Featured Review: Mathematica GuideBooks.

The Mathematica GuideBook for Numerics. *By Michael Trott.* Springer-Verlag, New York, 2006. $79.95. xxxvi+1208 pp., hardcover. ISBN 0-387-95011-7 (with DVD-ROM).

The Mathematica GuideBook for Symbolics. *By Michael Trott.* Springer-Verlag, New York, 2006. $79.95. xl+1453 pp., hardcover. ISBN 0-387-95020-6 (with DVD-ROM).

This review covers the GuideBooks for *Numerics* and *Symbolics*. Both are part of Michael Trott's mammoth four-volume set of *Mathematica* GuideBooks. The GuideBooks total 5,179 pages, 14 chapters, 107 sections, 4,700 illustrations and over 10,000 references, some of which overlap. There are 278 problems, all solved, resulting in 1650 pages of solutions. Each volume is available separately and deals with one of the key features of *Mathematica*: programming, graphics, numerics, and symbolics. Although the books can be read independently, Trott suggests reading the volumes in that order. I prefer to learn about the programming and symbolic capabilities of *Mathematica* before reading about the more applied numerical and visualization features. The overall scope and features of the guidebooks were addressed in an earlier review [11], which also covered the 1,000-page *Programming* and 1,300-page *Graphics* installments.

Numerics GuideBook. When Nick Trefethen launched the *SIAM 100-Digit Challenge* in the January/February 2002 issue of *SIAM News* [10], I wondered how well a team of *Mathematica* experts would fare in the contest. Like many, I often struggled with the numerous options of *Mathematica's* numerical functions to get a specified number of digits with confidence. To my delight, Michael Trott was a member of one of the 20 winning teams. Indeed, "Team Mathematica" computed at least 10 correct digits for all 10 problems, using the numerical features discussed in the numerics GuideBook.

The numerics volume has two chapters. The first chapter covers *Mathematica's* sledgehammer tools for handling numerical problems in floating-point arithmetic. Through an abundance of examples, one learns how to use *Mathematica* to solve systems of algebraic and differential equations, integrate numerically, compute Fourier transforms, and fit functions to data. The second chapter deals with exact calculations with integers and integer-valued functions.

In the 520-page Chapter 1, Trott describes the various options of *Mathematica* numerical functions in great detail. He visits the following issues in high-accuracy computation: interval arithmetic, appropriate scaling, numerical quadrature, and limit and series evaluations.

In the opening section, Trott gives striking examples to show that *Mathematica's* high-precision and exact arithmetic is superior to machine arithmetic. One example involves the numerical testing (up to many hundreds of digits) of a complicated exact solution of Chazy's equation; another shows how to find bounds on step sizes in a forward Euler discretization method to compute trajectories of the Rössler system.

In section 1.1, Trott discusses precision and accuracy, in user-guided or automated fashion. He shows in great detail how to work with $MinPrecision, $MaxPrecision, $MaxExtraPrecision, SetPrecision, WorkingPrecision, PrecisionGoal, AccuracyGoal, and SetAccuracy. This is where I knew the least and could have learned the most. Yet I failed because I lost interest in this highly technical material. Nonetheless, sections 1.1 and 1.6 (on monitoring precision internally) are required reading for anyone who wants to use *Mathematica* for reliable numerical computations. In contrast, I enjoyed reading about the various strategies and formulas to compute π. Aficionados of this subject should consult [2, 4, 13] for additional references.

In section 1.1.2, Trott turns to interval arithmetic which is currently only available for elementary functions and for real arguments. The computation of a global attractor of a rational map is used to illustrate the use of interval arithmetic.

Often one can take advantage of the exact arithmetical capabilities of *Mathematica* by converting approximate numbers into exact numbers. For example, successive iterations of a Newton-Raphson solver can be performed in exact arithmetic, which can be too slow. Alternatively, using the RATIONALIZE function between iterations, one can convert the floating-point decimal output into a rational input with a desired high accuracy. Trott does not discuss this example. Instead, in section 1.1.3, he gives several applications involving continued fractions. Under the title "When N Does Not Succeed" in section 1.1.4, Trott explains how to avoid frustrations when *Mathematica's* key function, i.e., N, fails to give the specified precision. Section 1.1.5 covers packed arrays with an application to string searches in Shakespeare's *Hamlet*. This is a nice example showing the power of the packed array approach.

Section 1.2 on fitting data and interpolation based on powerful commands such as FIT, FINDFIT and INTERPOLATINGPOLYNOMIAL, will be of great interest to whoever has to process data analytically or graphically. The function INTERPOLATINGPOLYNOMIAL only works for univariate polynomials, but Trott shows how to use *Mathematica* cleverly to carry out multivariate interpolation. There is a lot to learn from the challenging examples in this section.

The material in section 1.3 on compiling and optimizing functions is quite technical. Yet Trott prevents it from being boring by showing fast computations of Julia sets, Gosper curves, and periodic attractors. I especially enjoyed the graphical analysis of the behavior of a sand pile as extra grains of sand are added.

In section 1.4 on linear algebra, Trott focuses on commands for computations with matrices with numerical entries, such as LU and Schur decompositions, and the computation of singular values and null spaces. Using a large electrical grid with resistors, one can see the efficiency of *Mathematica*'s linear system solvers first hand. In the context of such applications, Trott offers tricks for computations with sparse matrices. Particular attention is paid to the precision of the built-in linear algebra functions such as EIGENVALUES, EIGENVECTORS and EIGENSYSTEM.

Catering to an audience of applied scientists and engineers, section 1.5 is devoted to data analysis with the discrete Fourier transform and LISTCONVOLVE and LISTCORRELATE functions. The numerical Fourier transform, based on the algorithm of the Fast Fourier Transform, is applied to numerous data sets in 1D and 2D from real-world problems in signal and image processing.

The rest of Chapter 1 covers the options and subtleties of the main numerical functions. NSUM and NPRODUCT are reviewed in section 1.7. Trott points to [7], a web site where one can submit the first few terms of an integer sequence in the hope of receiving a closed-form formula to build the general sequence. It works quite well and it is fun trying.

Section 1.8 is devoted to the numerical approximation of integrals, where Trott offers a rare glance behind the scenes at the methods used by NIntegrate. The solution of algebraic equations with NSolve, NRoots, FindRoot is covered in section 1.9 with ample examples, including graphs of Hofstadter's butterfly from quantum physics and Voderberg polygons.

Finding local extrema of functions with FindMinimum (and FindMaximum) is addressed in section 1.10 and applied to a spiral-shaped function. Little attention is devoted to the complicated problem of finding global extrema with the command NMinimize. As an aside, one of the problems of the *SIAM 100-Digit Challenge* was solved with a one-liner in *Mathematica* using NMinimize. The methods behind the various minimization functions are briefly outlined.

Section 1.11 deals with the numerical solution of ordinary and partial differential equations (PDEs) with NDSolve, which is the most complicated and most user-extensible function within *Mathematica*. Trott only discusses the options most relevant to the applied sciences and engineering. Currently, NDSolve solves standard initial-value problems, linear boundary-value problems, differential-algebraic systems, and partial differential equations in 1+1 dimensions. It does not yet solve Sturm-Liouville eigenvalue problems or delay differential equations.

Through carefully selected examples, including quite complicated systems of nonlinearly coupled and stiff differential equations, Trott showcases the capabilities and options of NDSolve. He solves differential equations describing spiral waves in chemical reactions, chaotic attractors in 4D, the motion of coupled pendulums, and many more. The most striking examples to me involve the numerical solution of Chazy's equation, the Ablowitz-Ladik system of differential-difference equations, and Kepler's 3D problem for the motion of three attracting bodies. With respect to PDEs, Trott solves the 1D wave equation, a time-dependent Schrödinger equation, and Fokker-Planck and Klein-Gordon equations. He also carries out experiments with nonlinear Schrödinger and Ginzburg-Landau equations. Through these examples, the reader can acquire the skills of solving differential equations in high-precision arithmetic.

Chapter 1 concludes with two larger applications. The first one deals with the calculation and visualization of electric and magnetic field lines; the second teaches how to visualize Riemann surfaces of algebraic functions. Both lead to stunning graphs, and by modifying the code one can create new art work.

Chapter 2 covers exact integer calculations and integer-valued functions used in classical analysis. Trott describes how calculations with integer and rational numbers with an arbitrary number of digits allows one to experimentally verify mathematical identities and conjectures. Clever use of exact arithmetic with hundreds of digits can help discover new identities. Occasionally, Trott shows multiple approaches for solving a problem or compares various implementations and programming styles.

Although I have used *Mathematica* for years, I knew little about using the Compile function (discussed in section 1.3) for fast computations with exact numbers. It is fun to read about Trott's experiments with Schönberg's Peano curve, Maurer roses, and de Bruijn medallions and friezes. The application of the Möbius μ function in the calculation of Fourier coefficients in section 2.2 is also delightful reading. In sections 2.3 and 2.4, one can learn about the Stirling, Euler, Bernoulli, and Fibonacci numbers, and the discussion is continued in the exercises, which offer a treasure-trove of identities and formulas.

It is, of course, impossible to discuss all options settings of the numerical functions in detail. For additional information, Trott refers to the Advanced Documentation in the Help Browser. Through carefully selected problems, often solved with multiple approaches, he shows how to avoid the pitfalls resulting in numerical round-off errors. My advice is that if you plan to tackle a new computational problem with *Mathematica* you should consult similar problems in the numerics GuideBook to get a grip on the appropriate functions (and their options).

Symbolics GuideBook. This guidebook has three chapters. In the first chapter, Trott reviews the symbolic capabilities of *Mathematica*. There is little overlap with the programming GuideBook which focuses on the structure of *Mathematica* expressions rather than on functions for "symbol crunching," such as FACTOR, SOLVE, DSOLVE, and REDUCE.

Chapters 2 and 3 showcase the classical orthogonal polynomials and special functions of mathematical physics respectively. Both are within Trott's areas of expertise for he holds a Ph.D. in theoretical physics and is one of the key developers of Wolfram's Functions Site [8].

The 800-page first chapter offers a tour of *Mathematica* commands for simplifications (section 1.1) and operations on polynomials (section 1.2), rational functions (section 1.3), and trigonometric expressions (section 1.4).

Trott uses creative applications from diverse fields to describe quantifier elimination and operations on polynomial inequalities. For example, with the RESOLVE function, one can prove inequalities and derive new ones. Although one does not learn how RESOLVE actually works, section 1.2.3 is worth reading.

In section 1.5, Trott offers many hints on how to steer *Mathematica* on the right path to a successful solution of algebraic equations. Here one learns why SOLVE only produces the solution $x = 0, y = 1$ of $x + y = 1, kx + y = 1$, and why REDUCE should be used instead. On pp. 127-128, one finds an excellent summary of which functions to use to analyze and solve linear systems with parameters.

One of the central functions in Chapter 1 is GROEBNERBASIS, with its many bells and whistles. Working with Gröbner bases in *Mathematica* is challenging, especially when systems with unknowns as well as parameters must be reduced and subsequently solved in a desired order [9]. By selecting the correct term ordering within GROEBNERBASIS, a computation that took hours may be reduced to a couple of seconds. Although Trott offers hints for playing with orderings and options in an interactive session, it is unclear how to do this in automated code.

The functions for symbolic calculus are reviewed in section 1.6. The differences between D, DT, and DERIVATIVE are clearly explained. The inserts on Riemann curvature tensors, Christoffel symbols, geodesic equations, evolutes, and phase integrals bring the text to life.

Section 1.6.2 discusses the symbolic evaluation of indefinite and definite integrals, single and multiple integrals alike. This section has excellent examples. I particularly enjoyed those involving the Korteweg-de Vries, Schrödinger, and Camassa-Holm equations, and the computation of minimal surfaces.

Trott alludes to the pitfalls and shows some integrals that were incorrectly computed, but fails to mention that INTEGRATE cannot handle simple expressions involving unspecified functions. For example, differentiate the following expression (with respect to x) : $f'^2 \cos f - g \cos f + cg'$, where $f(x)$ and $g(x)$ are differentiable functions and c is constant. Then, try to integrate the result to get back to the original expression. *Mathematica* fails unless $c = 0$. Yet this and similar integrals are straightforward to compute with an easy-to-implement homotopy method of integration [12]. The remaining subsections of 1.6 cover limits, series expansions (including ways to build $q-$Taylor series), residues, and sums. Many creative examples and exercises come from the work of Ramanujan.

I was particularly interested in how *Mathematica* would fare in solving differential and difference equations, as discussed in section 1.7. My former Ph.D. student, Ünal Göktaş, was hired by Wolfram Research Inc. to greatly enhance the capabilities of DSolve and RSolve. So, I was curious to see how things have changed. DSolve now solves the most popular classes of ODEs, some differential-algebraic equations, and a limited class of PDEs. Trott shows successes and failures and points out that DSolve does not yet reduce ODEs to the Painlevé normal form, nor does it recognize Painlevé-type equations. Difference equations (recurrence relations) can be solved with RSolve. Unfortunately, as with DSolve, there is no list of the types of equations Rsolve can handle. One just has to try and hope for the best!

Leaving smooth functions behind, section 1.8 covers unit-step and Dirac δ functions and Fourier and Laplace transforms. Two modern applications, one about compacton solutions of the nonlinear Schrödinger equation and the other about Adomian decomposition, brighten the rather technical presentation.

In section 1.9, under the title "Additional Symbolic Functions," Trott points to packages for vector analysis, variational methods, and the like. Chapter 1 ends with three large applications. The one about Gauss's ingenious method to express $\cos(2\pi/17)$ in closed analytic form, typically in terms of nested square roots, is quite interesting. In the late 1800s, Gauss's method was used to compute $\cos(2\pi/257)$ and $\cos(2\pi/65537)$. The latter took more than 10 years to calculate by hand whereas it now takes about a day on a modern workstation.

Chapters 2 gets us into classical orthogonal polynomials. After listing some common properties in section 2.1, the usual cast of polynomials appears section by section: Jacobi, Legendre, Hermite, Laguerre, Gegenbauer and Chebyshev polynomials. Trott emphasizes the formulas that define these polynomials and demonstrates how *Mathematica* can be used to compute series expansions, as well as to differentiate, integrate, and visualize them. To break the repetitiveness, most sections have one or more applications of orthogonal polynomials, often related to mathematical physics and quantum mechanics. The use of Gegenbauer polynomials in section 2.4 as a method to smooth the Gibbs phenomenon (encountered in Fourier analysis) is noteworthy. After a brief discussion of relationships among the orthogonal polynomials (section 2.9), Chapter 2 ends with an application of Hermite polynomials in the computation of the ground-state energy of a quantized quartic oscillator.

Chapter 3 covers a selection of special functions [8] that are commonly used in scientific applications. The chapter gets off to a slow start with 20 pages of introductory remarks, but then rapidly covers the Gamma, Polygamma, Beta, Error, Bessel, Airy, and Mathieu functions. The elliptic, exponential, sine, and cosine integrals are covered as well as Legendre and hypergeometric functions, and the product log function.

In every section one finds applications from physics, mathematics, and engineering. Paging through Chapter 3, I paused at the analyses and graphs of the oscillations of membrane surfaces of drums of various shapes. The final section of Chapter 3 deals with quintic polynomials for which some of the material can be found on the “Solving the Quintic with *Mathematica*” poster [3].

For more comprehensive treatments of orthogonal polynomials and special functions, I suggest visiting the (forthcoming) Digital Library of Mathematical Functions [1] at the National Institute of Standards and Technology (NIST), and the Encyclopedia of Special Functions [5] at INRIA, the French National Institute for Research in Computer Science and Control.

I enjoyed reading the symbolics Guidebook. More so than the numerics GuideBook, in part, because I am more interested in symbolic computing than number crunching. Furthermore, reading about the applications strengthened my problem-solving skills and taught me how to select *Mathematica* commands judiciously.

Looking back at all four volumes. After my excitement about the first two volumes [11], I became more critical as I plowed through the third and fourth volumes. Undoubtedly, the GuideBooks could have been better organized, with a clearer presentation in crisper language. This is particularly true for the numerics GuideBook, where one lacks tables summarizing the effect of option-settings on accuracy and precision. One could quibble about the notebook format in small font, for it makes the books look like the printout of a decade’s worth of *Mathematica* sessions, mixed with function descriptions, explanations, commentary, mathematics, physics, and an occasional historical tidbit.

Reflecting Trott’s background and expertise, a disproportionate number of applications deal with classical and quantum mechanics. The coverage is terse and lacks the background material to allow one to fully appreciate the aim of the applications. I would have preferred fewer examples, fewer applications, and more detail of the mathematics and physics needed to set up the problems.

The same can be said about the implementations. Less code would have been better. An expert in *Mathematica* can learn at lot from Trott’s code. A newcomer, however, will have to take the code apart to understand how it works. Presenting smaller pieces of code would have provided space for lucid descriptions of how the code was put together in the first place.

On the positive side, the GuideBooks take the reader on a thrilling tour of the features of *Mathematica*. I am impressed with the breadth and depth of Trott’s coverage and his profound understanding of the strengths and limitations of *Mathematica*. The GuideBooks offer a staggering number of original examples, creative applications, and engaging exercises. Often, I succumbed to the temptation to work on some of the many problems.

Each book includes a multiplatform DVD-ROM which allows the reader to experiment with code and view graphics in color. The small black and white pictures in the guidebooks pale in comparison to the high-resolution color graphs one can display on a monitor. In many ways, the DVD version of the GuideBooks is superior to the printed version. In addition to electronic features such as hyperlinking, navigation palettes, and animation, with the DVD one can experiment with and build upon Trott’s code.

The DVD is a terrific asset for whomever foregoes buying the complete set of GuideBooks. Indeed, buying one of the GuideBooks at the economical price of $79.95, gives one access on DVD to the complete text and executable codes from all four volumes. Surely, the books will age as *Mathematica* grows. Hopefully, Trott will continue to list corrections, offer updates, and add new material on the accompanying website [6].

Trott's four-volume set of *Mathematica* GuideBooks offers the most comprehensive discussion of *Mathematica* available. Subsuming Wolfram's *Mathematica Book* [14], the GuideBooks are destined to be an invaluable resource and classic reference for scientists who use *Mathematica* in teaching or research.

References

[1] Digital Library of Mathematical Functions, National Institute of Standards and Technology (NIST), `http://dlmf.nist.gov/`.

[2] Internet Resources for Computation of π and Bibliography, `http://math.fullerton.edu/mathews/n2003/PiComputationBib.html`.

[3] Solving the Quintic with *Mathematica*, MathSource 0207-122, `http://library.wolfram.com/examples/quintic/`.

[4] Talking about π, `http://www.cecm.sfu.ca/~jborwein/pi_cover.html`.

[5] The Encyclopedia of Special Functions, Institut National de Recherche en Informatique et en Automatique (INRIA), `http://algo.inria.fr/esf/`.

[6] The Mathematica GuideBooks Website, `http://www.MathematicaGuideBooks.org`.

[7] The On-Line Encyclopedia of Integer Sequences, `http://www.research.att.com/~njas/sequences`.

[8] The Wolfram Functions Site, `http://functions.wolfram.com/`.

[9] D. Baldwin et al., *Symbolic computation of exact solutions expressible in hyperbolic and elliptic functions for nonlinear PDEs*, J. Symb. Comp., 37 (2004), pp. 669-705.

[10] F. Bornemann et al., *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*, SIAM, Philadelphia, PA, 2004.

[11] W. Hereman, *Featured Review: Mathematica GuideBooks: The Mathematica GuideBook for Programming and the Mathematica GuideBook for Graphics*, SIAM Rev., 47 (2005), pp. 801-806.

[12] W. Hereman et al., *Continuous and Discrete Homotopy Operators and the Computation of Conservation Laws*, in Differential Equations with Symbolic Computation, Trends in Mathematics, D. Wang and Z. Zheng, eds., Birkhäuser Verlag, Basel, 2005, pp. 249-285.

[13] S. Walsh, Pi on the Web, `http://faculty.ed.umuc.edu/~swalsh/Math Articles/Pi.html`.

[14] S. Wolfram, *The Mathematica Book*, Fifth Edition, Wolfram Media, Champaign, IL, 2003.

Willy Hereman
Colorado School of Mines