# 1. Introduction

Welcome to a beautiful subject!—the constructive approximation of functions. And welcome to a rather unusual book.

Approximation theory is an established field, and my aim is to teach you some of its most important ideas and results, centered on classical topics related to polynomials and rational functions. The style of this book, however, is quite different from what you will find elsewhere. Everything is illustrated computationally with the help of the Chebfun software package in Matlab, from Chebyshev interpolants to Lebesgue constants, from the Weierstrass approximation theorem to the Remez algorithm. Everything is practical and fast, so we will routinely compute polynomial interpolants or Gauss quadrature weights for tens of thousands of points. In fact, each chapter of this book is a single Matlab M-file, and the book has been produced by executing these files with the Matlab "publish" facility. The chapters come from M-files called `chap1.m`,..., `chap28.m` and you can download them and use them as templates to be modified for explorations of your own.

Beginners are welcome, and so are experts, who will find familiar topics approached from new angles and familiar conclusions turned on their heads. Indeed, the field of approximation theory came of age in an era of polynomials of degrees perhaps O(10). Now that O(1000) is easy and O(1,000,000) is not hard, different questions come to the fore. For example, we shall see that "best" approximants are hardly better than "near-best", though they are much harder to compute, and that, contrary to widespread misconceptions, numerical methods based on high-order polynomials can be extremely efficient and robust.

This is a book about approximation, not Chebfun, and for the most part we shall use Chebfun tools with little explanation. For information about Chebfun, see `http://www.maths.ox.ac.uk/chebfun`. In the course of the book we shall use Chebfun overloads of the following Matlab functions, among others:

`CONV, CUMSUM, DIFF, INTERP1, NORM, POLY, POLYFIT, ROOTS, SPLINE`

as well as additional Chebfun commands such as

```
CF, CHEBELLIPSEPLOT, CHEBPADE, CHEBPOLY, CHEBPTS,
LEBESGUE, LEGPOLY, LEGPTS, PADEAPPROX,
RATINTERP, REMEZ.
```

There are quite a number of excellent books on approximation theory. Three classics are [Cheney 1966], [Davis 1975], and [Meinardus 1967], and a slightly more recent computationally oriented classic is [Powell 1981]. Perhaps the first approximation theory text was [Borel 1905].

A good deal of my emphasis will be on ideas related to Chebyshev points and polynomials, whose origins go back more than a century to mathematicians including Chebyshev (1821–1894), de la Vallée Poussin (1866–1962), Bernstein

(1880–1968), and Jackson (1888–1946). In the computer era, some of the early figures who developed "Chebyshev technology," in approximately chronological order, were Lanczos, Clenshaw, Good, Fox, Elliott, Mason, Orszag, Paszkowski, and V. I. Lebedev. Five books on Chebyshev polynomials are by Snyder [1966], Paszkowski [1975], Fox and Parker [1968], Rivlin [1990], and Mason and Handscomb [2003]. One reason we emphasize Chebyshev technology so much is that in practice, for working with functions on intervals, these methods are unbeatable. For example, we shall see in Chapter 16 that the difference in approximation power between Chebyshev and "optimal" interpolation points is utterly negligible. Another reason is that if you know the Chebyshev material well, this is the best possible foundation for work on other approximation topics, and for understanding the links with Fourier analysis.

My style is conversational, but that doesn't mean the material is all elementary. The book aims to be more readable than most, and the numerical experiments help achieve this. At the same time, theorems are stated and proofs are given, often rather tersely, without all the details spelled out. It is assumed that the reader is comfortable with rigorous mathematical arguments and familiar with ideas like continuous functions on compact sets, Lipschitz continuity, contour integrals in the complex plane, and norms of operators. If you are a student, I hope you are an advanced undergraduate or graduate who has taken courses in numerical analysis and complex analysis. If you are a seasoned mathematician, I hope you are also a Matlab user.

Each chapter has a collection of exercises, which span a wide range from mathematical theory to Chebfun-based numerical experimentation. Please do not skip the numerical exercises! If you are going to do that, you might as well put this book aside and read one of the classics from the 1960s.

To give readers easy access to all the examples in executable form, the book was produced using `publish` in LaTeX mode: thus this chapter, for example, can be generated with the Matlab command `publish('chap1','latex')`. To achieve the desired layout, we begin each chapter by setting a few default parameters concerning line widths for plots, etc., which are collected in an M-file called `ATAPformats` that is included with the standard distribution of Chebfun. Most readers can ignore these details and simply apply `publish` to each chapter. For the actual production of the printed book, `publish` was executed not chapter-by-chapter but on a concatenation of all the chapters, and a few tweaks were made to the resulting LaTeX file, including removal of Matlab commands whose effects are evident from looking at the figures, like `title`, `axis`, `hold off`, and `grid on`.

The Lagrange interpolation formula was discovered by Waring, the Gibbs phenomenon was discovered by Wilbraham, and the Hermite integral formula is due to Cauchy. These are just some of the instances of Stigler's Law in approximation theory, and in writing this book I have taken pleasure in trying to

cite the originator of each of the main ideas. Thus the entries in the references section stretch back several centuries, and each has an editorial comment attached. Often the original papers are surprisingly readable and insightful, at least if you are comfortable with French or German, and in any case, it seems particularly important to pay heed to original sources in a book like this that aims to reexamine material that has grown too standardized in the textbooks. Another reason for looking at original sources is that in the last few years it has become far easier to track them down, thanks to the digitization of journals, though there are always difficult special cases like [Wilbraham 1848], which I finally found in an elegant leather-bound volume in the Balliol College library. No doubt I have missed originators of certain ideas, and I would be glad to be corrected on such points by readers. For a great deal of information about approximation theory, including links to dozens of classic papers, see the History of Approximation Theory web site at `http://www.math.technion.ac.il/hat/`.

Perhaps I may add a further personal comment. As an undergraduate and graduate student in the late 1970s and early 1980s, one of my main interests was approximation theory. I regarded this subject as the foundation of my wider field of numerical analysis, but as the years passed, research in approximation theory came to seem to me dry and academic, and I moved into other areas. Now times have changed, computers have changed, and my perceptions have changed. I now again regard approximation theory as exceedingly close to computing, and in this book we shall discuss many practical numerical problems, including interpolation, quadrature, rootfinding, analytic continuation, extrapolation of sequences and series, and solution of differential equations.

Why is approximation theory useful? The answer goes much further than the rather tired old fact that your computer relies on approximations to evaluate functions like $\sin(x)$ and $\exp(x)$. For my personal answer to the question, concerning polynomials and rational functions in particular, take a look at the last three pages of Chapter 23, beginning with the quotes of Runge and Kirchberger from the beginning of the 20th century. There are also many other fascinating and important topics of approximation theory not touched upon in this volume, including splines, wavelets, radial basis functions, compressed sensing, and multivariate approximations of all kinds.

In summary, here are some distinctive features of this book:

- The emphasis is on topics close to numerical algorithms.
- Everything is illustrated with Chebfun.
- Each chapter is a publishable M-file, available online.
- There is a bias toward theorems and methods for analytic functions, which appear so often in applications, rather than on functions at the edge of discontinuity with their seductive theoretical challenges.

- Original sources are cited rather than textbooks, and each item in the bibliography is listed with an editorial comment.

At a more detailed level, virtually every chapter contains mathematical and scholarly novelties. Examples are the use of barycentric formulas beginning in Chapter 5, the tracing of barycentric formulas and the Hermite integral formula back to Jacobi in 1825 and Cauchy in 1826, Theorem 7.1 on the size of Chebyshev coefficients, the introduction to potential theory in Chapter 12, the discussion in Chapter 14 of prevailing misconceptions about interpolation, the presentation of colleague matrices for rootfinding in Chapter 18 with Jacobi matrices for quadrature as a special case in Chapter 19, Theorem 19.5 showing that Clenshaw–Curtis quadrature converges about as fast as Gauss quadrature, the first textbook presentation of Carathódory–Fejér approximation in Chapter 20, the explanation in Chapter 22 of why polynomials are not optimal functions for linear approximation, the extensive discussion in Chapter 23 of the uses of rational approximations, and the SVD-based algorithms for robust rational interpolation and linearized least-squares fitting and Padé approximation in Chapters 26 and 27.

All in all, we shall see that there is scarcely an idea in classical approximation theory that cannot be illustrated in a few lines of Chebfun code, and as I first imagined around 1975, anyone who wants to be expert at numerical computation really does need to know this material.

Dozens of people have helped me in preparing this book. I cannot name them all, but I would like to thank in particular Serkan Gugercin, Nick Higham, Jörg Liesen, Ricardo Pachón, and Ivo Panayotov for reading the whole text and making many useful suggestions, Jean-Paul Berrut for teaching me about rational functions and barycentric formulas, Folkmar Bornemann for bringing to light historical surprises involving Jacobi, Cauchy, and Marcel Riesz, and Volker Mehrmann for hosting a sabbatical visit to the Technical University of Berlin in 2010 during which much of the work was done. I am grateful to Max Jensen of the University of Durham, whose invitation to give a 50-minute talk in March 2009 sparked the whole project, and to Marlis Hochbruck and Caroline Lasser for testing a draft of the book with their students in Karlsruhe and Munich. Here in the Numerical Analysis Group at Oxford, Endre Süli and Andy Wathen have been the finest colleagues one could ask for these past fifteen years, and the remarkable Lotti Ekert makes everything run smoothly. Finally, none of this would have been possible without the team who have made Chebfun so powerful and beautiful, my good friends Zachary Battles, Ásgeir Birkisson, Toby Driscoll, Pedro Gonnet, Stefan Güttel, Nick Hale, Ricardo Pachón, Rodrigo Platte, Mark Richardson, and Alex Townsend.

**Exercise 1.1. Chebfun download.** Download Chebfun from the web site at `http://www.maths.ox.ac.uk/chebfun` and install it in your Matlab path as instructed there. Execute `chebtest` to make sure things are working, and note the time taken. Execute `chebtest` again and note how much speedup there is now that various files

have been brought into memory. Now read Chapter 1 of the online *Chebfun Guide,* and look at the list of Examples.

**Exercise 1.2. The publish command.** Execute `help publish` and `doc publish` in Matlab to learn the basics of how the `publish` command works. Then download the files `chap1.m` and `chap2.m` from `http://www.maths.ox.ac.uk/chebfun/ATAP` and publish them with `publish('chap1','latex')` followed by appropriate LaTeX commands. (You will probably find that `chap1.tex` and `chap2.tex` appear in a subdirectory on your computer labeled `html`.) If you are a student taking a course for which you are expected to turn in writeups of the exercises, I recommend that you make it your habit to produce them with `publish`.

**Exercise 1.3. Textbook X.** Buy or borrow a copy of an approximation theory textbook, which we shall call $X$; good examples are the books of Achieser, Braess, Cheney, Davis, Lorentz, Meinardus, Natanson, Powell, Rice, Rivlin, Schönhage, Timan, and Watson listed in the References. As you work through *Approximation Theory and Approximation Practice*, keep $X$ at your side and get in the habit of comparing treatments of each topic between *ATAP* and $X$. (a) What are the author, title, and publication date of $X$? (b) Where did/does the author work and what were/are his/her dates? (c) Look at the first three theorems in $X$ and write down one of them that interests you. You do not have to write down the proof.