Preface

The research that led to the work described here was initiated during the winter of 2010 by informal discussions at the Rutherford Appleton Laboratory in the beautiful countryside surrounding Oxford. That numerical methods for optimization and the simulation of partial-differential equations using the finite-element method need to solve systems with similar structure is nothing new. Until recently, the selection of iterative methods for symmetric and quasi-definite systems was probably reduced to MINRES and SYMMLQ, which do not exploit the structure beyond symmetry. Linear algebra is a beautiful unifying topic in computational science, one that can be discussed and developed across disciplines and fields of applications. We feel that that is exactly how the present research materialized.

Background

Defining the background necessary to read this book is challenging because the material draws from the bottomless literature on linear least-squares problems and Krylov methods. We assume knowledge of basic linear algebra at an advanced undergraduate or early graduate level. We also assume that the reader has encountered a linear least-squares problem before and is able to recognize the optimality conditions of convex quadratic problems. No additional knowledge of applications, such as optimization or fluid dynamics, is required, though it might help understand the context of the numerical experiments of Chapter 7. Familiarity with factorization-based methods for least-squares problems will help the reader understand the core of the iterative methods and put things in perspective but is not strictly necessary. Part of the challenge of understanding Krylov methods is realizing that each method solves a well-defined optimization problem at each iteration. We organized the discussion so that sections describing implementation details can be skipped or skimmed if desired. This will let the reader have an overview of methods and their purpose and leave implementation details for later. Prior familiarity with one iterative method based on the Lanczos or Golub–Kahan process will greatly help but is not strictly necessary either.

Audience

The material described in the present book is suitable for a graduate audience and researchers in computational science. The application areas with which we are most familiar are numerical linear algebra, computational optimization, and computational finite-element methods. However, we encounter new applications areas every day where symmetric and quasi-definite systems occur, including machine learning, geophysics, medical imaging and numerous other fields where inverse problems play a central role, weather forecasting, and many more. The book could be used as the central reference in an advanced graduate-level course on computational methods for

least-squares problems or saddle-point systems. The material can also be selectively sampled to provide background on numerical methods in other fields. For example, Section 2.2, Chapter 4, and Sections 5.1 and 5.2 were used to provide concrete numerical methods used in a two-week accelerated course on factorization-free optimization.

Software

Computational scientists are often opinionated, and it can be difficult to convince them to use tools that are not already part of their tool set. One such tool is their favorite programming language. As it happens, we are enthusiastic about programming languages, and in an attempt to satisfy ourselves and the largest possible audience, we provide implementations in three programming languages to accompany and complement this book. Any knowledge of computational methods is incomplete and inaccurate until one got his or her hands dirty playing with an implementation, running it on test problems, examining the behavior of a method in finite-precision conditions, and understanding the results. Computational scientists rarely have the background necessary to produce the high-quality, high-usability, user-friendly software that computer scientists and computer engineers are trained to produce. We are no exception to the rule, but we did our best to make the software easy to access, easy to install, and easy to use. Time will tell if we succeeded. Readers will find implementations in MATLAB, Python, and Julia of all methods described in the book available from www.siam.org/books/sl03. We encourage all users to get in touch and let us know how they fare. We especially welcome reports letting us know what can be improved.

Acknowledgments

Mario Arioli wishes to express his gratitude to several institutions for inviting him to spend several sabbatical months in their mathematics and computer science departments: Emory University, which awarded him an Emerson fellowship and later appointed him as adjunct professor; the Berlin Mathematical School and the Technische Universität Berlin; the CIMI and the Paul Sabatier University (Toulouse); the INPT (Toulouse); and the Bergische Universität Wuppertal. He also thanks several friends for their valuable help and advice: Volker Mehrmann, Andreas Frommer, Iain Duff, and Daniel Ruiz.

Dominique Orban wishes to express his gratitude to the Rutherford Appleton Laboratory for hosting him for four sabbatical months in the Numerical Analysis Group, where this research was born. Work conducted while at Rutherford Appleton Laboratory was sponsored by EPSRC Grant EP/E053351/1. The remainder of the work was partially supported by NSERC Discovery Grant 299010-04. Much of the material here has greatly benefited from fruitful discussions with my good friend Michael Saunders. Indeed, iterative methods for linear systems and least-squares problems would not be where they are today without him.

We both also want to sincerely thank three referees who kindly provided their time as well as suggestions and advice on the text and the presentation.

Paige (1974) and Saunders (1995) have been constant sources of inspiration during the development of this research. We owe them much for paving the way ahead. We also wish to express our gratitude to Michele Benzi of Emory University for fruitful discussions.