Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference
Chicago, Illinois USA, September 2-6, 2003

DETC2003/DTM-48654

COMBINATORIAL LAWS FOR PHYSICALLY MEANINGFUL DESIGN

Vasu Ramaswamy
Spatial Automation Laboratory
University of Wisconsin-Madison
Email: ramaswam@sal-cnc.me.wisc.edu

Vadim Shapiro*
Spatial Automation Laboratory
University of Wisconsin-Madison
Email: vshapiro@engr.wisc.edu

ABSTRACT

A typical computer representation of a design includes geometric and physical information organized in a suitable *combinatorial* data structure. Queries and transformations of these design representations are used to formulate most algorithms in computational design, including analysis, optimization, evolution, generation, and synthesis. Formal properties, and in particular existence and validity of the computed solutions, must be assured and preserved by all such algorithms.

Using tools from algebraic topology, we show that a small set of the usual combinatorial operators: boundary (∂), coboundary (δ), and dualization (*) – are sufficient to represent a variety of physical laws and invariants. Specific examples include geometric integrity, balance and equilibrium, and surface smoothing. Our findings point a way toward systematic development of data structures and algorithms for design in a common formal computational framework.

1 Combinatorial representations in design

Given a rich variety of representations and algorithms for computational design, there appears to be little reason to believe that most of them may be formulated within a single combinatorial framework. We show that not only is such a framework possible [15], but it is formal [3] because it is based on the tools from algebraic topology [7], practical [5,14] because it arises naturally from computational considerations, and ubiquitous [25] because it spans most of the geometric and physical laws.

*Please direct all correspondence to this author at Mechanical Engineering, 1513 University Avenue, Madison, WI 53706 USA

1.1 Design as an oriented cell complex

Computational design of shapes in most cases involves working with both geometry as well as physics. It is common that shaping and analysis of artifacts rely on two separate representations. For example, in the evolutionary design of tables in [8] the assumed geometric primitives are voxels, but evaluating the structural integrity of the table requires that the voxel model be converted into another representation suitable for performing analysis. Furthermore, many different design procedures appear to rely on a variety of seemingly incompatible representations and techniques. Thus, shape-grammar rules may be implemented as transformations of truss structures [17], while subdivision and fairing rules are commonly applied to curve and surface representations [24] – both are quite distinct from the space-aligned three-dimensional voxel-based representation of tables.

It is well known that these and most other representations of designs are simple instances of an *oriented cell complex* – a standard combinatorial structure widely used in topology and more recently in geometric modeling. Perhaps less obvious is that most geometric, physical, and design quantities can be formally defined as algebraic objects called *chains* and *cochains* - intuitively vectors whose components are associated with individual cells. The immediate consequence of this fact is that most spatial, physical and design laws may be expressed algebraically and applied uniformly across a variety of design representations, tasks, and algorithms. Furthermore, most such laws are compositions of only a few primitive laws that are generic in nature. The purpose of this paper is to elucidate this algebraic topological view of computational design and to demonstrate its ubiquity. We substantially extend the earlier ideas proposed in [15] while drawing

heavily on efforts of others who have used algebraic topological tools for explaining analogies and classifying physical theories [3, 25] and improving numerical solution techniques [13]. Much of the present research in the network formulation of many physical systems is based on the earlier work by Kron and Roth. Kron modelled various physical systems including elastic and plastic structures as electric circuits and solved them piecewise by the method of tearing [11]. Roth related the various quantities in the physical systems as homology and cochomology sequences using his now well-known Roth diagrams [20]. Branin extended the algebraic-topological formulation of networks by linking it with standard operations in vector calculus [3]. Tonti classified most physical theories into a framework of topological and constitutive relationships using his own diagrams [25]. Acceptance of the proposed computational framework should streamline and simplify development of computational procedures in terms of standard computational primitives, lead to more efficient search algorithms that generate fewer invalid candidate designs, and encourage cross-fertilization of ideas and techniques across a wide range of application domains.

Combinatorially, cell complexes are compositions of oriented k-cells with k = 0(vertex), 1(edge), 2(surface), 3(volume), typically embedded in 3-D space without intersections and subject to the additional constraint that the closure of every cell is a union of other cells in the complex. For the purposes of this paper, we will draw on three substantially distinct but fairly common examples used in computational design, as illustrated in Figure 1. Voxel-based representation, such as used in evolutionary design of tables [8], are essentially 3-dimensional cubical cell complexes (Figure 1a), where 3-cells correspond to individual voxels, and lower-dimensional cells implicitly correspond to adjacency relationships (2-cells for face adjacency, 1-cells for edge adjacency, and 0-cells for corner adjacency). Truss and frame structures that are commonly used in shape-grammar based design [17] are 1-dimensional complexes composed of 0-cells (joints) and 1-cells (members) (Figure 1c). Boundary representations of solid models [18] and subdivision surfaces [24] (Figure 1b) are 2-dimensional geometric cell complexes with faces, edges, and vertices corresponding to 2-cells, 1-cells, and 0-cells respectively.

More generally, different types of cell complexes arise frequently in computational science and engineering. Finite elements, finite differences, and finite volume models in numerical analysis are defined on collections of k-cells forming a mesh or a grid [13]. Many graphical computations are performed on polygonal meshes, while manifolds are usually represented by underlying triangulations. In each case, the cellular decompositions of shapes and spaces are supported by graph-based data structures and algorithms to perform all relevant computations cell by cell. Furthermore, all spatial properties and functions of the represented spaces may now be treated algebraically as vectors over the decomposed spaces.

1.2 Properties as chains and cochains

The simplest operation that is performed on a cell complex is the identification and logical selection of certain cells for specific computational purposes. For example, the convex hull area computation in [8] requires isolating the corners (0-cells) and edges (1-cells) of faces of voxels at a certain height, while application of boundary conditions (tractions and displacements) requires isolation of 0-cells in a truss. Application of a smoothing procedure on a certain portion of a surface requires isolation of a subset of faces from the whole domain. In each case, a group of p-dimensional cells is selected among all cells of the complex for further processing. Such a group is represented algebraically as an elementary p-chain defined as a formal sum:

$$C_p = \sum_{i=1}^{n_p} f_i \cdot cell_p^i \tag{1}$$

where C_p is the p-chain, $cell_p^i$ denotes the ith cell of dimension p, n_p is the number of p-cells in the cell complex and $f_i \in \{-1, +1, 0\}$ is a coefficient that determines whether a particular oriented cell is selected with positive or negative orientation ($f_i = \pm 1$) or not selected at all ($f_i = 0$). The top row in Figure 2 illustrates the selection of three 2-cells (with coefficient $f_i = 1$ and shaded grey) on a surface as a chain C_2.

More general coefficients may be associated with the cells in the cell complex in order to capture other uses and properties of the cell complex algebraically. Thus, integers may correspond to how many times a particular p-cell is visited in a path, reals may represent a total mass associated with a cell, and vectors may be used to specify forces acting on certain cells, etc. The generalization of coefficients is accompanied by a subtle transition from elementary p-chains to *functions* on p-chains called *p-cochains.* A p-cochain can be defined as a formal sum:

$$C^p = \sum_{i=1}^{n_p} g_i \cdot cell_p^i \tag{2}$$

which is similar to Equation (1) with the difference that C^p is the p-cochain, and g_i denote the generic coefficients that captures specific properties of the associated cell, $cell_p^i$. Cochains in our examples include geometric properties (for instance, geometric locations of 0-cells of trusses, voxel-based table designs, and subdivision surface patches). Instances of physical and non-physical attributes include forces and displacements on a truss structure or voxel-based models of tables, and weights associated with 1-cells of a subdivision surface in [24]. Figure 3 shows a cell complex composed of 0, 1 and 2-cells (subscripts of the alphabets denote the dimension of the cell) and coefficients associated with all 1- and 2-cells. The corresponding 1-cochain C^1 and a 2-cochain C^2 are: $C^1 = 5 \cdot a_1 - 10 \cdot b_1 + 23 \cdot c_1 + 16 \cdot d_1 + 16 \cdot e_1 + 28 \cdot f_1 + 29 \cdot g_1 + 53 \cdot h_1 + 13 \cdot i_1 + 3 \cdot j_1 + 19 \cdot k_1 + 14 \cdot l_1$ and $C^2 = 80 \cdot a_2 + 67 \cdot b_2 + 76 \cdot c_2 + 50 \cdot d_2$. The coefficients in C^1 and C^2 typically represent discrete distributions of physical

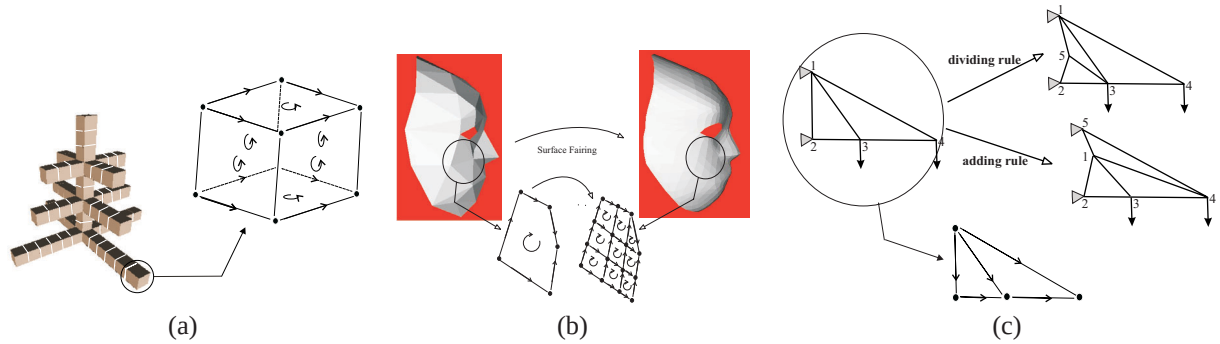

Figure 1. Common examples of combinatorial structures used in computational synthesis; each is an instance of an oriented cell complex.

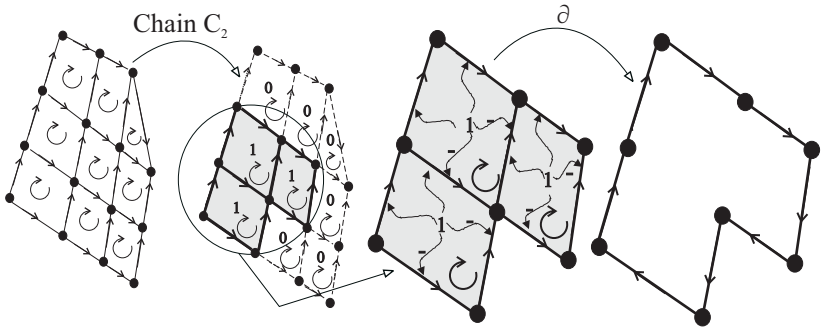

Figure 2. Dimensionally homogeneous portions of the cell-complex may be represented by chains. Here a 2-chain identifies three cells with non-zero coefficients; Boundary operation on this 2-chain produces an oriented 1-chain of edges bounding the selected area. Note the change in orientation of some 1-cells as ∂ reorients them based on their relative orientation with the 2-cells.

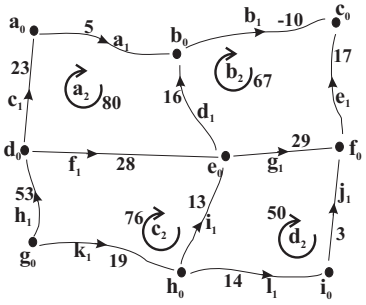

Figure 3. Cochains are collections of oriented cells along with their associated coefficients. These coefficients represent the integral quantities associated with cells. For example, vector-valued coefficients on 1-cells could denote relative displacements and scalar coefficients on 2-cells could represent their areas.

properties throughout the cell complex, such as displacements or areas respectively.

In topology, chains and cochains serve very different purposes, and it is customary to distinguish them as different types of objects. But, as explained by [7], every chain can be also viewed as a cochain (but not vice versa), and in fact p-cochains are usually defined in terms of (or evaluated on) the corresponding elementary p-chains. In engineering applications dealing with finite cellular structures, the distinction between the chains and cochains is not always critical, and the two concepts have sometimes been grouped under the name of "chains" [3, 5, 14, 15, 26]. We will also follow this practice, but will use the term *cochains* when referring to the most general class of objects that includes both chains and cochains. The main utility of cochains in computational applications lies in the fact they define a convenient class of objects for algorithmically constructing and simulating complex geometrical and physical models. Specifically, addition and multiplication operations are defined for p-cochains in a cell-by-cell fashion, very much like for the usual vectors. And in fact, all p-cochains defined over a cell complex form a linear vector space with dimension equal to the number of the p-cells. These and more general operations on cochains have been used in several physical modeling and simulation systems

that treated cochains as a basic data type [5, 14].

1.3 Combinatorial interpretation of physical laws

As witnessed by such cochain-based systems, emergence of cochains as a data type helped replace many cell-by-cell operations and algorithms by a powerful and elegant algebra. Our goal in this paper is to show that the same algebra is also useful in formulating, constructing, and enforcing a hierarchy of physical and design laws on combinatorial designs.

Generally, by "law" we refer to a rule that governs the behavior of the cochains (i.e. physical properties) by imposing constraints on the allowable values of coefficients (e.g., a maximum relative displacement in the 1-cochain or a limit on the total area represented by the 2-cochain in Figure 3). The coefficients in turn control existence, orientation, use, and properties of the cells in the cellular decomposition of space. Exhaustive classification of all possible laws is clearly impractical, but all such laws appear to be compositions of two types of primitive laws, broadly characterized as topological and measured (metric) [3, 25]. Topological laws may be formulated using strictly topological operations on chains and cochains, while metric laws involve measured and empirical relationships.

2 Spatial Laws and Operations

Let us first consider how topological laws manifest themselves in geometric and solid modeling. By definition, every three-dimensional solid is a topological polyhedron that may be represented as a finite three-dimensional cell complex [18]. Algebraically, a solid may be viewed as a 3-chain with coefficients 0,1,-1 indicating presence or absence of an oriented 3-cell in the complex. Let us now introduce the linear boundary map ∂ that operates on a k-chain and produces a $(k-1)$-chain. When the k-chain ($k > 0$) consists of a single cell $f_p \cdot cell_k^p$,

$$\partial(f_p \cdot cell_k^p) = \sum_{i=1}^{n_{k-1}} [cell_k^p, cell_{k-1}^i] \cdot f_p \cdot cell_{k-1}^i \qquad (3)$$

where f_p is the coefficient attached to the pth k-cell, n_{k-1} is the number of all $(k-1)$-dimensional cells, and $[cell_k^p, cell_{k-1}^i] \in \{0, -1, +1\}$ depending on the relative orientation between the pth k-cell and an incident $(k-1)$-cell. Informally, the resulting $(k-1)$-dimensional chain will have non-zero coefficients only on the incident (k-1)-cells of the given k-cell and each coefficient will be equal to $\pm f_p$ depending on the relative orientation between them. The definition of ∂ is extended linearly to any k-chain C_k:

$$\partial(C_k) = \partial(\sum_{i=1}^{n_k} f_i \cdot cell_k^i) = \sum_{i=1}^{n_k} \partial(f_i \cdot cell_k^i) \qquad (4)$$

The net effect is that all coefficients from k-cells are transferred to incident $(k-1)$-cells and summed cell-by-cell to obtain the $(k-1)$-chain. This means that every interior $(k-1)$-cell will get exactly two coefficients that will cancel each other, and the result of the operation is indeed the oriented $(k-1)$-dimensional chain representing the boundary of the solid. The bottom row in Figure 2 illustrates the boundary operation on a 2-chain C_2. In the figure, + and - indicate the positive and negative contribution of the coefficient ($f_i = 1$) of the 2-cells based on their relative orientation with the bounding 1-cells.

The above formulation using chains may appear to be a complicated method for constructing a boundary of a solid, but, in fact, it has had profound practical consequences for geometric modeling. Boundary representation of a solid is often constructed directly, and the above formulation immediately establishes the spatial laws that must be satisfied by any boundary representation. It is well known [7] that the boundary operator satisfies

$$\partial(\partial()) = 0 \qquad (5)$$

when applied to any k-chain, where 0 denotes a vanishing or null chain (a chain with all 0 coefficients). This implies that $\partial(boundary\ representation)$ must be 0 and, therefore, any valid boundary representation must be an orientable 2-cycle. In practical terms, every valid boundary representation must be consistently orientable and every edge must be shared by an even number of faces – a validity condition that is now enforced by boundary representations in all solid modeling systems.

The above validity condition is imposed on boundary representations based on the *assumed* concept of proper physical behavior. Other definitions are possible, and many of them are conveniently formulated by modifying the definition of chains, or operators, or both. For example, the formulation of boundary representation in [18] relies on chains with coefficients 0,1 that are added by mod 2 addition. While the condition (5) still holds, it no longer enforces that the 2-cycle be orientable. This is a weaker condition of validity that may be acceptable for modeling of non-orientable surfaces. These and other topological conditions on cell complexes and chains may be used to postulate and enforce spatial validity conditions on computed designs as well. Thus, it should be clear that every truss structure is a 1-cycle [21], a subdivision surface is an orientable manifold and often a 2-cycle, and a voxel model is a 3-manifold with boundary (which is a 2-cycle).

Chains and boundary operator also play a crucial role in formulating valid transformations and manipulations of cell complex models. For example, every *continuous* transformation of a cell complex g induces a family of the corresponding chain maps that must satisfy the necessary condition [16]

$$g_{k-1}(\partial(C_k)) = \partial(g_k(C_k)) \qquad (6)$$

Informally, the oriented boundary of every cell should be mapped to the oriented boundary of every non-collapsed image cell. While this condition may appear obvious in retrospect, it has been routinely overlooked by software vendors and researchers alike. One very unfortunate consequence of this omission is the

inability of the present day systems to guarantee the validity of regenerated parametric designs [16].

Similar conditions must be satisfied by many other transformations and design rules used in computational design. Examples include: Euler operators in solid modeling [12], productions in shape grammars for truss design [17], L-systems rules for design of tables [8], surface fairing by subdivisions [24], and other designer-defined shape operators [6].

3 Physical Laws

More general laws concern the distribution of assumed physical quantities over the cells in the complex. As such, they can be formulated as constraints on the coefficients of allowable cochains. These constraints may be topological, measure dependent, or compositions of the two.

3.1 Topological Laws via Coboundary

Topological physical laws are a generalization of the spatial laws and are exhibited by physical phenomena in the form of balance, equilibrium, compatibility, or conservation. In most cases, such laws take the form:

$$\sum_{\text{boundary}} \left\{ \begin{array}{c} \text{amount of the} \\ \text{physical quantity} \\ \text{THROUGH} \\ \text{the boundary to} \\ \text{the INSIDE} \end{array} \right\} = \left\{ \begin{array}{c} \text{amount of the} \\ \text{physical quantity} \\ \text{DESTROYED or} \\ \text{STORED} \\ \text{INSIDE a region} \end{array} \right\} \qquad (7)$$

It is well known that the intuitive process of transferring the physical quantity through the boundary is captured by a *coboundary* operation δ acting on the *k*-cochain of the boundary cells [3, 19, 25]. The linear operator δ is defined in a manner similar to the boundary operator ∂, except that it acts from lower-dimensional cells to their incident higher-dimensional cells [7]. For a single *k*-cell ($k \geq 0$):

$$\delta(f_p \cdot cell_k^p) = \sum_{i=1}^{n_{k+1}} f_p \cdot [cell_k^p, cell_{k+1}^i] cell_{k+1}^i \qquad (8)$$

where f_p is the coefficient attached to the p^{th} k-cell, n_{k+1} is the total number of $(k+1)$-cells, and $[cell_k^p, cell_{k+1}^i] \in \{0, -1, +1\}$ is the relative orientation between the p^{th} k-cell and the (k+1)-cells of the complex. Once again, the definition is extended to arbitrary *k*-cochains by linearity of δ. Informally, definition (8) says that all coefficients from a *k*-cochain are transferred to the incident $(k+1)$-cells where they are added cell-by-cell – precisely as required by (7).

One of the advantages of using δ to formulate physical laws is that it applies to any and all cell complexes. For example, assuming that the cells of the voxel-based table design example are endowed with physics of solid mechanics, the body force *F* (or 3-cochain) of each 3-cell is balanced by the traction forces *S* (or 2-cochain) on its incident 2-cells. The force balance can be compactly written as:

$$F = \delta S \qquad (9)$$

Figure 4 illustrates application of this law to a cochain defined by a pair of voxels. If $F_j = f_j \cdot cell_3^j$ and $S_i = s_i \cdot cell_2^i$ are single-cell cochains, then the cochains in (9) become $F = F_1 + F_2$ and $S = S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7 + S_8 + S_9 + S_{10} + S_{11}$. The balance equation for the voxel pair becomes $F = F_1 + F_2 = \{\delta(S_1) + \delta(S_2) + \delta(S_3) + \delta(S_4) + \delta(S_5) + \delta(S_6)\} + \{\delta(S_5) + \delta(S_7) + \delta(S_8) + \delta(S_9) + \delta(S_{10}) + \delta(S_{11})\} = \{\sum_{i=1}^{6} s_i \cdot [cell_2^i, cell_3^1] \cdot cell_3^1\} + \{s_5 \cdot [cell_2^5, cell_3^2] \cdot cell_3^2 + \sum_{i=7}^{11} s_i \cdot [cell_2^i, cell_3^2] \cdot cell_3^2\} = (s_1 - s_2 + s_3 - s_4 + s_5 - s_6) \cdot cell_3^1 + (-s_5 + s_7 + s_8 - s_9 + s_{10} - s_{11}) \cdot cell_3^2$. The relative orientation between the 2-cells and 3-cells is decided based on whether the arrow on the 2-cells is pointed toward or away from the 3-cells. The coboundary relation (9) between *F* and *S* is very general in that it is applicable to cell complexes composed of any type of cells - cubical, simplicial geometric, etc. It is also the discrete equivalent of the usual infinitesimal force equilibrium equation, $P_i + \sigma_{ji,j} = 0$, where P_i are the components of the body force and σ_{ji} are the components of the stress tensor. In fact, under a well defined subdivision process, each cochain in the limit becomes a unique differential form [28], and the coboundary process corresponds precisely to exterior differentiation [4]. It is well known that the coboundary δ is the discrete analogue of the vector operators gradient, curl, and divergence operators when applied respectively to 0-cochains, 1-cochains, and 2-cochains respectively [3, 27]. Just as the boundary operation ∂ is fundamental to enforcing integrity of geometric representations, the coboundary δ is central to enforcing physical laws. In particular, the well known identity $\delta(\delta()) = 0$ subsumes many vector calculus identities, including $grad(curl) = 0, curl(div) = 0$, and others. Balanced distribution of many physical properties require that $\delta() = 0$ (in which case, the cochain is called a cocycle), and the δ constraints must be preserved by continuous transformations.

3.2 Topological Laws via Boundary

An attempt to formulate all topological physical laws using the coboundary operation is likely to run into difficulties. For example, structural analysis of trusses and frames usually requires that the sum of member forces (represented by 1-cochains) and external forces applied to joints (represented by 0-cochains) should be 0. Since this is an instance of a balance law, we expect to see a coboundary formulation. But it is easy to see from Figure 5 that the force balance statement sums all 1-dimensional forces acting on the incident 0-dimensional cells. In other words, the balance law acts from a higher dimension to a lower dimension, implying some form of a boundary operation. In fact, the definition of boundary given by (3) and (4) generalizes to any cochain without modifications. The balance law becomes:

$$J = \partial M \qquad (10)$$

In the example of Figure 5 (top row), J is the 0-cochain of force J_1 at the joint and M is the 1-cochain of member forces $M_1 - M_2 + M_3 - M_4$, the relative orientation is decided based on

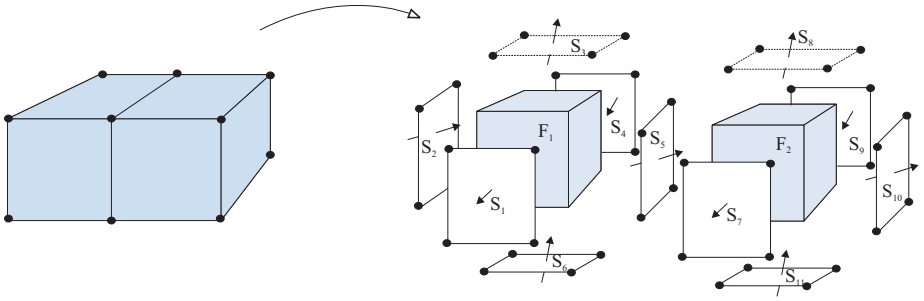

Figure 4. The force balance law can be applied via coboundary operation to any cellular structure, in this case to two adjacent voxels.

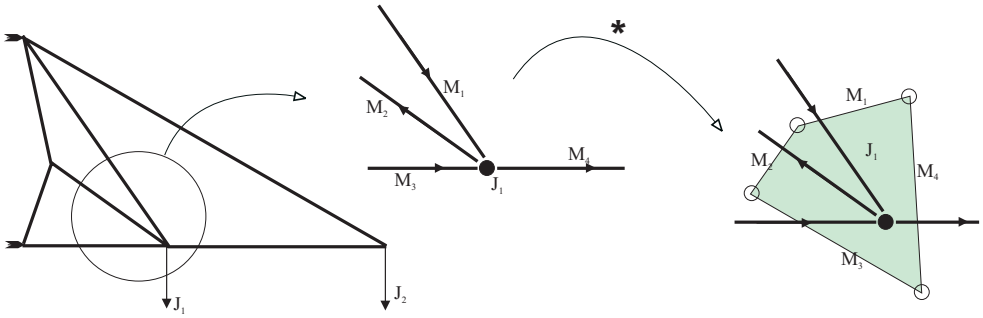

Figure 5. The force balance law may be more naturally specified using the boundary operation for some cellular structures. For trusses, the boundary of the 1-cochain of member forces is balanced to the 0-chain of forces acting on joints (top row). The coboundary formulation of the equilibrium of forces requires the association of the various cochains in their respective dual space (bottom row).

whether the 1-cells are oriented toward or away from the 0-cell. A similar situation can be observed in other physical domains, for example in analysis of electrical networks the Kirchoff's current law requires the currents on 1-cells equate to zero at an incident 0-cell [1]. Not only does such use of the boundary operation appear counter-intuitive, it also is conceptually problematic as the balance law does not appear to correspond to any differential equation. Some may also dispute the application of the boundary operation to a cochain (as opposed to a chain).

Fortunately, the difficulty is easily resolved. Because the boundary ∂ and the coboundary δ are adjoint or dual operators [7, 21], the two statements of the topological law are dual to one another. In other words they are representations of the same topological law in dual spaces. This duality Figure 6 is defined by associating a *p*-cell in an *n*-dimensional space with a dual $(n-p)$-cell and the * operator is used to transfer quantities back and forth between the dual spaces. Figure 7 explains the relationship between ∂ and δ in terms of a commutative diagram applied to the example in Figure 6. As a result, in the case of planar truss structures (Figure 5), the forces M on the joints should more properly be represented on the 2-dimensional cells dual to the joints by the dual 2-cochain $^*M$. The 1-cochain of forces J should be associated as $^*J$ with the (dual) cells bounding these 2-cells. Then relation (10) becomes an isomorphic coboundary statement of $^*J = \delta {}^*M$. That the two relationships are equivalent can be seen by applying the dual transformation to both sides:

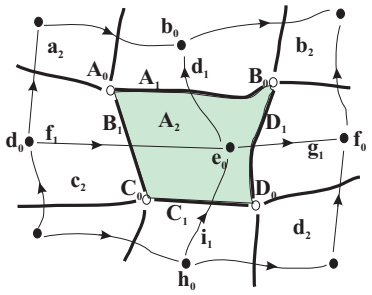

Figure 6. In the 2-D cell complex (n = 2), for every primal p-cell, there exists an (n-p)-dual cell. For example, A_0 is the dual 0-cell of the primal 2-cell a_2, B_0 is the dual to b_2 and so on. Similarly A_1 is the dual 1-cell of the primal 1-cell d_1, B_1 is dual to f_1 and so on. The 2-cell A_2 (in green) is dual to the primal 0-cell e_0.

$$J = {}^{**}J = {}^{*}\delta {}^{*}M = \partial M, \tag{11}$$

where $\partial = {}^{*}\delta {}^{*}$. Thus, every topological law on a cell complex may be equivalently expressed either using boundary or coboundary operation on cochains.

3.3 Measure dependent laws and relations

Many physical laws are established experimentally, based on experience, measurements and material properties. Such measure dependent laws include constitutive equations and equations coupling different physical phenomena. There is no reason to expect that such laws may be formulated using topological operations, but they can be expressed as functions of cochains. In principle, there are no restrictions on functions of cochains and many elegant formulations have been used in [14] and [5]. In most traditional numerical formulations, including Finite Element method, all physical quantities are approximated on a single discretization of the domain. For example in the finite element method the stiffness matrix K relates 0-cochains of forces and displacements associated with nodes of the elements. But the most common type of measured law is the constitutive relation involving cause and effect physical quantities (e.g Hooke's law between forces and displacements, Ohm's law between voltages and currents, etc.) and therefore are properly relating dual cochains – i.e., cochains defined by coefficients attached to the cells in the two decompositions of space that are dual to each other. Representing the dual physical quantities on a single cell complex obscures this duality and suppresses the geometric-physical essence of systems. Determining the correct type of cells for physical quantities is relatively straightforward based on the first principles and units of measurements. For example, the correct association of the weight of a body or the heat source within a domain should be with the volume or 3-cells as opposed to 0-cells [25].

If we accept that proper geometric type of every physical quantity is a basic principle, we will also conclude that the constitutive laws specify measured or weighted relationships between a k-cochain and a *dual* $(n-k)$-cochain associated with the dual of the cell decomposition. For example, in the table design example, the constitutive relation is between dual 2-cochain traction forces and 1-cochain relative displacements. In the truss design example, these equations relate dual 1-cochain member forces and 1-cochain member elongations. In general, such constitutive laws take the form of:

$$S = K({}^{*}E) = {}^{*}(KE), \tag{12}$$

where S and E are respectively the dual cause (source) and effect (configuration) cochains, and K is a measure/material dependent function. The two forms in Equation (12) imply that constitutive relation K may be formulated either on a single decomposition or on two decompositions of the same space that are dual to each other.

3.4 Composite Laws

A wide variety of broadly applicable laws may be constructed by combining the primitive measure-dependent and topological laws. That many such laws follow similar patterns has been observed by others. Figure 8 shows a diagram from [3] based on the earlier work by Roth [19] summarizing possible primitive laws and their compositions between cochains on a three-dimensional cell complex. The Roth diagram suggests that in most physical theorries there appears to be two types of dual physical quantities modeled by cochain sequences on two cellular decompositions of space that are dual to each other. The cochains within the same cellular decomposition are related by coboundary relationships. Applying the coboundary to the i-cochains C^i assigned to the primal cells gives the $(i+1)$-cocycles $M^{(i+1)}$ for every dimension i. Similarly the coboundary operation applied to the cochains D^i in the dual space gives the $(i+1)$-cocycles $N^{(i+1)}$ on the dual cells. The measure dependent constitutive relations usually exist between the pairs of dual p-cochains. A typical composite law is constructed by combining the primitive relationships indicated by arrows. For example, starting from a 0-cochain C^0, coboundary law gives the derived 1-cocycle M^1, the constitutive relation may transform 1-cochains $M^1 + C^1$ to the dual space to give a 2-cochain $D^2 + N^2$ which, when operated on by another coboundary gives the 3-cocycle N^3. Combining these transformations, it is easy to check that the primal 0-cochains and dual 3-cochains are always related as $N^3 = \delta {}^{*}K\delta C^0$ (right side of Figure 8).

Similar patterns are clearly exhibited in each of the computational design examples considered in this paper. In the shape grammar based planar truss design [17], the displacements u are 0-cochains and applied joint forces J are the dual 2-cochains. The 1-cochain of member elongations e is obtained by applying the coboundary operation on the displacement cochain u as $e = \delta u$. The Hooke's law for trusses relates the dual quantities, in this case member elongations e to member forces M as $M = {}^{*}Ke$. Since M is now a dual cochain, the structural balance between the member forces and the applied joint forces is given by $J = \delta M$, where J is a 2-cochain on the dual cell complex. Combining the above equations results in the familiar factored form:

$$J = \delta {}^{*}K\delta u \tag{13}$$

In the voxel-based table design example, the coboundary relation between the 0-cochain u of displacements and 1-cochain e of relative displacement defines the compatibility condition as $e = \delta u$; the dual coboundary statement defines the balance between the 2-cochain M of traction forces and 3-cochain J of body forces on the dual decomposition as $J = \delta M$. These two sets of cochains are related by means of Hooke's law, $M = {}^{*}Ke$. Composition of these three laws yields the form identical to (13), though clearly with a very different meaning.

Subdivision methods of surface design often involve a smoothing step that may be described in terms of a discrete

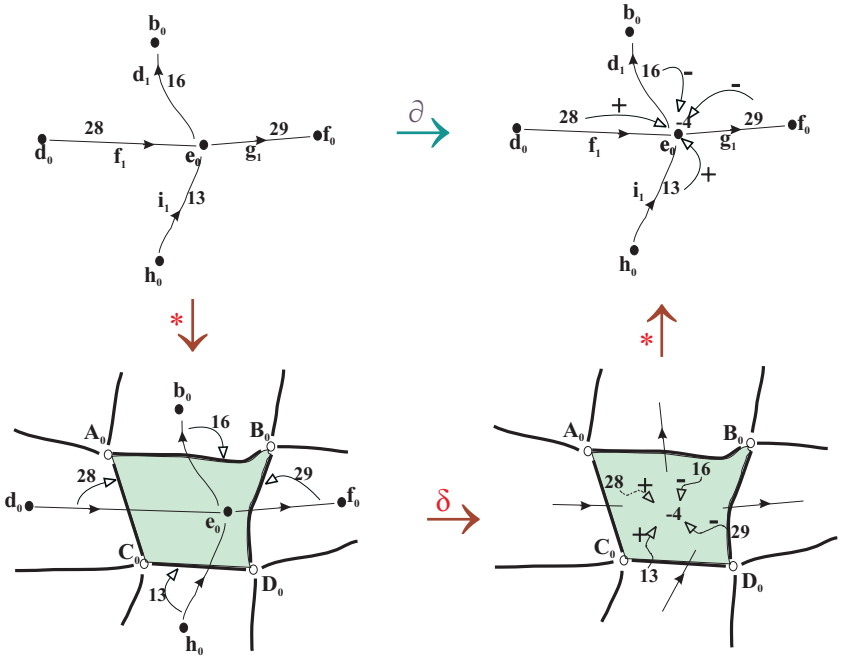

Figure 7. The boundary ∂ operation on the primal cell complex is equivalent to the three-operation sequence ${}^*\delta{}^*$. The operation ∂ (top row) transfers the coefficients from 1-cells onto the 0-cell e_0. This procedure is equivalent to applying a *operation (left column), which transfers the quantities from the primal cells to their dual, followed by a coboundary operation δ (bottom row), which transfers the coefficients from the dual 1-cells to the dual 2-cell A_2. Application of another *operation (right column) then transfers the coefficient on A_2 on to its dual, which is the primal 0-cell e_0.

Laplacian operator on positions of vertices in the surface mesh [24]. The same operator may be obtained as a composition of three primitive steps starting with a 0-cochain u of vertex positions: (1) assign the sum of vertex positions onto incident edges or $e = \delta u$; (2) multiply each edge coefficient by an (empirically determined) weight K and transfer the result to the dual edge M on the dual mesh or $M = {}^*Ke$; (3) finally, assign the sum of the coefficient from the dual edges J to the incident dual faces M or $J = \delta M$. The composition of the three steps is the same discrete Laplacian that evidently follows the pattern of (13).

The ubiquity of the patterns implied by the Roth diagram in Figure 8 has been recognized in other fields as well. Notably, Tonti [25] provides extensive analysis, classification, and factoring of physical theories in terms of the primitive topological and measure-dependent laws. Strang, while pointing out that such an approach may be too general, also observed the pervasiveness of the "equilibrium pattern" A^TKA [23]. But it is easy to show that it too follows directly from (13). If we choose to represent the result of the composite law on the primal cell complex, then the result of (13) must be transformed once more by the *operator to obtain:

$$ {}^*J = {}^*\delta{}^*K\delta u = \partial K\delta u \qquad (14) $$

It is worth noting that the linear coboundary and boundary operators may be represented in principle by incidence matrices [2]. Specifically, an incidence matrix between k-cells and $(k-1)$-cells can act as a boundary operator when applied to a k-cochain, while its transpose corresponds to the coboundary operator when applied to the $(k-1)$-cochain (see Figure 9). Then the composition $\partial K\delta$ becomes exactly the equilibrium pattern A^TKA in [23] with A^T and A carrying the boundary and coboundary information of incidence relationships. This idea is also disguised in the FEM formulation of field problems where the globally assembled stiffness matrix corresponds to the equilibrium pattern A^TKA [23].

4 Conclusion

The proposed framework extends well beyond the few examples and the equilibrium patterns illustrated in this paper. As shown in [25, 27], most physical laws may be represented in a factored form as a composition of the primitive topological and measured relationships. The formal connection with the vector calculus assures that the laws are formulated from first principles whenever possible and includes combinatorial versions of fundamental principles such as conservation of energy and Stokes theorems [22, 25]. Efforts to develop an interactive geometric language for describing, combining, and editing combinatorial laws are currently under way [4]. The factored form of physical laws

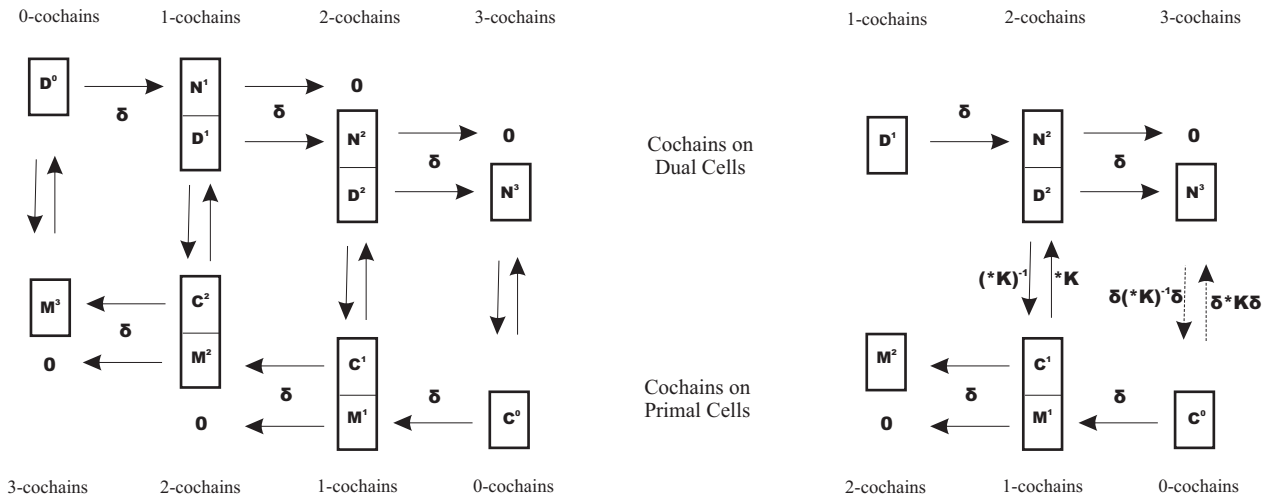

Figure 8. Roth diagram (left) shows some possible relationships between cochains on a three-dimensional cell complex. An example of composite law is shown on the right.

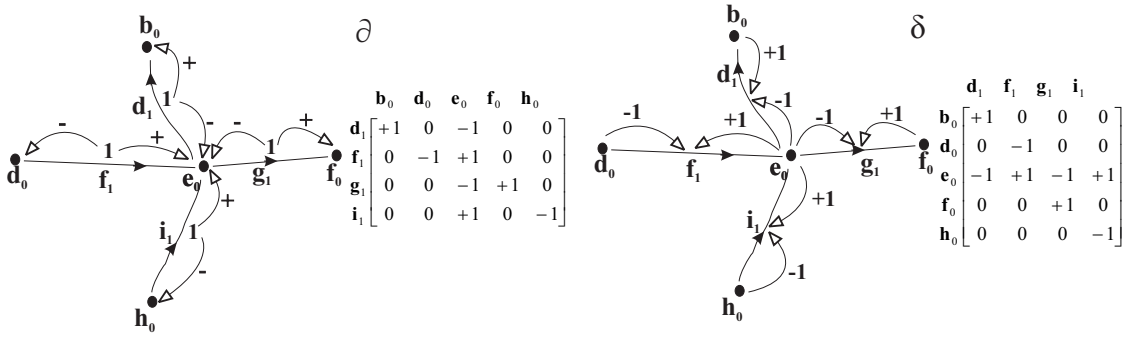

Figure 9. The incidence relationships between the 0 and 1-cells via boundary and coboundary can be represented as matrices that are transpose of each other. Orientation of a 1-cell toward an incident 0-cell is considered positive and negative otherwise.

may also be used directly to formulate, compare, and improve various numerical discretization and solution schemes [9, 13].

Furthermore, the same approach may be used to represent and enforce a variety of synthetic or user-defined laws, relations, and constraints. These include various constraints on values of, and relations between cochains, such as constraints on maximal forces in a truss structure, the fitness function of table designs evaluated from area, length, and/or weight properties of the voxels in [8], and application of the Laplacian of a discrete surface signal to surface fairing [24]. An even greater variety of laws may be constructed using operators proposed by [14] and [5], though their properties have not been studied formally.

In the context of computational design, each represented law restricts the space of admissible solutions, leading to more efficient algorithms that produce better candidate designs. A hierarchy of laws – from most general (geometric and physical) to more and more specific user-defined – suggests a corresponding hierarchical approach to design in terms of classes of possible solutions [10].

Reformulating design representations and algorithms in algebraic topological terms introduces a non-trivial conceptual overhead, but this additional investment is well worth it. The most immediate benefit is a unifying combinatorial framework for computational design and an algorithmic toolbox for constructing a variety of laws and invariants. The combinatorial representation is standard in that it is a straightforward extension of the current geometric modeling practices and systems. The proposed formulation of laws and invariants is broadly applicable across diverse design domains and computer representations. Implementation should be practical because the primitive laws are

quite simple. Thus, a promising direction for future research is to reformulate the common techniques of computational synthesis (including L-systems, shape grammars, genetic algorithms, and subdivision) in terms of law-preserving transformations on algebraic topological chains and cochains. We have shown in this paper that, depending on a particular problem domain, equivalent formulations of such techniques may rely either on one or two (dual to each other) cellular decompositions of space.

5 Acknowledgements

This work was supported in part by the National Science Foundation grants DMI-0115133, CCR-0112758 and by the National Institute of Standards grant 60NANB2D0126.

REFERENCES

[1] Paul Bamberg and Schlomo Sternberg. *A Course in Mathematics for Students of Physics*. Cambridge University Press, Cambridge, England, 1988–1990.

[2] Donald W. Blackett. *Elementary Topology: A Combinatorial and Algebraic Approach*. Academic Press Textbooks in Mathematics. Academic Press, New York London, 1967.

[3] Franklin H. Branin. The algebraic-topological basis for network analogies and the vector calculus. In *Proceedings of the Symposium on Generalized Networks*, volume 16, pages 453 – 491, Brooklyn, New York, 1966. Polytechnic Institute of Brooklyn.

[4] Jeffrey A. Chard and V. Shapiro. A multivector data structure for differential forms and equations. *IMACS Transactions Journal, Mathematics and Computers in Simulation*, 54:33–64, 2000.

[5] R. Egli and N.F. Stewart. A framework for system specification using chains on cell complexes. *Computer-Aided Design*, 31(11):669–681, 1999.

[6] J.A. Heisserman. *Generative Geometric Design and Boundary Solid Grammars*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 1991.

[7] John G. Hocking and Gail S. Young. *Topology*. Dover, 1988.

[8] Gregory S. Hornby and Jordan B. Pollack. The advantages of generative grammatical encodings for physical design. *Congress on Evolutionary Computation*, pages 600–607, 2001.

[9] 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.

[10] H. Ilies and V. Shapiro. On the synthesis of functionally equivalent mechanical designs. In *2003 AAAI Spring Symposium on Computational Synthesis*. March 24–26, 2003, Stanford, AAAI, 2003.

[11] Gabriel Kron. *Diakoptics – The Piecewise Solution of Large-Scale Systems*. The Electrical Journal, London, 1957-1959. A series of 20 articles beginning June 7, 1957.

[12] M. Mantyla. *An Introduction to Solid Modeling*. Computer Science Press, Maryland, USA, 1988.

[13] Claudio Mattiussi. The finite volume, finite element, and finite difference methods as numerical methods for physical field problems. *Advances in Imaging and Electron Physics (P. Hawkes, Ed.)*, 113:1–146, 2000.

[14] R. S. Palmer. Chain models and finite element analysis: An executable chains formualtion of plane stress. *Computer Aided Geometric Design*, 12:733–770, 1995.

[15] R.S. Palmer and V Shapiro. Chain models of physical behavior for engineering analysis and design. *Research in Engineering Design*, 5:161–184, 1993.

[16] S. Raghothama and V. Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Transactions on Computer Graphics*, 17(4):259–286, October 1998.

[17] G. Reddy and J. Cagan. An improved shape annealing algorithm for truss topology generation. *ASME Journal of Mechanical Design*, 117(2(A)):315–321, 1995.

[18] A. A. G. Requicha. Mathematical models of rigid solid objects. Tech. Memo 28, Production Automation Project, University of Rochester, Rochester, NY, November 1977.

[19] 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.

[20] J. Paul Roth. Existence and uniqueness of solution to electrical network problem via homology sequences. In *SIAM-AMS Proceedings, Vol. III*, pages 113–118. American Mathematical Society, 1971.

[21] Alf Samuelsson. *Linear Analysis of Frame Structures by use of Algebraic Topology*. PhD thesis, Chalmer Tekniska Hogskola, Goteborg, 1962.

[22] W. Schwalm, B. Moritz, and M. Giona. Vector difference calculus for physical lattice models. *Physical Review E*, 59(1), January 1999.

[23] Gilbert Strang. A framework for equilibrium equations. *SIAM Review*, 30(2):283–297, June 1988.

[24] G. Taubin. A signal processing approach to fair surface design. *Siggraph '95 Conference Proceedings*, pages 351–358, August 1995.

[25] Enzo Tonti. *On the Formal Structure of Physical Theories*. Istituto Di Matematica Del Politecnico Di Milano, Milan, 1975.

[26] Enzo Tonti. The reason of the analogies in physics. In F.H. Branin Jr. and K. Huseyin, editors, *Problem Analysis in Science and Engineering*, pages 463–514. Academic Press, 1977.

[27] Enzo Tonti. *Finite Formulation of Field Laws*. unpublished, 1999.

[28] Hassler Whitney. *Geometric Integration Theory*. Princeton University Press, Princeton, New Jersey, 1957.