Scientific Programming 15 (2007) 117–119
IOS Press


# Book Review

**Understanding and Implementing the Finite Element Method** by Mark Gockenbach, SIAM, Philadelphia, USA, 2006. ISBN-10: 0-89871-614-4, ISBN-13: 978-0-898716-14-6

In *Understanding and Implementing the Finite Element Method*, Mark Gockenback addresses a critical current issue in the teaching of finite element methods. Classes based on texts such as Strang and Fix [15], Brenner and Scott [5], or Ciarlet [6] typically focus heavily if not exclusively on mathematical analysis and do not prepare students to make actual computations. On the other hand, classes taught from engineering texts such as Hughes [13], Zienkewicz [16] or the first volume of the Texas Finite Element series [4] do not provide a thorough enough theoretical presentation. Unless they intend to specialize in numerical computation, many students only get one or two classes to learn about numerics. What should they learn? Gockenbach strikes a balance between essential theory and very practical, hands-on details. His book is quite accessible, the theory is well-motivated, and prepared students should not have difficulty following it.

This book divides into four sections and fifteen chapters. The first part provides a basic motivation and introduction variational and finite element formulations of some model steady PDE. This is all developed using the standard mechanics of vector calculus such as Green's identities. A welcome feature of the introductory material is that it goes beyond the Poisson equation to include the system of linear elasticity. Part I introduces Galerkin methods, piecewise polynomial spaces, and provides an overview of the basic convergence theory. This includes a survey of results on Hilbert spaces up through the Riesz Representation Theorem and illustrations of how the model PDE considered satisfy the coercivity condition.

Part II considers the basic data structures and implementation of the finite element method. Starting from the perspective of elementwise construction of the stiffness matrices and load vectors, he develops necessary data structures for storing mesh connectivity information. Essentially, these are integer arrays much as would be used in a FORTRAN implementation, but glued together in a single MATLAB data structure. Then, he deals with constructing such meshes by various techniques, including refinement of existing meshes and MATLAB's built-in Delaunay triangulation algorithm. Following this, Gockenbach provides a thorough treatment of many important details in the context of linear Lagrange triangles, including local-to-global mapping, quadrature for forming element matrices and vectors, and imposing various kinds of boundary conditions. This is very concrete and well-presented, if a translation of FORTRAN into MATLAB. Chapter 8 purports to deal with arbitrary order Lagrange elements, and includes some of the more interesting as well as some of the more troubling aspects of the book. It includes a nice treatment of isoparametric elements, often missing from basic texts. However, his approach to arbitrary order Lagrange polynomials entails expressing them as linear combinations of the standard monomial basis, inverting a Vandermonde matrix on an equispaced lattice along the way. While this is fine for low order elements, something more is needed to overcome poor conditiong as the element degree becomes "arbitrary".

After handling the implementation of the mesh and forming the system of equations, Gockenbach turns to linear solvers in Part III. Much of this material is standard fare but accessibly written, covering direct methods before proceeding to conjugate gradient, stationary iterations, and multigrid algorithms. Two nice features that are somewhat nonstandard for books of this ilk are treatments of hierarchical basis conjugate gradients and handling the null space resulting from a pure Neumann problem. Gockenbach's treatment of solvers is good so far it goes, but suffers from the same omission as many other finite element books. Lawrence Evans writes in the introduction to his book on PDE [10] that we know too much about linear problems and not enough about nonlinear ones. I claim that something analagous is true of symmetric versus nonsymmetric matrices. While the theory of symmetric matrices is much simpler and leads to elegant analysis of powerful algorithms, many important problems in the world are nonsymmetric. It is a shame that students are not provided with a couple of comprehensible sections overviewing GMRES, giving them an iterative solver capable of handling nonsymmetric systems.

Finally, Part IV handles basic issues related to adaptive refinement. This breaks naturally into two chapters – one on handling mesh data structures and another on some simple but practical error estimators and indicators. Much as before, the discussion is practical, includes code, and focuses on essential issues. While readers will not come away with a survey of the literature of *a posteriori* error estimation, they will gain a few examples of the kinds of estimates, how to store a nested mesh, and see the power of adaptive algorithms on a sequence of nontrivial examples.

Gockenbach's attempt to marry theory and practice in a single volume is not the first. The 2004 book of Guermond and Ern [9] entitled *Theory and Practice of Finite Elements* covers similar ground with some notable differences of scope and audience. The analysis in Guermond and Ern book is more advanced, including stability and convergence theory for mixed methods. They also include a wider range of equations (including first order equations and time-dependent problems) and methods (including least squares and discontinuous Galerkin). The implementation portion of *Theory and Practice* centers less on MATLAB and does not include source code as Gockenbach's book does, but on the other hand includes coverage of some other very practical issues such as reordering linear systems and basic parallelization. It provides a wider and deeper range of topics than Gockenbach, but fewer "nuts and bolts". If Guermond and Ern is a book attempting to get mathematicians to compute, Gockenbach's book is an attempt to get beginning engineering and science graduate students a brief introduction to mathematics before diving into practical issues.

Despite the provided code base, one could also argue that this book, like many others, implicitly encourages the detrimental practice of "roll-your-own" software so prevelant in scientific computing. The community around a journal such as *Scientific Programming* should emphasize bringing more modern and powerful computing practices to bear on numerics. There are many excellent, freely available solver libraries such as PETSc [2] and Trilinos [12]. Students gaining familiarity with such libraries have a considerable benefit, as they have access to the latest solvers ready to run on parallel machines. While MATLAB encourages software reuse in the form of calling its (quite good) existing solvers, it does not provide such an easy entry point to the realm of high-performance parallel computing in the same way that PETSc or Trilinos does. Additionally, one could point to the emergence of several high-level open source finite element packages such as DEAL.II [3], Sundance [14], or FEniCS [1] and domain-specific languages such as GetDP [7] or FreeFEM [11]. While there is definite pedagogical value in having students see low-level code for meshes and element integration, this must be weighed against the time constraints. Finite element methods are very intricate to implement. A one-semester class might be better focused on having students learn basic theory (as Gockenbach covers), surveying some implementation details with reference to existing code, and using the high-level code to highlight basic stability and convergence theory and delve into some other applications. The 1996 text, *Computational Differential Equations* by Eriksson et al. [8], attempts this with a code called FEMLab. Perhaps a similar treatment with more modern solver technology, a similar basic mathematical focus, and more powerful freely available available code, would fill the niche for which Gockenbach aims even better than this practical and readable text.

## References


[1] The FEniCS Project. http://www.fenics.org.

[2] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L. Curfman McInnes, B.F. Smith and H. Zhang, PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[3] W. Bangerth, R. Hartmann and G. Kanschat, deal.II Differential Equations Analysis Library, Technical Reference. http://www.dealii.org.

[4] E.B. Becker, G.F. Carey and J. Tinsley Oden, *Finite elements. Vol. I. The Texas Finite Element Series,* I. Prentice Hall Inc., Englewood Cliffs, NJ, 1981. An introduction.

[5] S.C. Brenner and L. Ridgway Scott, *The mathematical theory of finite element methods, volume 15 of Texts in Applied Mathematics,* Springer-Verlag, New York, second edition, 2002.

[6] P.G. Ciarlet, *The finite element method for elliptic problems,* Studies in Mathe-matics and its Applications, Vol. 4, North-Holland Publishing Co., Amsterdam, 1978.

[7] P. Dular and C. Geuzaine, *GetDP: a General Environment for the Treatment of Discrete Problems,* http://www.geuz.org/getdp.

[8] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, *Computational Differential Equations,* Cambridge University Press, Cambridge, 1996.

[9] A. Ern and J.-L. Guermond, *Theory and practice of finite el-ements, volume 159 of Applied Mathematical Sciences,* Springer-Verlag, New York, 2004.

[10] L.C. Evans, *Partial differential equations, volume 19 of Graduate Studies in Mathematics,* American Mathematical Society, Providence, RI, 1998.

[11] F. Hecht and A. Le Hyaric, *FreeFEM++ Web page,* http://www.freefem.org/ff++/index.htm.

[12] M. Heroux, R. Bartlett, V.H.R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tumi-naro, J. Willenbring and A. Williams, An Overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[13] T.J.R. Hughes, The finite element method. Linear static and dynamic finite element analysis, With the collaboration of Robert M. Ferencz and Arthur M. Raefsky, Prentice Hall Inc., Englewood Cliffs, NJ, 1987.

[14] K. Long, Sundance, *A rapid prototyping tool for parallel PDE-constrained optimization,* In Large-Scale PDE-Constrained Optimiza-tion, Lecture notes in computational science and engineering. Springer-Verlag, 2003.

[15] G. Strang and G.J. Fix, *An Analysis of the Finite Element Method,* Prentice-Hall Series in Automatic Computation, Prentice-Hall Inc., Englewood Cliffs, NJ, 1973.

[16] O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method,* Vol. 1. Butterworth-Heinemann, Oxford, fifth edition, 2000. The basis.

*Robert C. Kirby*
Department of Mathematics and Statistics
Texas Tech University
Lubbock, TX, USA