Preface

The advent of high-speed computers and sophisticated software tools has made the computation of derivatives for functions defined by evaluation programs both easier and more important. On one hand, the dependence of certain program outputs on certain input parameters can now be determined and quantified more or less automatically, i.e., without the user having to append or rewrite the function evaluation procedure. On the other hand, such qualitative and quantitative dependence analysis is invaluable for the optimization of key output objectives with respect to suitable decision variables or the identification of model parameters with respect to given data. In fact, we may juxtapose the mere *simulation* of a physical or social system by the repeated running of an appropriate computer model for various input data with its *optimization* by a systematic adjustment of certain decision variables and model parameters. The transition from the former computational paradigm to the latter may be viewed as a central characteristic of present-day scientific computing.

Optimization nowadays already forms an important application area for exact derivative calculation using algorithmic differentiation. This includes the provision of gradients and Hessians for the unconstrained case, as, for example, in nonlinear finite element calculations [Wri08], the optimal laser control of chemical reactions [BCL01] or the optimization of low-pass analog filters [Alk98]. In the constrained case, algorithmic differentiation can be used to compute the required Jacobians, Hessians, or Hessian-vector products. Here, the applications areas include chemical engineering [AB04], race car performance [CS$^+$01], and industrial production processes [KW00]. Since 1997 the Network Enabled Optimization Server (NEOS) at Argonne National Laboratory has been using algorithmic differentiation to compute gradients and Jacobians of remotely supplied user code for optimization objectives and constraints [CMM97]. The modelling language AMPL and GAMS have incorporated both modes of algorithmic differentiation to provide first and second derivatives for various optimization solvers. Also multicriteria optimization benefits from exact derivatives, e.g., to optimize medical radiotherapy [JMF06]. For the field of parameter estimation, algorithmic differentiation is used to improve, for example, weather models [GK$^+$06] or the simulation of the climate [KHL06]. The cover shows the independence between various vector quantities in optimal design as discussed in Chapter 15 superimposed on a global sensitivity map. It was provided to us by Patrick

Heimbach of MIT and represents the derivative of the poleward heat transfer with respect to local maritime temperatures.

Moreover, even the pure simulation may be improved by the provision of exact derivative information. In this context, algorithmic differentiation may be used to compute exact Jacobians for the integration of stiff ordinary differential equations (ODEs). Numerical quadrature is probably the first—and numerical integration of ODEs and differential algebraic equations (DAEs) are still the most important—application areas for third and higher-order derivative evaluations. For example, derivatives of arbitrary high-order may be required for the integration of DAEs modeling multibody systems [NP05]. Many computational schemes for nonlinear problems are known to work well if certain higher-order terms, often error estimates, are small. The heuristic procedures employed to estimate these terms are quite unreliable, so the ability to evaluate them up to working accuracy should greatly improve algorithmic performance and robustness. In other words, the availability of selected analytical derivatives can facilitate the intelligent use of adaptive and higher-order methods, which may otherwise be somewhat unreliable or simply wasteful in unfavorable circumstances that could not be properly detected beforehand. Subsequently, the simulation may be subject of a sensitivity analysis, where the required derivatives can be provided exactly by algorithmic differentiation. Here, current research topics cover, e.g., the analysis of financial instruments like options [Gil07]. Algorithmic differentiation is used successfully also in the context data assimilation, e.g., for flood modeling [CD$^+$06].

Of course, our distinction between simulation and optimization is not a sharp dichotomy. Many practical calculations have always incorporated optimization methods, for example, to determine stable equilibria of certain subsystems with phase transitions. However, what may be characteristic for modern optimization calculations is that the number of decision or design variables can grow so large that estimating their effect on the objective function(s) becomes the most expensive task of the overall calculation. This happens in particular when unknown functions like geometrical shapes or distributions of material properties need to be discretized, which leads to a virtually unlimited number of variables or parameters [AC92]. Using a discrete analog of adjoint differential equations, we will be able to calculate the gradient of a scalar-valued function at a temporal complexity not exceeding that of five function evaluations, regardless of the number of independent variables. This is being used extensively in weather forecasting based on four-dimensional data assimilation [Tal08].

This book should be almost invaluable to the designers of algorithms and software for nonlinear computational problems. It should also be especially useful for the users of current numerical software, which usually still places on them the responsibility for providing derivative values, sparsity patterns, and other dependence information. Even without reading the more theoretical sections of the book, the reader should gain the insight necessary to choose and deploy existing algorithmic differentiation software tools to best advantage. Finally, we hope that there will be a significant number of readers who are interested in algorithmic differentiation for its own sake and who will continue

or begin to contribute to this growing area of theoretical research and software development.

The earlier books *Automatic Differentiation: Techniques and Applications*, by Louis Rall [Ral81], and *Numerical Derivatives and Nonlinear Analysis*, by Kagiwada et al. [KK+86], cover only the *forward* or *direct* mode of what we call here *algorithmic differentiation*. Most material in this book concerns the "reverse" or "adjoint" mode as well as hybrids between forward and reverse that promise lower operation counts at acceptable storage requirements. Many newer results and application studies were published in the proceedings [GC91], [BB+96], [CF+01], [BC+06], and [BB+08] of the international workshops on algorithmic, or computational, differentiation held in Breckenridge (1991), in Santa Fe (1996), in Nice (2000), in Chicago (2004), and in Bonn (2008). They also contain excellent introductory articles and an extensive bibliography. Apart from some papers of historical interest, we have therefore listed in our bibliography only those sources that are directly pertinent to developments in the text.

Acknowledgments

The plan for writing the original version of this book was hatched with Bruce Christianson. We are indebted to Olaf Vogel, Klaus Röbenack, Mike Giles, and many colleagues, who have commented on the first edition and made suggestions for the second. Section 6.2 on source transformation and section 6.3 on parallelism were largely provided by Christele Faure and Jean Utke, respectively. To help improve the readability for novices in the field, the second author reorganized the book's contents. As with the first edition, Mrs. Sigrid Eckstein composed and revised the LaTeX source with its many tables and diagrams.