The Book Review Column[1]

by Frederic Green
Department of Mathematics and Computer Science
Clark University
Worcester, MA 02465
email: fgreen@clarku.edu

In this column, the following books are reviewed:

1. **Games and Mathematics: Subtle Connections**, by David Wells. Reviewed by S. C. Coutinho. This is a (mostly non-technical) book that explores the mathematical nature of games. In addition, and more extensively, it discusses the game-like nature of mathematics through a variety of mathematical examples.

2. **Jewels of Stringology**, by Maxime Crochemore and Wojciech Rytter. Reviewed by Shoshana Marcus. This is a book about string algorithms, with an emphasis on those that are fundamental and elegant: i.e., "algorithmic jewels." Written at the advanced undergraduate/beginning graduate level.

3. **Algorithms on Strings**, by Maxime Crochemore, Christophe Hancart and Thierry Lecroq. Reviewed by Matthias Gallé. Another book on stringology, aimed more at the graduate level. Although there is some overlap with "Jewels" (e.g., algorithms and data structures for pattern matching), the two books have a substantial symmetric difference.

4. **Polyhedral and Algebraic Methods in Computational Geometry**, by Michael Joswig and Thorsten Theobald. Reviewed by Brittany Terese Fasy and David L. Millman. This book treats a wide array of topics, beginning from the basics (e.g., convex hulls) and proceeding through the fundamental tools of computational algebraic geometry (e.g., Gröbner bases), and containing a section on applications as well.

5. **A Mathematical Orchard - Problems and Solutions**, by Mark Krusemeyer, George Gilbert, and Loren Larson. Reviewed by S.V. Nagaraj. This is a collection of challenging mathematical problems, published by the MAA.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. *ReCombinatorics: The algorithmics of ancestral recombination graphs and phylogenic networks* by Gusfield.
2. *Distributed Systems: An algorithmic approach (second edition)* by Ghosh.
3. *Tractability: Practical approach to Hard Problems* Edited by Bordeaux, Hamadi, Kohli.
4. *Recent progress in the Boolean Domain* Edited by Bernd Steinbach
5. *Distributed computing through combinatorial topology* by Herlihy, Kozlov, and Rajsbaum.

Programming Languages

1. *Selected Papers on Computer Languages* by Donald Knuth.

Miscellaneous Computer Science

1. *Introduction to reversible computing* by Perumalla.
2. *Algebraic Geometry Modeling in Information Theory* Edited by Edgar Moro.
3. *Digital Logic Design: A Rigorous Approach* by Even and Medina.
4. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective* by Srikant and Ying.
5. *CoCo: The colorful history of Tandy's Underdog Computer* by Boisy Pitre and Bill Loguidice.

Cryptography

1. *The Mathematics of Encryption: An Elementary Introduction,* by Margaret Cozzens and Steven J. Miller.

Miscellaneous Mathematics

1. *Asymptopia,* by Joel Spencer with Laura Florescu.
2. *Spectra of Graphs,* by Andries E. Brouwer and Willem H. Haemers.

Mathematics and History

1. *Professor Stewart's Casebook of Mathematical Mysteries* by Ian Stewart.
2. *The Golden Ratio and Fibonacci Numbers* by Richard Dunlap.
3. *Mathematics Galore! The first five years of the St. Marks Institute of Mathematics* by Tanton.
4. *An Episodic History of Mathematics: Mathematical Culture Through Problem Solving* by Krantz.
5. *Proof Analysis: A Contribution to Hilbert's Last Problem* by Negri and Von Plato.

Review of
Games and Mathematics: Subtle Connections [2]
Author: David Wells
Publisher: Cambridge University Press, 2012, 246 pp.
ISBN NUMBER: 978-1-107-69091-2, PRICE: US$19.99

Review by: S. C. Coutinho (collier@dcc.ufrj.br)

1 Overview

The connections between mathematics and games are probably as old as mathematics itself. Simple arithmetical riddles can be found in ancient Egyptian papyrus and clay tablets from Mesopotamia. The *Greek Anthology*, a collection of epigrammes of ancient and Byzantine origin, contains 46 mathematical problems of this type, the best known of which concerns the age of the Hellenistic mathematician *Diophantus*. Although these are mostly exercises in arithmetic, there was also a far more distinguished tradition concerned with using mathematics to analyse a game, or even to develop new games. Thus, Archimedes, the greatest mathematician of ancient Greece, devoted a tract to a tangram-like puzzle called the *Stomachion* – Greek for bellyache, from which one was supposed to suffer in trying to solve the puzzle. In their book *The Archimedean Codex*, Reviel Netz and William Noel point out that

> Archimedes was famous for hoaxes, enigmas, circuitous routes. These were not some external features of his writing: they characterised his scientific personality. Science is not – mathematics is not – dry and impersonal. It is where ones imagination is allowed to roam freely [1] (p. 54).

More recent examples of the use of mathematics to analyse games include Euler's solution of the Königsberg bridge puzzle and Sir William Hamilton's *Icosian* game. The 20th century witnessed a great variety of works in this tradition, the best known of which is probably *Winning ways for your mathematical plays* by Elwyn R. Berlekamp, John H. Conway and Richard K. Guy, originally published in 1982.

The popularity of puzzles and games with a mathematical bent led to the publication of whole collections of them. One of the earliest seems to have been the 9th century *Propositions to Sharpen the Young*, attributed to Alcuin of York, where we find the puzzle on the wolf, the goat and the cabbage, that a boatman must transport across a river in a small boat, never leaving the wolf alone with the goat, or the goat with the cabbage. A more recent collection, which discusses this problem in its first chapter, is *Récréations Mathématiques*, by Édouard Lucas, a French teacher well-known to students of number theory and cryptography for his primality tests. Lucas's book include chapters on chess problems, peg solitaire, the 15-game and the Chinese Rings puzzle, for which he found a solution using binary codes. Lucas also invented the Towers of Hanoi puzzle, together with the story of the 64 golden disks moved between three posts by priests in an Indian Temple.

Of course not everything that we call a game can be analysed mathematically. As Wells (p. 18) notes, the games that are liable to such an analysis have three characteristics: they are abstract, they can be played in the head and it is possible to prove ones conclusions about them. By abstract, he means that one can make the pieces with which the game is played of any size or any material one likes. One is reminded of Hilbert's saying that, given the axioms of geometry,

> one must be able to say "tables, chairs, beer mugs" each time in place of "points, lines, planes";

and still arrive at the same conclusions. This brings us to the 20th century, when a new theme was introduced in the story of the interaction between mathematics and games: the idea that mathematics itself is a game, which consists in the manipulation of symbols that are subject to given rules. The most extreme expression of this theme is probably Bertrand Russel's definition of mathematics as "the subject where we never know what we are talking about, nor whether what we are saying is true". This view of mathematics as a game is one of the key themes in Wells's book.

2 Summary of Contents

The book has two parts. The first, called *Mathematical Recreations and abstract games* is mostly a discussion of a number of abstract games and the way mathematics can be used to analyse them. The games mentioned by Wells include the bridges of Königsberg, knight tours in chess and the towers of Hanoi in chapter 1, and Nine Men's Morris, Hex, Chess and Go in chapter 2. Chapter 1 also contains the characterisation of abstract games, mentioned above. Two properties of these games are then considered in more detail: the fact that they can be "played in the head" (chapter 3) and the rôles of checking and proving what one has learned (chapter 5). The very short chapter 4, called *Why chess is not mathematics* deals with the differences between mathematics itself and mathematical games like Chess and Go.

Part II, entitled *Mathematics: game-like, scientific and perceptual*, is very different from part I. To begin with, it is almost twice as long as part I. Moreover, mathematical games play a very minor rôle in part II, which is concerned mostly with examples that illustrate the game-like characteristics of mathematics. The examples chosen by Wells include sums of powers (chapter 7), Ceva's theorem and Dandelin spheres (chapter 7), sums of series (chapters 8 and 9), group theory (chapter 10), the seven circle theorem (chapter 11), prime numbers (chapter 12), the four colour theorem (chapter 13), the cycloid (chapter 14), minimum paths (chapter 15), polygonal numbers (chapter 16), coordinate geometry (chapter 17) and Steiner's porism (chapter 18). These topics are used to discuss a large number of themes, like different styles of doing mathematics, computer aided proofs, experimental mathematics, applications of mathematics in physics and the notion of mathematical structure. Part II ends with two chapters of a more philosophical tenor. Chapter 19 is concerned with mathematical beauty, while in the final chapter Wells takes (p. 8)

> a step back to look at society and culture as a whole and to see what game-like features we can find in it – both illuminating the role of mathematics and helping to explain why maths exists at all.

3 Opinion

The aim of the book, which is explained at the introduction and further elaborated in the chapters of part I, can be summed up as: to identify characteristics that are common to mathematics and (abstract) games and to explore the game-like aspects of mathematics through examples. This much is clear; what is not quite so clear is what its intended public is. Those who come to this book looking for new mathematical puzzles are likely to be disappointed. Although many games and puzzles are mentioned in the book, they are not Wells's main theme. His real aim is to show that mathematics itself can be thought of as an abstract game. In order to do that he collects an impressive array of examples, most of them elementary, from number theory, geometry, algebra and analysis. This suggests that the ideal reader of the book is probably a games

enthusiast with a smattering of calculus. The book provides pointers to several areas of mathematics that are likely to fascinate such a person.

Unfortunately the book is not always reliable, specially when it comes to the history of mathematics. For example, Wells explains in page 116 that many attempts were necessary before mathematicians found the most general definition of polyhedron that would satisfy Euler's relation between faces, edges and vertices. In the very next page, he states that no such thing happened in the case of group theory:

> Early group theorists made many errors but – unlike the case of Euler's relation – they were game-like errors, not errors of definition.

If by 'game-like errors' he means errors in applying the rules of the game, his statement is very far from being true. As Hans Wussing explains in [2], groups were originally defined as sets of permutations closed under composition – which, being finite, are indeed groups in our sense of the word. However, Lie used the same definition in his first papers on (infinite) transformation groups published in 1874, and it was only in 1876 that he felt the need to prove that, under the assumptions he was making, all his transformation groups actually contained an identity and each element had an inverse. Other historical errors are more obvious, like the statement (p. 108) that Archimedes was "the first (known) genius in the history of mathematics" – which ignores Eudoxus – or referring to Mandelbrot in the present tense, when he had died two years before the book was published. Given the author's obvious enthusiam for Archimedes, it is also surprising that the *Stomachion* is not mentioned at all in the book.

Of course these slips can be easily fixed in future editions. What really matters is whether the author has managed to achieve the aims that he set himself in writing his book, and the answer to that is clearly yes. Indeed, the many examples collected in the book illustrate quite clearly the importance of play to the practice of mathematics. Let me finish with Wells excellent advice (p. 59) on the importance of play in mathematics:

> Students who do not 'play around' with such new ideas, but stick to the textbook explanation and do no more than answer a few exercises, will never develop deep intuition, and never become real mathematicians.

References

[1] R. Netz and W. Noel, *The Archimedes Codex*, Weidenfeld and Nicolson (2007).

[2] H. Wussing, *The genesis of the abstract group concept*, Dover (1984).

The Author, David Wells, Responds:

I shall indeed correct the historical errors to which the reviewer draws attention in any future edition. I would like to disagree with the reviewer on the following point only:

> Unfortunately the book is not always reliable, specially when it comes to the history of mathematics. For example, Wells explains in page 116 that many attempts were necessary before mathematicians found the most general definition of polyhedron that would satisfy Eulers relation between faces, edges and vertices.
>
> In the very next page, he states that no such thing happened in the case of group theory:

> Early group theorists made many errors but unlike the case of Eulers relation they were game-like errors, not errors of definition.

I believe that there is an important distinction here. The failures of the definition of a polyhedron were signaled by proofs that failed, again and again, as Lakatos documents. The definition of a group changed but not, as I understand it, because it led to proofs which were falsified by the discovery of surprising groups that did not fit the definition - indeed - that was not possible because the idea of a group, as it were, did not exist before the definition. The point about the polyhedron example is that mathematicians did have what seemed to them a clear idea of what a polyhedron 'was', and they were trying to formalise, clarify, state plainly and concisely, what they thought they already 'knew'. That was not the case with the definition of a group.

Review of[3]
Jewels of Stringology
by Maxime Crochemore and Wojciech Rytter
World Scientific Publishing Company, 2003
320 pages, Softcover

Review by
Shoshana Marcus shoshana.marcus@kbcc.cuny.edu
Kingsborough Community College of the City University of New York, Brooklyn, NY, 11235

1 Introduction

This book is a collection of beautiful and classical algorithms for working with strings. Hence, the title, *Jewels of Stringology*. A string is one of the basic data types in computing. A string, or *text* in other contexts, is simply a sequence of symbols. Many of the algorithms presented in this book solve practical problems like string matching and data compression. Others address problems that are of more theoretical interest, relating to symmetries and repetitions in words. However, these problems are all interesting from an algorithmic perspective and enable the reader to appreciate the importance of combinatorics on words. This book organizes and elucidates sets of related research achievements that were never compiled before, all in an easily accessible format.

String algorithms are employed by many diverse domains. Application areas that utilize text algorithms for fundamental tasks include spam filters, anti-virus programs, word processors, web search engines, computational molecular biology, and feature detection in digitized images. This book seeks to elucidate the essential components and algorithmic tasks faced by string algorithms, which lay the foundation for many computational processes to run effectively.

2 Summary

The book begins by introducing the pattern matching problem, which is to find all occurrences of a pattern inside a larger text. Pattern matching arises in many computing domains, ranging from word processing to computational biology. The main focus of this book is on the basic pattern matching problem and its variations, such as approximate matching, multiple pattern matching, two-dimensional matching, finding regularities in text, and data compression. Combinatoric properties of strings form the basis for many techniques to solve these problems. Thus, concepts like periodicity and the notion of a primitive word are explained in the introductory first chapter.

The first algorithms described in this book are the basic pattern matching algorithms that were developed in the 1970's, those of Knuth, Morris and Pratt and of Boyer and Moore. Both of these algorithms run in linear time and space. These algorithms each consist of a preprocessing stage that computes information about the pattern, such as borders, followed by a searching phase that identifies occurrences of the pattern in the text. The book introduces the basic premises of these algorithms and the combinatoric ideas that underlie their analyses.

It is often useful to index the text ahead of time so that membership queries to find occurrences of a pattern in the text can be answered in time proportional to the size of the pattern, without considering the

entire text. The suffix tree is one such data structure that facilitates efficient solutions to many common string problems. Several of the problems that are directly solved with a suffix tree are alignment of strings, finding the longest common factor of several strings, all-pairs suffix-prefix matching, and locating all maximal repetitions in a string. The suffix tree is a compact trie that represents all suffixes of the underlying text. The suffix tree for $T = t_1 t_2 \cdots t_n$ is a rooted, directed tree with n leaves, one for each suffix. A special character "$" is appended to the string before construction of the suffix tree to guarantee that each suffix ends at a leaf in the tree. Each internal node, except the root, has at least two children. Each edge is labeled with a nonempty substring of T and no two edges out of a node begin with the same character. The path from the root to leaf i spells suffix $T[i \ldots n]$.

The suffix tree can be constructed in linear time and space with respect to the string it represents. This book gives a clear and concise explanation of two linear time suffix tree construction algorithms. Diagrams and examples clarify the exposition particularly to a reader who has no previous exposure to the techniques. Ukkonen's linear time algorithm is online and can update the data structure as new data arrives. It uses several implementation tricks to maintain a linear time complexity even though the algorithm builds incomplete, implicit suffix trees at each stage of construction before the last character is considered. The text also includes a brief overview of McCreight's incremental suffix tree construction algorithm. This book also gives an overview of Farach's suffix tree construction algorithm which uses suffix sorting techniques and runs in linear time, independently of the size of the alphabet.

After introducing suffix tree construction algorithms and applications of suffix trees, the text discusses another data structure that serves a similar role, the directed acyclic word graph (DAWG). A DAWG results from identifying and eliminating isomorphic subtrees in a suffix trie. A DAWG can be constructed online and a straightforward algorithm is described in the text. Since they are both compact representations of the same trie, suffix trees and DAWGs are closely related. A compacted DAWG can be constructed by identifying and merging isomorphic subtrees in a suffix tree. A suffix tree can be transformed into a compacted DAWG in linear time.

A chapter is devoted to algorithms that deal with regularities in strings. One kind of regularity occurs when one text segment is an exact copy of another. For example, squares are of the form xx. Another kind of regularity allows segments of text to be symmetric copies of each other. Palindromes are of this nature. Even palindromes are of the form xx^R and odd palindromes are of the form xax^R. A palstar is a composition of palindromes. Algorithms that discover these types of regularities in text are interesting in and of themselves. They also form useful components within many other string algorithms.

The most intriguing algorithms are perhaps those that are efficient with respect to both time and space. Several different linear-time yet constant space string matching algorithms have been developed. Amazingly, these algorithms are quite simple, since they all rely on simple combinatoric properties of periodicities in strings. This text is a wonderful resource for understanding this set of elegant yet simple algorithms. It is probably the first book to compile them and explain how they evolved in a clear yet concise manner.

Data compression allows data to be represented in a reduced form. This book discusses lossless compression techniques that are reversible. Compressing data reduces storage space requirements and allows faster data transmission. Data compression methods attempt to eliminate redundancies, regularities, and repetitions. Thus, these algorithms fit well into the topics of this book. The text focuses on the theoretical frameworks that underlie Huffman statistical encoding and Ziv-Lempel compression.

In many pattern matching applications, approximate matching is more relevant than exact matching. For instance, pattern matching of DNA sequences needs to address the mutations that frequently occur. The approximate matching scheme discussed in this book is edit distance, i.e., the minimal number of local edit operations that are required to transform one string into another. After computing an edit distance matrix

with dynamic programming, edit distance can be viewed as a shortest path problem or as the problem of finding the length of a least weighted path in a graph from a source to a sink. Finding a longest common subsequence between two strings is closely related to the problem of finding the edit distance between them. String matching with errors differs only slightly from the edit distance problem. Here we are given a pattern and a text and want to find the minimum edit distance between the pattern and any factor of the text. A straightforward algorithm with optimal asymptotic time complexity has been developed for the scenario in which the number of allowed errors is bounded ahead of time by some constant k.

Vishkin introduced pattern matching by dueling in 1985. It uses techniques that are closely related to borders of words and the Knuth-Morris-Pratt algorithm. When the pattern is preprocessed, a witness table is constructed to store for each pattern position a position that does not align with it. Then, for every pair of close positions in the text, a duel is performed to eliminate at least one of them as a candidate for a pattern occurrence. After the duels are performed a consistent set of candidates remain. In the final step, each text position is compared to a corresponding position in any of the candidates in a consistent set. The dueling paradigm has many applications in optimal parallel string matching and in two-dimensional pattern matching.

Pattern matching generalizes to the two-dimensional setting, in which both the pattern and the text are matrices. The first solution to two-dimensional pattern matching was proposed independently by Bird and by Baker. Their algorithm converts the two-dimensional pattern matching problem into a one-dimensional problem. The pattern is viewed as a set of metacharacters and each column is named, so that distinct columns are given different names and recurrences of the same metacharacter are given the same name. Then the text is named with the same naming scheme and searched for the one-dimensional pattern of names. It is fascinating that this technique readily generalizes to an efficient solution to multiple pattern matching of two-dimensional data.

Similar to the one-dimensional case, the most interesting algorithms for exact two-dimensional matching are connected to periodicities. However, the structure of two-dimensional periodicities is much more complex and intricate. This book gives a nice overview of these concepts and of algorithmic techniques that compute the two-dimensional periodicity of a matrix.

For the more advanced reader, this book includes a wonderful chapter on parallel text algorithms. These algorithms are elegant and their time complexities are incredible.

The book concludes with several other string problems that are offshoots of the simple pattern matching problem, such as computing the shortest common superstring of strings and parameterized matching. Another algorithm that is covered in the final chapter is the classic Karp-Rabin string matching algorithm, which has fascinated algorithms researchers for decades. This algorithm uses hashing to compute fingerprints of strings. On average, the algorithm is fast. Furthermore, it leads to a straightforward randomized optimal parallel algorithm because the process reduces to prefix computations. Hashing extends to the problems of finding repetitions in strings and arrays, by looking for repetitions of fingerprints.

3 Opinion

Jewels of Stringology was written by experts who continue to shape the field of text algorithms. Thus, it is not surprising that the authors did a fantastic job selecting algorithms for inclusion in this book that are elegant yet fundamental. True to its name, this book is truly filled with algorithmic gems. This book is an invaluable resource for researchers working in the field of stringology, whether they are designing new algorithms or whether they are implementing existing text algorithms.

Most textbooks on algorithms and data structures focus on computing subfields like graph theory, sort-

ing, and searching and they barely give notice to string algorithms. Until this book was written, many interesting text algorithms were only accessible in journal articles, in a form directed at experts in the field. This book covers a gap in the algorithmic literature and makes a wonderful set of fundamental and efficient algorithms accessible to wider audiences. The exposition in this text expects only a basic understanding of the techniques used in the design and analysis of algorithms. Furthermore, the ideas are clearly organized and the material is elucidated in a concise, often pictorial, manner. Thus, this book serves as a wonderful accompaniment to an undergraduate or graduate algorithms curriculum.

Review of[4]
Algorithms on Strings
by Maxime Crochemore, Christophe Hancart and Thierry Lecroq
Cambridge University Press, 2007
383 pages, Hardcover, US$ 104

Review by
Matthias Gallé matthias.galle@xrce.xerox.com
Xerox Research Centre Europe

1 Introduction

The title says it all: this is an algorithmic-centric book concerned with sequences of symbols. Pattern matching over sequences is a central point, with many variations and with different data-structures, although not the only one. The authors are very well known in the field of *stringology* and have published several other related books. This book is an updated translation from the original (2001) French "Algorithmique du texte"[5], and – for the metric-lovers – both versions have an accumulative citation count of ≈ 340 (Google Scholar). Thirteen years ago there was no definitive reference for strings algorithms, besides the still famous "Algorithms on Strings, Trees and Sequences" of Gusfield which, however, focused mostly on the suffix tree data-structure. The present book tackles an overlapping set of problems, but the algorithms rely mostly on automata and suffix arrays. All algorithms are given with many details together with a rigorous analysis of them, and the presentation of them is around problems. Different solutions to each problem are presented, relying on lemmas that are introduced as needed or on data-structures explained in their own chapters.

2 Summary

The book consists of nine chapters. Each chapter finishes with a historical section (references are given only there) and a set of exercises. Except a few chapters (e.g., the first for definition, or the fourth introducing suffix arrays) they can be read independently. Most of the definitions are intuitive and the more esoteric ones are well-referenced. All this means that you could pick up the chapter of your choice and just start reading!

Chapter 1 gives definitions, basic lemmas and some implementation choices which will impact the complexity analysis of future algorithms. It is not always clear why some things are included and why others aren't and I would have preferred a bit more of intuition before the introduction of some of the propositions.

Chapter 2 expands on the section of Chapter 1 dealing with implementation details of automata, taking care of many details. After a discussion of the pros and cons of each of them, two scenarios are presented: pattern matching on a single string and one on a set of strings.

Chapter 3 deals with exact pattern matching without a particular index structure. The famous algorithms of Boyer-Moore and Knuth-Morris-Pratt are presented in a global framework, together with many others.

Chapter 4 is about the suffix array and its associated data-structures. It is telling that this data-structure is introduced before the suffix tree, highlighting its importance in the remainder of the book.

[5] Whose full text can be downloaded on the authors' websites

Chapter 5 presents two similar data-structures. Besides the well-known suffix tree, this chapter also introduces the suffix automaton and relates both structures.

The remaining four chapters are concerned with applications of these data-structures. **Chapter 6** uses suffix automata and arrays for classical applications like finding an exact pattern and its position; compute all repeats, conjugates, longest repeats, minimal forbidden string, etc. **Chapter 7** is about alignments of strings, starting with Hamming and Levenshtein distance, continues with alignments and finishes with the famous bioinformatics BLAST algorithms. **Chapter 8** introduces some variations of approximate string matching and **Chapter 9** ("Local periods") deals with applications of partioning the prefixes of the suffixes of a string (periods, squares and construction of suffix arrays).

3 Opinion

The book has already been around for seven years (13 if counting the original edition, although the current one has some updates): while the theorems are still true (!), the book obviously doesn't cover more recent work, like recent advances in suffix array construction, and domains which became more popular in recent years (like compressed indexes). However, still today there are only a handful of books trying to cover a wide range of string algorithms. My feeling is that the authors tried to find the balance between exhaustive reviews of sub-domains (like exact string matching, of which one of the authors has a dedicated book) with a general overview of the area.

The book is targeted towards an aid for lectures at a master's course level, and this seems about right. I would not recommend this book as a stand-alone textbook, but it would be a great companion for students digging into the details of the algorithms presented during class, and wanting to explore different algorithms for solving the same problem. My reasons for this are that many of the notations are not intuitive, and the flow of the book does not always give hints as to where it is heading. Also, many well-known algorithms are referred to by their original name only in the notes, making it harder to link it to previous knowledge the reader may have.

The content tries to be application-agnostic, mentioning only a few times obvious applications in bioinformatics for example. It is clearly not aimed at practitioners, like a software engineer who is not satisfied with existing string matching libraries and wants to implement his or her own. While several times there are claims of what works in practice, benchmarking of some implementations is beyond the scope of this book.

A course entirely based on this book however may miss some recent advances, or risk not to satisfy the appetite of more application oriented students (like those looking for tools to use in computational biology). On the other hand, the chapters are rather independent and some of them can be replaced without problems. Most of the material seems appropiate for a masters level, excepting maybe the last two chapters which are a bit more advanced. At the same time these will be the most interesting ones for readers more familiar with basic notions of stringology.

Review[6] of
Polyhedral and Algebraic Methods in Computational Geometry
Authors: Michael Joswig and Thorsten Theobald
Springer (2013)
250 pages, softcover

Review by
Brittany Terese Fasy brittany@fasy.us
and David L. Millman dave@cs.unc.edu

1 Introduction

Polyhedral and Algebraic Methods in Computational Geometry by Michael Joswig and Thorsten Theobald (and translated by Theresa Szczepanski and the authors) is a book that covers the standard topics in computational geometry and algebraic geometry, grounded in their connection to polygons in $\mathbb{R}^2$ and polytopes and polyhedra in $\mathbb{R}^d$.

2 Summary

Introducing first polyhedral surfaces followed by linear programming, this book presents topics by building up from the fundamentals. After linear programming, this book goes on to cover both the standard topics in computational geometry, such as Voronoi diagrams and Delaunay triangulations, before moving on to topics in algebraic geometry, such as nonlinear surfaces and solving systems of polynomial equations. In addition to using figures to illustrate technical definitions and proofs, the authors provide both algorithms and code in `polymake`, `Maple`, and `Singular` to allow the interested reader to dig deeper into the materials.

2.1 Part I: Linear Computational Geometry

Part I focuses on geometric objects that can be described by linear constraints. This book begins with a discussion of projective geometry, which is a convenient model of computation. In particular, instead of treating the intersection of two parallel lines and the intersection of two transverse lines as two different cases, new points are added to the space, called ideal points or points at infinity, that are used to denote the intersection of parallel lines. (To visualize this, think about standing on railroad tracks. On the horizon, it looks like the two tracks join at one point, even though we know that they remain a constant distance apart. The point where they appear to meet is the point at infinity.)

Projective geometry provides a means for defining terms such as the affine transformation, which is a projective transformation that maps ideal points to ideal points. Then, they continue with defining an affine combination, affine hull, convex combination, convex hull (which is discussed in more detail in a later chapter), and orientation.

In Chapter 3, the book provides an in-depth description of polytopes and polyhedra. A polytope is defined to be the convex hull of finitely many points. More generally, a polyhedron is the intersection of a finite number of closed affine half-spaces. A pointed polyhedron is one that contains no affine lines. They show that every polytope is a polyhedron, as one can take the intersection of half spaces defined by the facets

of the polytope. This chapter also defines what it means for two polytopes to be equivalent (there exists an isomorphism of the face lattices), as well as introduces the concepts of polarity and duality. Several theorems and lemmas are proven, including a combinatorial proof of Euler's formula (that the alternating sum of the number of faces of each dimension is zero).

The next chapter covers Linear Programming (LP): given a nonemepty but bounded polyhedron $P = \{x : Ax \leq b\}$, and an *objective function* c, find a vector x that maximizes (or minimizes) the quantity cx, or determine that no such x exists. This chapter highlights the close connection between linear algebra and computational geometry, as illustrated by the well-known Simplex algorithm for solving a linear program. This chapter also contains a brief discussion on how to find an initial solution, from which one would optimize in order to find the correct solution. In addition, the dual of a linear program is defined.

Chapter 5 is the first chapter focusing specifically on computational geometry, by covering convex hulls. There are two equivalent representations of the convex hull: the *$\mathcal{V}$-representation* and the *$\mathcal{H}$-representation*. The $\mathcal{V}$-representation or *inner description* represents the convex hull as a set of points (vertices), and the $\mathcal{H}$-representation or *outer description* describes the convex hull as the intersection of finitely many closed half-spaces. The task of *computing the convex hull* is the task of finding the $\mathcal{H}$-representation. Then, the special case of the convex hull in the plane is discussed. In particular, the convex hull can be described by a cyclic ordering of the vertices (or of the edges) on the boundary of the convex hull. The Preparata and Hong $O(m \log m)$ algorithm to compute the convex hull of m vertices is given, and proven to be correct.

In Euclidean space, if we are given a set of m vertices, called sites, the set of points closest to a particular site x forms a polyhedron. We call this polyhedron the Voronoi cell for x. The collection of all m Voronoi cells forms the Voronoi complex, which is seen as an example of a polyhedral complex. After defining these concepts, the intricate relationship between convex hulls and Voronoi diagrams is explored. Then, Fortune's beach line algorithm for computing a Voronoi diagram is given, including a detailed description of the data structures used by the algorithm. To conclude this chapter, the book describes how to use a trapezoidal decomposition in order to perform nearest neighbor queries in $O(\log m)$ time.

Finally, Delone (Delaunay) triangulations are discussed. The Delone triangulation is the dual of the Voronoi diagram. The Delone triangulation satisfies the empty ball property: for each cell of the Delaunay triangulation, the smallest enclosing ball (the circumsphere) of the vertices contains no other vertices of the triangulation. In addition, the Delone triangulation minimizes the maximal radius of all such circumspheres. Then, the special case of Delone triangulations in $\mathbb{R}^2$ is discussed. In particular, they prove that the minimal angle of a triangulation of a given point set is maximized by using the Delone triangulation in $\mathbb{R}^2$. In addition, a comparison between Fortune's algorithm and the flip algorithm is given.

2.2 Part II: Non-linear Computational Geometry

Part II of the book slowly builds up all of the tools (namely, Gröbner bases) needed to solve non-linear systems of equations. This chapter starts with a discussion of linear algebra, and computing the zeros of a polynomial. Polynomials can be explicitly factored if the degree of the polynomial is at most four. Another problem related to factoring is how to determine if two polynomials have a common factor. If the degree of the polynomials is five or more, then factoring will not always work. Instead, one technique is to compute the determinant of the corresponding Sylvester matrix, which is a matrix consisting of coefficients of the polynomials and which has determinant zero if the two polynomials have a common factor. Section 8.3 provides a proof that this technique works.

Chapter 8 continues to define an affine-algebraic curve: $C = V(f) = \{(x, y) \in \mathbb{C}^2 : f(x, y) = 0\}$, where $f \in \mathbb{C}(x, y)$. Study's lemma relates point sets in the complex plane to divisibility of polynomials:

namely, if f is irreducible, non-constant, and $V(f) \subseteq V(g)$, then f divides g. This is a special form of the Nullstellensatz, which is covered in Chapter 10.

Studying point sets in the plane does not give a full, concise description. For example, a line and a parabola can intersect zero, one, or two times in the complex plane. However, in projective space, a line and a parabola intersect exactly twice. More generally, the intersection of curves of degrees n and m intersects mn times, as long as the corresponding Sylvester matrix has a nonzero-determinate. This is known as Bézout's Theorem.

In Chapter 9, the Euclidean algorithm (Euclid's GCD algorithm) is presented to compute the greatest common divisor of two polynomials. They give a proof of correctness, as well as discuss its relationship to the common multiple (lcm):

$$\operatorname{lcm}(f, g) = \frac{fg}{\gcd(f, g)}.$$

The major goal of Part II is to define the Gröbner basis, which, at a high level, can be described as a set of polynomials with desirable properties. We start with an ideal I, which is a set of polynomials over n variables such that for all pairs $f, g \in I$, the following polynomials are also in I: $f + g$ and fg. Each polynomial contains a special monomial called the *leading term*, which is the first term of the polynomial when the monomials are sorted. The *Gröbner* basis G is a subset of an ideal I such that the leading terms of the polynomials in G generate the initial ideal of I. In section 9.3, the authors prove that every ideal has a Gröbner basis and that every ideal has a finite generating system; the latter is known as the Hilbert basis corollary. Buchberger's theorem to compute the Gröbner basis incrementally is given in section 9.4. Chapter 9 finishes by using a Gröbner basis to prove that the intersection point of the three medians of a triangle occurs at a single point, dividing each median 2:1.

Chapter 10 begins with the computation of the Gröbner basis in `Maple` and `Singular`. Then, they cover how to extend a partial solution to a full one (and to determine if it is possible to do so). Section 10.4 covers Hilbert's Nullstellensatz. The weak form of the Nullstellensatz is as follows: if I is an ideal in $\mathbb{C}[x_1, \ldots, x_n]$ such that $V(I) = \emptyset$, then $1 \in I$. For example, this theorem implies that in order to prove f, g do not have a common root, it suffices to show that there exists a, b such that $1 = af + bg$. In this case, the pair (a, b) is called the *certificate*. The strong form of the Nullstellensatz is given as follows: for an ideal I in $\mathbb{C}[x_1, \ldots, x_n]$ and $f \in \mathbb{C}[x_1, \ldots, x_n]$ a polynomial which vanishes at all points of $V(I)$, then there exists a natural number $s \geq 1$ such that $f^s \in I$.

Section 10.5 is the climax of the book, combining everything leading up to it in order to solve a system of polynomial equations using a Gröbner bases. Then, section 10.6 uses the Gröbner basis in order to solve integer linear programs.

2.3 Part III: Applications

Part III focuses on applications of the results presented throughout the book, including graphics, robotics, and curve reconstruction.

In Chapter 11, the problem of curve reconstruction is presented. Given a sample from a curve parameterized by $f\colon [0, 1] \to \mathbb{R}^2$, the goal is to reconstruct the curve itself. To do this, the authors present the NN-Crust algorithm, which can reconstruct a densely sampled curve in $O(m \log m)$ time, where m is the number of sample points. In this chapter, the medial axis and the local feature size are also covered.

Most of the book deals with $\mathbb{R}^2$. In Chapter 12, higher dimensional spaces are considered, with particular attention paid to $\mathbb{R}^3$. Plücker coordinates are introduced as a generalization of the Grassmannian. The Grassmanian of K^n is the set of its k-dimension subspaces; for example, the 1-Grassmanian of K^n is the

set of points. An application covered in this chapter is using these tools to intersect lines with surfaces in $\mathbb{R}^3$ (i.e., computer graphics).

Chapter 13 covers applications of non-linear computational geometry. First, the generalization of Voronoi diagrams of line segments, which include parabolic arcs as bisectors, in addition to line segments and rays. One of the several applications discussed in this section is GPS positioning: how to position given inaccurate (although close) distances to GPS satellites. In order for a position to be calculated, at least four satellites are needed; however, given more than four satellites, the position can be computed with greater accuracy.

3 Opinion

This book is accessible to to advanced undergraduate or beginning graduate student in mathematics, computer science, and engineering. The authors claim that it assumes only linear algebra and calculus; however, we would argue that the reader should also be familiar with abstract algebra (in particular, fields and rings). Some of the exercises even require knowledge of topology.

The biggest shortcoming of this book is that the fact that this book was translated from German to English is very transparent. Sentence structure, word choice, and lack of commas make the book difficult to read at times for a native English speaker. As we like the topics covered in this book, we hope that a future edition will correct this issue.

We found a couple other issues with the book as well; for example, the Definition 2.1(a) should read that the projective space $P(V)$ should be the set of one-dimensional subspaces of V *which contain the origin*. Furthermore, although projective geometry is covered in Chapter 1, oriented projective geometry is not mentioned at all. Furthermore, sometimes definitions are given in either proofs or exercises, which makes them hard to pick out (see exercise 7.16, for example).

We would like to point out that there is currently no other book written from this perspective, which makes this book a useful resource despite the shortcomings mentioned above. Furthermore, the book very concisely covers a large breath of topics, and provides references to further information and to current research in the remarks at the end of each chapter. In sum, we would recommend this book for someone looking to approach algebraic and computational geometry with examples stemming from the study of polytopes.

Review of[7]
A Mathematical Orchard - Problems and Solutions
by Mark Krusemeyer, George Gilbert, and Loren Larson
Mathematical Association of America, 2012
397 pages, Softback

Review by
S.V. Nagaraj svnagaraj@acm.org

1 Introduction

Mathematics is a very engrossing subject which offers many charming problems with elegant solutions. With this in mind, Mathematical Association of America (MAA) published this problem book in its Problem Books Series. The book contains many improvements over a book titled *The Wohascum County Problem Book* previously published by the same authors with MAA in the year 1993. The new book is available in both print and ebook formats. The ISBN of the softback version is 978-0-88385-833-2 and its list price is US $49.95 while the ISBN of the ebook is 978-1-61444-403-9 and its list price is US $27.

2 Summary

The book includes a total of 208 problems along with their solutions. The earlier version of the book had 130 problems. The problems in the earlier book have been retained while 78 new ones have been added. The problems occupy just 43 pages and have been aggregated at the beginning of the book, followed by solutions which form the bulk of the book. Since this is a problem book, there are just problems and solutions. There are no chapters unlike conventional mathematics textbooks.

The problems in the book are well organized. All problems are numbered consecutively. Every problem in the problems section of the book includes the page number in which the solution is found. Many problems in the book have more than one solution included. This provides an insight to the reader about the variety of approaches that may be followed for solving a particular problem. The beauty of each such approach can thus be appreciated by the reader. The problems in the book are also arranged to some extent according to the level of difficulty in solving them. Thus problems which occur earlier are often more difficult to solve than those that come later. The solutions for the problems are elaborate and educative. Solutions may contain *lemmas, ideas* or *comments*. *Ideas* help a reader to have a quick look at the key ideas inherent in a solution. *Comments* provide a wider view of a problem.

In order to aid the reader, there are two appendices, one of which highlights the mathematical prerequisites for every problem in the book while the other classifies problems by subject. To illustrate, the prerequisites could be one or more of the following: basic properties of integers, elementary algebra, differential calculus, pre-calculus, geometric series, elementary geometry, infinite series, determinants, basic counting methods, integral calculus, elementary probability, limits of sequences and so on. Thus the reader will be able to get some hints about the math background needed for a particular problem in the book. Nevertheless, it should be noted that the prerequisites mentioned are just a guidepost to aid the reader. It may so happen that the reader may come up with a solution that follows some other approach. The information in

the appendices are supplementary. The reader is free to ignore it. The second appendix classifies problems by the subject area. Thus readers interested in a particular area say algebra and trigonometry can find a listing of problems in that area. The subject areas comprise, broadly, algebra and trigonometry, calculus, discrete mathematics, geometry, linear and abstract algebra, and number theory. Thus enthusiasts of number theory may just focus on problems involving number theory. The book includes a very helpful topic index which gives information about the occurrence of a topic in the statement or the solution(s) of a problem, information about a topic occurring in just the comment of a problem, information about a topic occurring in just the idea of a problem, and information about a topic occurring in a solution of a problem.

3 Opinion

The book presents many interesting mathematical problems along with their solutions. For some problems, more than one solution technique is demonstrated. The book in its earlier avatar was very popular. In its new incarnation, it has a very catchy title (A Mathematical Orchard) that will surely attract young readers. The readers will be tempted to taste the fruits in the mathematical orchard. The authors are highly experienced mathematics teachers who have been involved in problem setting for math competitions such as the Putnam competition. The book will continue to be absorbing to high school and college students as well as mathematics teachers. The book will be useful for students preparing for math competitions. There are many problem books such as this book. However, this book and its earlier edition are unique for their collection of original problems. A large number of problems in the book are not mundane problems. The background required is often just some algebra and calculus. The book will be useful for teachers who wish to stimulate their students. This problem book will provide joy for those who like to perfect their mathematical abilities and also for experienced problem solvers. The authors and MAA have done a good job in producing this book.