

Chapter 1

Discrete Calculus: History and Future

Abstract The goal of this book is to bring together three active areas of current research into a single framework and show how each area benefits from more exposure to the other two. The areas are: discrete calculus, complex networks, and algorithmic content extraction. Although there have been a few intersections in the literature between these areas, they have largely developed independently of one another. However, we believe that researchers working in any one of these three areas can strongly benefit from the tools and techniques being developed in the others. We begin this book by outlining each of these three areas, their history and their relationship to one another. Subsequently, we outline the structure of this work and help the reader navigate its contents.

1.1 Discrete Calculus

The term “*discrete calculus*” is one of many expressions, along with “discrete exterior calculus” and “mimetic discretization”, that describes the body of literature that has focused on finding a proper set of definitions and differential operators that makes it possible to operate the machinery of multivariate calculus on a finite, discrete space. In contrast to traditional goals of finding an accurate *discretization* of conventional multivariate calculus, discrete calculus establishes a separate, equivalent calculus that operates purely in the discrete space without any reference to an underlying continuous process. Therefore, the purpose of this field has been to establish a fully *discrete* calculus rather than a *discretized* calculus. The standard setting for this discrete calculus is a cell complex, of which a graph or network is a special case.

Although the tools of discrete calculus have risen to prominence more recently, the concepts in discrete calculus were historically developed in parallel with the conventional calculus. In fact, the origins of both conventional vector calculus and discrete calculus have their origins in studies of *spatial* representations and relationships as well as the description of physical systems associated with space. In order to understand the relationship of conventional calculus to discrete calculus, we believe that it is useful to briefly examine the history of development in both areas.

The term “discrete calculus” appears to be a recent invention which we have reluctantly adopted. Our reluctance is due to possible confusion with *discretized* methods, which have a different goal. The term *combinatorial calculus* might be more appropriate, but “discrete calculus” seems established at this point and the term “combinatorial calculus” has been used previously in a different context.

1.1.1 *Origins of Vector Calculus*

Modern, conventional calculus consists of several components. One component of calculus employs the notion of the *infinitesimal*, using limits and infinite series to develop the theory. Naturally, these concepts are associated with an underlying continuous space. In contrast, other components of calculus do not depend on the concept of the infinitesimal. For example, the Fundamental Theorem of Calculus describes essentially a topological relationship, given by the operation of integration, between the integrand and the domain of integration. In the context of describing a discrete calculus, in which the domain is discrete and finite, the aspects of conventional calculus that focus attention on the infinitesimal will play a smaller role in this exposition. Instead, our focus in reviewing conventional calculus will be in those aspects that can be used to describe *space* and the behavior of functions defined in space.

Historically, univariate or one-dimensional calculus that was developed by Newton and Leibniz was extended to describe two-dimensional space by using the real and imaginary parts of a complex number to represent the two dimensions of the plane (known as the **complex plane**). This development has variously been attributed to Caspar Wessel or Jean-Robert Argand, both of whom worked in the late 18th century (and as a result the complex plane is sometimes called the **Argand plane**). Other sources attribute to Gauss the introduction of the complex plane to represent two-dimensional space, although Gauss’ work was mainly in the early 19th century.

Unfortunately, physical space is three-dimensional and therefore the two-dimensional representation by the complex plane was insufficient to describe all physical processes. Furthermore, it was unclear how to extend the concept of complex numbers to three dimensions. This problem was finally solved in 1844 by William Hamilton who defined *four*-dimensional complex numbers called **quaternions** [185]. Quaternions formed the basis for modern vector calculus by defining a *scalar* quantity as the real part of a quaternion and a *vector* quantity as the imaginary part of a quaternion. Hamilton’s student Peter Tait continued to develop and promote quaternions after Hamilton’s death in 1865, but later researchers Josiah Gibbs and Oliver Heaviside (independently) stripped out the quaternion focus of the work and presented a simplified form of vector calculus. This simplified form, without any explicit reference to complex numbers or quaternions, is today the conventional vector calculus taught in school. However, it is important to note that, because of its origin, conventional vector calculus was derived explicitly to describe space in *three dimensions*. This point will be emphasized again in Chap. 2 as we proceed to develop the tools of discrete calculus.

During the mathematical development of quaternions (and later vector calculus), James Maxwell was developing his theory of electromagnetism. Maxwell immediately recognized the value of quaternions in his work and seized upon this new mathematics to help him describe the physical behavior of electric and magnetic fields. Therefore, the description of space provided by vector calculus was immediately used by Maxwell to describe the behavior of *functions* associated with that space. In fact, the use of vector calculus in physics became so successful that connections were made between various physical theories, showing that quantities in one area of physics behaved analogously to quantities in a different area of physics. These analogies were later explained in 1976 by Enzo Tonti who suggested that the reason for these analogies was that each analogous quantity was associated with the same unit of *space* [378]. Consequently, we see again the close connection between the mechanics of vector calculus and the mathematical description of space. Ultimately, it is this connection that will allow for the development of a discrete calculus on a discrete domain.

Vector calculus was further generalized to describe surfaces and also extended beyond three dimensions. The development of calculus on surfaces belongs to the classical discipline of *differential geometry*. The abstraction of calculus and extension to higher dimensions is sometimes called *exterior calculus* or the *theory of differential forms*, which was first developed by Élie Cartan in the early 20th century. This more abstract and generalized form of calculus is where we begin our exposition in Chap. 2 to develop the discrete calculus.

From an early stage in the development of vector calculus, there was interest in *discretizing* the equations of vector calculus so that they could be solved in pieces. A major motivation for this approach is that many of the differential operators are *linear*. In this case, linearity implies that the act of applying an operator to a function may be subdivided into small, local operators and then reassembled to produce the result. In 1928, Courant, Friedrichs and Lewy published the **finite differences** approach to discretizing differential equations [92], which became the standard method for discretization and was heavily developed during the middle of the 20th century. Courant later planted the seed for what later became known as the **finite element method** in 1943 [91] and it was later formalized [361].

The rise of ubiquitous computing has propelled a sustained interest in the discretization of differential equations to model everything from airplanes to medical implants. More recently, development of discretization has moved toward formulation of differential calculus on a more general cell complex rather than at a series of point locations, which is sometimes known as *mimetic discretization* [40, 47, 178, 249, 275].

For more details on the history of the development of vector calculus, see [95].

1.1.2 Origins of Discrete Calculus

The origins of discrete calculus also began with a study of *space* in the context of graph theory [37]. Specifically, the study by Euler of the Königsberg Bridge Prob-

lem modeled the two banks and islands of Königsberg as *nodes* in a graph and the bridges connecting them as *edges* [126]. Therefore, from its earliest beginnings, graph theory was also modeling space and the neighborhood connections between different areas.

The first application of graph theory to the modeling of physical systems came from Kirchhoff, who both developed the basic laws of circuit theory and also made fundamental contributions to graph theory [232]. Kirchhoff's work on applying graph theory to model circuits in 1841 predated the development of quaternions, vector calculus and Maxwell's Laws. At the end of the 19th century Poincaré published his work on *analysis situs* [307] in which he analyzed simplicial complexes, simplicial homology and laid the foundation for the subject of algebraic topology. Poincaré also was concerned with representing space by discrete elements and, in fact, the term *analysis situs* is Latin for "analysis of position" or "analysis of location". Algebraic topology was further developed in the early 20th century by many contributors, including Whitney, de Rham, Cartan and Lefschetz (see [107, 217] for more history on this development).

Circuit theory continued to develop using graph theory and the concepts of algebraic topology [118, 400]. In 1955, Roth directly connected algebraic topology to electrical circuits and used the theory to establish conditions under which a circuit will have a solution (i.e., be realizable) [323]. This achievement, coupled with the unconventional work of Kron [250], caused electrical engineers to begin viewing electrical circuits as an alternative to conventional vector calculus in which all of the laws of vector calculus were discrete. This viewpoint came together in the review article by Branin [59] who explicitly posited that electrical circuits (and "higher-dimensional" circuits) had the same structure as conventional vector calculus.

As technology has allowed increased representation and computational power, these tools from discrete calculus have received recent attention. In particular, the area of computer graphics has seen a strong interest in the concepts of discrete calculus [103, 105, 200, 258, 374, 421], although the interest has been by no means limited to that field [81, 123, 161, 219, 420]. The rise of linear algebra packages, such as MATLAB, make the use of discrete calculus operators and algorithms quite convenient, since the primary operators take the form of large, sparse matrices. Additionally, the demonstrated ability of parallel computational devices such as GPUs to efficiently solve problems in linear algebra with sparse matrices holds strong promise for discrete calculus operations in the future as these parallel computing devices become increasingly common.

1.1.3 Discrete vs. Discretized

It is remarkable that both conventional vector calculus and discrete calculus developed around the representation of *space* and the manipulation of functions defined on that space. As we see in Chap. 2, the definition of the underlying space actually *defines* the structure of the differential operators in both conventional vector

calculus and discrete calculus. Additionally, it is remarkable that both continuous vector calculus and discrete calculus were initially adopted by researchers seeking to understand the behavior of electricity, with Maxwell initially adopting the early conventional vector calculus to describe electromagnetism and Kirchoff (and later researchers) adopting the early discrete calculus to describe circuit theory.

The focus of historical study in vector calculus and partial differential equations has been on producing *analytical*, closed-form solutions to problems. In contrast, the finite nature of discrete calculus and rise of computational power has driven that area to be less focused on analytical solutions and to place more emphasis on algorithms for finding solutions. In Chap. 2, we give the discrete calculus expressions for such classic topics as integration by parts and Green's Theorems. The truth is that these fundamental tools in conventional, analytical calculus are simply not as important in discrete calculus because there is less need to find analytical solutions to equations in the discrete calculus setting. However, these classical techniques can be useful in the sense that the intuition behind these concepts in conventional vector calculus can be re-used in the discrete calculus setting, and also these classical analytic tools can sometimes be used in proving behavioral properties about certain algorithms (see, e.g., [161]).

Before moving on, we want to stress again the importance of distinguishing *discretization* of conventional calculus from the *discrete* calculus treated in this work. In the first case, the goal is to compute a solution to some problem on a continuous space. However, an analytic solution is too difficult to find and so a discretization strategy is employed that allows for a computer to produce an approximate solution. Therefore, the main goal in discretization methods is the fidelity of the discretized, computed approximation to the desired analytical solution. Consequently, an important technique for proving the correctness of a discretization strategy is to show that as the discretization becomes finer and finer (i.e., closer to the continuum) that the solution obtained by discretization in the limit approaches the known analytical solution. This discretization approach is commonly used in mimetic discretizations [103, 105, 200, 258, 374, 421] and in modern finite element methods [40, 47, 178, 249, 275].

In contrast, *discrete* calculus treats a discrete domain (e.g., a graph) as entirely its own entity with no reference to an underlying continuum. For example, a social network (such as a citation network) is not associated with any continuous space in the sense that the network is not viewed as a discretization or sampling of an underlying continuum. However, the tools of discrete calculus can still be used to analyze the structure of the network and the behavior of functions associated with the network. Consequently, traditional discretization concerns about approaching a continuous solution in the limit are meaningless in the context of discrete calculus.

Neither conventional calculus nor discrete calculus are subordinate to each other. Both frameworks can be used to describe physical systems, e.g., with conventional vector calculus describing the behavior of electromagnetic fields and discrete calculus describing the behavior of electrical circuits. Chapter 3 goes into greater detail about the connection between discrete calculus, circuit theory, and other discrete systems. Additionally, the history of 20th century physics has shown that there are

legitimate philosophical questions about the appropriateness of treating space, and quantities associated with that space, as continuous or discrete entities. We go no further in addressing these issues except to state that the focus of this work will be on discrete calculus, its relationship to conventional calculus, and the occasional intersections with discretization methods.

1.2 Complex Networks

The term “complex network” is used to describe any non-trivial network.¹ Examples of “trivial” networks are **regular graphs** (where every node has the same number of incident edges), lattices, or random graphs. Traditionally, trivial networks were the focus of study because they are easier to study analytically. However, networks obtained from the real world are often nontrivial, and the availability of modern computers has allowed us to represent and analyze huge networks.

The current level of interest in complex networks began in the late 1990s. During this period the Internet (along with the World Wide Web) was on the rise and there were many groups looking at network structure for purposes of designing a more secure, efficient network as well as techniques for analyzing the structure of the network for tasks like Internet search. During this same period, a series of influential papers by Watts, Strogatz, Albert and Barabási [20, 362, 396] spurred interest in the description of complex networks.

One major effect of the interest in complex networks has been the recognition that complex networks may be used to model a huge array of phenomena across all scientific and social disciplines. Examples include the World Wide Web, citation networks, social networks (e.g., Facebook), recommendation networks (e.g., Netflix), gene regulatory networks, neural connectivity networks, oscillator networks, sports playoff networks, road and traffic networks, chemical networks, economic networks, epidemiological networks, game theory, geospatial networks, metabolic networks, protein networks and food webs, to name a few. The ubiquity of complex networks and the importance of understanding their structure has been the focus of several books in popular science [21, 64, 77, 398].

This book is not about complex networks directly. However, there are ideas which have been developed in the field of complex networks throughout the book (particularly in Chap. 8). Instead, the examples used in the applications chapters borrow heavily from the problems which have been studied in this field. In effect, our goal is to show how the tools of discrete calculus and the algorithms developed here can be applied to the vast array of problems which have been uncovered in the literature on complex networks, as well as show how some of the concepts developed in the complex network literature relate to discrete calculus. Furthermore, in contrast to image processing or computer graphics applications, we have been careful to develop all

¹Throughout this work we use the terms *network* and *graph* interchangeably to mean exactly the same thing—a 1-complex comprised of nodes and edges. A complex will be defined in Chap. 2.

of our tools *without assuming a network embedding* so that the tools developed here may be applied to an arbitrary complex network.

1.3 Content Extraction

The third area addressed in this book is content extraction. The term **content extraction** has a broad meaning that can encompass many different problems and disciplines. In our case, we use the term *content extraction* to indicate any algorithm in which the goal is to extract information from a dataset and/or network.² Examples of content extraction algorithms covered in the book include filtering (denoising), clustering, manifold learning, ranking, and network characterization.

Content extraction can be used to analyze the structure of data associated with a network (sometimes called *attributed graphs*) or the structure of the network itself. An important methodology described in this book for analyzing data associated with a network is to use the data to define *weights* on the network and then use an algorithm that analyzes the structure of the network to draw conclusions about the data. Chapter 4 describes how weights may be generated from the data. For example, to perform clustering of data associated with nodes, in Chap. 6 we show how that data may be used to establish edge weights, after which any algorithm that clusters a weighted network may be applied to produce a clustering of the data. In particular, image content may be clustered using this approach.

Many of the algorithms developed for content extraction were developed in the context of image processing or computer graphics. In both of these cases, traditional one-dimensional signal processing must be substantially modified to operate in multiple spatial dimensions (generally in two or three dimensions). Consequently, the algorithms developed in these fields explicitly account for spatial interactions. In many ways, the work in this book may be viewed as continuing the development of the variational algorithms based on *active contours* [222] and *level sets* [339] which dominated image processing (among other fields) for many years. These methods cast content extraction problems as energy minimization problems in which the optimum solution to the energy minimization problem provided a solution to the content extraction problem. Level sets provided a mechanism for optimizing these energies, using tools from the study of partial differential equations. The book by Sethian [339] demonstrated the remarkable number of applications that could be treated by the energy minimization methodology of level set techniques. Similar to this body

²In many disciplines the network structure is itself considered the dataset (since this information must be collected). Our distinction here between *data* and *network* is simply due to the fact that in several of the disciplines we consider there is data associated with each node or edge in the network (as in the case of image processing), in which case the network defines the spatial domain upon which the data is processed. However, we address algorithms that can be used to process data associated with a network and algorithms that can be used to analyze the structure of the network itself. Consequently, algorithms that process data associated with networks or algorithms that analyze the network structure itself are both considered to be content extraction algorithms.

of work, we also equate energy optimization with the solution to content extraction problems. However, the use of discrete calculus to formulate the energy minimization problems affords us the major advantage of generalizing the utility of these content extraction methods to arbitrary discrete domains (e.g., graphs). This generalization allows us to apply the energy minimization methodology to tackle the problems of the future being defined in the field of complex networks. Additionally, this formulation in terms of discrete calculus may also be applied in the same areas that were conventionally treated by level sets by viewing a Cartesian domain as a special case of the more general framework (i.e., a lattice). In fact, recent work has demonstrated that energies which were conventionally formulated using vector calculus and optimized with level sets could be dramatically outperformed by formulating the same energies using discrete calculus and performing the optimization using techniques in combinatorial optimization [163].

1.4 Organization of the Book

In the first part of the book, we present a brief review of discrete calculus with a focus on those key concepts that are required for the successful application of discrete calculus. This is by no means an exhaustive treatment of this topic, but is included to establish the notation and terminology used throughout the subsequent chapters, and to make our treatment reasonably self-contained. We provide reference throughout to the literature for readers who would like to delve deeper into the vast topics of differential forms and discrete calculus.

In the second part of the book, we redevelop many of the standard tools in image processing on a generalized, unembedded network. In these chapters, the generalized Laplacian operator plays a central, consistent role. Specifically, we show how the discrete calculus provides a natural definition of “low-frequency” on a discrete space, which then yields filtering and denoising algorithms. These algorithms are also developed from the standpoint of local interaction models between neighboring nodes. We then show how filtering algorithms can give rise to clustering algorithms. Clustering algorithms are then used to develop manifold learning and data discovery methods. Finally, ranking algorithms and algorithms for analyzing the structure of a network are also addressed. In addition to generalizing this set of tools to arbitrary networks, we believe that the context of discrete calculus has allowed us to unify very many standard image processing algorithms into a common framework. Therefore, the reader who is interested purely in image processing will find a unified framework for viewing a wide variety of standard algorithms in filtering, clustering, and manifold learning.

1.5 Intended Audience

This book is intended for graduate students, researchers, and engineers who are familiar with the basics of vector calculus, graph theory, and linear algebra. For

researchers interested in discrete calculus, we intend for this book to tie algorithms and applications to the theory. For researchers in the domain of complex networks, we intend this book to provide an introduction to the foundations of discrete calculus on a network and a set of theoretical and algorithmic tools for analyzing networks. For researchers interested in image processing and computer graphics, we intend to introduce the foundations of discrete calculus, argue why algorithms should be developed on a more general graph, and demonstrate how to reformulate traditional algorithms defined in the continuum onto a discrete structure.

In each of the applications chapters in the second part of the book, we present several worked examples of how to use discrete calculus to analyze real data, in which multiple algorithms are applied. Naturally, the best algorithm for any given data set will depend on the application and on the nature of the data. Therefore, our intention is not so much to determine which algorithm is the best for a particular application, but rather to demonstrate the wide applicability of the framework and to present multiple processing strategies to give the reader a sense for the performance and behavior of the algorithms.

The primary content of this book is a review of work which has occurred in several fields and an attempt to bring them all into the same framework with a standardized notation. However, there are also aspects of a research monograph in the sense that some of the material has not previously appeared in the literature to the knowledge of the authors. Significant new material includes our generalization of algorithms and concepts used to analyze nodes and node data to novel analyses of edges and edge data. Additionally, we view the running unification of several ideas (and algorithms) into a single framework as a useful contribution. Finally, our goal is to provide the reader with the ability to understand the concepts being described and also an idea of how to implement them. Where algorithms are not fully described, citations are provided such that the interested reader may find more details.

Chapter 2 forms the basis of our exposition for discrete calculus and therefore every subsequent chapter depends to some degree on this chapter. Chapter 3 extends the exposition of discrete calculus to the description of physical systems, with a focus on circuit theory. Although some concepts from circuit theory will reappear in later chapters (e.g., effective resistance), this chapter primarily stands on its own. Chapter 4 marks the beginning of the application sections, in the sense that it details how a weighted edge set or cycle set may be derived for a particular application. The usefulness of this chapter ultimately depends on the particular application that a reader may be pursuing. Chapter 5 introduces the concept of filtering on an arbitrary graph, which then forms the basis for Chap. 6 on clustering. Chapter 7 continues to build directly on the clustering and filtering concepts to introduce manifold learning and ranking techniques. Chapter 8 breaks from the stream of the previous three chapters to provide various methods for measuring connectivity, separability, and topological and geometric properties of a network. Appendix A contains useful notes for the implementation of the algorithms described in the text and Appendix B provides an introduction to the set of optimization techniques used throughout the book. Finally Appendix C ties most closely back to Chap. 2 by going into further details on the Hodge Decomposition.