Technical note

Linear algebraic representation for topological structures

Antonio DiCarlo [a], Alberto Paoluzzi [a,*], Vadim Shapiro [b]

[a] *Università Roma Tre, Italy*
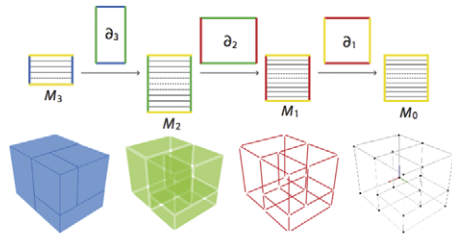[b] *University of Wisconsin, Madison, USA*

GRAPHICAL ABSTRACT

A **complex** C is a sequence $\cdots \longrightarrow C_{d+1} \xrightarrow{\partial_{d+1}} C_d \xrightarrow{\partial_d} C_{d-1} \longrightarrow \cdots$ of **linear** spaces C_d and **linear** boundary maps ∂_d, where $\partial_{d+1} \circ \partial_d = 0$, for all d

Chain and cochain complex

A **chain** complex C is a complex of **chain spaces** and **boundary maps**:

Characteristic matrices in CSR matrix form

Unit d-chains (single d-cell subsets), are the **standard bases** (M_d rows) of d-chain spaces d-cells as **subsets** of vertices   CSR form of a **binary** matrix

HIGHLIGHTS

- A proper mathematical model for all topological structures is a (co)chain complex.
- We propose a linear algebraic representation (LAR) scheme for representing such complexes.
- The LAR scheme is fully implemented using sparse matrices.
- The LAR scheme provides efficient support for topological queries and constructions.

ARTICLE INFO

ABSTRACT

With increased complexity of geometric data, topological models play an increasingly important role beyond boundary representations, assemblies, finite elements, image processing, and other traditional modeling applications. While many graph- and index-based data structures have been proposed, no standard representation has emerged as of now. Furthermore, such representations typically do not deal with representations of mappings and functions and do not scale to support parallel processing, open source, and client-based architectures. We advocate that a proper mathematical model for all topological structures is a (co)chain complex: a sequence of (co)chain spaces and (co)boundary mappings. This in turn implies all topological structures may be represented by a collection of sparse matrices. We propose a Linear Algebraic Representation (LAR) scheme for mod 2 (co)chain complexes using CSR matrices and show that it supports a variety of topological computations using standard matrix algebra, without any overhead in space or running time. A full open source implementation of LAR is available and is being used for a variety of applications.

* Corresponding author.
E-mail address: apaoluzzi@me.com (A. Paoluzzi).

1. Introduction

1.1. Goal: An efficient and unified representation of topology

Present-day computational problems in science and technology must deal with increasingly complex geometric information and applications. The complexity of geometric information stems from a dramatic increase in size, diversity, and complexity of geometric data: point clouds, boundary meshes, NURBs representations, finite element meshes, CT scans, and so on. The complexity of applications is apparent in increasingly complicated semantics, usually expressed in terms of incidences and relations involving geometric data: large-scale assemblies, topology of microstructure, image segmentation, multi-physics simulation, to name a few. Emerging applications require the convergence of shape synthesis and analysis from computer graphics, computer imaging, and computer-aided geometric design, with discrete meshing of domains used for physical simulations. The goals of unification, scalability, and massively parallel distributed computing call for rethinking of the foundations of geometric and topological computing.

The evolution of 3D geometric representations can be generally understood in terms of graph-based data structures representing one of several possible cell complexes partitioning either the boundary or the interior of the represented model. A variety of assumptions about cell complexes and graph representations make standardization difficult, complicate the issues of data exchange and transfer, and lead to the proliferation of incompatible algorithms. Graph-based structures are difficult to parallelize: e.g. boundary representation algorithms are dominated by graph searching algorithms (boundary traversals) that tend to force serial processing. In short, the specialized data structures based on cell complexes, that have driven the evolutionary development of the field, are no longer adequate for dealing with the emerging challenges and opportunities in geometric computing.

We observe that all types of cell complexes and functions over cell complexes are properly represented by a (co)chain complex, that captures all combinatorial relationships of interest in solid and physical modeling formally and unambiguously. Based on classical results from algebraic topology techniques, we show that a (co)chain complex and all associated combinatorial operations are readily represented using standard techniques from linear algebra, giving rise to a Linear Algebraic Representation (LAR) scheme. In this paper, we focus specifically on mod 2 (co)chain complexes because they are sufficient for many solid modeling applications [1]. In particular, we describe a fully implemented LAR data structure and algorithms using compressed sparse row (CSR) matrices that introduce no computational overhead and are asymptotically as efficient as (and usually better than) many other popular topological data structures. Our presentation will focus on chain complexes and operations. The extension to cochain complexes is straightforward because the two are isomorphic. For example, it is well known that boundary and coboundary matrices are transposes of each other.

1.2. Related work and state of the art

The literature on *representation schemes* in solid modeling started with the foundational paper [1], where a mathematical framework for the important aspects of *computer representations of solids* was introduced. The ground-breaking paper [2] had previously supplied the first but already efficient *boundary representation* scheme. It also introduced *Euler operators*, elementary operations for step-wise building well-formed polyhedra, using atomic software actions satisfying the Euler–Poincaré formula at each stage.

The *Quad-Edge* data structure, providing efficient primitives for planar subdivisions and Voronoi diagrams, was proposed in [3], and is still largely used in computational geometry algorithms and in geometric libraries. Variations of the *radial-edge* non-manifold representation by [4] have been embodied in almost every commercial CAD system that uses a non-manifold boundary representation. The *Cell–Tuple* structure [5] was introduced as a simple, uniform representation of finite and *regular* CW-complexes over subdivided *d*-manifolds.

More general set-theoretic and topological operators were provided by [6] to represent inhomogeneous objects, building upon the *Selective Geometric Complex* (SGC), a superset of *CW-complexes* allowing for cells non homeomorphic to open balls, proposed to handle dimension-independent models of point sets with internal structures and incomplete boundaries. A dimension-independent generalization of simplicial schemes, and various operators and algorithms were discussed by [7]. The foundational ideas underlying these and other representations are reviewed in [8], pointing out that solid modeling was conceived as a *universal technology* for developing engineering languages and systems with guaranteed geometric validity.

Despite the tremendous amount of research done and the progress made, most modern industrial systems still follow the basic approach established twenty years ago, centered on boundary representations, graphs, and non-manifold data structures represented using complex and inefficient (usually redundant) pointer structures. Computations in such systems, without preprocessing, usually rely on sequential "traversal" (graph or boundary) algorithms that depend on specific pointer structures, and do not readily scale. For example, handling large 3D unstructured meshes becomes problematic, but the situation is much more challenging for other higher-dimensional topological structures.

More recently, the geometric and physical modeling fields started moving towards computing with functions defined over more general cellular decompositions [9,10] and direct integration between geometry and physics [11–14]. These latest developments provided the motivation for the present paper; in particular, we will draw heavily on the ideas, concepts, and definitions in [12] and [13] to propose a new linear-algebraic approach to representing and computing with topological structures.

2. LAR scheme

2.1. Summary

In this section we introduce the *Linear Algebraic Representation* (LAR) scheme. The aim is to provide a representation that supports all topological constructions and queries that arise in typical cellular decomposition of space (mesh, image, boundary, etc.). Formally, LAR relies on standard definitions [15,16]: in the mod 2 cellular complexes used here, *d*-chains are *sets* of *d*-cells; the standard basis of the $\mathbb{Z}_2$-linear space C_d of *d*-chains is provided by *singletons* of *d*-cells; each *d*-cell is represented by a map $C_d \rightarrow \mathbb{Z}_2 C_0$, i.e. by a *row* of a binary characteristic matrix M_d. Of course, every *d*-chain in C_d may be generated by a ($\mathbb{Z}_2$)-linear combination of M_d rows. The formulation may be extended to *d*-cochains that represent any possible field over the chains, and boundary and coboundary operators provide the computational tools needed by the discrete version of the generalized Stokes theorem for integration of fields over *d*-dimensional domains [13]. Due to space limitations, in this paper we will focus on chain complexes only, but the extension to cochains is straightforward [13].

Below, we show that the characteristic matrices of a chain complex, i.e. the binary matrices that encode the incidence of *d*-cells with 0-cells, provide a convenient tool for computing boundary (and coboundary) operators and answering queries concerning

the topological relations between cells. Characteristic matrices are *very* sparse for actual chain complexes used in applications, and can be represented in a standard CSR (Compressed Sparse Row) format. The product and transposition of such CSR matrices, needed to compute the boundary, adjacency and incidence operators between such linear spaces, are intrinsically efficient, since the sparse matrix–vector (SpMV) multiplication is linear in the size of the output.

2.2. Representation scheme

Let a finite cellular complex $\Lambda(X)$ be given, such that $X = \Lambda_0 \cup \cdots \cup \Lambda_d$. Let also $M_p \in \mathbb{Z}_2^{m \times n}$ ($0 \le p \le d$) be binary characteristic matrices, with number of rows equal to the number of p-cells, and number of columns equal to the number of 0-cells:

$$m = k_p = \sharp \Lambda_p, \qquad n = k_0 = \sharp \Lambda_0.$$

Each $m_{ij} \in M_p$ tells us whether (the vertex) $\nu_j \in \Lambda_0$ is contained on the boundary of the cell $\lambda_i \in \Lambda_p$ or not.

The *LAR scheme* is a map from mathematical models of solids to their computer representations:

$$\text{LAR} : \mathbb{C}\text{h}(\Lambda(X)) \to (M_p)_{p=1}^d.$$

In other words, LAR represents *chain complexes* $\mathbb{C}\text{h}$, supported by a finite cellular complex $\Lambda(X)$, by d-tuples of binary *compressed sparse row* (CSR)[1] matrices, where d is the dimension of the space X. By definition, a LAR representation $(M_p)_{p=1}^d$ is *valid* if and only if it represents a valid chain complex, i.e. its boundary operators form an exact short sequence

$$[\partial_{p-1}][\partial_p]\mathbf{1}_{k_p} = \mathbf{0}_{k_{p-2}} \pmod 2, \quad 1 \le p \le d \tag{1}$$

where $k_h = \sharp \Lambda_h$, and the matrices $[\partial_h]$ ($h = p, p-1$) are boundary operators. We give a detailed algorithm for constructing boundary operators in Section 3.3.

Observe that, for a given cellular d-complex $\Lambda(X)$, all the M_p matrices ($1 \le p \le d$) have the same number of columns, and can be operated by standard matrix transposition and multiplication. This fact provides a convenient tool for computing boundary and coboundary operators and topological relations between cells. In particular, in Section 3.3, we show that a boundary operator may be constructed by filtering the matrices $M_{h-1} M_h^t$.

Reduced LAR. We now consider a special but common situation where a topological structure is *regular d-complex*, where every cell is contained in some d-cell. In this case, a LAR representation can be simplified further under mild assumptions:

– LAR contains both the d-cells of a regular decomposition of a d-space and of its complement;
– any two cells intersect on a connected cell.

In this case, the highest dimensional CSR(M_d) matrix is a valid Reduced LAR representation in the sense that all lower-dimensional M_p matrices ($p \le d$) and (co)boundary operators may be computed from the matrix M_d. Therefore, CSR(M_d) fully characterizes the chain complex induced by $\Lambda(X)$, and produces a *valid* LAR. For a given decomposition, no ambiguity may arise. The algorithm to compute the matrix M_{p-1} from M_p ($1 \le p \le d$) is given in Section 3.2. The only requirement for the validity of M_d is the usual one for cell-complexes, i.e. that the intersection of any two d-cells (considered as sets of 0-cells) is a *single connected cell*.

Compressed LAR. If regular d-complexes are additionally characterized from d-cells with the same number of vertices, a specially compressed version of the CSR representation may be used, since

every row of M_d contains the same number of non-zero elements, and therefore it is not necessary to store the index to the beginning of the storage of the column indices of each row. It is worth noting that the above holds for the important and common case of simplicial and cuboidal d-complexes, where the number of ones per row of the characteristic matrix is equal to $d+1$ and to 2^d, respectively. Of course, we do not need to store the values of non-zero elements in M_d, since it is binary.

Compactness of representation. Notice that a reduced and compressed LAR is highly compact. For example, the boundary rep of a triangulated 3-manifold using LAR needs only space $|FV| = 2|E|$ (compare [19]), where $|E|$ is the number of boundary edges. Recall that the minimal *complete* (in the sense of Requicha [1]) BRep of 3-manifolds needs space $|EF| + |EV| = 4|E|$. Notice also that the reduced LAR exactly coincides with the bulk of common representations in Computer Graphics—for instance, the OBJ and PLY file formats.

3. Mappings and algorithms

In this section, we show that common topological operations and queries on CSR represented LAR structures, including incidence, boundary, star, and product of spaces, reduce to a sequence of SpMV operations that are asymptotically linear and, with a few exceptions, output sensitive.

3.1. Incidence/adjacencies

For simplicity and concreteness, we first consider the incidence queries that typically arise in a cellular decomposition $\Lambda(X)$ of a 2D space, such as 2D triangulation or the boundary of a solid 3-shape. There are 9 incidence relations between pairs of cells in such a V, E, F decomposition, and traditional graph-based representations are chosen by optimizing the trade-off between the space requirements and efficiency of queries [19]. With LAR, *all* such queries are sparse matrix–vector multiplications without any additional space requirements. The chain *operators* corresponding to the incidence relations $VV \subset V \times V$, $VE \subset V \times E$, and $VF \subset V \times F$ are given below:

$$\mathcal{V}\mathcal{V} : C_0 \to C_0, \qquad \mathcal{E}\mathcal{V} : C_0 \to C_1, \qquad \mathcal{F}\mathcal{V} : C_0 \to C_2;$$
$$\mathcal{V}\mathcal{E} : C_1 \to C_0, \qquad \mathcal{E}\mathcal{E} : C_1 \to C_1, \qquad \mathcal{F}\mathcal{E} : C_1 \to C_2;$$
$$\mathcal{V}\mathcal{F} : C_2 \to C_0, \qquad \mathcal{E}\mathcal{F} : C_2 \to C_1, \qquad \mathcal{F}\mathcal{F} : C_2 \to C_2.$$

The corresponding CSR matrices are readily computed:

$$\mathcal{V}\mathcal{V} = \mathcal{V}\mathcal{E} \circ \mathcal{E}\mathcal{V} = \mathcal{E}\mathcal{V}^\top \circ \mathcal{E}\mathcal{V} \Rightarrow [\mathcal{V}\mathcal{V}] = M_1^t M_1$$
$$\mathcal{V}\mathcal{E} = \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{V}\mathcal{E}] = M_1^t$$
$$\mathcal{V}\mathcal{F} = \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{V}\mathcal{F}] = M_2^t$$
$$\mathcal{E}\mathcal{V} \quad [\mathcal{E}\mathcal{V}] = M_1$$
$$\mathcal{E}\mathcal{E} = \mathcal{E}\mathcal{V} \circ \mathcal{V}\mathcal{E} = \mathcal{E}\mathcal{V} \circ \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{E}\mathcal{E}] = M_1 M_1^t$$
$$\mathcal{E}\mathcal{F} = \mathcal{E}\mathcal{V} \circ \mathcal{V}\mathcal{F} = \mathcal{E}\mathcal{V} \circ \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{E}\mathcal{F}] = M_1 M_2^t$$
$$\mathcal{F}\mathcal{V} \quad [\mathcal{F}\mathcal{V}] = M_2$$
$$\mathcal{F}\mathcal{E} = \mathcal{F}\mathcal{V} \circ \mathcal{V}\mathcal{E} = \mathcal{F}\mathcal{V} \circ \mathcal{E}\mathcal{V}^\top \Rightarrow [\mathcal{F}\mathcal{E}] = M_2 M_1^t$$
$$\mathcal{F}\mathcal{F} = \mathcal{F}\mathcal{V} \circ \mathcal{V}\mathcal{F} = \mathcal{F}\mathcal{V} \circ \mathcal{F}\mathcal{V}^\top \Rightarrow [\mathcal{F}\mathcal{F}] = M_2 M_2^t.$$

3.2. Reduced LAR: computing $(d - 1)$-chains from d-chains

It should be apparent that the above formulation and computation of incidence relations generalizes to all valid chain complexes. In this section, we rely on the incidence queries to show how to compute the matrix M_{d-1} starting from M_d in a valid reduced LAR

---

[1] Compressed Sparse Row (CSR) format, for which efficient implementations on high-performance hardware exist. See [17,18].
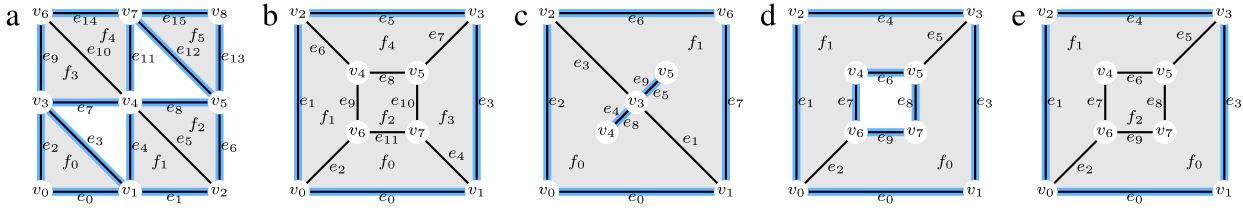
Fig. 1. Cellular 2-complexes and their boundaries (blue): (a) non-manifold simplicial complex; (b) cuboidal complex; (c) complex with internal boundaries; (d, e) complexes with non-convex 2-cells.

of a space. In other words, we give a map $\phi : M_d \mapsto M_{d-1}$. Recall that reduced LAR starts with a cell partition of the whole space, i.e. $\Lambda_d = \Lambda_d(X) \cup \Lambda_d(Y \setminus X)$ where Y is the embedding space of X.

The adjacency relation between d-cells is the subset of cell pairs that share some vertex (0-cell). The adjacency matrix A_d can be computed by matrix multiplication between the characteristic matrix $M_d(\Lambda_d)$ and its transpose:

$$A_d = M_d M_d^t$$

$A_d = (a_{i,j})$ is integer-valued and symmetric. Each $a_{i,j}$ gives the *number* of 0-cells shared between the cells $\lambda_i, \lambda_j \in \Lambda_d$. When the set intersection of two cells $\lambda_i, \lambda_j \in \Lambda_d$ contains a number of vertices greater than or equal to the dimension d of cells, they share a common facet $\mu \in \Lambda_{d-1}$, that can be computed as a component-wise integer product (or binary intersection) of rows i and j of M_d. The generated characteristic vector gives a row of the matrix M_{d-1}. In symbols:

$$a_{i,j} = \sharp(\lambda_d^i \cap \lambda_d^j)$$

$$a_{i,j} \geq d \quad (i \neq j) \Rightarrow \exists\, M_d(i) \wedge M_d(j) = \mu \in \Lambda_{d-1}.$$

Computations with CSR matrices are efficient because zeros are not stored, and each row of CSR(A_d) actually stores only the numbers of vertices shared by a d-cell with the incident ones. The component-wise product of two binary CSR rows i and j of M_d takes at most $O(\sharp\lambda_i + \sharp\lambda_j)$ time.

Examples. A detailed example of computation of the M_1 characteristic matrix starting from M_2 is given in Section 4.2 for the regular polytopal complex shown in Fig. 1(b). For instance, the edge $e_2 = (v_0, v_6)$ is generated from the term $a_{0,1} = 2$ of the cell adjacency matrix $A_2 = M_2 M_2^t$ given in Eq. (3). The edge e_2 is computed by the entrywise product (intersection) of rows $M_2(0)$ and $M_2(1)$, related to the 2-cells f_0 and f_1 of Fig. 1(b):

$$(1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1) \cdot (1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0)$$
$$= (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0).$$

3.3. Boundary/coboundary matrix

Consider the *incidence map* $\mathcal{I}_{p-1}^p : C_p \to C_{p-1}$ and its matrix

$$[\mathcal{I}_{p-1}^p] := M_{p-1}^p = M_{p-1} M_p^t.$$

The entry $M_{p-1}^p(i,j)$ stores the number k of common vertices (the cardinality of the intersection) between the cells μ_{p-1}^i and λ_p^j. In fact, we get

$$M_{p-1}^p(i,j) = \sum_{h=0}^{k_0-1} M_{p-1}(i,h) \cdot M_p(j,h) = \sharp(\mu_{p-1}^i \cap \lambda_p^j) = k,$$

thus computing the *number k of vertices of the intersection*, i.e. of the common face, between $\mu_{p-1}^i \subset \Lambda_0$ and $\lambda_p^j \subset \Lambda_0$. This common face coincides with μ_{p-1}^i if and only if

$$\sharp(\mu_{p-1}^i \cap \lambda_p^j) = \sharp\mu_{p-1}^i.$$

In this case, we have

$$\mu_{p-1}^i \wedge \lambda_p^j = \mu_{p-1}^i \in \partial\lambda_p^j \quad \text{and} \quad \lambda_p^j \in \delta\mu_{p-1}^i,$$

with $[\delta_{p-1}] = [\partial_p]^t$ the $(p-1)$-coboundary map. Therefore, as a computational procedure to calculate the matrix of the *unoriented boundary* operator, we have the algorithm:

(i) Compute $\text{CSR}(M_{p-1}^p) := \text{CSR}(M_{p-1})\,\text{CSR}(M_p)^t$ as the product of sparse matrices.
(ii) Filtering procedure: for each $0 \leq i \leq k_{p-1} - 1$,
    (a) compute the number $k := \sharp\mu_{p-1}^i$ of non-zero elements stored in row i of $\text{CSR}(M_{p-1})$;
    (b) for each $0 \leq j \leq k_p - 1$:

$$[\partial_p](i,j) := 1 \quad \text{if } M_{p-1}^p(i,j) = k \quad \text{else} \quad [\partial_p](i,j) := 0.$$

By duality, the same procedure may be used to compute the $(p-1)$-coboundary operator δ_{p-1} as the transpose of the boundary ∂_p.

Example. Here we compute the $[\partial_2]$ boundary matrix for the regular polytopal complex shown in Fig. 1(b), from characteristic matrices M_2 and M_1 given in Section 4.2. According to step (i) of the algorithm above, we get

$$M_1^2 = M_1 M_2^t.$$

Then, we filter the matrix M_1^2 according to the criterion given in step (ii), producing the matrix of $\partial_2 : C_2 \to C_1$.

$$[\partial_2]^t = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (2)$$

The rows of $[\partial_2]$ are the (characteristic vectors of) generators of the C_1 linear space, i.e. the edges of the complex, indexed by faces; the columns give the (characteristic vectors of) generators of the C_2 linear space, i.e. the faces indexed by edges, as it can be verified in Fig. 1(b).

3.4. Boundary queries

The knowledge of a d-boundary matrix is very useful, since it encodes the boundaries of *all possible chains*, i.e. of *all possible subsets* of d-cells. It is also computationally efficient, since the calculation of $[\mu] = [\partial_d][\lambda]$, where $[\mu], [\lambda]$ are the characteristic vectors of chains $\mu \in C_{d-1}$ and $\lambda \in C_d$, is actually performed using only the non-zero terms stored in their CSR matrices. In particular, the boundary of the partition $\Lambda_d(X) \cup \Lambda_d(\complement X)$, of a space and its complement, is performed by setting the characteristic vector $[\Lambda_d]$ with $\sharp\Lambda_d(X)$ ones and $\sharp\Lambda_d(\complement X)$ zeros.

Example (*Boundary of the Total Chain*)**.** The regular complex of Fig. 1(b) is represented as a 2-chain over the shown cellular decomposition:

$$[\Lambda_2] = [1, 1, 1, 1, 1, 0]^t$$

and therefore we get for the boundary 1-chain:

$$[\partial_2][\Lambda_2] = [1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0]^t \quad (\text{mod } 2)$$

which corresponds to the boundary 1-chain shown in Fig. 1(b) as a set of 1-cells (edges), $\{e_0, e_1, e_3, e_5\}$.

Example (*Boundary of Subset Chain*)**.** If we remove from $\Lambda_2(X)$ the central cell f_4, and insert it in the exterior space partition $\Lambda_2(\complement X)$, we set

$$[\Lambda_2] = [\Lambda_2(X) \cup \Lambda_2(\complement X)] = [1, 1, 1, 1, 0, 0]^t, \text{ and get}$$

$$[\partial_2][\Lambda_2] = [1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1]^t \quad (\text{mod } 2)$$

giving the boundary chain $\{e_0, e_1, e_3, e_5, e_8, e_9, e_{10}, e_{11}\}$.

3.5. Product of cellular complexes

Let $A = \Lambda(X)$ and $B = \Lambda(Y)$ be cell complexes of dimension m and n, respectively. Then, $C = A \times B = \Lambda(X \times Y)$ is a cell complex of dimension $m+n$, with the product $\lambda^h \times \lambda^k$ as cells, with $\lambda^h \in \Lambda_m(X)$ and $\lambda^k \in \Lambda_n(Y)$. Assuming $X \subset \mathbb{E}^c$ and $Y \subset \mathbb{E}^d$, it is $X \times Y \subset \mathbb{E}^{c+d}$. Common examples of products in geometric and physical modeling are extrusions, sweeps, and time stepping (e.g. in morphing or animation). If $V(X)$ and $V(Y)$ are position vectors,

$$V(X \times Y) = V(X) \otimes V(Y)^t.$$

The LAR implementation of product operation is trivial, amounting to a double loop on cells and vertices.

3.6. Star operator

The star of a cell λ in a cellular complex $\Lambda(X)$, denoted St λ, is the set of all cells in Λ that have at least a face in λ. Informally, the star represents a combinatorial neighborhood of λ. In algebraic terms, it is a linear endomorphism of $\mathcal{C} = \oplus_p C_p$ such that St : $\mathcal{C} \to \mathcal{C}$. Computation of a star is straightforward in the LAR scheme, by treating λ as an elementary chain $\{\lambda\} \in C_p$. We simply pre-multiply all matrices $M_{p,q} := M_p M_q^t$ by the CSR of characteristic vector $[\lambda]^t$, and take the union of the resulting sets of cells.

An application of the star operator arises in allocation of large models across a distributed environment. To execute any distributed simulation, the (image of the) star of each submodel (as a chain) must be allocated on each pair of processes sharing a portion of the submodel boundary. Stars are also used for enforcing continuity of mappings between the complexes, as well as in physics and in mesh processing (e.g., Laplace operators operate on open neighborhoods, and can be easily generated by this operator).

4. LAR examples

Some elementary examples aim to show the simple computations in the LAR scheme. A non-trivial extraction of complex topological models from 3D image data follows. FV and EV give a CSR rep of the binary matrices M_2 and M_1.

4.1. Non-manifold simplicial complex

Consider the $\Lambda(X)$ decomposition in Fig. 1(a), given by the simplicial complex $\Lambda = \Lambda_0 \cup \Lambda_1 \cup \Lambda_2$, with $k_0, k_1, k_2 = 9, 16, 6$, that is locally *non-manifold*.

```
FV = [[0,1,3],[1,2,4],[2,4,5],[3,4,6],[4,6,7],[5,7,8]]
```

$FV = CSR(M_2)$ encodes the linear space $C_2(\Lambda)$ of 2-chains through a standard basis (isomorphic to 2-cells). These are combinations of the standard basis of 0-chains (isomorphic to 0-cells).

The linear space $C_1(\Lambda)$ of 1-chains is encoded by $EV = CSR(M_1)$. Note that FV is the only input for a simplicial complex: EV may be computed in time linear with the output size, by using either a well-known combinatorial formula [7], or the method given in Section 3.2.

4.2. Convex-cell complex

The cellular complex shown in Fig. 1(b) is composed of five quads, described symbolically by quadruples of vertex indices (FV), and is embedded in $\mathbb{E}^2$ by assigning a pair of coordinates to position vectors of vertices (V).

```
V = [[0.,0.],[1.,0.],[2.,0.],[0.,1.],[1.,1.],[2.,1.],
     [0.,2.],[1.,2.],[2.,2.]]
FV = [[0,1,6,7],[0,2,4,6],[4,5,6,7],[1,3,5,7],[2,3,4,5]]
```

To construct a reduced LAR representation, we must append to FV the description of one or more external 2-cells, in this case denoted by the quadruple [0, 1, 2, 3]. The resulting representation $FV := CSR(M_2)$ becomes:

```
FV = [[0,1,6,7],[0,2,4,6],[4,5,6,7],[1,3,5,7],[2,3,4,5],
      [0,1,2,3]]
```

Using the technique introduced in Section 3.2, we first convert the input FV into the (binary) characteristic matrix M_2 such that $CSR(M_2) = FV$, and into the adjacency matrix A_2:

$$M_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A_2 = M_2 M_2^t = \begin{pmatrix} 4 & 2 & 2 & 2 & 0 & 2 \\ 2 & 4 & 2 & 0 & 2 & 2 \\ 2 & 2 & 4 & 2 & 2 & 0 \\ 2 & 0 & 2 & 4 & 2 & 2 \\ 0 & 2 & 2 & 2 & 4 & 2 \\ 2 & 2 & 0 & 2 & 2 & 4 \end{pmatrix}$$

(3)

and for each item $a_{ij} \in A_2$ $(i < j)$ such that $a_{ij} \geq d = 2$, compute the binary intersection between the rows i and j of M_2, and put the binary result as a row of the characteristic matrix of the 1-cells, named M_1. Therefore we get:

$$M_1^t = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

and

$$EV = CSR(M_1).$$

```
EV = [[0,1],[0,2],[0,6],[1,3],[1,7],[2,3],[2,4],[3,5],
      [4,5],[4,6],[5,7],[6,7]]
```

4.3. Practical application: models from 3D images

In addition to the existing implementation in python, LAR is being developed as a web service in a web-based computational environment, using some Khronos's APIs (OpenCL server-side and WebCL client-side) for industry standard heterogeneous computing. In particular, LAR is now being used in biomedical
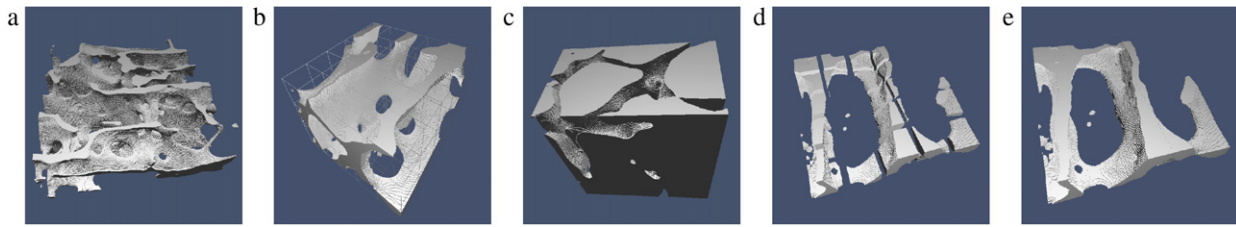
Fig. 2. Models extracted from 3D images of spongy bone: (a, b) from a portion of a $500 \times 500 \times 150$ pixel; (c) a portion of exterior space; (d) stream of chunks generated by work-units using a 25 MB $[\partial_3]$ matrix in constant memory in the OpenCL implementation; (e) partial model.

applications which require fast performances with big geometric data for topological tasks such as model extraction from 3D images.[2] For instance, density values in medical images represent scalar fields (cochains) over cubical cellular complexes, and LAR is used to guarantee topologically correct 3D image segmentation as well as to extract a/an (enumerative) solid model from the image, which is subsequently smoothed out (see Fig. 2).

5. Conclusions

In this paper we introduced LAR, a simple linear algebraic representation for cellular complexes, using a CSR (Compressed Sparse Row) form for characteristic matrices of linear spaces of (co)chains. LAR allows us to efficiently compute and query any model topology through sparse matrix algebra, and supports all topological incidence structures, including enumerative (images), decompositive (meshes) and boundary (CAD) representations. LAR is dimension-independent, not restricted to regular complexes, and allows for internal/external structures, including cracks and fibers of lower dimensions. Also, LAR enjoys a neat mathematical format—being based on *chains*, the domains of discrete integration, and *cochains*, the discrete prototype of differential forms, so naturally integrating the geometric shape with the supported physical properties. Therefore, LAR appeared to offer a sound basis for a standardization of topological structures and computations.[2] Other promising research directions point towards LAR-based multicore/heterogeneous computing and web computations with distributed models and parallel algorithms.

Acknowledgments

References

[1] Requicha AG. Representations for rigid solids: theory, methods, and systems. ACM Computing Surveys 1980;12(4):437–64.
[2] Baumgart BG. Winged edge polyhedron representation. Tech Rep Stan-CS-320, CA, USA: Stanford; 1972.
[3] Guibas L, Stolfi J. Primitives for the manipulation of general subdivisions and the computation of Voronoi. ACM Transactions on Graphics 1985;4(2):74–123.
[4] Weiler KJ. The radial edge structure: a topological representation for non-manifold geometric modelling. In: Geometric modelling for CAD applications. North-Holland; 1988. p. 3–12.
[5] Brisson E. Representing geometric structures in *d* dimensions: topology and order. In: 5th Annual symposium on computational geometry, SCG'89. ACM; 1989. p. 218–27.
[6] Rossignac JR, O'Connor MA. SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In: Geometric modeling for product engineering. North-Holland; 1990.
[7] Paoluzzi A, Bernardini F, Cattani C, Ferrucci V. Dimension-independent modeling with simplicial complexes. ACM Transactions on Graphics 1993; 12(1):56–102.
[8] Shapiro V. Solid modeling. In: Farin G, Hoschek J, Kim S, editors. Handbook of computer aided geometric design (chapter 20). Elsevier Science; 2002. p. 473–518.
[9] Feng X, Wang Y, Weng Y, Tong Y. Compact combinatorial maps in 3D. Graphical Models 2012;75(3):149–56.
[10] Wang K, Li X, Li B, Xu H, Qin H. Restricted trivariate polycube splines for volumetric data modeling. IEEE Transactions on Visualization and Computer Graphics 2012;18(5):703–16.
[11] Desbrun M, Hirani AN, Leok M, Jerrold, Marsden E. Discrete exterior calculus. Tech Rep 2003.
[12] DiCarlo A, Milicchio F, Paoluzzi A, Shapiro V. Solid and physical modeling with chain complexes. In: Proceedings of the 2007 ACM symposium on solid and physical modeling, SPM '07. New York, NY, USA: ACM; 2007. p. 73–84.
[13] DiCarlo A, Milicchio F, Paoluzzi A, Shapiro V. Chain-based representations for solid and physical modeling. IEEE Transactions on Automation Science and Engineering 2009;6(3):454–67.
[14] Cottrell J, Hughes T, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA. Wiley; 2009.
[15] Hatcher A. Algebraic topology. Cambridge Univ. Press; 2002.
[16] Hausmann J-C. Mod two homology and cohomology, Book project, 2012. http://www.unige.ch/math/folks/hausmann/.
[17] Buluç A, Gilbert JR. Parallel sparse matrix–matrix multiplication and indexing: implementation and experiments. SIAM Journal of Scientific Computing (SISC) 2012;34(4):170–91.
[18] Lokhmotov A. Implementing sparse matrix–vector product in OpenCL, In: OpenCL tutorial, HiPEAC'11, 2012.
[19] Woo T. A combinatorial analysis of boundary data structure schemata. IEEE Computer Graphics & Applications 1985;5(3):19–27.

[2] This work is carried out within the framework of the IEEE-SA Project P3333.2—Standard for "Three-Dimensional Model Creation Using Unprocessed 3D Medical Data".