Preface

This book is an introduction to implicit filtering, one of many derivative-free optimization methods which have been developed over the last twenty years. The audience for this book includes students who want to learn about this technology, scientists and engineers who wish to apply the methods to their problems, and specialists who will use the ideas and the software from this book in their own research.

Implicit filtering is a hybrid of a projected quasi-Newton or Gauss–Newton algorithm for bound constrained optimization and nonlinear least squares problems and a deterministic grid-based search algorithm. The gradients for the quasi-Newton method and the Jacobians for the Gauss–Newton iteration are approximated with finite differences, and the difference increment varies as the optimization progresses. The points on the difference stencil are also used to guide a direct search.

Implicit filtering, like coordinate search, is a **sampling method**. Sampling methods control the progress of the optimization by evaluating (sampling) the objective function at feasible points. Sampling methods do not require gradient information but may, as implicit filtering does, attempt to infer gradient and even Hessian information from the sampling.

imfil.m is a MATLAB implementation of the implicit filtering method. This version differs in significant ways from our older Fortran code [29]. This document is a complete reference to version 1.0 of **imfil.m**, covering installation, testing, and its use in both serial and parallel environments.

As implicit filtering has evolved since its introduction [131], so have several related approaches. The current version of implicit filtering, as reflected in this book and in **imfil.m**, uses ideas from [7, 38, 69, 92].

The plan of the book is that Chapters 2, 6, and 8 will also serve as the stand-alone users' guide to **imfil.m**.

Chapter 3 has two functions. The first is to give students who are using the book in a course or in their research enough background to enable them to put the algorithms in perspective and to understand what some of the options for **imfil.m** do. The second purpose is to introduce the notation needed to follow the algorithmic and theoretical development in Chapters 4 and 5.

Chapter 4 is an overview of the algorithms in **imfil.m** and an explanation of some of the design decisions. This chapter also illustrates the details of many of the algorithmic parameters one sets in the `options` structure. We develop the

convergence theory for **imfil.m** in Chapter 5.

Finally, in Part IV of the book we show how **imfil.m** can be applied in the context of a few problems. One of these problems is intended to be very simple (Chapter 8), and the source code should be easy for the reader to modify and play with. The other two case studies arose from research projects [28, 28, 53], and the codes for the applications were not written with this book in mind. Hence, the application code for Chapters 9 and 10 should be viewed as "black boxes." The driver codes which call **imfil.m** to solve the problems, on the other hand, should be easy to modify.

This book owes its existence to my students and collaborators who worked on the algorithm, the Fortran code [29], and the applications which drove the development. This list of implicit filtering heroes includes Astrid Battermann, Griff Bilbro, Greg Characklis, Tony Choi, Todd Coffey, Gilles Couture, Robert Darwin, Joe David, Karen Dillard, Owen Eslinger, Matthew Farthing, Dan Finkel, Katie Fowler, Joerg Gablonsky, Paul Gilmore, Deena Hannoun, Lea Jenkins, Brian Kirsch, Anna Meade, Casey Miller, Dave Mokrauer, Alton Patrick, Jill Reese, Dan Stoneking, Mike Tocci, Bob Trew, and Tom Winslow.

Sampling methods like implicit filtering have evolved significantly in the last several years, and **imfil.m** has been vastly improved by my interaction with master searchers and interpolators such as Mark Abramson, Charles Audet, Andrew Booker, Andrew Conn, John Dennis, Genetha Gray, Tammy Kolda, Michael Lewis, Jorge Moré, John Nelder, Chung-Wei Ng, Mike Powell, Luis Rios, Nick Sahinidis, Katya Schienberg, Christine Shoemaker, Virginia Torczon, Luis Vicente, Stefan Wild, and Margaret Wright.

The development of implicit filtering has been supported by several grants from the National Science Foundation and the Army Research Office, most recently NSF grants DMS-0707220, CDI-0941253, and OCI-0749320, ARO grants W911NF0910159, W911NF-07-1-0112, and W911NF-06-1-0412, and USACE contracts W912HZ-10-P-0221 and W912HZ-10-P-0256.

C. T. Kelley
Raleigh, North Carolina
March 2011