

Preface

This book is aimed to be both a textbook for graduate students and a starting point for applications scientists. It is designed to show how to implement spectral methods to approximate the solutions of partial differential equations. It presents a systematic development of the fundamental algorithms needed to write spectral methods codes to solve basic problems of mathematical physics, including steady potentials, transport, and wave propagation. As such, it is meant to supplement, not replace, more general monographs on spectral methods like the recently updated “Spectral Methods: Fundamentals in Single Domains” and “Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics” by Canuto, Hussaini, Quarteroni and Zang, which provide detailed surveys of the variety of methods, their performance and theory.

I was motivated by comments that I have heard over the years that spectral methods are “too hard to implement.” I hope to dispel this view—or at least to remove the “too”. Although it is true that a spectral code is harder to hack together than a simple finite difference code (at least a low order finite difference method on a square domain), I show that only a few fundamental algorithms for interpolation, differentiation, FFT and quadrature—the subjects of basic numerical methods courses—form the building blocks of any spectral code, even for problems in complex geometries. I present the algorithms not only to solve problems in 1D, but 2D as well, to show the flexibility of spectral methods and to make as straightforward as possible the transition from simple, exploratory programs that illustrate the behavior of the methods to application programs.

I assume that the reader has a basic knowledge of numerical algorithms. The most important topics include interpolation, quadrature and the numerical integration of ordinary differential equations. An understanding of other methods for the solution of PDEs, such as finite difference or finite element methods is very helpful.

Although I assume some background in numerical methods, I have tried to make the presentation and the collection of algorithms self contained. The idea is to make it as straightforward as possible to implement the methods without the need to search for auxiliary routines. Of course, some of the routines (like FFTs, for example) should be later replaced by well-tested and optimally designed versions in the programmer’s programming language of choice. Also, since I emphasize the implementation of spectral methods, I recommend having one of the many more general books, such as the two mentioned above, available to consult on the theory of the methods.

I have chosen to present the algorithms in a detailed pseudocode, rather than in a specific language such as C or Fortran or application such as Matlab, Maple or Mathematica. The idea is to present the algorithms unencumbered by a language syntax that the reader might not know, but that can nevertheless be quickly translated into the programmer’s favorite computer language. I hope that this will make

the methods useful to the widest possible audience and for the widest range of applications. This is not a programming book, however, and the translation process of the pseudocode will depend on the particular computer language chosen. Fortran programmers will most likely replace loops with vector operations. C/C++ programmers will often have to adjust array indices. The object oriented concepts used for many of the algorithms will be natural to C++ programmers, but a number of tutorials on the web for Fortran programmers show how to implement object oriented ideas using modules, at least until F2003 compilers become available.

The book consists of two parts. The first is a quick introduction to spectral approximation to provide a reference point and to define the notation. It is followed by the development of the basic algorithms that form the building blocks of spectral codes. These algorithms include how to use the complex FFT to compute real transforms, how to compute Chebyshev and Legendre polynomials and then how to use them to approximate integrals and derivatives.

The second part presents algorithms to approximate the solutions of PDEs. It includes a short survey of spectral approximations that shows how to use the building blocks in one space dimension. Our main interest, however, is how to solve problems in complex geometries in two space dimensions, so we put the emphasis on collocation and nodal versions of Galerkin spectral methods. The development of the algorithms starts from basic approximation on the square, then moves to more complex geometries through the introduction of boundary-fitted mappings, and ends with spectral multidomain methods.

I am grateful for the efforts of my students and colleagues who did so much to help me prepare this work. I'd like to thank my students Wuming Zhu, James DeMarco, Matt Willyard and Cesar Acosta who gave suggestions on what they wanted to see in a book on spectral methods and implemented algorithms. Tom Zang deserves many thanks for taking the time to read through an entire draft of the manuscript and provide detailed comments. Last, but certainly not least, I want to thank Yousuff Hussaini, who started my research in spectral methods. He has mentored me and encouraged my career for two and a half decades. Many of the topics covered here came out of projects that we have worked on together.

Tallahassee, FL

David A. Kopriva
October 2008

Implementing Spectral Methods for Partial Differential
Equations

Algorithms for Scientists and Engineers

Kopriva, D.

2009, XVIII, 397 p., Hardcover

ISBN: 978-90-481-2260-8