (a) duality between K_2 and D_0 ; (b) duality between K_1 and D_1 .

the boundary of some dual cells is not in D . However, there are several standard ways to complete D as a cell complex, see for example [Bra66].

In the $K \leftrightarrow D$ 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 5.1.

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 [Mun84] for details). This produces the commutative diagram

$$\begin{array}{ccc} C^{p-1} & \xrightarrow{\phi} & C_{n-p+1} \\ \delta \downarrow & & \downarrow \partial \\ C^p & \xrightarrow{\phi} & C_{n-p} \end{array} \quad (5.4)$$

implying that boundary and coboundary operators are related as follows:

$$\partial = \phi \circ \delta \circ \phi^{-1}, \quad \delta = \phi^{-1} \circ \partial \circ \phi. \quad (5.5)$$

Figures 5.2 and 5.3 illustrate these relationships. Figure 5.2 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 5.3

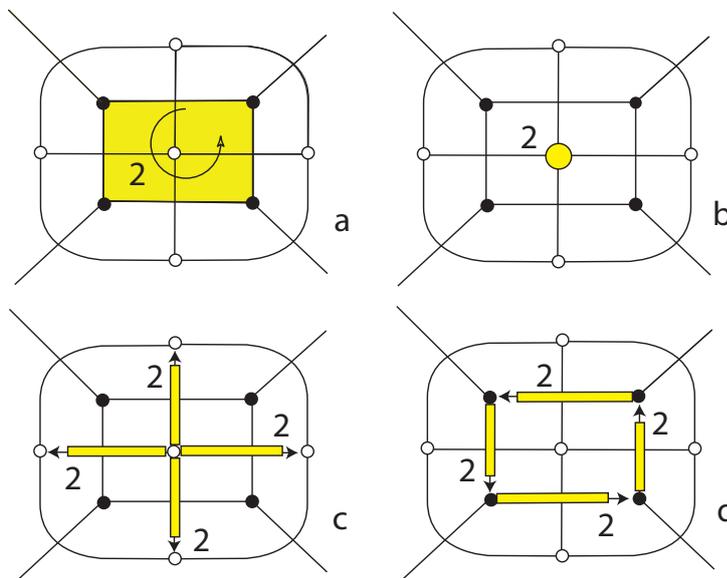


Figure 5.2: Dual computation of $\partial(2\sigma)$: (a) the elementary chain 2σ ; (b) its dual cochain value $\phi^{-1}(2\sigma)$; (c) the coboundary value $\delta(\phi^{-1}(2\sigma))$; (d) back to K : $\partial(2\sigma) = \phi(\delta(\phi^{-1}(2\sigma)))$.

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 (5.4) and (5.5), the combinatorial version of many physical laws may be factorized, as shown in [RS04], 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.

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 association could be misinterpreted, implying that the hypothetical physical measurement of that quantity is so ac-

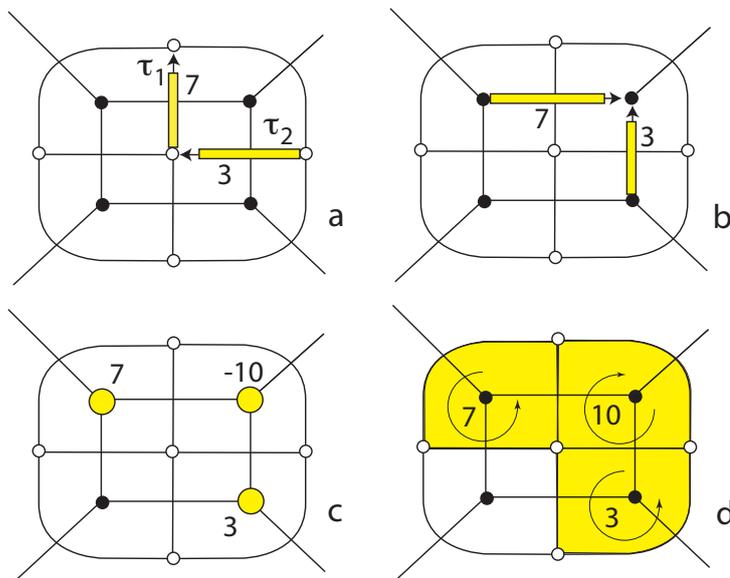


Figure 5.3: Dual computation of $\delta(7\tau_1^* - 3\tau_2^*)$: (a) the cochain value $\gamma = 7\tau_1^* - 3\tau_2^*$; (b) its dual chain $\phi(\gamma)$; (c) the boundary of the dual chain $\partial(\phi(\gamma))$; (d) back to D : $\delta(\gamma) = \phi^{-1}(\partial(\phi(\gamma)))$.

curate 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, as presented in Sections 3.1–3.3.

Lexicon

A variety of well-known physical theories may be constructed by combining primitive topological and metrical maps. In fact, many authors, including [Rot55, Bra66, Ton75, Mat97, CS00] and [RS04], 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 2.1, notice that a field $F : \mathcal{D} \rightarrow G$ is piecewise approximated over the representative cell complex K by a chain c_p , i.e. a map $K_p \rightarrow G$.

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 $\Omega \in Hom(C_{p-1}(K; G), C^{p-1}(K; G))$ 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 (5.7) is the coboundary δ . This corresponds to an *integration* on the cell decomposition of the domain, producing a p -cochain γ^p .

Existence of Canonical Form

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

$$C_p \xrightarrow{\partial} C_{p-1} \xrightarrow{\Omega} C^{p-1} \xrightarrow{\delta} C^p, \quad (5.6)$$

so that the equivalent functional formulation is expressed as

$$\langle \gamma^p, c_p \rangle = (\delta \circ \Omega \circ \partial)(c_p), \quad (5.7)$$

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*, presented in Sections 3.2, 3.3, and 3.1 respectively. There are authors that in their work recognized some underlying structure of these methods, notably [Ton75, Mat97, CS00, ES04] and [RS04], 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 previously introduced in equation (5.6).

Any given finite method supplies a way of expressing complex differential equations approximating them with a system of linear equations $M\xi = b$. A system of linear equations is then the reduction of differential relations to finite differences equations: a finite method, being it a direct finite difference, or more sophisticated approaches like Bubnov-Galerkin finite elements or Godunov finite volumes, produce a system of difference equations. The difference between these proposed and well-established methods is *how* to produce a linear problem, each yielding different numerical errors and solution stability. The following theorem proves that separating topology from metrical-physical informations, all finite methods are expressible in the canonical form (5.7): in this sense all finite methods are equivalent.

Theorem 1. *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. 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. $\text{Ker}(A) = \{0\}$.

5.2. DISCRETIZATION OF FIELD PROBLEMS

69

Proof. The necessity of $Ker(A) = \{0\}$ follows directly from the fact that M , provided by a finite method, should be 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 C 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 $\Omega = C^T M C$:

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

□

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 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.

Finite Differences

Finite difference methods (see Section 3.1) 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: $2\vartheta = \min\{\text{Vol}(F)|F \in K_{p-1}\}$.

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 5.4.

Finite Elements

Classical finite elements (see Section 3.2) replace the original problem, expressed usually under a differential formulation, with an equivalent version—the *weak form* presented in equation (3.5)—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 5.4.

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 $u'' = -f$ in (3.4) by expressing its variational form $u'' \rightarrow -u'w'$, where u and w are the field and variation functions, respectively, as expressed in (3.5).

To achieve better numerical approximations FEMs usually make use of higher order interpolating functions, influencing the $\omega_{i,j}$ coefficients that es-

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.

Finite Volumes

Similarly to the finite difference method, the finite volume method (see Section 3.3) 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. In contrast 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, 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.

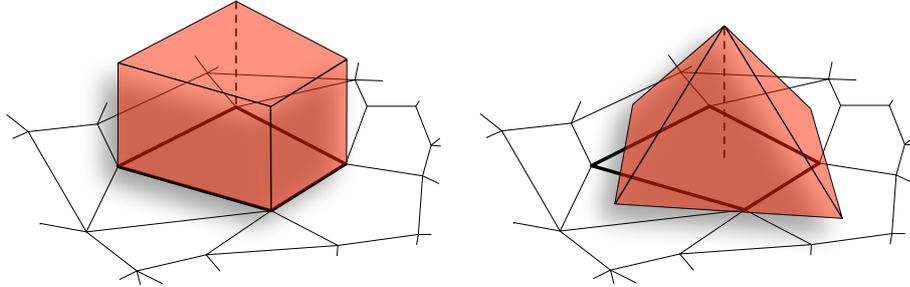


Figure 5.4: Different shape functions on the same mesh.

5.3 Adaptive Graph Representation

In the following sections we will reintroduce the Hasse diagram presented in Section 4.1, showing that this data structure can be employed for 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, described in Section 4.5 will be presented as an efficient mesh refining tool, and analyzed from the viewpoint of our proposed method.

Although the algorithm has been shown to be comfortably applied to the Hasse matrix, and already presented in its matrix-based formulation in Figure 4.7, it will be presented in the following sections in its graph-based version for its self-explanatory simplicity.

The Hasse Diagram

Let us recall the basic concepts presented in the previous chapter. 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 [Ski90]. 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 [BP96], together with an efficient SPLIT algorithm for splitting a convex cell (and its boundary faces) with an hyperplane, as shown in Figures 5.5 and 5.6. Such a spitting is the very basic operation when building a progressive BSP-tree as described in [PPS04].

5.3. ADAPTIVE GRAPH REPRESENTATION

73

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

$$C_k(K, G) = (N_k \times G, +), \tag{5.8}$$

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 $E_n := \{(n, n_j) \in E | n_j \in N_{k-1}\}$, and $N_n := \{n_j \in N_{k-1} | (n, n_j) \in E\}$. 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:

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

In other words, the boundary of the elementary chain gn is obtained simply by summing the (properly signed) coefficient transferred 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

$$C^k(K, G) = (N_k^* \times G, +). \tag{5.9}$$

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 algorithm as seen in [BP96] and will be described in Section 5.3.

Mesh Refining

The framework presented Section 5.2 applies to general meshes, and does not require any particular cell shape. Representing a finite method in its canonical form (5.6) 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 [PPS04], providing an effective method of adaptive geometrical refinement of complex shapes.

In the present section we will extend the SPLIT algorithm described in Section 4.5 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 5.5, 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 $c_2 = t_1 C_1 + t_2 C_2 + \dots$ 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

$$c_2 = t_1 C_1 + t_2 C_2 + \dots \longrightarrow \xi = [C_1, C_2, \dots]^T$$

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 with a product of matrices as $A_2^T \Omega A_2 \xi$. The thermal tension between adjacent cells

5.3. ADAPTIVE GRAPH REPRESENTATION

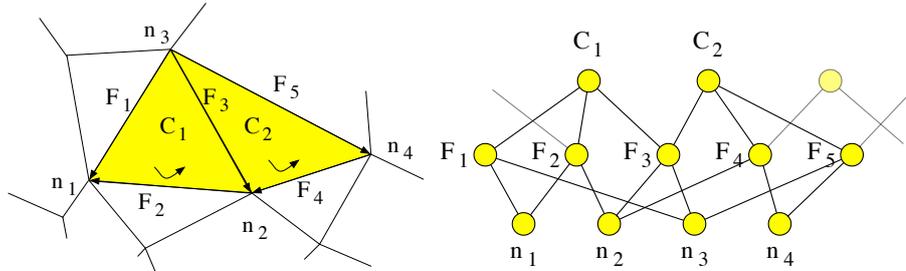


Figure 5.5: 2-cell complex with the corresponding Hasse diagram, focusing on two cells, C_1 and C_2 .

is then given by the 1-chain f_1 :

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

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 :

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

The SPLIT algorithm 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 5.6, 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,

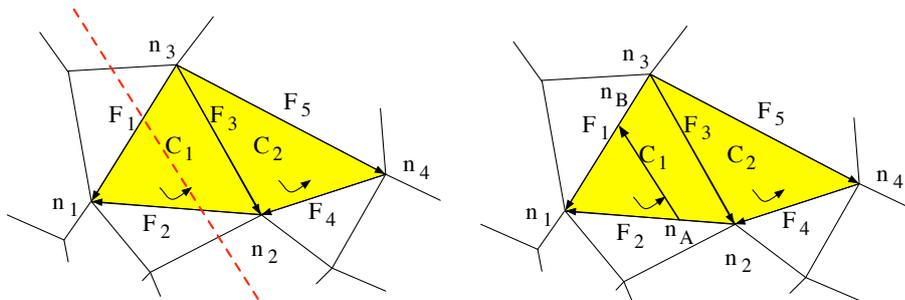


Figure 5.6: (a) Splitting the C_1 simplex with an hyperplane; (b) non-simplicial cells are generated.

our algebraic-topological approach allows the use of cells of different shapes 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 5.6 and 5.7. 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 . Using the matrix representation of the above complex as explained in Section 4.1, we would update the coboundary matrix δ_0 (i.e. expressing incidence relations between 0 and 1-cells), so that the new $\tilde{\delta}_0$ would be the sum of three contributions:

$$[\tilde{\delta}_0] = (S_1 \quad S_2 \quad S_3) [\delta_0] \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}$$

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 $\tilde{\Omega}$, of dimension $|\tilde{K}_1| \times |\tilde{K}_1|$, where $|\tilde{K}_1| = |K_1| + 3$, will then reflect the newly created cells that replace F_1 and

5.3. ADAPTIVE GRAPH REPRESENTATION

77

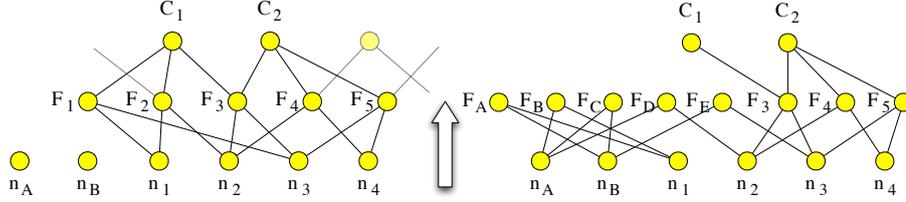


Figure 5.7: (a) Creation of two new 0-cells n_A and n_B ; (b) the subsequent creation of 1-cells F_A, F_B, F_C, F_D, F_E , replacing F_1 and F_2 .

F_2 :

$$\tilde{\Omega} = \begin{bmatrix} \omega_{F_A, F_A} & \omega_{F_A, F_B} & \dots & \omega_{F_A, F_3} & \omega_{F_A, F_4} & \omega_{F_A, F_5} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{F_E, F_A} & \omega_{F_E, F_B} & \dots & \omega_{F_E, F_3} & \omega_{F_E, F_4} & \omega_{F_E, F_5} & \dots \\ \omega_{F_3, F_A} & \omega_{F_3, F_B} & \dots & \omega_{F_3, F_3} & \omega_{F_3, F_4} & \omega_{F_3, F_5} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{|\tilde{K}_1| \times |\tilde{K}_1|}$$

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 \tilde{A}_2 with dimensions $|\tilde{K}_1| \times |\tilde{K}_2|$, where $|\tilde{K}_2| = |K_2| + 1$. With reference to Figure 5.8, the updated boundary chain \tilde{f}_1 can be expressed with the following product of matrices:

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

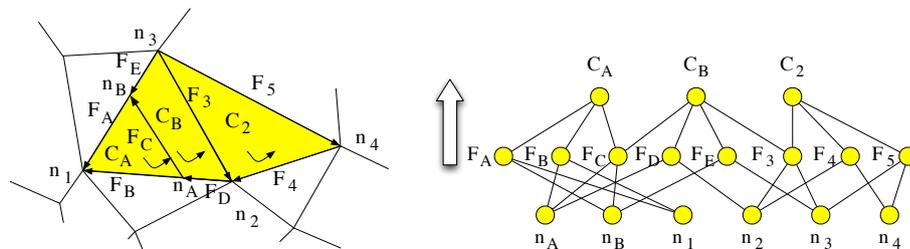


Figure 5.8: The final cell complex and its relative Hasse diagram.

5.4 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.

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 5.9, 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 5.9):

$$\begin{cases} q_{15} = (\lambda_{15}/h_y)(T_1 - T_5)(-1) \\ q_{65} = (\lambda_{65}/h_x)(T_6 - T_5)(-1) \\ q_{95} = (\lambda_{95}/h_y)(T_5 - T_9) \\ q_{45} = (\lambda_{45}/h_x)(T_5 - T_4) \end{cases} \quad (5.10)$$

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

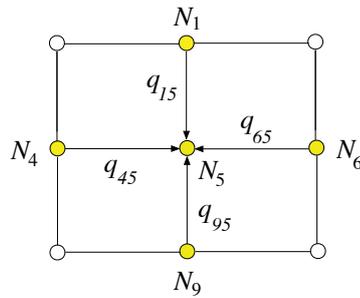


Figure 5.9: Five-point finite-difference scheme.

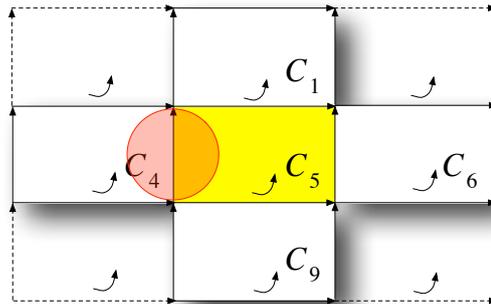


Figure 5.10: Cell-based finite difference scheme with its metrics.

quantity, say zero:

$$\sum q_{ij} = q_{15} + q_{65} + q_{95} + q_{45} = 0. \tag{5.11}$$

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 5.10 shows the oriented cell complex corresponding to the FD mesh in Figure 5.9.

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

$$T = \sum_i T_i C_i \quad (5.12)$$

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

$$\begin{cases} F_{15} = T_1 - T_5 \\ F_{65} = T_5 - T_6 \\ F_{95} = T_5 - T_9 \\ F_{45} = T_4 - T_5 \end{cases} \quad (5.13)$$

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 5.10, our choice is to map each 1-face with all the 1-cells within an euclidean distance of $2\vartheta = \min\{\text{Vol}(F) | F \in K_1\}$ 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} :

$$\Omega = [\omega_{i,j}], \quad \omega_{i,j} = \begin{cases} 0, & C_i \cap C_j = \emptyset \\ \sigma_{ij}, & C_i \cap C_j \neq \emptyset \end{cases} \quad (5.14)$$

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

$$\begin{cases} q_{15} = \sigma_{15}(T_1 - T_5) \\ q_{65} = \sigma_{65}(T_5 - T_6) \\ q_{95} = \sigma_{95}(T_5 - T_9) \\ q_{45} = \sigma_{45}(T_4 - T_5) \end{cases} \quad (5.15)$$

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 \tilde{Q}^2 which provides the total heat flux entering each cell. Focusing as before on cell C_5 , this gives the equation:

$$\delta q = \sum \tilde{q}_{ij} = -\tilde{q}_{15} + \tilde{q}_{65} + \tilde{q}_{95} - \tilde{q}_{45} = 0 \quad (5.16)$$

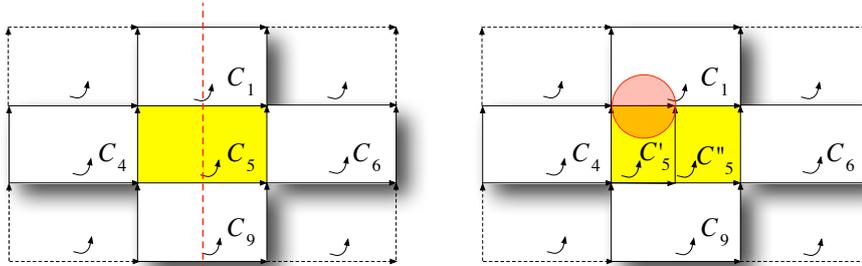


Figure 5.11: Splitting the cell C_5 yields a change in the distance used to describe face relationships, affecting the $\sigma_{ij} = \sigma(\text{Vol}(F_{ij}))$ parameters.

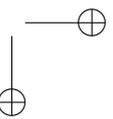
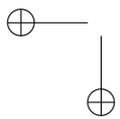
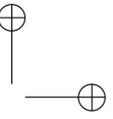
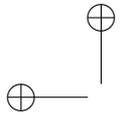
which coincides with Equation (5.11), provided that

$$\begin{cases} \sigma_{15} &= \lambda_{15}/h_y \\ \sigma_{65} &= \lambda_{65}/h_x \\ \sigma_{95} &= \lambda_{95}/h_y \\ \sigma_{45} &= \lambda_{45}/h_x \end{cases} \quad (5.17)$$

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 $\lambda_{ij} \propto \text{Vol}(F_{ij})$.

In case of a SPLIT, the structure of our problem remains the same. With reference to Figure 5.11, 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 5.2, these metrical informations are stored and comfortably updated in the Hasse diagram as the SPLIT algorithm proceeds.

In the next chapter, we will outline a simple metrical problem, providing a numerical example of our proposed algebraic topological approach on a anisotropic version of the mesh presented in this section.



Chapter 6

A Metrical Example

In this chapter we will illustrate the algebraic topological approach outlined in Section 5.2. In the following the case formulated in Section 5.4 will be taken into account as the prototype of our examples.

6.1 The Normal Gradient

The finite methods described in Sections 3.1–3.3 are techniques meant to produce a system of linear equations, given a partial differential equation, or a system of PDEs. In practice, much of the attention is payed on the solution provided by such systems, with finite methods seen as “black boxes”. The entire process is constituted by steps:

- Partial differential equations;
- Mesh generation;
- Finite Method: system of linear equations $A\xi = b$;
- Solution inverting the A matrix: $\xi = A^{-1}b$

In this section we will focus on the “finite method” step that yields the matrix A from partial differential equations. The last step, the solution of the system of linear equations, can be achieved independently of the applied finite method with various techniques, such as the Jacobi or the Gauss-Seidel methods.

In order to show the algebraic topological approach proposed in Section 5.2, we will concentrate on a numerical example applied to a simple differential equation involving the *gradient*. Let us then suppose to have a physical phenomenon described by the following differential relation:

$$u = \nabla f, \tag{6.1}$$

where f is a scalar field on a domain \mathcal{D} , and u is a vector field coincident with the gradient of the scalar field. As we previewed, we will produce a system of linear equation for (6.1) on the domain \mathcal{D} , leaving the matrix inversion to a separate and independent task.

In our test case we will focus on a domain constituted by a *toroidal* surface depicted in Figure 6.1. The surface will be then decomposed into a cell complex topologically equivalent to the rectangular grid introduced in Section 5.4.

The regularity of a solution, its stability, and numerical accuracy rely in part on the chosen decomposition of the domain \mathcal{D} . It is known that irregular meshes lead to numerical problems, for instance on a two dimensional domain approximated by a simplicial mesh, triangles with high anisotropy are not good candidates for physical computations. In other words, cells that “look like a needle” should be rejected due to probable numerical errors of obvious etiology.

In our tests, we will deliberately use a highly distorted version of a rectangular mesh as pictured in Figure 6.2, where the cell complex is pictured with the classic topological torus unfolding; the geometry of a cell in the complex is showed in detail in Figure 6.3.

We have already seen in Section 5.2 that a non-singular matrix provided by any finite method, can be reduced in the canonical form

$$A\xi = b \longrightarrow (\delta\Omega\partial)\xi = b,$$

where the topological operator coboundary δ and boundary ∂ are represented by the incidence matrix of the oriented cell complex, and its transpose. In the following, with reference to Figure 6.3, we will analyze the *normal gradient*

$$u \cdot n = \nabla f \cdot n,$$

or better, the normal component of the gradient field with respect to the cellular decomposition we have previously described. Both our tests will consider an two-dimensional field represented by an harmonic function on the toroidal domain depicted in Figure 6.1.

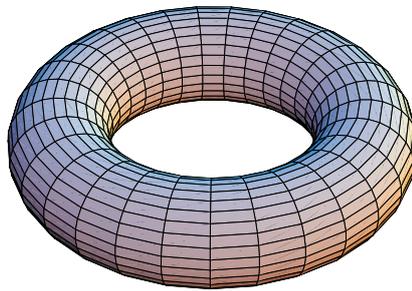


Figure 6.1: The domain \mathcal{D} of equation (6.1).

6.2 A First Approximation

With reference to Figures 6.2 and 6.3, let us consider the toroidal domain \mathcal{D} whose topological unfolding is the set $[0, \ell] \times [0, m]$, and let us consider the function

$$f(\xi, \eta) := \sin\left(\pi \frac{\xi}{\ell}\right) \sin\left(\pi \frac{\eta}{m}\right). \quad (6.2)$$

The function, defined on the domain \mathcal{D} with $\ell = 50$ and $m = 25$, is pictured in Figure 6.4. The discrete domain $\tilde{\mathcal{D}}$ will consist of a 50×5 grid, with a 2-chain

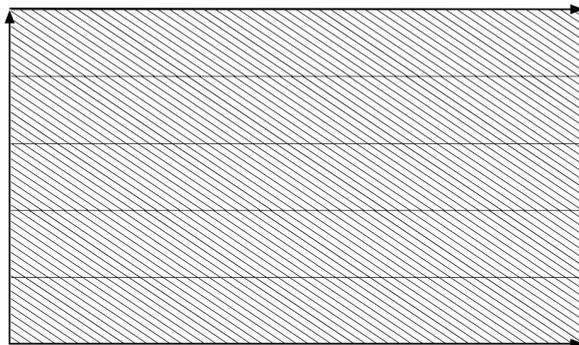


Figure 6.2: The mesh corresponding to the domain \mathcal{D} .

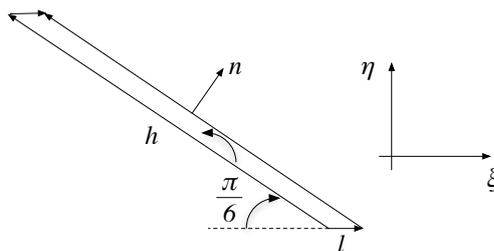


Figure 6.3: A cell in the discretized version of the domain \mathcal{D} .

representing the average of the field as described in the previous chapter. The sampled field is drawn in Figure 6.5, showing cell values.

To approximate the normal component of the gradient function—pictured in Figure 6.6 where the hue has been used to indicate the magnitude—we will use the metric described in Section 5.4, based on the euclidean distance.

In this example we chose to mimic the five-point finite difference method by establishing relations between adjacent cells. As we have previously showed, expressing a five-point FDM in its canonical form, we will relate each 1-cell only with itself, in other words, we consider “adjacent” cells with an euclidean distance ϑ such that $2\vartheta = \min\{\text{Vol}(F)|F \in K_1\}$. This choice is reflected on the Ω matrix, the only part in the canonical form that is aware of metrics and

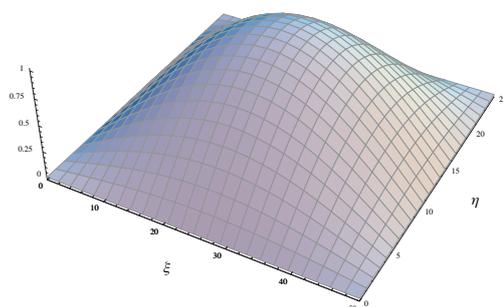


Figure 6.4: Image of the function f in equation 6.2, with $\ell = 50$ and $m = 25$.

physics. The Ω matrix connecting 1-cells will be a diagonal matrix:

$$\Omega = [\omega_{i,j}], \quad \omega_{i,j} = \begin{cases} 0, & F_i \cap F_j = \emptyset \\ \frac{1}{5}, & F_i \cap F_j \neq \emptyset. \end{cases}$$

The results of computing the normal gradient using the above matrix, although employing a coarse grid and a naïve metrics for Ω , are encouraging, as we can see comparing Figures 6.7 and 6.8. We stress the fact that this example has the same result as a finite difference approximation of the differential equation (6.2) presented in the canonical form.

6.3 A Finer Estimate

In this section we will consider the equation (6.2) on the domain $\mathcal{D} = [0, \ell] \times [0, m]$, with $\ell = 50$ and $m = 250$. In this example we will approximate the

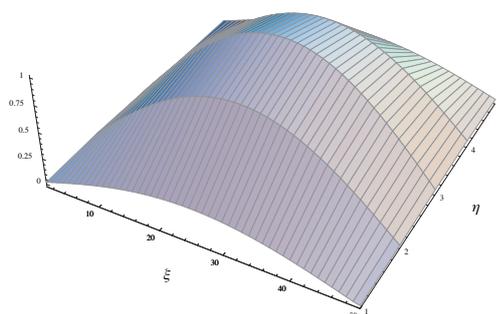


Figure 6.5: Samples of the field in Figure 6.4.

normal gradient of the function

$$f(\xi, \eta) := \sin\left(\pi \frac{\xi}{\ell}\right) \sin\left(\pi \frac{\eta}{m}\right),$$

using the a numerical approximation of the field constituted by a 50×50 samples grid. The topology of the example is essentially the same as the previous, and the mesh is composed of elements already described in Section 6.2. In Figure 6.9 we picture the image of the field f , and Figure 6.10 shows the sampled version of the function, represented by a 2-chain \tilde{c}_2 .

In order to approximate the normal gradient we will influence the coefficients of the Ω matrix by relating each face to a sufficient number of “neighbors”. The previous example showed a simple metrical relation that associates each face to itself, or in other words, considering in the approximation of the normal gradient, simply the results from the application of the boundary operation. We are now considering a more “sophisticated” metrical relation, where

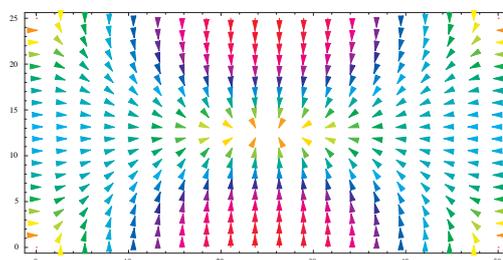


Figure 6.6: The gradient function of Equation (6.2).

the influence distance depends on the volume of the face itself. Moreover, the metric involved in the calculus is not the classic euclidean distance: we will consider a distance function induced by a linear 1-covector based at a point $x \in F, \forall F \in K_1$. Such a function is pictured in Figure 6.11. In other words, we will consider all the cells within a *linear* distance

$$2\vartheta_F = \text{Vol}(F). \tag{6.3}$$

In this particular cellular decomposition, all the faces have the same volume, allowing ϑ to be constant on the cell complex. Let us establish the face numbering as follows: numbers from 1 to 250 will indicate the long edges of each cell in the complex, and 251 to 500 for the short edged faces, as indicated in Figure 6.12.

We have introduced the coefficients ω_{F_i, F_j} of Ω as a result of metrical (and physical) quantities involved in the problem formulation, measuring the “influ-

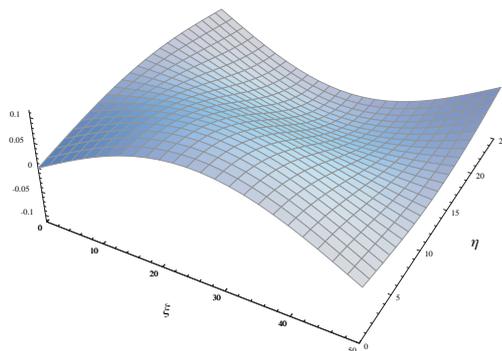


Figure 6.7: The normal gradient function f with $\ell = 50$ and $m = 25$.

ence” between the 1-faces F_i and F_j . Such values will obviously have an impact on the numerical approximation of the problem illustrated in Section 6.1.

The proposed example uses an unsophisticated measure of the influence of neighboring cells in the approximation of the normal gradient, although not as simple as in the example in Section 6.2. The metric involved in the formulation lead to define our Ω matrix as the block matrix

$$\Omega = \frac{1}{5} \begin{bmatrix} L & S \\ 0 & 0 \end{bmatrix}.$$

The matrices L and S contain the weights regarding the long edged faces and the small edged ones, respectively. Due to the toroidal nature of the domain \mathcal{D} , both L and S will consist of a “shifted” version of a vector of weights, or better, a “rotated” version of such values, mimicking the *Rotate Right* bit-wise operation instead of a *Shift Right*. In other words, shifting a value (on the

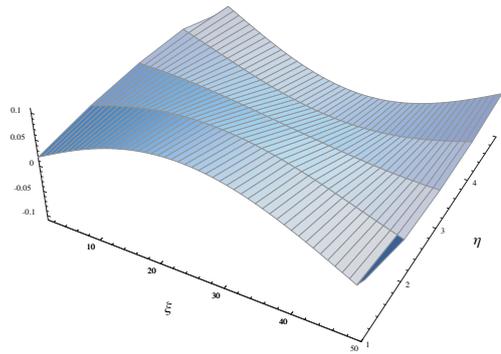


Figure 6.8: Normal gradient with a naïve metrical approach.

right) will not discard the exceeding value, as we can see for instance in the FE example in Section 3.2, instead it will wrap on the right side of the matrix.

The resulting normal gradient is pictured in Figure 6.14. The L and S matrixes used in the computation are as follows:

$$L = \begin{bmatrix} 1 & 0.5 & 0.1 & 0.01 & 0 & \dots & 0 & -0.01 & -0.1 & -0.5 \\ -0.5 & 1 & 0.5 & 0.1 & 0.01 & 0 & \dots & 0 & -0.01 & -0.1 \\ -0.1 & -0.5 & 1 & 0.5 & 0.1 & 0.01 & 0 & \dots & 0 & -0.01 \\ \vdots & & & & & & & & \ddots & \vdots \\ 0.5 & 0.1 & 0.01 & 0 & \dots & 0 & -0.01 & -0.1 & -0.5 & 1 \end{bmatrix},$$

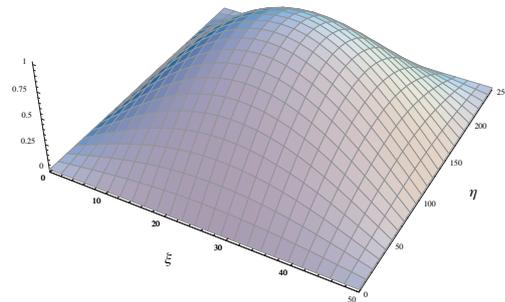


Figure 6.9: Image of the function f , with $\ell = 50$ and $m = 250$.

$$S = \begin{bmatrix} 0.05 & 0.01 & 0.001 & 0 & \dots & 0 & 0.001 & 0.01 & 0.05 \\ 0.05 & 0.05 & 0.01 & 0.001 & 0 & \dots & 0 & 0.001 & 0.01 \\ 0.01 & 0.05 & 0.05 & 0.01 & 0.001 & 0 & \dots & 0 & 0.001 \\ \vdots & & & & & & & \ddots & \vdots \\ 0.01 & 0.001 & 0 & \dots & 0 & 0.001 & 0.01 & 0.05 & 0.05 \end{bmatrix}.$$

6.3. A FINER ESTIMATE

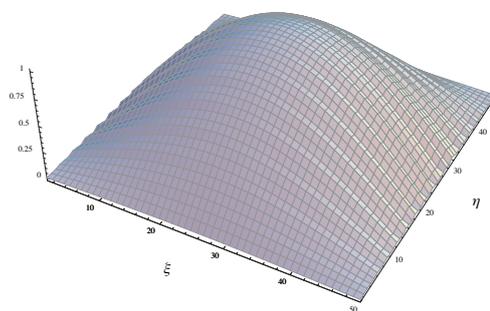


Figure 6.10: Samples of the function pictured in Figure 6.9.

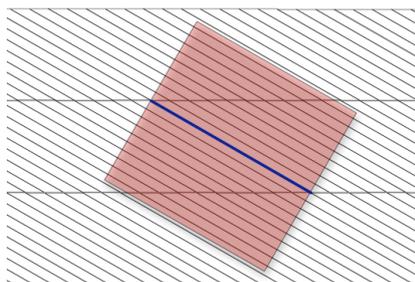


Figure 6.11: The metrical relation of equation (6.3).

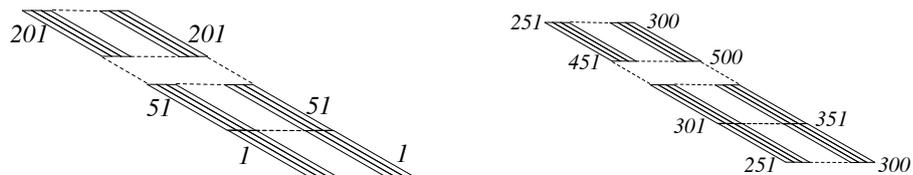


Figure 6.12: Numbering of 1-cells of the mesh.

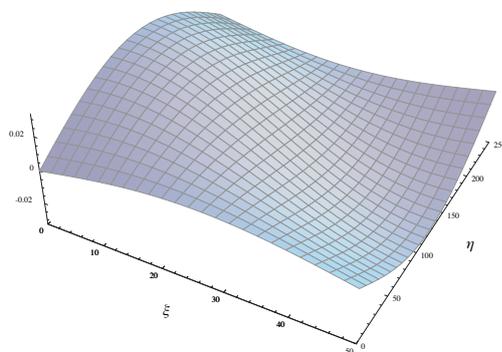


Figure 6.13: Normal gradient $\nabla f \cdot n$, with $\ell = 50$ and $m = 250$.

6.3. A FINER ESTIMATE

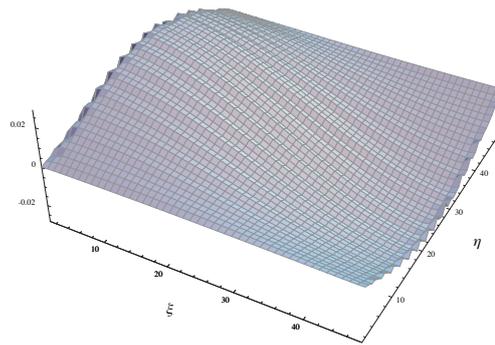
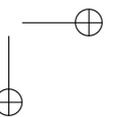
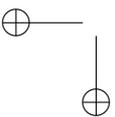
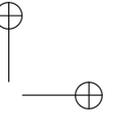
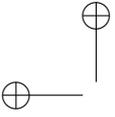


Figure 6.14: The resulting normal gradient.



Conclusions

The quest for a physically sound representation of physical phenomena is far from being concluded. We have presented in the previous chapters a different view over the algebraic topological foundations of finite methods. We also provided a review of a well-known mesh refinement algorithm employing chains and cochains, showing its importance also as a field approximation instrument.

Final Remarks

Historically, the development of boundary representation schemes in solid modeling was driven by limited computational resources, and the usual space-time trade-offs [Woo85]. A typical boundary representation was chosen

1. to save memory, when RAM was small and expensive,
2. to spare disk access times, by giving efficient answers to topological queries.

Contrary to what might appear at first sight, the proposed splitting algorithm involving the Hasse matrix (see Section 4.1) does not imply higher theoretical complexity, since the number of non-zero elements in the Hasse matrix $H(K)$ is essentially of the same order as the number of adjacency pointers in a typical graph-based representation of the cell complex K . Furthermore, the Hasse matrix serves as a unifying standard for all boundary representations; the difference between different graph structures amount to different methods [Dav06] for encoding a subset of the sparse matrix $H(K)$.

We also note that the chain complex is a standard tool for representing and analyzing topological properties of arbitrary cellular spaces. It follows that the proposed Hasse matrix and transformations may codify much more general models, without restrictions on orientability, (co)dimension, manifoldness, connectivity, homology, and so on.

The proposed canonical form is a physically valid and theoretically straightforward interpretation of finite methods based on an algebraic-topological formulation 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 the Hasse matrix to support mesh (and field) refinement in case it is needed, and only where it is required.

The resulting framework, centered on a matrix representation of the domain of interest, unifies several geometric and physical finite formulations, and supports local (progressive) refinement and coarsening.

Future Work

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 5.2.

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.

As an example of such a possibility, let us refer to a common forced convection problem of a fluid over a plate. It is a known result of thermodynamics that in a simplified model of such a problem, we have a subdomain where the fluid moves in a laminar manner, a turbulent subdomain and a mixed-state one. The influence of each region of the space on the others, for instance the

FUTURE WORK

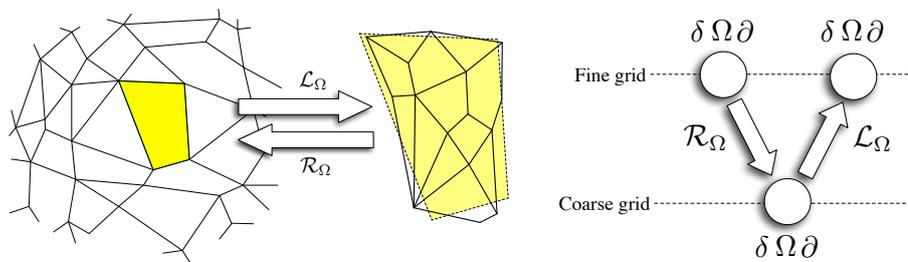


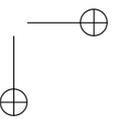
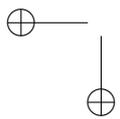
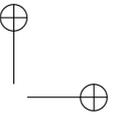
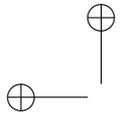
Figure 1: A mixed multi-grid/multi-physics approach mesh (a) and V-Cycle (b).

laminar with the mixed-state subregion, will drive the coefficients expressed in the sub-matrix $\Omega_{i,j}$.

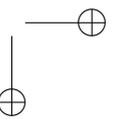
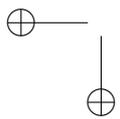
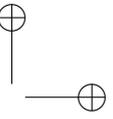
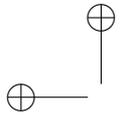
Another promising research direction is the application of our framework in a mixed multi-grid/multi-physics environment. The current multi-grid approach approximate solutions on multiple meshes at different granularity, interchanging results with relaxation \mathcal{R} (fine-to-coarse grid) and interpolator \mathcal{L} operators (coarse-to-fine grid), see e.g. [mgr88] and [mgr03]. 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}_Ω and \mathcal{R}_Ω will modify the Ω matrix at each stage of computation as the subscript suggests. Then a typical multigrid 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 .



Appendices



Appendix

Adjacency matrices

In graph theory, the adjacency matrix of vertices is one of the possible representations of a graph $G = (N, E)$ which is, by definition, a 1-complex $K = (K_0, K_1)$.

The well-known relation between the incidence matrix of a graph, its transpose and the adjacency matrix of its vertices can be generalized to boundary and coboundary operators of every order, and to the adjacency of p -cells in K_p , for any dimension p .

The topology of the 3-complex K depicted in Figure 2 is represented by the matrices $[\delta_0] = [\partial_1]^\top$, $[\delta_1] = [\partial_2]^\top$, and $[\delta_2] = [\partial_3]^\top$.

Definition 20. *The symmetric matrices*

$$[\partial_{p+1}][\delta_p] \quad \text{and} \quad [\delta_{p-1}][\partial_p]$$

define the adjacency between p -cells through $(p + 1)$ -cells and $(p - 1)$ -cells, respectively.

We stress here that such a representation makes use of the *standard* metric on K , by which each elementary chain 1σ is identified with the elementary cochain 1σ . The metric information introduced in this way becomes important when introducing and computing *adjacency matrices*, which imply the successive application of the boundary and coboundary operators (or viceversa).

It is worth mentioning that the discrete *Laplace-De Rham operators*

$$[\partial_{p+1}][\delta_p] + [\delta_{p-1}][\partial_p]$$

are just sums of adjacency matrices. They depend essentially on the metric carried by the matrix representation of boundary and coboundary operators.

A sample calculation of these operators, based on the complex pictured in Figure 2, is as follows:

$$[\partial_1][\delta_0] = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{pmatrix},$$

$$[\partial_2][\delta_1] = \begin{pmatrix} 2 & -1 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 3 & 1 & -1 & 1 & -1 & 0 \\ 0 & -1 & 1 & 1 & 3 & -1 & 0 & 1 & -1 \\ -1 & 0 & 1 & -1 & -1 & 3 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 2 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 2 \end{pmatrix},$$

$$[\delta_0][\partial_1] = \begin{pmatrix} 2 & 1 & 1 & 1 & 0 & 1 & -1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 1 & -1 & 0 & -1 & 0 \\ 1 & 1 & 2 & -1 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 2 & 1 & 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 & 2 & -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 2 & -1 & 1 & 0 \\ -1 & 0 & 0 & -1 & 0 & -1 & 2 & 1 & 1 \\ 0 & -1 & 0 & 0 & -1 & 1 & 1 & 2 & 1 \\ 0 & 0 & -1 & 1 & 1 & 0 & 1 & 1 & 2 \end{pmatrix},$$

$$[\partial_3][\delta_2] = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 1 & 1 \end{pmatrix},$$

$$[\delta_1][\partial_2] = \begin{pmatrix} 3 & -1 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & -1 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & -1 \\ -1 & -1 & -1 & 3 & 1 & 1 & 1 \\ -1 & 0 & 0 & 1 & 3 & -1 & -1 \\ 0 & -1 & 0 & 1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 & -1 & -1 & 3 \end{pmatrix},$$
$$[\delta_2][\partial_3] = \begin{pmatrix} 4 & -1 \\ -1 & 4 \end{pmatrix}.$$

Bibliography

- [Ada75] R. A. Adams. *Sobolev Spaces*. Academic Press, 1975.
- [AFW06] D.N. Arnold, R.S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.
- [Ale98] P. S. Alexandroff. *Combinatorial Topology*. Dover, New York, 1998.
- [Bae03] John Baez. Euler characteristic versus homotopy cardinality. Lecture at the Program on Applied Homotopy Theory, Fields Institute, Toronto, September 2003.
- [Bau72] B. G. Baumgart. *Winged-Edge Polyhedron Representation*. Technical Report Stan-CS-320, Artificial Intelligence Laboratory, Stanford University, CA, 1972.
- [Boa05] David Boal. *Mechanics of the Cell*. Cambridge University Press, Cambridge, UK, 2005. Third edition.
- [Boo03] George Boole. *A treatise on the calculus of finite differences*. Macmillan, 1872 (Dover Publications, 2003).
- [Bos88] A. Bossavit. Whitney forms: a class of finite elements for three-dimensional computations in electromagnetism. *Science, Measurement and Technology, IEE Proceedings A*, 135(8):493–500, 1988.
- [BP96] Chandrajit L. Bajaj and Valerio Pascucci. Splitting a complex of convex polytopes in any dimension. In *SCG '96: Proceedings of the twelfth annual symposium on Computational geometry*, pages 88–97, Philadelphia, PA, 1996. ACM Press.

- [Bra66] F. H. Branin. The algebraic-topological basis for network analogies and the vector calculus. In *Proceeding of the Symposium on Generalized Networks*, volume 16, pages 453–491. Polytechnic Institute of Brooklyn, 1966.
- [Bri93] E. Brisson. Representing geometric structures ind dimensions: Topology and order. *Discrete and Computational Geometry*, 9(1):387–426, 1993.
- [BS90] Paul Bamberg and Schlomo Sternberg. *A Course in Mathematics for Students of Physics*. Cambridge University Press, Cambridge, England, 1988,1990.
- [BT82] R. Bott and L. Tu. *Differential forms in algebraic topology*. Springer-Verlag (GTM 82), 1982.
- [CFL28] R. Courant, K. O. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.
- [CHSS95] J. Castillo, J. Hyman, M. Shashkov, and S. Steinberg. High-order mimetic finite difference methods on nonuniform grids. In *International International Conference on Spectral and High Order Methods*, 1995.
- [CS00] Jeffrey A. Chard and Vadim Shapiro. A multivector data structure for differential forms and equations. *IMACS Transactions Journal, Mathematics and Computer in Simulation*, 54(1):33–64, 2000.
- [Dav06] T. A. Davis. Direct methods for sparse linear systems. In *Fundamentals of Algorithms*. SIAM, 2006.
- [ES00] R. Egli and N. F. Stewart. A framework for system specification using chains on cell complexes. *Computer Aided Design*, 32(9):447–459, 2000.
- [ES04] R. Egli and N. F. Stewart. A framework for system specification using chains on cell complexes. *Mathematics and Computers in Simulations*, 66:449–468, 2004.
- [EW79] C.M. Eastman and K.J. Weiler. *Geometric Modeling Using the Euler Operators*. Institute of Physical Planning, Carnegie-Mellon University, 1979.

- [God59] S. K. Godunov. A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–306, 1959.
- [GS85] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi. *ACM Transactions on Graphics (TOG)*, 4(2):74–123, 1985.
- [Har83] Donald G. Hartig. The Riesz representation theorem revisited. *American Mathematical Monthly*, 90(4):277–280, 1983.
- [Hat02] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.
- [Hof89] C.M. Hoffmann. *Geometric and solid modeling: an introduction*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1989.
- [HS97] J.M. Hyman and M. Shashkov. Natural discretizations for the divergence, gradient and curl on logically rectangular grids. *International Journal of Computers and Mathematics with Applications*, 33(4):81–104, 1997.
- [Joh88] Claes Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1988.
- [Kin93] L. C. Kinsey. *Topology of Surfaces*. Springer-Verlag, 1993.
- [Kre91] Erwin Kreyszig. *Differential Geometry*. Dover Publications, 1991.
- [Kro45] Gabriel Kron. Numerical solution of ordinary and partial differential equations by means of equivalent circuits. *Journal of Applied Physics*, 126:172–186, March 1945.
- [LBZ⁺00] Harvey Lodish, Arnold Berk, Lawrence S. Zipursky, Paul Mutsaers, David Baltimore, and James Darnell. *Molecular Cell Biology*. W. H. Freeman, New York, NY, USA, 2000. Fourth edition.
- [Lev02] Randall J. Leveque. *Finite volume methods for hyperbolic problems*. Cambridge texts in applied mathematics. Cambridge University Press, 2002.
- [LL92] H. Levy and F. Lessman. *Finite difference equations*. Dover Publications, 1992.

- [Män88] M. Mäntylä. *Introduction to Solid Modeling*. WH Freeman & Co. New York, NY, USA, 1988.
- [Mas93] H. Masuda. Topological operators and Boolean operations for complex-based nonmanifold geometric models. *Computer Aided Design*, 25(2):119–29, 1993.
- [Mat97] Claudio Mattiussi. An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology. *Journal of Computational Physics*, 133:289–309, 1997.
- [Max90] James Clerk Maxwell. On the mathematical classification of physical quantities. In W.D. Niven, editor, *The Scientific Papers of James Clerk Maxwell*, pages 257 – 266. Cambridge University Press, 1890.
- [mgr88] *Multigrid Methods*. Lecture Notes in Pure and Applied Mathematics. Dekker, 1988.
- [mgr03] *Matrix-Based Multigrid: Theory and Applications*. Numerical Methods and Algorithms. Kluwer Academic Publishers, 2003.
- [Mun84] James R. Munkres. *Elements Of Algebraic Topology*. Addison Wesley, Reading MA, 1984.
- [Nak90] M. Nakahara. *Geometry, topology and physics*. Institute of Physics Publishing, 1990.
- [OR90] M. A. O’Connor and J. R. Rossignac. SGC: A dimension independent model for pointsets with internal structures and incomplete boundaries. In *IFIP/NSF Workshop on Geometric Modeling, Rensselaerville, NY, 1988*. North-Holland, September 1990.
- [Pal95] Richard S. Palmer. Chain models and finite elements analysis: an executable chains formulation of plane stress. *Computer Aided Geometric Desisgn*, 12(7):733–770, 1995.
- [PPS04] A. Paoluzzi, V. Pascucci, and G. Scorzelli. Progressive dimension-independent boolean operations. In *ACM Symposium on Solid Modeling and Applications*, Genova, June 2004. Eurographics, Proc. of Int. Convention on Shapes and Solids.

- [PS93] R. Palmer and Vadim Shapiro. Chain models of physical behavior for engineering analysis and design. *Research in Engineering Design*, 5(3):161–184, 1993.
- [Req77] A. A. G. Requicha. Mathematical models of rigid solid models. Production automation project, University of Rochester, 1977.
- [Req80] A. A. G. Requicha. Representations for rigid solids: theory, methods and systems. *ACM Computer Surveys*, 12(4):437–464, 1980.
- [Rie07] F. Riesz. Sur une espèce de géométrie analytiques des systèmes de fonctions sommables. *C. R. Acad. Sci. Paris*, 1907.
- [Rie09] F. Riesz. Sur les opérations fonctionelles linéaires. *C. R. Acad. Sci. Paris*, 1909.
- [Rot55] 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.*, 41:518–521, 1955.
- [RS04] V. Ramaswamy and V. Shapiro. Combinatorial laws for physically meaningful design. *Journal of Computing and Information Science in Engineering*, 4(3):3–10, 2004.
- [SF73] G. Strang and G. J. Fix. *An analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [Sil81] C.E. Silva. Alternative definitions of faces in boundary representations of solid objects. Technical report, Production Automation Project, University of Rochester, 1981.
- [Ski90] Steven Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Addison-Wesley, Reading, MA, 1990.
- [SMG99] W. Schwalm, B. Moritz, and M. Giona. Vector difference calculus for physical lattice models. *Physical Review E*, 59(1), January 1999.
- [Str88] Gilbert Strang. A framework for equilibrium equations. *SIAM Review*, (30):283–297, 1988.
- [Ton75] Enzo Tonti. On the formal structure of physical theories. Technical report, Istituto di Matematica del Politecnico di Milano, 1975.

- [Ton01] Enzo Tonti. A direct discrete formulation of field laws: The cell method. *Computer Modeling in Engineering & Sciences*, 2(2):237–258, 2001.
- [Wil82] T. J. Willmore. *Total Curvature in Riemannian Geometry*. Ellis Horwood, 1982.
- [Woo85] TC Woo. Combinatorial analysis of boundary data structure schemata. *IEEE Computer Graphics and Applications*, 5(3):19–27, 1985.
- [YK95] Y. Yamaguchi and F. Kimura. Nonmanifold topology based on coupling entities. *Computer Graphics and Applications, IEEE*, 15(1):42–50, 1995.