Chain Models of Physical Behavior for Engineering Analysis and Design

Richard S. Palmer*
Vadim Shapiro

TR 93-1375
(replaces 93-1336)
August 1993

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

* This author's work was supported by the Advanced Research Projects Agency of the Department of Defense under ONR Contract N00014-92-J-1989, by ONR Contract N00014-92-J-1839 and by NSF Contract IRI-9006137.

Chain Models of Physical Behavior for Engineering Analysis and Design

Richard S. Palmer*
Department of Computer Science
Cornell University
Ithaca, NY 14053
rick@cs.cornell.edu

Vadim Shapiro
Analytic Process Department
General Motors R & D Center
Warren, MI 48090-9055
shapiro@gmr.com

August 11, 1993

Abstract

The relationship between geometry (form) and physical behavior (function) dominates many engineering activities. The lack of uniform and rigorous computational models for this relationship has resulted in a plethora of inconsistent (and thus usually incompatible) computer aided design (CAD) tools and systems, causing unreasonable overhead in time, effort, and cost, and limiting the extent to which CAD tools are used in practice. It seems clear that formalization of the relationship between form and function is a prerequisite to taking full advantage of computers in automating design and analysis of engineering systems.

We present a unified computational model of physical behavior that explicitly links geometric and physical representations. The proposed approach characterizes physical systems in terms of their algebraic-topological properties: *cell complexes*, *chains*, and operations on them.

1 Introduction

1.1 Motivation

Modeling the interaction of geometric shape and physical behavior dominates many engineering and scientific activities. Broadly, such activities can be divided into two categories:

- *Analysis:* representing, simulating, and predicting the physical behavior of existing objects and devices; and
- *Design:* synthesis of (e.g., the shape of) new objects and devices that exhibit a desired physical behavior.

For the purposes of this paper, we identify an object's "form" with (perhaps partial) information about its geometric shape and an object's "function" with physical aspects of its behavior. Thus, both analysis and synthesis deal with the interaction of form and function. Our research is focused on developing a unified theory, computer representations, algorithms, and systems to support a wide range of such activities. Historically, physical and geometric modeling have evolved independently and in isolation from each other, as can be seen from the following considerations.

*This author's work was supported by the Advanced Research Projects Agency of the Department of Defense under ONR Contract N00014-92-J-1989, by ONR Contract N00014-92-J-1839 and by NSF Contract IRI-9006137.

Typically, a physical phenomenon is modeled as a boundary/initial value problem of mathematical physics, where given a geometric domain and boundary/initial conditions, a time history of assumed mathematical quantities (such as forces, velocities, displacements, energy, etc.) can be computed, at least in principle. The physical behavior is then viewed as being described by integral, partial and/or ordinary differential equations, which must be solved given some initial and/or boundary conditions. The mathematical properties of boundary/initial value problems have been studied extensively (e.g., see [CH89]). Of course, such models cannot generally be solved exactly, and require numerical approximations that are often based on discretization and/or approximation of the geometric domain. Much of the vast and mature field of numerical analysis concerns representing, manipulating, and solving such approximations; the original model of the physical phenomenon and the process by which such numerical approximations are derived are rarely (if ever) explicitly represented on a computer.

Perhaps the most perceptible aspect of a physical object is its shape. It is therefore not surprising that geometric modeling is one the most developed disciplines related to modeling physical objects. In fact, until recently [RV81] the solid models were viewed as *the* models of physical objects. Increasingly, it is argued that physical objects may be also modeled as more general point sets [Wei86, RO90, GCP90]. Geometric models support a variety of spatial computations (such as rendering, interference detection, and motion planning), as well as computations of certain physical properties that can be reduced to geometric computations in restricted situations (such as inertial properties, assuming homogeneous distribution of material [RV82]). They can also be used to compute the discrete data (for example, finite element meshes) needed by various numerical methods to find approximate solutions to boundary/initial value problems. However, geometric models do not explicitly represent more general physical quantities, such as heat, stresses, or magnetic flux. Such quantities are simply not part of the assumed mathematical model.

Thus, most geometric modeling activities assume that physical modeling is a largely disjoint discipline, while the majority of physical analysis and simulation codes treat geometric information as a given input. Below we examine some consequences resulting from the lack of closer coupling between geometric form and physical behavior, both for engineering analysis and design.

1.2 Analysis and simulation

Centuries of scientific progress has produced mathematical models for variety of physical phenomena. Since the advent of the computer, a vast array of numerical methods and libraries has been developed which can be applied to solve particular problems of mathematical physics, few of which admit closed-form solutions. Broadly, numerical methods for analysis and simulation of such models can be divided in two categories: those focused on solving distributed parameter systems (for example, using finite differences, finite and boundary elements), and those dealing with lumped parameter systems (such as modeling by analogy, network-based modeling). The value of these numerical methods for science and engineering can hardly be overestimated, yet these methods do not generally address (or take advantage of) the physical aspects of computations, or the representational issues described in this paper.

For example, it is generally recognized that the finite element method is formulated as a means to approximate continuous functions, and not to model or represent physical phenomena [CME89, ZL89, TO93]. The lack of such an explicit representation limits the flexibility of typical finite element codes. To quote from [TO93]: "In the majority of finite element codes, it is assumed that all decisions and choices have already been made and the program follows a prescribed procedure to produce final results or, quite frequently, give error messages or just stop execution." Today, these decisions and choices are made by human experts and deal with selection of a mathematical model of the phenomenon, discretization of the involved geometric domains, methods of approximation, and sequence of solution steps. Developing rigorous mathematical foundations to model, represent, and automate some or all of these steps is an important research problem [TO93].

An important line of work in physical modeling grew out of the apparent similarities between electrical circuits and other kinds of physical systems. A nice historical survey by Evans and Dixhoorn [EvD74] suggests that initially, physical analogies were used to better understand electrical circuits, but with the introduction of better tools for analyzing circuits (particularly the oscilloscope), there was a move, initiated by Nickle[Nic25] in 1925, to use electrical circuits to model other physical domains. Kron used "equivalent electrical circuits" to model distributed parameter systems, and argued that the information described by equations can be also captured on a more fundamental level [Kro39, Kro45, Kro59]. His pioneering network-based approach to modeling and solving physical systems by analogy with electrical networks stimulated a new direction of research in physical modeling that is still active today. Once an analogy was defined, Kron built circuits to model a given physical system, and used circuit analyzers to determine its properties. This occurred years before the first digital computers were built.

One network-based formalism, the bond-graph, was developed by Henry Paynter[Pay61]. Using this methodology, a physical system is "reticulated" into a bond-graph by systematic "lumping" of spatially distributed physical quantities, so that the system's dynamic behavior (time-frequency response) can be simulated and analyzed. Bond graphs have been proposed as a universal language for engineering, but they do not represent geometric information (which has been abstracted into a small number of real numbers [SV89]), and are thus not informationally complete.[1]

Because the mathematical model itself and the process of discretization are not explicitly represented on a computer, the current state of the art in analysis and simulation of physical processes comprises painstakingly created special purpose tools. Since these representations and tools are usually created independently, great difficulty arises in attempting to make them work together. Thus, differences in assumed physical behaviors, type of models, and solution methods lead to significant difficulties in computer modeling and simulating problems involving multiple interacting physical phenomena [WTC93] or both lumped and distributed models of behavior. A common method of dealing with such problems is based on "coupling" the output of one program with input of another, which requires special-case handing and/or places significant restrictions on the solution techniques.

We conclude our (brief and necessarily incomplete) remarks on analysis and simulation by noting that Kron's work also served as a point of departure for Branin [Bra66] and Tonti[Ton75], who have advocated (as did Kron[Kro59]) that algebraic topology forms a common and universal basis for classification of physical theories. In this paper, drawing on ideas on Branin and Tonti, we propose a unified formal model of physical behavior that explicitly links geometric and physical representations.

1.3 Design of form from function

A restricted, but important, line of research in design concerns the process of "generating geometry" having a desired physical behavior [SV89]. Such an informal description of this task assumes the existence of a computer representation (or language) in which the design specification can be expressed, from which the process of generating form can be systematically and algorithmically car-

[1] A representation is informationally complete (with respect to a given task), when it contains all information needed for a task to be carried automatically, at least in principle, without additional information or human guidance.

ried out on a computer, with or without additional information or human guidance. But as Voelcker observed in [Voe88], " One of the major gaps in the understanding of design is the lack of means for modeling mechanical "function" in a manner that links function to form." Because of that, synthesis, the process of inducing form from function, is generally viewed as a poorly understood creative activity performed by humans. And more specifically, we do yet not have adequate formal models for "function," or for form-function interaction that could support engineering synthesis.

In current engineering practice, physical information is usually represented as a geometric object together together with textual and symbolic information, including equations to be solved, descriptions of functionality, and any other information deemed relevant to the modeled situation. Advanced and sophisticated systems strive to organize this information on a computer in hierarchical and relational fashion [Hen93, GM93]. The semantics (meaning) of these representations is often informally specified, and their use generally requires an "intelligent agent" (e.g., human) who knows what to do with them. This 'geometric approach' to physical modeling is historically based on common engineering practices that assume that geometric information is the primary communication medium, and that other aspects of a physical system can be communicated informally or implicitly. Annotated drawings of mechanical parts (as opposed to descriptions of their function) have been traditionally used to relate design and manufacturing activities. Because these models have no precisely represented meaning, it is impossible to determine whether such descriptions are unambiguous, and under what conditions they represent feasible physical situations.

Given the primary role geometry plays in representing physical objects, it seems clear that all other information must somehow be "attached" to the geometric model. Thus, (representations of) geometric entities are often grouped together to form a "geometric feature." Numerous definitions and uses of features have been put forward [Sha91a], and many challenging technical problems remain [DLH90]. Features are intended to capture and represent engineering "knowledge" and "function," but there are currently no agreed means for doing so.

Many researchers have attempted to integrate geometric, topological, and physical information for various purposes and with varying degree of success [SG92, MCKP91, US89, DD88, CA91, SV89]; a comprehensive survey is outside the scope of this paper, but see [FD89]. A common limitation of approaches described in the design research literature is the lack of a complete and explicit model of physical behavior. Domain restrictions, physically impossible results, and case-by-case analysis are some manifestations of this limitation.

1.4 Our goals

The current situation in physical modeling is reminiscent of the problems that plagued the geometric modeling community in 1960's and early 1970's, before the now common taxonomy of geometric models and representations emerged. From the lessons learned in geometric modeling[RV82], it seems clear that the desired advances in computer analysis and synthesis of physical systems can be achieved only when appropriate mathematical models of physical behavior are identified, so that formal properties of various computer representations can be studied.

Informally, it is clear that any useful mathematical model and computer representation of a physical object and system should be capable of predicting the behavior of a real physical object in response to some set of initial stimuli. Ideally, we would like our model to apply to (at least) the same broad range of physical phenomena that can be modeled by differential/integral equations. At the same time, we seek a *computational* model, one that translates directly into a computer representation. Annotated geometric models, geometric features, bond graphs, and finite elements — all are important examples of computational tools for analysis and design, yet each by itself is unable to model some important aspect of physical behavior.

In this paper, building on work of Branin [Bra66] and Tonti[Ton75], we characterize physical systems in terms of their algebraic-topological properties: finite *cell complexes, chains,* and operations on them. In addition of possessing many attractive computational properties, we believe that the proposed models will allow us to formally define, represent, and unify much of the information that is currently either assumed or missing in computer systems for analysis, simulation, and design in engineering, In particular, we argue that such models support specification, refinement, and synthesis of engineering designs.

More generally, one may view the formalism described in this paper as a step towards defining a computer language that includes concepts from physics, geometry, discretization, and numerical solution. Such a language would bridge the gap between these disciplines, and enable physical systems to be directly expressed in terms of computer data structures, in much the way that geometry is currently expressed. One might then expect to define, store and retrieve computational models of physical systems using sentences in a well defined computer language. As an example, a syntactic form similar to that used in geometry for finding the intersection of two solids could be used to query a physical model for regions of maximum stress.

In Section 2 we investigate the properties required of mathematical models and computer representations of physical systems, and conclude that the key to physical modeling is the ability to effectively compute with distributions of physical quantities in space and time. In Section 3, we use algebraic topology to develop mathematical models of such distributions, based on cell complexes, chains, and certain topological operations on chains, which we propose as the basis of informationally complete models and computer representations of physical systems and behaviors. In Section 4, we define *physical elements,* which are computational structures for representing and manipulating (models of) physical behaviors. We conclude in Section 5 by discussing how the proposed formal machinery applies to problems in engineering analysis and design.

2 Towards effective representation of physical systems

2.1 From physical objects to computer representations

We have argued that much of the difficulty in applying computers to modeling physical systems can be explained by the lack of appropriate computer representations. In what follows, we attempt to systematically investigate mathematical models and computer representations of physical behaviors and systems, which we hope will lead to rigorous formulation of many computational problems in engineering design and analysis.

Our initial approach is analogous to the modeling paradigm that Requicha and Voelcker [RV81] employed to develop mathematical models and representations of (geometric) solid objects. Figure 1 distinguishes between three classes of objects. Physical objects and systems are the "real things" that we are trying to model on a computer. They are first abstracted using appropriate mathematical models, which are then represented on a computer using data structures and programming languages. The mathematical models of real objects establish precisely what is to be represented on a computer. These mathematical models must be defined before formal properties of various representations such as validity, completeness, and uniqueness can be studied. In summary, the computer model of a physical object is a computer representation of a mathematical model that is an abstraction of a real thing.

Perhaps the most important and difficult task in computer modeling is to choose an appropriate mathematical model that captures the real world properties of interest. Mathematical models are

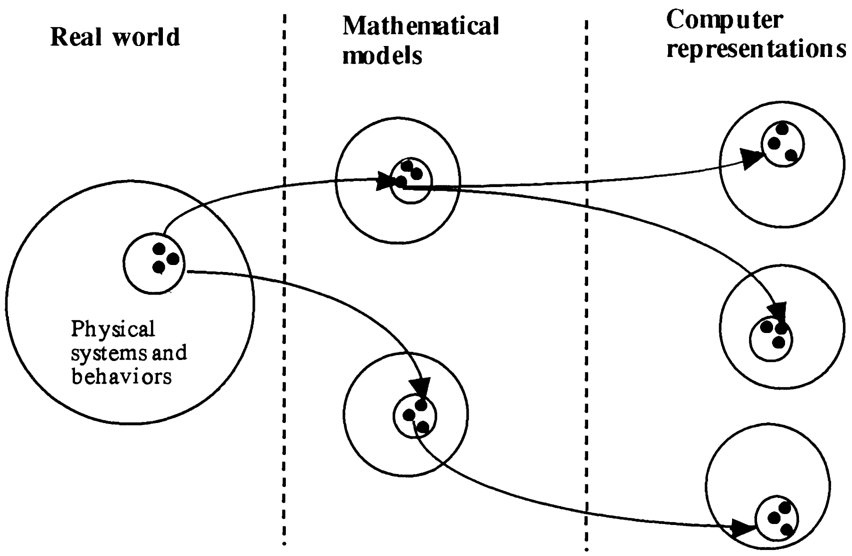


Figure 1: Physical systems, mathematical models, and computer representations

not unique, and their selection is a delicate and often subjective matter. For example, depending on computational requirements, solids have been defined as r-sets[Req77], manifolds with boundary [Hof89], and as open-regular sets[Arb90].[2] A mathematical model of a physical object must be capable of predicting the behavior of a physical object in a variety of situations (e.g., initial and boundary conditions). Thus, such a model may be viewed as a map from stimuli to responses. A computer representation of a physical object or system is set of data structures that support algorithms to implement this map, so that the behavior of a physical object can be predicted without human intervention.

The choice of an appropriate mathematical model depends on more than the ability to predict behavior in principle – there must be a well defined map between the data structures of the computer implementation and the mathematical model. For practical computer implementations, the data structures in the computational implementation of the model must support efficient algorithms. In other words, the mathematical models must be *computationally effective.* While this "computational effectiveness" is a relative and ephemeral term (rapidly improving computer hardware is constantly expanding the class of effectively computable problems), it is indisputable that the utility of a computational method is intimately related to the organization of its data and operations.

2.2 Physical objects and behaviors

Let us consider possible models of a simple physical object, a steel ball found in a ball bearing. If we wish to view the shape of the ball on a CAD system, a geometric model of the surface is sufficient. But if we wish to study the (rigid body) dynamics of the ball, we need no geometric model at all, since the distribution of mass within a rigid body can be expressed by an inertia tensor – a few numbers. Finally, if we are interested in elastic properties of the ball, we may require an appropriate model of deformation (such as linear elasticity, which we might approximate using

[2]It is also important that the relationship between these models is well understood.

the finite element method), so that stress and strain distributions can be computed. Yet another model may be required for thermal analysis, and so on.

In each case, we focused on a distinct aspect of the ball's physical behavior. What is common to these various models of this object? They all consist of *distributions* of physical quantities in time and space. A non-exhaustive list of physical quantities includes mass, momentum, energy, force, position, velocity, entropy, charge, etc. Even geometric shape is nothing more than a static distribution of homogeneous material in space. To be more precise, we assume that the mathematical model of a physical quantity q is a tensor (e.g., scalar, vector, matrix) derived from an observation or measurement in physical space and time, and make the following definition.

Definition 1 A *system (or object)* is a set of quantities $\{Q\}$ distributed in space and time.

To make this definition precise, mathematical models of space, time, and distributions must be chosen, which we do in Section 4.1. Not every collection of distributed quantities corresponds to a physical behavior. The distributions of these quantities must also satisfy certain constraints, called physical laws. Conservation laws, equilibrium properties, Hook's and Ohm's laws are all examples of such physical laws.

Definition 2 A *physical* system (object) is a system that satisfies some *physical laws*, which are constraints on the values of these distributions.

The nature of physical constraints is discussed in detail in Section 4.2. According to the above definitions, the state of a physical system (object) is characterized by specifying the distribution of all physical quantities throughout the domain S at a given time. Note that our definition equates systems and objects, even though a conceptual distinction may be useful. A physical object is a physical system viewed as a whole, with behavior described in terms of the whole. Thus a given physical object may on one hand be viewed as a system (as when a radio is viewed as a system of electronic components), while all or parts of a physical system may be viewed as an object (for example, the output circuit, or the fluid in a given region of space).

It is often necessary to talk about classes of physical objects that exhibit the same behavior. Thus the term "physical behavior" is associated with various patterns or properties of distributions of such physical quantities.

Definition 3 A *physical behavior* is the class of all physical objects (or systems) satisfying a given set of physical laws. A given physical object is said to *exhibit* a behavior if it is in the class.

Thus for instance, we may define a class of conservative systems to include those distributions that satisfy the constraint of energy conservation; similarly, linear-elastic objects are those distributions of stresses and strains that are constrained by equilibrium, compatibility, and generalized Hook's laws; etc.

The above definitions should not be surprising; they are intended to capture the intuitive meanings of physical behaviors and systems, so that effective mathematical models (and computer representations) can be developed. But before we attempt to develop such mathematical models, consider how common engineering definitions and problems can be interpreted in terms of distributions.

In practice, it is impossible to know the values of all physical quantities at all locations and at all times. Instead, the physical quantities are observed at some times and at some places. Given these observables, it is the purpose of engineering analysis and simulation to determine the distribution of the same quantities in all places and times of interest, using the applicable physical laws. Thus boundary/initial value problems of mathematical physics concern the existence, uniqueness, and determination of distributions of physical quantities that are governed by physical laws, boundary conditions, and other constraints in time and space. Integral and partial differential equations are constraints on allowable distributions of physical quantities. Finite element analysis is one way to approximate distributions of physical quantities in space that satisfy the given integral/differential constraints. Bond graphs specify the incidence relationship between adjacent lumps of predefined physical quantities and transformations between these quantities corresponding to the selected physical laws.

If the properties of a physical object are constraints on its behavior, the purpose of engineering design is to produce (synthesize) a physical object (system) from these constraints. In terms of our definitions, this physical object is simply a set of physical quantities distributed in time and space that satisfy the constraints. For instance, we could specify the required static behavior of a simple bracket by defining inequality constraints on the displacement and applied loads, as well as perhaps an enclosing volume in which the material comprising the bracket must be contained. Thus, a design specification should be a formal statement placing constraints on distribution of physical properties of interest, including the object's shape, boundary conditions, and so on. It is usually the case that there are many physical objects satisfying the same specification. Thus we cannot expect to formulate the synthesis problem so as to guarantee a unique solution.

There are also common engineering terms that do not currently have precise meaning. Design intent (the intention of a designer) is, by definition, an inherently inexact and amorphous term that cannot be explicitly represented or computed. It is nevertheless the goal of the design specification to capture the semantics of the design intent. An engineering "feature" is another term lacking a formal definition. It is possible that viewing design intent and features in terms of appropriate quantity distributions will lead to a better understanding of these and other similar concepts.

2.3 Effective models of physical systems

The key element in mathematical modeling of physical behavior and systems is modeling of distributions of physical quantities. There are many ways to define such distributions; our goal is to define computationally effective model of distributions that support a variety of tasks needed in physical modeling. We must be able to describe, compute, update, synthesize, measure, and query spatial distributions of physical quantities, as well as their evolution in time.

A classical way to model distributions of physical quantities is by "sufficiently smooth" functions defined over continuous space and time that satisfy a given partial differential equation. A physical model is 'solved' when distributions of assumed physical quantities are known at all places and times of interest. This can be achieved by exhibiting an *explicit* function of the form

$$p = f(x, y, z, t, \ldots),$$

describing the behavior of every physical property p as a function of position, time, and perhaps other parameters. Such continuum equations are declarative: they can be formulated to specify unique solutions, but they do not offer a direct means of evaluation in general. To compute an (approximate) solution to such a continuum problem, discretization must be performed (of time, space, or function space), a numerical solution method must be chosen, the problem transformed to the discrete domain, and solved there. But the information necessary to perform these steps is not contained in the equations. Since our goal is to define computer representations for physical objects that support prediction of behavior without human intervention, it is clear that we must add some information to the traditional models to make them effective.

In the next section we describe computer representable mathematical models of distributions of physical quantities. These models are called *chains* in algebraic topology. With chains we will construct finite mathematical models of physical behavior that can be exactly represented on a computer, and are consistent with classical continuum models, in a sense we shall describe in Section 3.5.

3 Computational Models of Distributions

We have defined a physical system as a distribution of physical quantities in time and space that satisfy certain constraints, such as conservation laws, and discussed the notion of effective representation of such systems. In this section we develop computational models of distributions and show how they can be used for modeling geometric and other physical properties. Our discussion is self-contained but relies on notions from point-set and algebraic topology, as well as differential geometry, such as can be found in standard texts[Mun84, BG80, Bur85, BS90].

3.1 Cells and complexes

The concept of 'distribution of a physical quantity' implies the ability to characterize and hence distinguish the value of quantities in different regions of n-dimensional Euclidean space, E^n ($n \leq 3$ in this paper). To distinguish regions of space, we employ standard concepts from algebraic topology: cells, complexes, chains, and operations on them.

We shall assume that space and time are continuous, and that physical quantities may be either continuous or discrete. We restrict our attention to phenomena occurring in regions of space that are finitely describable and have well-behaved boundaries. Such a space $S \subseteq E^n$ can be decomposed into a finite number of cells of various dimensions. A common cell decomposition is the triangulation, which exists for most sets of interest in engineering (e.g., for all semi-analytic sets and for all manifold objects); in this case, all cells c_i are (possibly curved) simplices (see [Mun84]) of dimension $n = 0, 1, 2, 3$. Other cell decompositions may be chosen based on the requirements of a given application, as described below.

Definition 4 An *n-cell* c is a set that is homeomorphic[3] to a closed unit n-ball B^n. The closed unit n-ball is a subset of $\Re^n$: $B^n = \{x \in \Re^n \mid \|x\|_2 \leq 1\}$. An n-cell c is said to have *dimension* n.

Thus, we define an n-cell to be a closed set that is topologically equivalent to B^n. A given function (i.e., homeomorphism) h defines a cell c: $c = \{x \mid h(x) \in B^n\}$, and thus h may be viewed as a representation of c.

Definition 5 The *boundary* of an n-cell c is the set $\partial(c) = \{x| \ \|h(x)\|_2 = 1\}$, where h is a homeomorphism defining c.

The above definitions are a matter of convenience; they can be easily modified to accommodate other kinds of spatial decompositions. For example, open cells could be defined similarly. Another computationally useful decomposition of space (called *stratification*) subdivides the space into a finite collection of open submanifolds of various dimension (strata) [RO90, Sha91b]. We have defined cells that are (at least implicitly) embedded in space, although we may, on occasion, view cells as abstract combinatorial objects as well.

[3]Two sets S_1 and S_2 are homeomorphic if there is a continuous bijection $h : S_1 \rightarrow S_2$ [Mun84].

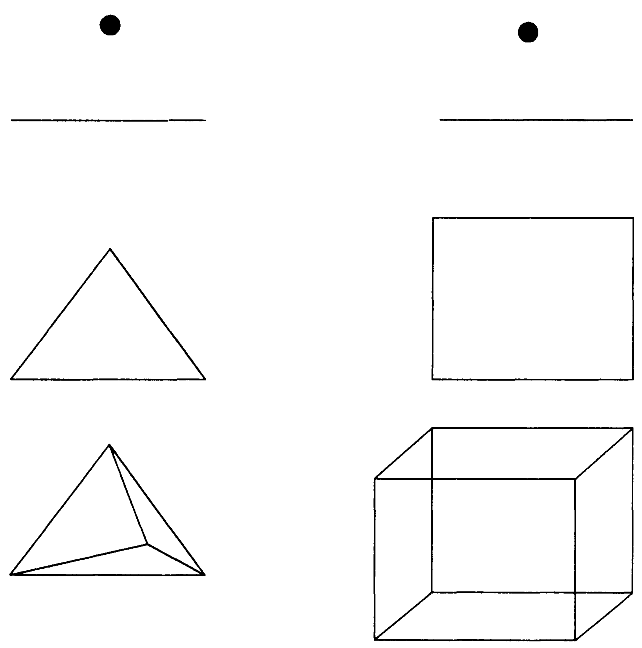


Figure 2: n-simplices and n-cubes for n less than four

Definition 6 A *cell complex* K is a set of cells that satisfy the following properties:

1. The boundary of each n-cell c is a finite union of $(n-1)$-cells in K: $\partial(c) = \bigcup_j c_j$
2. The intersection of any two cells c_i,c_j in K is either empty, or is a unique cell in K.

Because we are interested in computational representations, the cell complexes described in this paper will always be finite. Several kinds of cell complexes are useful for physical systems modeling. When all cells are simplices, K is called a simplicial complex. Other types of cells give rise to other types of complexes, such as cubical[BG80], CW[Mun84], geometric [RO90]. Figure 2 shows the first four n-simplices (point, line segment, triangle, and tetrahedron), and first four n-cubes (point, line segment, quadrilateral and 3-cube). While our examples will rely mostly on simplicial and cubical cell complexes, we stress that the arguments apply to any cell complexes as defined above. Furthermore, nothing in the definition of cell complex requires that we restrict complexes to any one of these cell types, and in fact we may find it computationally advantageous to use heterogeneous cell complexes, e.g., complexes containing both simplices and cubes.

Definition 7 A cell complex K is said to *decompose* a region R if R is equal to the union of the cells of K.

The assumption that a given domain can be decomposed into a finitely describable cell complex is common in solid and geometric modeling [Req77, Hof89] to support development of algorithms and to assure the validity of the computed results. Similarly, domain decompositions are crucial to the development of the analysis and simulation tools, such as those based on finite elements, finite differences, network analysis, and so on.

When a cell complex K decomposes R, every point in R is contained within a unique lowest dimensional cell. It follows that there is a unique cell representing the intersection of any two (intersecting) cells in K. These two facts assume fundamental importance in representing physical systems, for they provide us a well defined mathematical object with which to associate any of the physical quantities in a given region in space.

Definition 8 The *faces* of an n-cell $c \in K$ are the $(n-1)$-cells in K comprising its boundary. If f is a face of c, then c is a *coface* of f.

Thus, the cofaces of a n-dimensional cell c_i are the $(n+1)$-cells of which c_i is a face. The face/coface relationships between cells capture all incidence information in a cell complex.

Definition 9 An *oriented* cell is a pair $c = (u, o)$, where u is an (unoriented) cell, and $o \in \{1, -1\}$.

We use the notation $\sigma(c_i, c_j)$ to represent the orientation relationship between an oriented cell $c_i = (u_i, o_i)$ and one of its faces $c_j = (u_j, o_j)$. It is defined as $\sigma(c_i, c_j) = o_i o_j$ and can assume values of 1 or -1. Thus, the orientation relationship defines a map between p-cells (the cells of dimension p) and $(p-1)$-cells of a cell complex. When $\sigma(c_i, c_j) = 1$ cells c_i and c_j are oriented consistently. The definition of orientation given here is strictly algebraic, and does not require embedding. From now on we shall assume that all cells are oriented.

3.2 Chains

Where cells and cell complexes provide the mathematical machinery for decomposing space into simple regions, with sufficient structure to represent the relationships between these regions, *chains* are used to associate distributed physical quantities with these regions. For computer representations, we use finite chains, i.e., chains defined over finite cell complexes.

Definition 10 A *p-chain ch* defined over a complex K, and a vector space G is a formal sum $\Sigma_{c_i \in p\text{-cells}(K)} g_i c_i$ of *p-cells* of K with coefficients $g_i \in G$. We use the notation $Ch(c_i)$ for the value of the coefficient associated with the cell c_i in Ch.

A p-chain can be viewed as a map from the p-cells (cells of dimension p) of a complex K to some domain G; it assigns an element of G to each p-cell $c_i \in K$. The elements of G will be used to represent physical quantities (such as mass, momentum, or charge) associated with a given p-cell. More generally, the chains we use in representing physical systems will be tensor (e.g., scalar, vector, or matrix) valued. We will see some examples of such chains later in this section of the paper.

Finite chains have many attractive computational properties. Algebraically, the set $Ch(K, G, p)$ of all p-chains over a given complex K with coefficients in G form a vector space, which enables us to use the all of the vector operations on chains. Thus, we may add two p-chains, or multiply a p-chain by a scalar. In addition, p-chains support algebraic-topological operations, including the *boundary* and *coboundary* operators defined below.[4] These operators will be used to represent physical laws such as conservation of mass and energy.

Definition 11 The *boundary* $\partial(ch)$ of a p-chain $ch \in Ch(K, G, p)$ is a $(p-1)$-chain defined as follows: $\partial(ch) = \Sigma_i g_i c_i$, where $g_i = \Sigma_{cf \in cofaces(c_i)} \sigma(cf, c_i) ch(cf)$.

[4]Because we are concerned only with finite cell complexes, we shall not distinguish between chains and cochains, as they are isomorphic. See [BG80], p. 199.

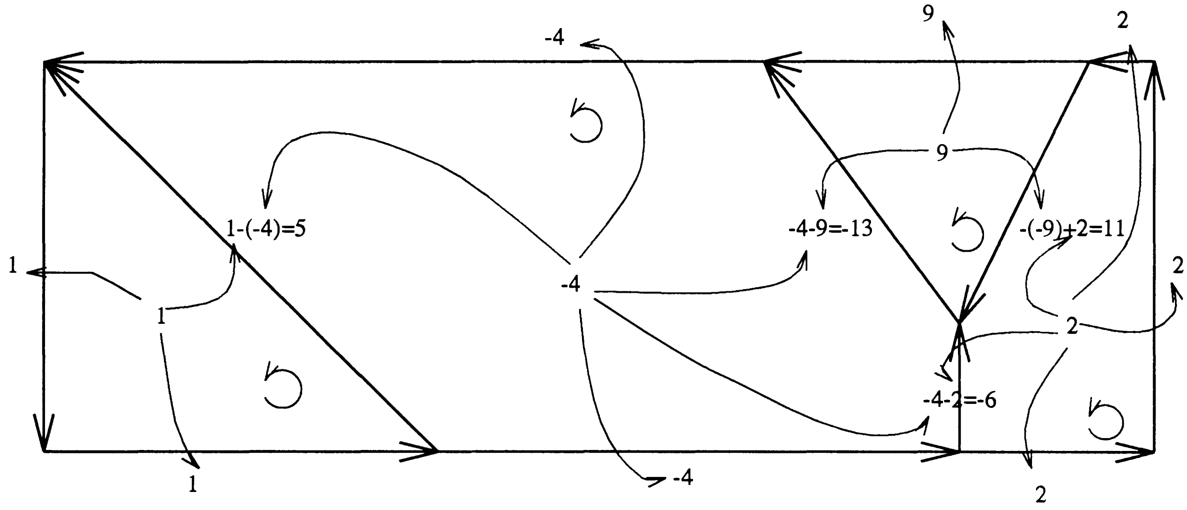


Figure 3: The boundary operator applied to a 2-chain.

This algebraic-topological operation defines a $(p-1)$-chain in terms of a p-chain, and should not be confused with the geometric boundary of a point set. However, we shall see in Section 3.3 that the two concepts are closely related: the geometric boundary can be formulated as a special case of the boundary operator on chains.

Definition 12 The *coboundary* $\delta(ch)$ of a p-chain $ch \in Ch(K,G,p)$ is a $(p+1)$-chain defined as follows: $\delta ch(K,G,p) = \Sigma_i g_i c_i$, where $g_i = \Sigma_{f \in faces(c_i)} \sigma(f, c_i) ch(f)$.

Thus, the boundary can be viewed as an operator that associates, with each p-cell $c_i \in K$, the oriented sum of all the $(p+1)$-cells of which c_i is a face, while the coboundary associates, with each p-cell $c_i \in K$, the oriented sum of all the $(p-1)$-cells that are faces of c_i. Figure 3 illustrates the boundary operator applied to a 2-chain represented in the figure by the coefficients (reading left to right) $1, -4, 9, 2$. The arrowhead on a 1-cell indicates the positive orientation of that cell, as does the circular arrow in the interior of a 2-cell in the figure indicate the orientation of c, while the circular arrow in the interior of a 2-cell indicates the orientation of that cell (in this figure all 2-cells are oriented counterclockwise). A 1-cell c_j is consistently oriented with one of its cofaces c_i if the direction of c_j is the same as circular arrow associated with c_i. In this case, $\sigma(c_j, c_i) = 1$, otherwise -1. In this figure, the lighter curved arrows indicate (oriented) values being transferred from the 2-cells to their faces.

Figure 4 illustrates the coboundary operator applied to a 1-chain represented in the figure by the coefficients associated with 1-cells. The lighter curved arrows indicate values being transferred from the 1-cells to the 2-cells. The resulting 2-chain has coefficients (reading left to right) $6, 21, 4, 16$.

Since chains $Ch(K,G,p)$ form a vector space over G, any given function f defined on the elements of G can be extended to chains by applying f to coefficients of individual cells.

Definition 13 Given a function $f : G \to S$, and a chain $ch = \Sigma_{c_i \in p\text{-cells}(K)} g_i c_i$, we define $f(ch) = \Sigma_{c_i \in p\text{-cells}(K)} f(g_i) c_i$.

More generally, we may form expressions using operations such as multiplication or division of the elements of two p-chains. These operations are performed on an element by element basis.

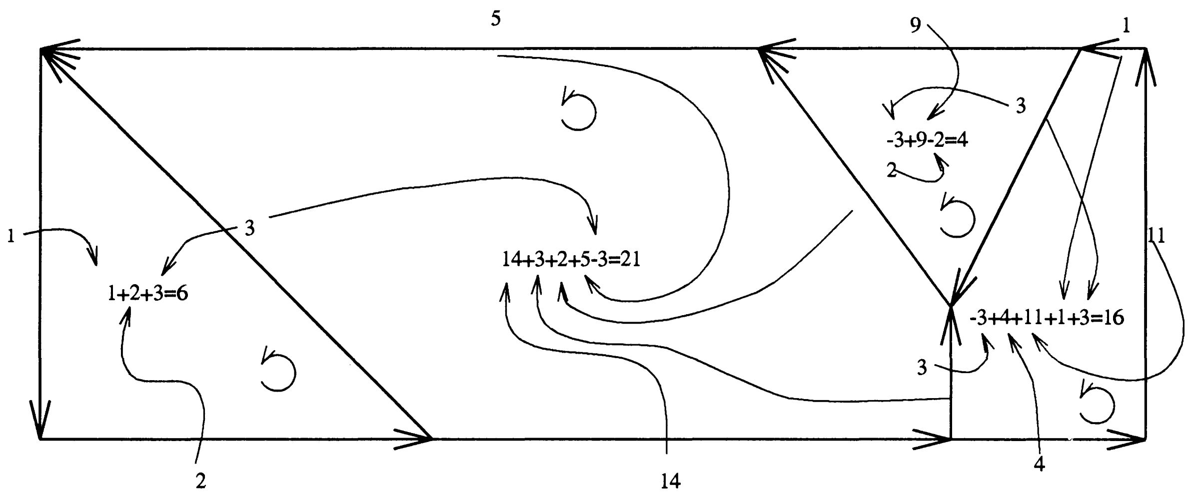


Figure 4: The coboundary operator applied to a 1-chain.

For instance, we may define (elementwise) multiplication: $ch = ch_n ch_d$ is defined to be the chain $f(ch) = \Sigma_{c_i \in p\text{-cells}(K)} ch_n(c_i) ch_d(c_i) c_i$

Finally, we define operation of *equality*, which has value 'true' whenever two p-chains $ch1$ and $ch2$ are defined over the same G and K, and $ch1(c_i) = ch2(c_i)$, for all $c_i \in p\text{-cells}(K)$.

The above operations (boundary, coboundary, addition, multiplication by a scalar, function application, and equality) on p-chains defined over a complex K and a vector space G comprise the first set of tools we use to represent physical systems. We now consider how chains are used to model distributions of physical quantities.

3.3 Chains in geometric modeling

Geometric modeling is concerned with representing points, curves, surfaces, and volumes in Euclidean space, and with algorithms to analyze and manipulate them. The physical quantity of interest in geometric modeling is shape, which can be characterized by the presence or absence of some homogeneous material in every cell in a decomposition of Euclidean space. To algebraically distinguish full cells from empty ones, we may associate the integer 1 with each non-empty cell, and 0 with each empty cell. This simple machinery is sufficient to define and compute many properties important in solid modeling, such as (geometric) boundary, cyclicity, connectivity.

Any solid S is an orientable topological polyhedra [Req77, Hof89]. In other words, S can be defined as a homogeneously 3-dimensional cell complex K. Furthermore, all 3-cells of K can be oriented *coherently*. For example, if S is a manifold solid with boundary, then it can be shown that every 2-cell is a face of either one or two 3-cells in K. Let a 2-cell c be a face of both 3-cells a and b, i.e. $a \cap b = c$. Then a and b are oriented coherently if $\sigma(a, c) = -\sigma(b, c)$. Note that this definition does not depend on the orientation of the 2-cells in K.[5]

Using coherently oriented 3-cells of K, we may redefine S as a p-chain $\Sigma_i a_i c_i$, where all $a_i = 1$. If a 2-cell c is a face of two 3-cells c_j, $j = 1, 2$, then applying the boundary operator to S produces a 2-chain that associates a sum $\Sigma_j \sigma(c_j, c)$ with every 2-cell $c \in K$. Because all 3-cells c_j are

[5]A similar argument can be used for a non-manifold solid S, where every 2-cell is a face of an even number of coherently oriented 3-cells in K.

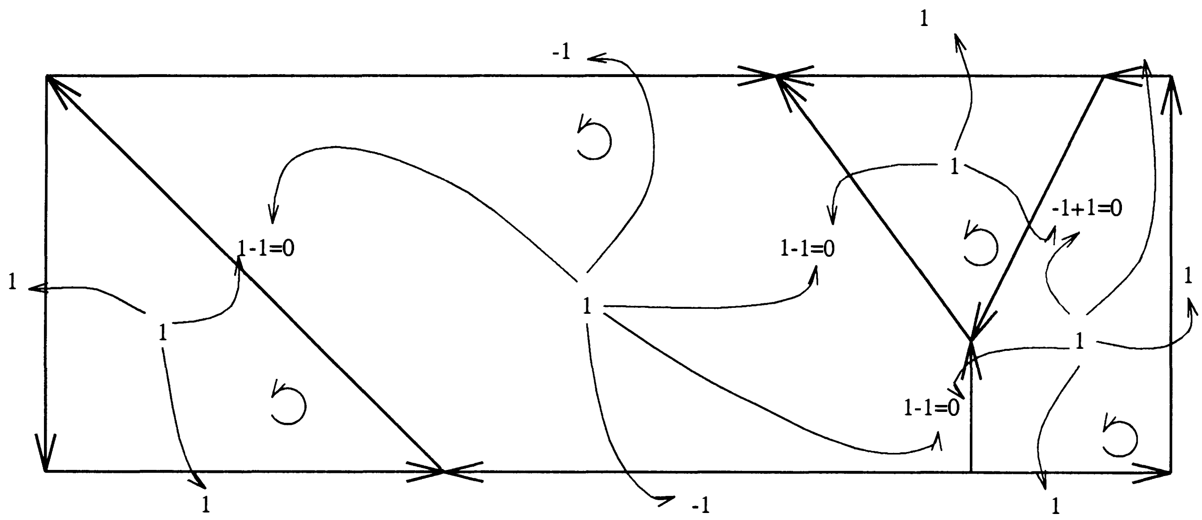


Figure 5: Chain representation of geometry.

coherently oriented, the resulting 2-chain assigns non-zero coefficients only to 2-cells that are faces of an odd number of 3-cells. In other words, all interior 2-cells in the chain will cancel out, and only geometric boundary cells of S will remain (see Figure 5). This explains the sense in which the operation of computing the geometric boundary of a solid is a special case of a more general topological boundary operator defined in Definition 11.

A p-chain whose boundary is 0 (i.e., the coefficient of every $(p-1)$-cell is 0) is called a p-cycle. Intuitively, p-cycles correspond to the class of closed curves and surfaces. For example, if a three-dimensional solid S is given as a 3-chain, its boundary is a uniquely defined 2-cycle. More generally, the boundary of every chain is also a cycle. Conversely, every oriented 2-cycle can be shown to be a boundary of some solid.[6]

3.4 Chain models of distributions of physical quantities

Suppose we wish to represent the distribution of mass in some region of space R. We first define a cell complex K that subdivides R into convenient 3-cells. Using these cells, the mass distribution is modeled by a 3-chain that associates with every cell a real number representing the total amount of mass contained in that cell. To be more precise, the chain is a map $M : 3\text{-}cells(K) \rightarrow \Re$, where $M(c_i)$ is the integral of the mass contained in the 3-cell c_i. Figure 6 illustrates a 2-chain representing a distribution of mass in a body. Such chain models are consistent with algorithms to compute mass and inertial properties of solids [LR82].

Many important physical systems are described by distributions of physical quantities that vary in time. For example, let complex K decompose a region of space containing a fluid. At any given time t, the mass distribution in K may be described by the 3-chain $M(K)$. If K is fixed in R, the time history of the mass distribution in K is a function of t. The mass in a given 3-cell c_i in K at a time t is given by $M(c_i)$, and thus the rate of change in mass is given by $\frac{d}{dt}M(c_i)$.

This changing mass distribution cannot be arbitrary, because it must satisfy the physical law of conservation of mass, which states that the change in mass in a given region results from flow

[6]Additional conditions on 2-cycles are usually imposed; see [Req77, Hof89].

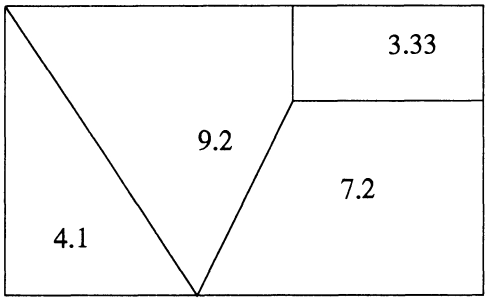


Figure 6: A 2-chain representing the mass distribution in a region of 2-space.

through its boundary. To state the law in terms of chains, we need another chain to represent this flow of mass through the boundary of every 3-cell in K. Thus we define a 2-chain $M_f(K)$, which represents the signed mass flux through each oriented 2-cell c_j of K. (That is, if c_j is a face of a 3-cell c_i, then $\sigma(c_j, c_i)M_f(c_i) > 0$ means that mass is flowing through c_j into c_i.)

The conservation of mass constraint requires that the rate of change of mass in c_i is equal to the sum of all fluxes through the boundary of c_i. In other words, if every 3-cell c_i in K contains $M(c_i)$ mass, then $\frac{d}{dt}M(c_i)$ is given by the sum

$$- \Sigma_{c_j \in faces(c_i)} \sigma(c_j, c_i) M_f(c_j). \tag{1}$$

Comparing this expression with Definition 12, the change in mass $\frac{d}{dt}M(c_i)$ in every 3-cell c_i is exactly the coboundary of the flux 2-chain bounding c_i. Thus once the 3-chain of mass and 2-chain of mass flux is defined, conservation of mass can be expressed as a simple topological relationship:

$$\frac{d}{dt}M(K) = -\delta(M_f(K)). \tag{2}$$

Mass is an example of a quantity that is naturally associated with volume in a sense that the value of the quantity depends on the *extent* of the volume in question. We may use a 3-chain to represent any other *volume extensive* quantities in a physical model (including momentum, energy, entropy, charge, and so on), and 2-chains to represent fluxes of these quantities. Distributions of all such quantities satisfy similar conservation or balance laws, such as conservation of momentum and energy. Any and all such conservation laws can be expressed in terms of the relationship between a *flux* 2-chain F and a *volume extensity* 3-chain E defined over a cell complex K:

$$\frac{d}{dt}E(K) = -\delta(F(K)). \tag{3}$$

Thus, in the example above, mass M is E and mass flux M_f is F.

It is important to note that Equation 3 is *not* an approximation; for any conserved quantity in a region R, and the complex K that decomposes R, Equation 3 must hold exactly. [7]

3.5 Properties of distributions modeled by chains

Finite decompositions of a region $R \subseteq E^n$ are never unique; the space can always be decomposed differently, and any given decomposition can be subdivided further to yield finer decompositions. This has two important implications:

[7] Which is not to say that Equation 3 can be exactly measured or computed for a given system.

- There are infinitely many finite chains corresponding to the distribution of a given physical quantity in R. (For instance, if we assume that cell complexes K are fixed with respect to some inertial coordinate frame, there is one for each K embedded in R.) If the quantities associated with a given cell c_i are exact integrals of the distribution within C_i, then there is a sense in which each such chain is an exact model of the distribution.

- Any given chain models (exactly) any number of physical distributions. These distributions form an equivalence class.

Thus the relationship between distributions of physical quantities and chains is many-to-many. We shall see in Section 5.1 that this property of chains is crucial for their role in design. But it also raises the issue of whether chains are well suited for modeling physical behavior. In particular, what is the relation between two distinct chains representing the same distribution? What does it mean to say that two distributions are the same?

When distributions of physical quantities are modeled by continuous functions, it is possible to talk about a physical behavior and state at a point. For example, in continuum mechanics it is common to study the state of stress or strain at a point, even though any physical measurement involves a finite volume of material. One advantage of such continuum model of distribution is that it is unique; that is, two distributions are the same if their values agree at every point. However, this advantage is mitigated somewhat, because it is impossible to either compute or measure continuum distributions.

A rigorous development of the relationship between traditional continuum models of physics and models defined by finite chain models is best expressed in terms of the calculus of differential forms [Ton75, BG80, BS90], which is beyond the scope of this paper. But if we assume that space and time are continuous, we may show that a finite chain model of physics is consistent with traditional continuum models in the following common (but somewhat restricted) case.

Suppose $u : \Re^3 \rightarrow \Re$ is a sufficiently smooth function that represents the distribution of a physical quantity (for example it may satisfy some partial differential equation $Lu = f$) over a region R. If a cell complex K decomposes R, we may define a chain $ch \in Ch(K, \Re, 3)$ by associating with each 3-cell $c \in K$, the integral of u in the cell. Let u_{max} and u_{min} be the maximum and the minimum values respectively of the function u inside the cell c. Defining $\epsilon = u_{max} - u_{min}$, it is clear that

$$x \in c \;\Rightarrow\; \|u(x) - ch(c)/V(c)\|_2 \leq \epsilon,$$

where $V(c)$ is the volume of c. Furthermore, given any such small ϵ and a cell complex K decomposing R, we can recursively (barycentrically) subdivide each cell in K until the variation of u in every cell does not exceed ϵ. As cells (and ϵ) get smaller and smaller, there is a precise sense in which discrete coefficients of the chains approach the values of u.

Thus, one may view the relationship between continuum models and finite chain models in one of two complimentary ways: the continuum model may be viewed as the limit of an infinite sequence of finite chain models; alternatively, a finite chain model can be viewed as a particular discretization of the continuum model.

Finally, chain models are compatible with the concept of measurement, in the following sense. We need not have a continuum model of the physical behavior of a given object – instead we may choose to perform a finite number of measurements and construct a chain model whose coefficients are determined from the physical measurements.

4 Physical Elements

4.1 Chain models of physical systems and behaviors

We have defined a physical system (object) as a collection of physical quantities distributed in space and time together with a set of laws that distributions of these quantities must satisfy. If chains are used to model distributions of physical quantities, then physical laws can be modeled as constraints on the coefficients of the chains. We have already seen in Section 3.4 that the law of conservation of mass (and other physical quantities) can be modeled by equating the coboundary of a 2-chain representing the mass flux through the 2-cells and the time derivative of the 3-chain representing the mass contained in the 3-cells. We shall discuss other collections of chains and constraints below, but we first address the question of validity of such models. Based on Definition 1, Definition 2, and the discussion in the previous section, we now redefine (mathematical models of) systems, physical systems, and behaviors in terms of chain models.

Definition 14 A *system* (a distribution of quantities in time and space) is a set of chains Q defined over a cell complex K representing (or embedded in, depending on one's point of view) time and space.

Definition 15 A *physical system* is a pair $S = (Q, C)$, where Q is a system satisfying a set C of *chain constraints* on Q.

Corresponding to Definition 3, we define (a mathematical model of) physical behavior to include all those physical systems that are subject to common physical laws.

Definition 16 A physical behavior $PB(C)$ is the set (equivalence class) of all physical systems (Q, C_i) that satisfy C, i.e., $PB(C) = \{(Q, C_i) \,|\, C_i \Rightarrow C\}$.

When defining mathematical models of real world objects, one is invariably faced with the question of how good such models are. In the context of physical behavior, it is important to know that the modeled constraints "faithfully" correspond to physical laws. Discussing the range of possible physical laws and theories is an enormous undertaking that is beyond the scope of this paper. A physical theory is "good" to the extent that it predicts or mimics the "real physical world." This in turn depends on what aspects of the real physical world (and the model) are being measured. Thus, while the appropriateness of models of physics is a question of fundamental importance, we shall not attempt to comment on what physical theories are appropriate, but rather, we give examples of the physical theories that can be represented using finite chain models.

The discussion in the last section indicates that an integral statement can be replaced by a collection of chains whose coefficients correspond to evaluated integrals of the corresponding integrable functions over the cells. This fact, together with the fact that chain models are capable of representing measured behavior directly, suggest that there are "good" chain models for the vast majority of engineering problems. Of course, chain coefficients can be constructed in many other ways as well. For instance, they could be polynomials specifying the embedding of the corresponding cells in Euclidean space.

For the reader interested in the structure of physical theories, there are many classical writings in the philosophy of physics (e.g., [Mar77]), as well as an extensive literature describing analogies between various physical domains [Kro45, Ols43, Pay61, Bra66, Ton75, Fra85]. Notably, Tonti[Ton75] describes a classification scheme for physical theories in algebraic-topological format, representing them in terms of canonical graphical schema. The main components of these schema are dual quantities (configuration and source), "structural" equations (conservation, balance, etc.), and constitutive or phenomenological equations.

Definition 16 provides a definition of physical behavior. We now use the chain operations described in Section 3.2, together with the combinatorial properties of cell complexes, to create a model of physical behavior that is computational and can be used for constructing chain models.

4.2 Constraint Elements

Definition 16 can be interpreted as stating that if a given physical system in a region R exhibits a behavior C, then the chains Q corresponding to any and all spatial decompositions of R will satisfy C. The key to making this definition computational is in finding a way to express the physical constraints without explicitly referring to all possible chains. This is possible under the following conditions:

- Constraints can be imposed only on chain coefficients associated with incident cells or adjacent cells; and
- All cells in the decomposition of space are similar in the sense of being able to "implement" the specified constraints.

It is commonly accepted (e.g., [Ton75, Bra66, Fra85]) that most physical laws, and therefore constraints on chain models, fall into two broad categories:

1. *Structural* laws (conservation, balance, equilibrium), which are based on topological invariants and can be expressed using operations of boundary and coboundary; clearly, these operations constrain incident cells.
2. *Constitutive* laws (such Ohm's and Hook's), that represent phenomenological (macro) constraints corresponding to material properties; these are obtained and defined by local measurements.

In both cases, all such physical laws are formulated locally and can be formulated as constraints on incident or adjacent cells. Structural laws are pure topological statements that apply to all cells in all decompositions, and do not depend on how cells are embedded or what their shape is. Constitutive laws depend on ability to measure, and hence on the metric of the space, as well as on the cell types.

We may use the structural and constitutive laws to define computational devices called *constraint elements,* each of which is defined in terms of classes of chains and cells. For instance, we shall see that structural laws are independent of cell types, while constitutive laws are expressed in terms of a particular type of cell. This allows us to define chain models of physics that are independent of any particular cell complex K.

Rather than attempt to be exhaustive, we consider a few important examples of constraint elements, corresponding to the two categories of the physical laws.

1. When a structural constraint element relates coboundary (boundary) of a chain ch_1 to chain ch_2, it is sufficient to represent the chain ch_1 using a single cell, and chain ch_2 using the faces of ch_1.

   For example, Figure 7 shows a *conservation* (constraint) element. As described in Section 3.4, this constraint has the form $\frac{d}{dt}\textit{3-chain} = \delta(\textit{2-chain})$. Any one 3-cell and 2-cells that are faces of this 3-cells is sufficient to model that statement.

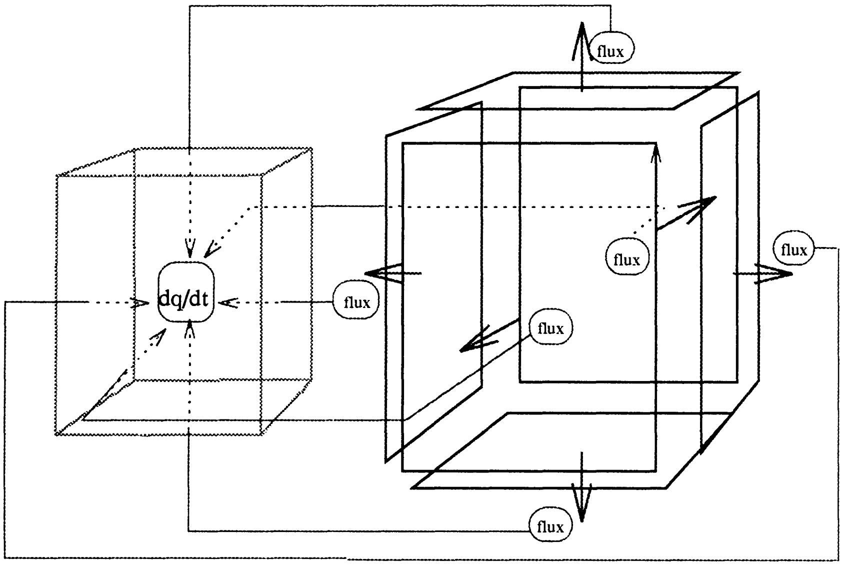


Figure 7: A constraint element for conservation

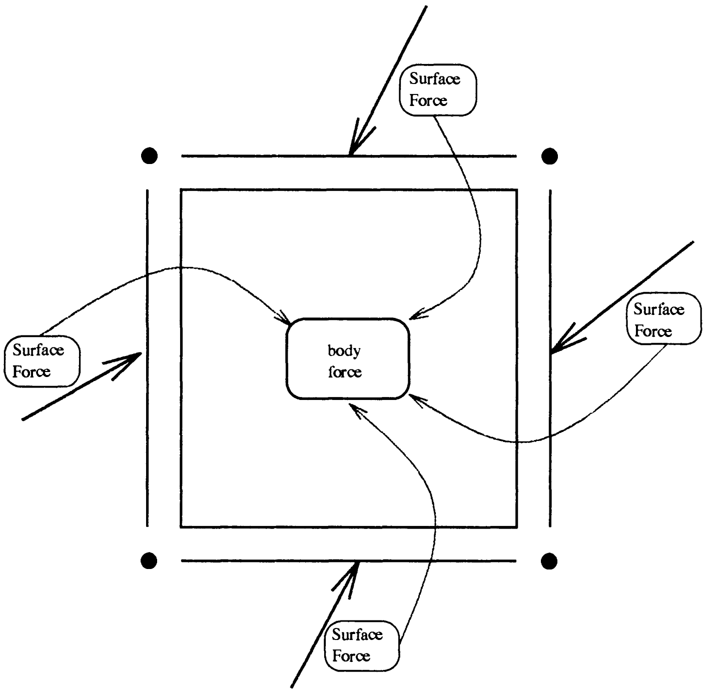


Figure 8: A constraint element for balance

In a similar fashion, Figure 8 illustrates a *balance* element. In this case, K is a 2-complex. If sf represents the force through the 1-cells of K, and bf represents the body force associated with the 2-cells, then $bf = -\delta(sf)$ is an equilibrium constraint for K.

2. Because constitutive constraint elements represent the behavior of the system *within* a region of space (i.e., in the interior of some cell), they require a pattern cell complex, with which chains may be associated. Constraints may then be specified in terms of particular (prototype) cells. For example, a Hook (constraint) element may be defined to relate the force through a set of faces of a cell to the deformation of that cell. In this case, we choose a particular type of cell, say a simplex or a cube, and represent this constraint in terms of constraints on the chain coefficients on the individual faces. Section 4.4 describes this process in detail.

It is not a coincidence that the pictures depicting constraint elements are reminiscent of those found in standard engineering texts for the purpose of deriving differential and integral relationships. There is a close relationship between the operators of vector calculus (such as gradient, divergence, curl) and operations of boundary and coboundary [Bra66, Ton75], which is why a variety of other structural constraint elements can be defined in a straightforward fashion. These include compatibility conditions, circuital laws, various versions of Stokes theorem, and so on.

Constraint elements are specified abstractly, in the sense that they define relationships between chains without reference to particular cells, in the same way that tensorial relationships are specified without considering any particular coordinate systems. Analogously to tensorial representations, this allows physical behavior to be specified in a very abstract sense – laws may be defined independently of a particular choice of cell complex.

Constraint elements are primitive computational building blocks, from which models of physical systems and behaviors can be constructed. The application of a constraint element to a complex K produces appropriate chains that must satisfy the specified constraint. On the other hand, a constraint element may or may not correspond to a complete model of physical behavior, and it may not be obvious how a given collection of constraint elements should be interpreted to obtain a complete model of physical behavior. This is the subject of the following section. It is perhaps worth remarking that the ability to formulate, apply, and combine the constraint elements depends entirely on the primitive chain operations defined in Section 3.2: boundary, coboundary, addition, multiplication by a scalar, function application, and chain equality.

4.3 Physical elements

We now have all the ingredients needed to define computational models of physical behavior called *physical elements.* In essence, a physical element is a definition of the behavior in a single n-cell of a physical system, which can be applied to each n-cell in a region decomposed by a cell complex K. Accordingly, it is the most primitive "complete" chain model of a physical behavior in the sense of the following definition.

Definition 17 A *physical element* is

1. The cell complex K generated by a single n-cell c_n of a particular cell type (e.g, simplex, cube).

2. A set of p-chains on K that represent physical state.

3. A set of constraints elements defining the constraints on the p-chains of K.

The cell c_n (together with the cell complex it generates) serves as a *prototype*; it is used for declaring the type of p-chains and allows definition of the structural and constitutive constraints. The relationships defined in the physical element can be *applied* to any cell of the same type.

Intuitively, a physical element captures all knowledge about the physical behavior in a given simple region of space (i.e., in a cell) in such a way that the behavior can be operated on computationally (without human intervention). Once a physical element has been defined, it becomes possible to create models for regions of space representable by the kind of cells used in the element. To represent physical behavior in more complex regions, it may be necessary to combine the single-cell models. (Even if the space in which a physical element is to be applied is homeomorphic to a n-ball, the space may need to be subdivided in order for the model to predict physical behavior satisfactorily.) Computationally, the only requirement for applying a physical element models to two adjacent cells is that their chains be identified appropriately.

It is possible to operate on physical elements themselves. For instance, a physical element that represents the general Navier-Stokes equations for fluid flow can be operated on (through additional constraints) to create, for instance, a lubrication model. Similarly, we may create a "beam" element from a more general elastic element. Simplification, modification, and combination of physics models by operating directly on physical element representations makes the relationship between the models explicit. One use of this technology is to create "physics editors" relying on a graphical user interface (and not just symbolic mathematics) to create and modify computational models of physics.

Physical elements can be viewed as "object oriented" components for building computational models of physical systems. They are object oriented in the sense that 1) physical behavior is defined in terms of a few definitions (like the notion of "class" in object oriented programming) which may then be "instantiated" by applying to regions of space, 2) they interact through predefined, well defined interfaces, which simplifies working with them – previously defined models of physics (say of fluid flow) need not be redefined when a new physical element (say of elasticity) is introduced. In fact, once each of these types of element has been defined, we may represent systems that contain interactions between elastic solids and fluids without introducing additional elements.

4.4 Example: a physical element to represent elastic solid behavior

Recall that linear elasticity is a statics problem, and is based on the assumption that force distribution in a body is dependent only on the deformation from some initial "rest state," and thus that in the absence of external forces, the body will return to this initial state. The continuum formulation (for example, see [Lov27]) of elasticity expresses the behavior in terms of four basic quantities: displacement u, strain ξ , force f, and stress σ. The strain tensor defines the local deformation properties of a body at a point, while the stress tensor represents forces acting through infinitesimal planes passing through a point. These four quantities are related by three basic physical laws: kinematics conditions relating displacement and strain, equilibrium between body force and stress, and a constitutive relation (generalized Hook's law) relating strain and stress.

In this section we develop our model of elastic behavior in a finite region of space by defining relationships between quantities corresponding to measurements. This chain model is consistent with the continuum model in the sense that as the sizes of its cells approach zero, the chain model approaches the continuum model. It should be clear that there is no "correct" physical element for elasticity – the best choice for a given situation depends on the physical behavior being modeled, as well as the computational resources available. Even if the test of "correctness" is that the model converge to some given continuum model (as opposed to modeling some given measured behavior well), there is a variety of physical element models that will satisfy this criterion.

From Definition 17, we know that to define a physical element, we must choose a cell type, define a set of chains, and specify the behavior of the element by constructing an appropriate set of constraint elements (Section 4.2).

4.4.1 A prototype cell for elasticity

Figure 9 shows our choice of cell for this example. In this case we have one 3-cell, marked $b0$, which has six faces, 2-cells marked $s0, \ldots s5$. These 2-cells in turn have 12 1-cells as their faces, which are marked $e0, \ldots, e11$. Finally, there are eight 0-cells, labeled $v0, \ldots, v7$.

4.4.2 Chain quantities for elasticity

The chains required for our chain model of elasticity (illustrated in Figure 10) are as follows:

1. A 0-chain u represents the displacement of the 0-cells of K from the initial *rest state*. So $u \in Ch(K, \Re^3, 0)$, that is, the coefficients of u are elements of $\Re^3$.

2. A 1-chain $ud \in Ch(K, \Re^3, 1)$, which represents the change in u along the 1-cells of K. (The symbol "ud" is a mnemonic for "u difference.") The coefficients of ud are in $\Re^3$.

3. A 2-chain sf, which represents the resultant of the force acting through the 2-cells of K. This chain models *contact forces* – local force interactions between adjacent cells. The domain of sf is a six dimensional vector that represents both the translational and rotational components of the force on a body. We may view the range of sf as being a six dimensional space representing the three components of translation and three components of the moment, which we may refer to (in non-standard notation) as G^6.

4. A 3-chain $bf \in Ch(K, G^6, 3)$, which represents the generalized *body force* acting on (the contents of) the 3-cells of K. The body force models interactions that occur "at a distance," i.e., any force that is not viewed as a "contact force." Typical examples are gravitational and electrostatic forces. The range of bf is also G^6. (The moments of bf are often zero, for instance for small regions in a gravitational field. However, since we are defining models for finite regions of space, the body force can include moments.)

4.4.3 Constraint elements for elasticity

Once we have chosen the prototype cell and defined the set of chains, we define the constraint elements. In this problem, there are two structural constraints:

The first relates the surface forces to the body force: an equilibrium constraint: $bf = -\delta(sf)$. This is the chain formulation of the statement that for a body in equilibrium, the sum of the forces on the body must be zero. In this case we have partitioned all of the forces acting on the body into two classes, those acting locally, sf, and those acting at a distance, bf, and stated that the sum of these chains is zero.

The second structural constraint element defines the relationship between the displacement u and the relative displacement ud, which is defined to be $ud = \delta(u)$. Thus ud represents the amount that the displacement u changes along the 1-cells of K.

Finally, we define the constitutive relation between ud and sf. In theory, the relationship between ud and sf can be defined by any set of constraints that completely constrain them. Of course, our goal is to represent physical behavior, which constrains our choices.

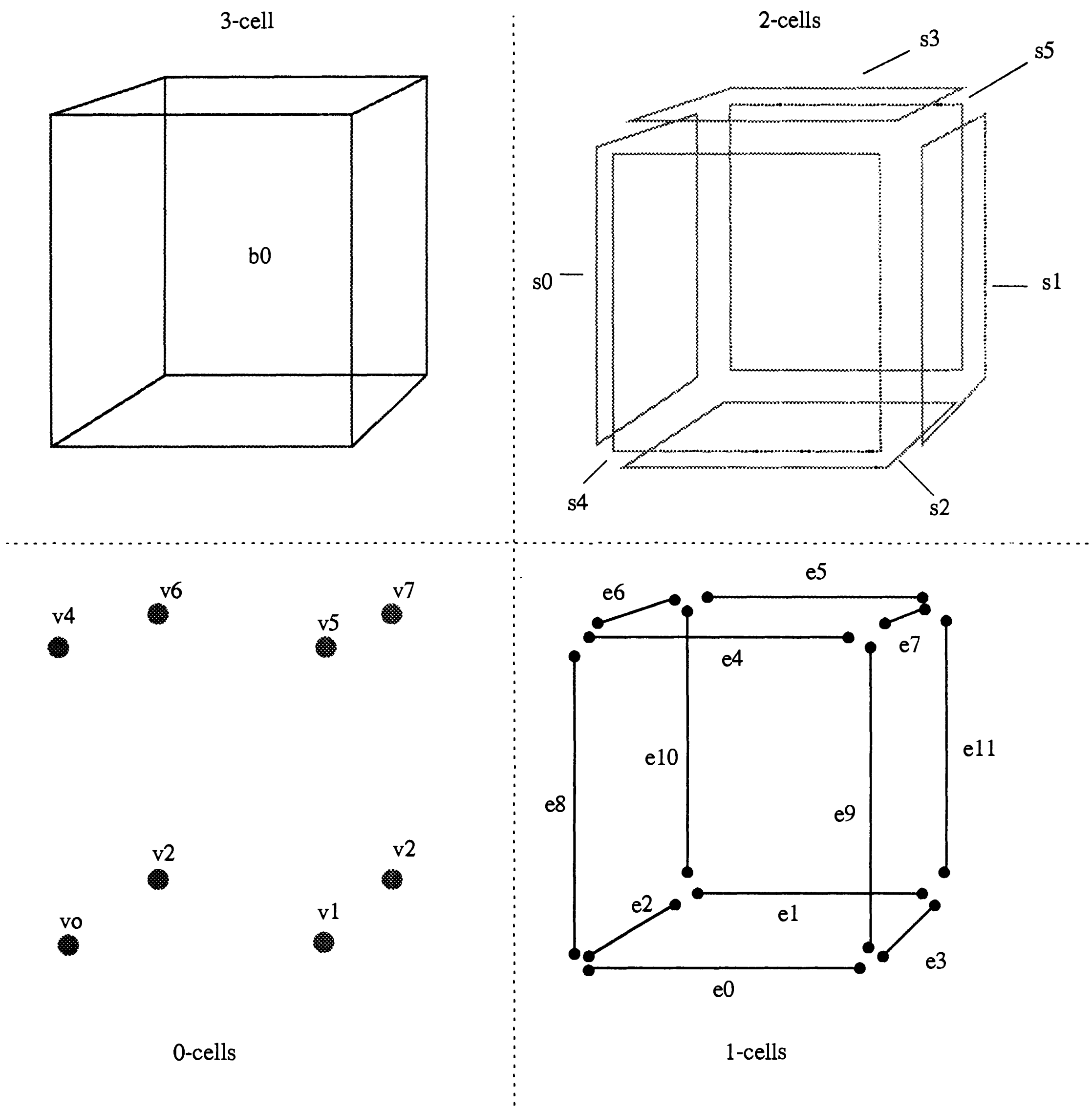


Figure 9: The complex generated by a single 3-cube.

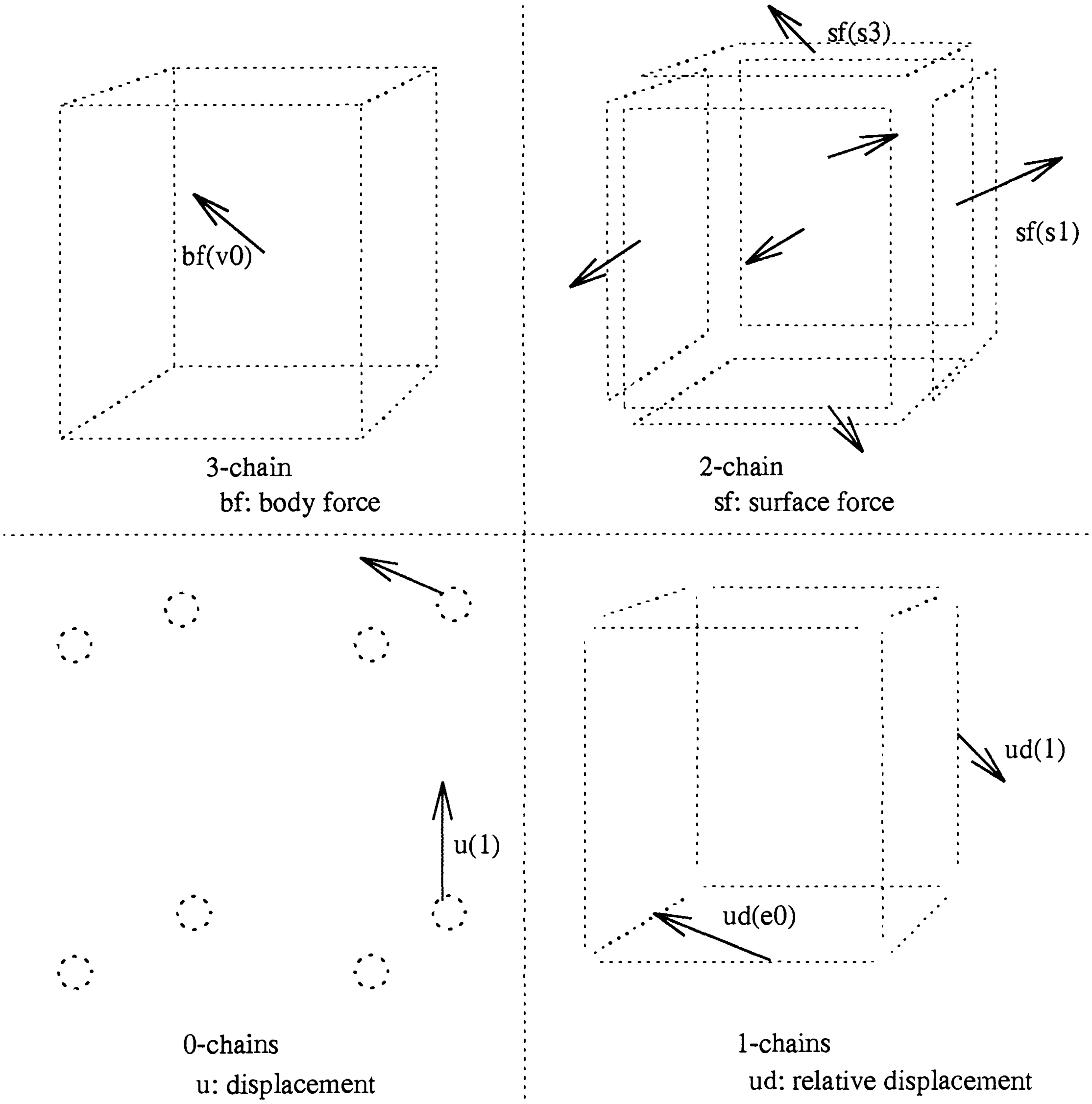


Figure 10: The chain quantities associated with a physical element for elasticity

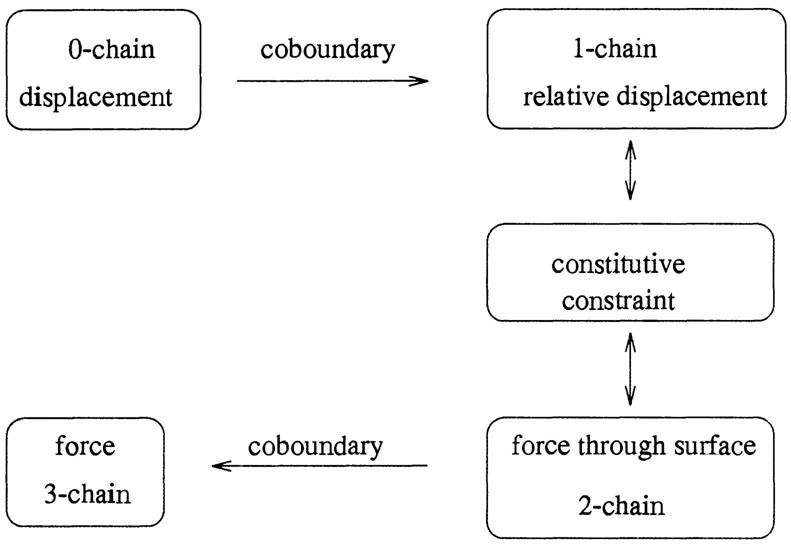


Figure 11: Relationships between the chains in chain model of elasticity.

So let us consider the number of degrees of freedom of ud and sf. K contains six 2-cells, and the range of sf is G^6, which is six dimensional, so there are $6 \times 6 = 36$ for sf. On the other hand, K contains twelve 1-cells, and the range of ud is $\Re^3$, so there are $12 \times 3 = 36$ degrees of freedom in ud.

There are 36 degrees of freedom in each of ud and sf. Therefore the constitutive constraint element for our three dimensional elasticity element can be represented by a matrix D, of dimension 36×36 . In practice, because of symmetry, additional assumptions, and other physical considerations the number of independent parameters defining D is usually much smaller. Linear elastic models of homogeneous isotropic materials need only two independent parameters to completely determine D. (For instance, E, the modulus of elasticity in tension, and ν, the Poisson's ratio.) The metric properties of the problem enter the constitutive element as well.

4.4.4 Discussion of the example

The relationships between the various chains in this model of elasticity are illustrated in Figure 11. Classical models of elasticity (infinitesimal, homogeneous, isotropic, no moments, etc.) can be derived from this physical element by making additional assumptions found in the corresponding theories, and letting the 3-cell shrink to an infinitesimal element.

There is a close relationship between measurement and the number of independent parameters in the constitutive constraint element. In essence, we need one measurement for each independent parameter. The cells used to represent this relationship also imply which measurements are modeled.

In the above example, the fact that we are using a 1-chain to describe the state of deformation implies that we are measuring the relative displacement between two points. Similarly, using a 2-chain for sf implies that we are capable of applying a (generalized) force to a surface. Or turning the problem around, we can apply deformation by changing the relative displacement of a discrete set of (12) pairs of points, and measure the resulting force across the (6) surfaces of the cube. Thus the ability to define *any* constitutive relation implies the ability to relate the stimuli and response.

Of course this doesn't guarantee that we can extrapolate the behavior we've measured in a given case to any *other* problem. However, it is clear that once a method of measuring a system has been chosen, the structural aspects of the constitutive relation are completely determined.

5 Applications of chain models

The definitions of chain models in this paper provide a formal means of specifying, transforming, manipulating, and computing with physical systems. They point the way to seamless integration of physical and geometric information, and have applications in many areas of computer-aided engineering analysis and design.

It seems clear to us that most engineering computations can be formulated in terms of chain models. If the relationship between chain models and finite element analysis is apparent (a given finite element may be viewed as a constitutive constraint element), the connection to synthesis and optimization of shape may be less so. We argue below that the chain models and the physical elements provide a basis for formal description of both (physical) functionality and spatial embedding (i.e. shape). More generally, chain models may be used to create a language for specifying, building, subcontracting, buying and selling engineering artifacts (i.e., products).

5.1 Chain models in design

In practice, formal design specifications are rarely available. When they are available, the specifications are rarely fixed; they constantly change and are subject to numerous manufacturing, service, cost, time, and other constraints. For illustrative purposes, we focus on a particular aspect of design; we assume that the purpose of design is to synthesize a shape (domain) that interacts with the predetermined environment in a specified manner.

A design specification must capture the physical behavior desired of the artifact being designed. Given our definition of physical behavior (Definition 16), a design specification can be expressed as a set of physical constraints on finite chains. The goal of design is then to find an (embedded) cell complex K and a set of chains on K that satisfy these constraints, perhaps in some optimal sense. Below, we use the two-dimensional bracket example from [SV89] to show how chain models can be used to formalize the specification of a simple synthesis problem, and how different solutions to this problem might be expressed in terms of the chain model formalism.

5.1.1 Formal functional specification

Informally, the functionality of a bracket is usually described by specifying how the bracket should interface with mating parts as well as its strength characteristics. Figure 12 illustrates such a specification for a simple bracket whose interfaces are the three holes of given diameters, and whose mechanical behavior is defined by specifying the relationship between the applied forces on and the displacements of the holes. While such design specifications are common in literature [SV89, SG92, MCKP91], they are rarely accompanied by formal specification (or computer representation) of their semantics. The formal semantics of such a design specification can be represented in a straightforward fashion using chain models. For the purposes of illustration, we will make a number of (somewhat arbitrary) assumptions and then describe how they may be represented in terms of chains.

At the conceptual stage of design, one may assume that the holes undergo rigid body motions and remain undeformed, while the position, applied forces, and displacement of each hole is associated with the hole's center. Figure 13 illustrates a chain model of such a specification. Each hole

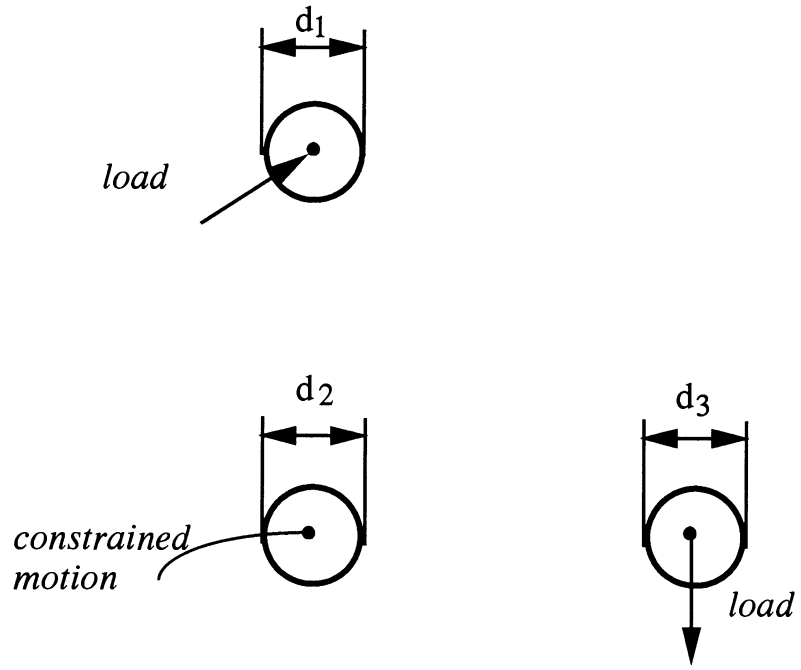


Figure 12: Simple bracket specification.

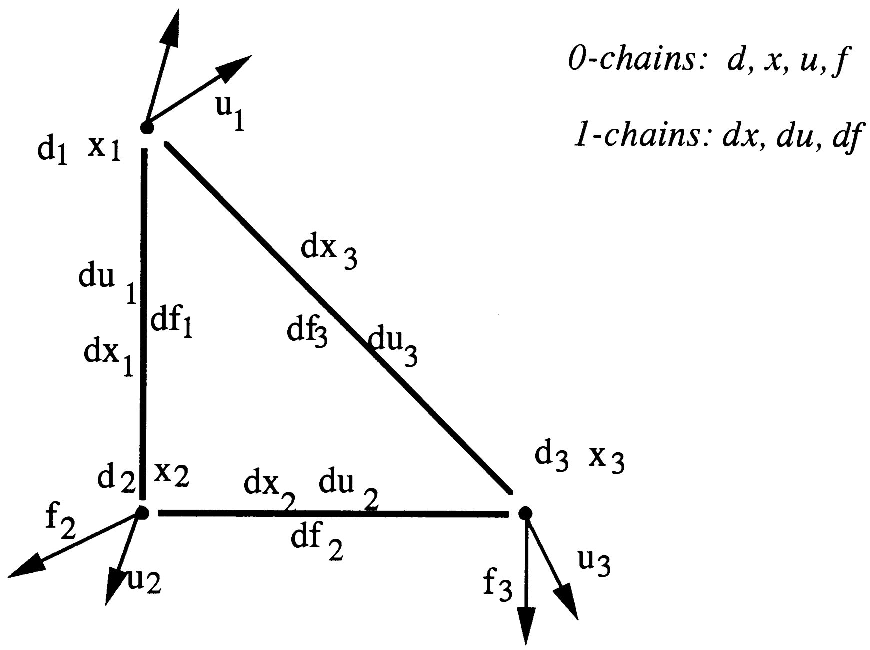


Figure 13: A abstract chain model bracket specification.

is represented by a 0-cell (a node), giving raise to a number of 0-chains: hole positions x, forces f, displacements of holes u. The 0-cells can be used to define an *abstract* cell complex by defining a 1-cell for each pair of 0-cells. Using the coboundary operator, we induce three 1-chains: relative position $dx = \delta(x)$, relative displacement $du = \delta(u)$, and relative force $df = \delta(f)$. Clearly, this chain model does not represent the geometry (or material properties) of any given solution to the bracket problem. Rather, it can be used to represent the functionality of the bracket as a *class* of solutions.

In order to define the class of chains (corresponding to the behavior of the bracket), we define constraints on the values of the chains in the specification. To be consistent with the requirement that boundary value problems be well posed, it is a common practice to prescribe specific values of either forces or displacement (but not both) at the interfaces [SG92, MCKP91] and perhaps put bounds on other physical quantities of interest. Clearly we can impose similar conditions on chain coefficients in Figure 13, but the proposed chain model can be used to specify more general constraints useful in design. For instance, we may specify any consistent relationship between the chains of forces f, relative forces df, displacements u, and relative displacements du. One can think of this relationship as a "generalized constitutive relation" which may be closely tied to the notion of measurement. Given any real physical object, we may choose a set of measurements that determine the type of cells and the abstract complex (e.g., measuring at a point defines a 0-chain, measuring the difference between two points defines a 1-chain, etc.). We may then perform a set of measurements, and use these to define a constitutive relation between the chains. As an illustration, we may specify that the absolute value of the dot product of the relative displacement du and relative force df is less than some bound, which may also be expressed as a 1-chain. This would result in a simple set of linear inequalities, but any testable set of inequalities could be used. In this example, the generalized constitutive relations are static, but they could also specify time-dependent constraints that are similar to those modeled by bond graphs [Pay61]. We might also require that forces f must satisfy the equilibrium condition, that relative forces df do not exceed some threshold value, and so on.

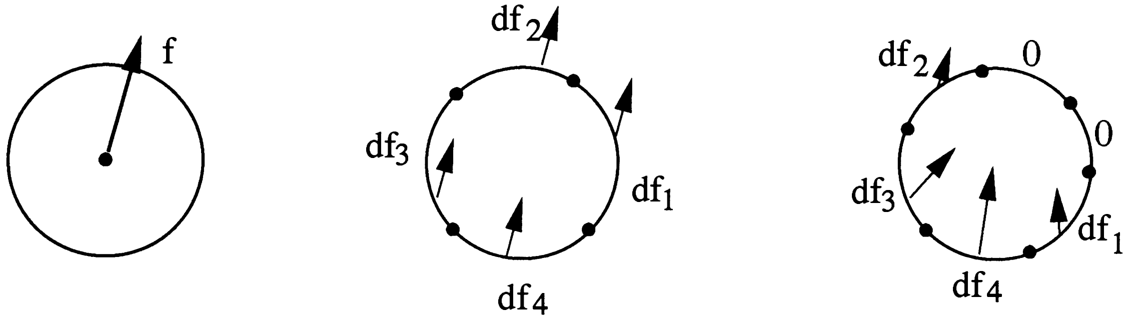
5.2 From function to form

The abstract chain model in Figure 13 specifies the relationship between idealized forces and displacements, and may be appropriate at early stages of design. It assumes (but does not specify) the geometry of the holes and does not model the detailed geometry of the bracket. This abstract chain model can be systematically mapped into a more detailed chain model that is geometrically embedded in space.

In this case, the geometry of each two-dimensional hole is represented by a 1-complex which will represent a subregion of the bracket's boundary. The geometry of bracket itself will be represented by a 2-complex, and the goal of synthesis is to construct and embed such a 2-complex consistent with the functional specification above. For the sake of illustration, we shall assume that the bracket is to be constructed of elastic material whose behavior has already been modeled by a two-dimensional [8] cubical physical element such as described in Section 4.4. Thus, as the first step, we need to relate the 0-chains associated with holes in the abstract chain model to the 0-chains and 1-chains associated with hole boundaries. The nature of this relationship is determined by the specifics of the problem; Figure 14 illustrates some possibilities.

We have used chain models to formally specify the bracket's function, and to translate that specification into a partial specification of the bracket's geometry. The solution to the synthesis

[8] Thus, surface forces sf are modeled by 1-chains and body forces are represented by 2-chains.

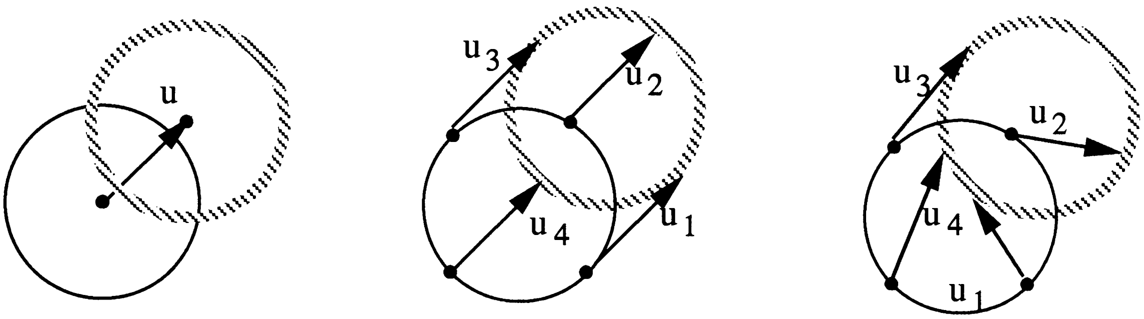


Concentrated force f *at the center of the hole*

Force df *is uniformly distributed on the surface of the hole*

Force df *is distributed based on surface area and and direction of contact*

$$f = df_1 + df_2 + df_3 + df_4 = \text{coboundary (df)}$$



Displacement u *of the center of the hole implies rigid body motion*

The hole undergoes rigid body translation
$u = u_1 = u_2 = u_3 = u_4$

The hole undergoes translation and rotation
$u_i = u + (r\cos\theta, r\sin\theta)$

Figure 14: Relationships between abstract and spatially distributed chains.

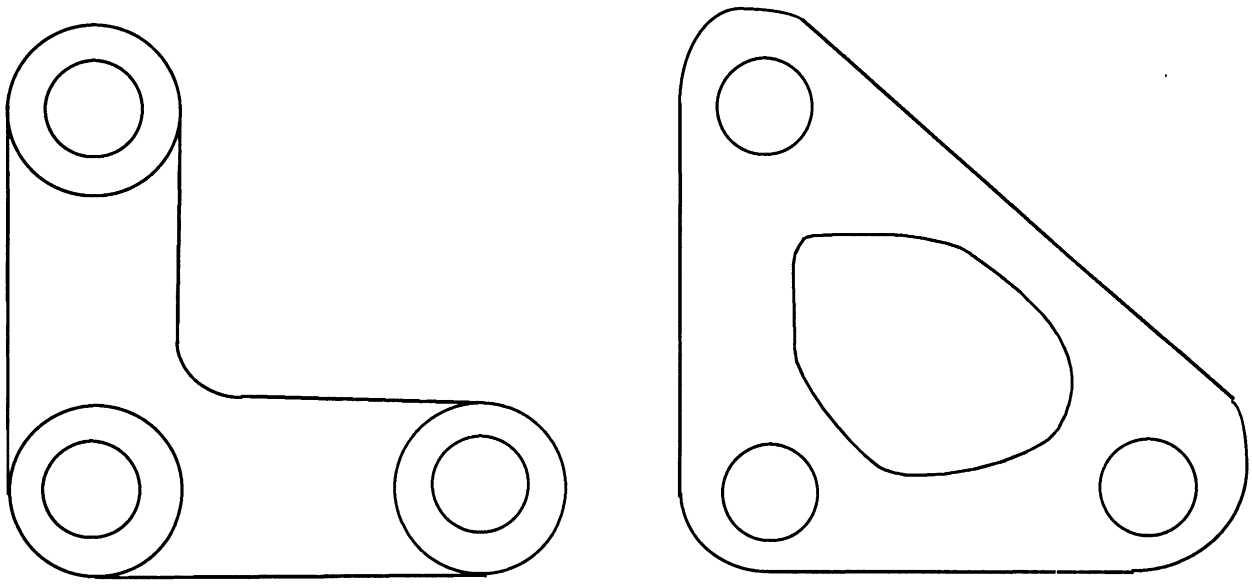


Figure 15: Bracket designs resulting from different design paradigms.

problem consists of an embedded cell complex (defining the shape of rest of the bracket) together with chains defined over it, which represent physical behavior. If the physical elements used to define the chain model represent the behavior of some particular material, the chain model will both satisfy the design specification and have a physical realization.

As discussed in [SV89], solutions to such design problems are not unique, by the very nature of the synthesis problem. Several different approaches have been described in the literature, and Figure 15 illustrates some possible chain models of the completed bracket. In [SG92], it is suggested that the holes can be connected by "support regions" whose size is proportional to the applied forces. In [MCKP91], the synthesis is completed by filling all available space surrounding the holes and performing shape optimization. These methods are likely to result in distinctly different bracket designs. Formulating such methods in terms of operations on chain models should lead to a better understanding, formalization, and unification of various synthesis techniques.

5.2.1 Other uses in design

The above example illustrates how the repeated and systematic application of chain model methodology can provide a mathematically well defined, intuitive means for an individual to define models appropriate to their specific task. We do not suggest that the choices made here are in any sense optimal for any given task.

There are other design scenarios and tools in which chain models could be systematically used. Thus the central purpose of this work is not to automate the process of design, but rather to provide a uniform framework in which automated procedures (e.g., shape optimization, lumped parameter analysis) may be freely mixed with human engineering knowledge (e.g., "this worked well on that project we did four years ago") to improve the design process. Chain models may also provide a means of defining engineering features in terms of their physical behavior.

5.3 Analysis and simulation using chain models

One of the primary motivations in the development of this theory has been the desire to define a computer language for engineering physics – a language whose expressions correspond directly to real world physical objects. The SIMLAB[PC91] simulator generation system defines such a language for lumped parameter physical systems (i.e., those that can be represented by ordinary differential equations such as n-body problems, rigid body dynamics, electrical circuits, etc.), and creates simulators directly from computer representations of "primitive" elements and the physical interactions between them.

We believe that chain models could provide a means of expanding the coverage of SIMLAB to distributed parameter systems. This would allow automatic construction of simulation software for a variety of mixed domain phenomena, such as a fluid flow interacting with an elastic solid which is in turn positioned by a (lumped modeled) hydraulic actuator. Because each of the phenomena can be expressed in terms of chain models, and because chain models can be combined automatically (we have not described how to do this here), we are moving towards a truly "object oriented" system for defining simulation and, more generally, for specifying physical systems. In this case, the "objects" are physical elements: models of physical behavior, which can be applied to various regions of space, and "know" how to interact with other physical phenomena occurring in either the same or an adjacent cell.

Finally, chain models provide a path to parallel and distributed computation. Because the interactions in chain models are locally defined, and because these interactions are explicitly represented in the combinatorial structure, this structure can be taken advantage of in constructing parallel and distributed software to simulate (or otherwise operate on) chain models. This assumes paramount significance, as the ability to parallelize computation is the key to solving large problems.

5.4 Communication and standardization

Another potential use of chain models is as a common language for communicating engineering and scientific knowledge. A standardized chain model language could form the core of a true *product model* language useful in both the commercial and engineering worlds: for instance, a chain model could serve as part of a contractual specification to build a product. The product model would form both the definition of what product is to be built, and the validation mechanism for determining whether a product satisfies the design criteria. Chain specifications could also provide a means of "hiding" information about the product, which is important when competitors cooperate in joint contracts. Information can be hidden precisely because it is possible to specify the "generalized constitutive behavior" of a product in terms of relationships between chains, which in and of themselves offer no means of constructing an object that satisfies the constraint. (For example, there is not generally enough information in a functional description of the input/output behavior of an analog electrical circuit to provide a means of creating a circuit with that behavior.) Thus it would become possible for competitors to limit the exchange of information to *exactly* that required for interaction.

A common language allows information to flow freely, and such a language for representing physical objects and behavior would allow engineering designs and specifications to be transmitted, stored, and retrieved. Beyond this, a specification language with an executable semantics allows these same specifications to take the place of real physical objects in a variety of engineering applications. Chain models, because they unite geometry and physical behavior, are one form such a language could take.

References

[Arb90] F. Arbab. Set models and boolean operations for solids and assemblies. *IEEE Computer Graphics and Applications*, pages 76–86, November 1990.

[BG80] Richard L. Bishop and Samuel I. Goldberg. *Tensor Analysis on Manifolds*. Dover, New York, 1980.

[Bra66] 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.

[BS90] Paul Bamberg and Schlomo Sternberg. *A Course in Mathematics for Students of Physics*. Cambridge, Cambridge, England, 1988,1990.

[Bur85] William L. Burke. *Applied Differential Geometry*. Cambridge, Cambridge, England, 1985.

[CA91] J. J. Cox and D. C. Anderson. Single model formulation that link engineering analysis with geometric modeling. In J. Turner, J. Pegna, and M. Wozny, editors, *Product Modeling for Computer-Aided Design and Manufacturing*. Elsevier Science Publishers, North-Holland, 1991.

[CH89] Richard Courant and David Hilbert. *Methods of Mathematical Physics*. Wiley, New York, 1989.

[CME89] R.D. Cook, D. S. Malkus, and Plesha M. E. *Concepts and Applications of Finite Element Analysis*. John Wiley and Sons, third edition, 1989.

[DD88] M. R. Duffey and J. R. Dixon. Automating extrusion design: A case study in geometric and topological reasoning for mechanical design. *Computer-Aided Design*, 20(10):589–596, 1988.

[DLH90] J. R. Dixon, E. C. Libardi, and Nielsen E. H. Unresolved research issues in development of design-with-features systems. In M. Wozny, J. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 183–196. North Holland, 1990.

[EvD74] F. J. Evans and J. J. van Dixhoorn. Towards more physical structure in systems theory. In F. J. Evans and J. J. van Dixhoorn, editors, *Physical Structure in Systems Theory*, pages 1 – 15, London, 1974. Academic Press.

[FD89] S. Finger and J. A. Dixon. A review of research in mechanical engineering design. *Research in Engineering Design*, 1(1 and 2), 1989. Parts I and II.

[Fra85] O. I. Franksen. The nature of data – from measurements to systems. *Bit*, 25:25–50, 1985.

[GCP90] E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. Wozny, J. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 107–130. North Holland, 1990.

[GM93] Jin-Kang Gui and Marti Mantyla. New concepts for complete product assembly modeling. In J. Rossignac, J. Turner, and G. Allen, editors, *Second Symposium on SOlid Modeling and Applications*, pages 397–406, Montreal, Canada, May 19-21, 1993.

[Hen93] M. R Henderson. Representing functionality and design intent in product models. In J. Rossignac, J. Turner, and G. Allen, editors, *Second Symposium on SOlid Modeling and Applications*, pages 387–396, Montreal, Canada, May 19-21, 1993.

[Hof89] C. H. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann Publishers, 1989.

[Kro39] Gabriel Kron. *Tensor analysis of networks.* Wiley, New York, 1939.

[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.

[Kro59] 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.

[Lov27] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity.* Cambridge University Press, New York, 1927.

[LR82] Y. T. Lee and A. A. G. Requicha. Algorithms for computing the volume and other integral properties of solids: Parts i and II. *Communications of the ACM*, 25(9):635–650, 1982.

[Mar77] H. Margenau. *The Nature of Physical Reality.* Ox Bow Press, Woodbridge, Connecticut, 1977.

[MCKP91] Bremicker M, M. Chirehdast, N. Kikuchi, and P.Y. Papalambros. Integrated topology and shape optimization in structural design. *Mechnical Structures and Machines*, 19(4):551 – 587, 1991.

[Mun84] James R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, New York, 1984.

[Nic25] C. A. Nickle. Oscillographic solution of electro-mechanical systems. *Transactions of the American Institute of Electrical Engineers*, 44:844–856, 1925.

[Ols43] H. F. Olson. *Dynamical Analogies.* Van Nostrand, Princeton, 1943.

[Pay61] Henry M. Paynter. *Analysis and Design of Engineering Systems.* The M.I.T. Press, Cambridge, Massachusetts, 1961.

[PC91] Richard S. Palmer and James F. Cremer. Simlab: Automatically creating physical systems simulators. Computer Science Technical Report TR91–1246, Cornell University, November 1991.

[Req77] A. A. G. Requicha. Mathematical models of rigid solid objects. Tech. Memo 28, Production Automation Project, University of Rochesteri, Rochester, NY, November 1977.

[RO90] J. R. Rossignac and M. A. O'Connor. SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In M. Wozny, J. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 145–180. North Holland, 1990.

[RV81] A. A. G. Requicha and H. B. Voelcker. An introduction to geometric modeling and its applications in mechanical design and production. In J. T. Tou, editor, *Advances in Information Systems Science, Vol. 8.* Plenum Publishing, 1981.

[RV82] A. A. G. Requicha and H. B. Voelcker. Solid modeling: A historical summary and contemporary assessment. *IEEE Computer Graphics and Applications*, pages 9–24, March 1982.

[SG92] K. Shimada and D. Gossard. Automated shape generation of components in mechanical assemblies. In *ASME Design Automation Conference*, Tempe, AZ, September 1992.

[Sha91a] Jami J. Shah. Conceptual development of form features and feature modelers. *Research in Engineering Design*, 2:93–108, 1991.

[Sha91b] V. Shapiro. *Representations of Semi-Algebraic Sets in Finite Algebras Generated by Space Decompositions.* PhD thesis, Cornell University, Cornell Programmable Automation, Ithaca, NY, 14853, February 1991.

[SV89] V. Shapiro and H. B. Voelcker. On role of geometry in mechanical design. *Research in Engineering Design*, 1(1):69–73, 1989.

[TO93] W. W. Tworzydlo and J. T. Oden. Towards an automated environment in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 104:87 – 143, 1993.

[Ton75] Enzo Tonti. *On the Formal Structure of Physical Theories.* Istituto Di Matematica Del Politecnico Di Milano, Milan, 1975.

[US89] K. T. Ulrich and W. P. Seering. Synthesis of schematic descriptions in mechanical design. *Research in Engineering Design*, 1(1):3 – 18, 1989.

[Voe88] H. B. Voelcker. Modeling in the design process. In W. Dale Compton, editor, *Design and Analysis of Integrated Manufacturing Systems.* National Academy Press, Washington, D.C., 1988.

[Wei86] K. J. Weiler. *Topological Structures for Geometric Modeling.* PhD thesis, Rensselaer Polytechnic Institute, 1986.

[WTC93] T. C. Wilson, J. A. Talbert, and J. J. Cox. Modeling primitives: an object oriented formulation of boundary value problems in a solid geometric modeling context. In J. Rossignac, J. Turner, and G. Allen, editors, *Second Symposium on SOlid Modeling and Applications*, pages 441–448, Montreal, Canada, May 19-21, 1993.

[ZL89] O. C. Zienkiewicz and Taylor R. L. *The Finite Element Method.* McGraw-Hill, fourth edition, 1989.