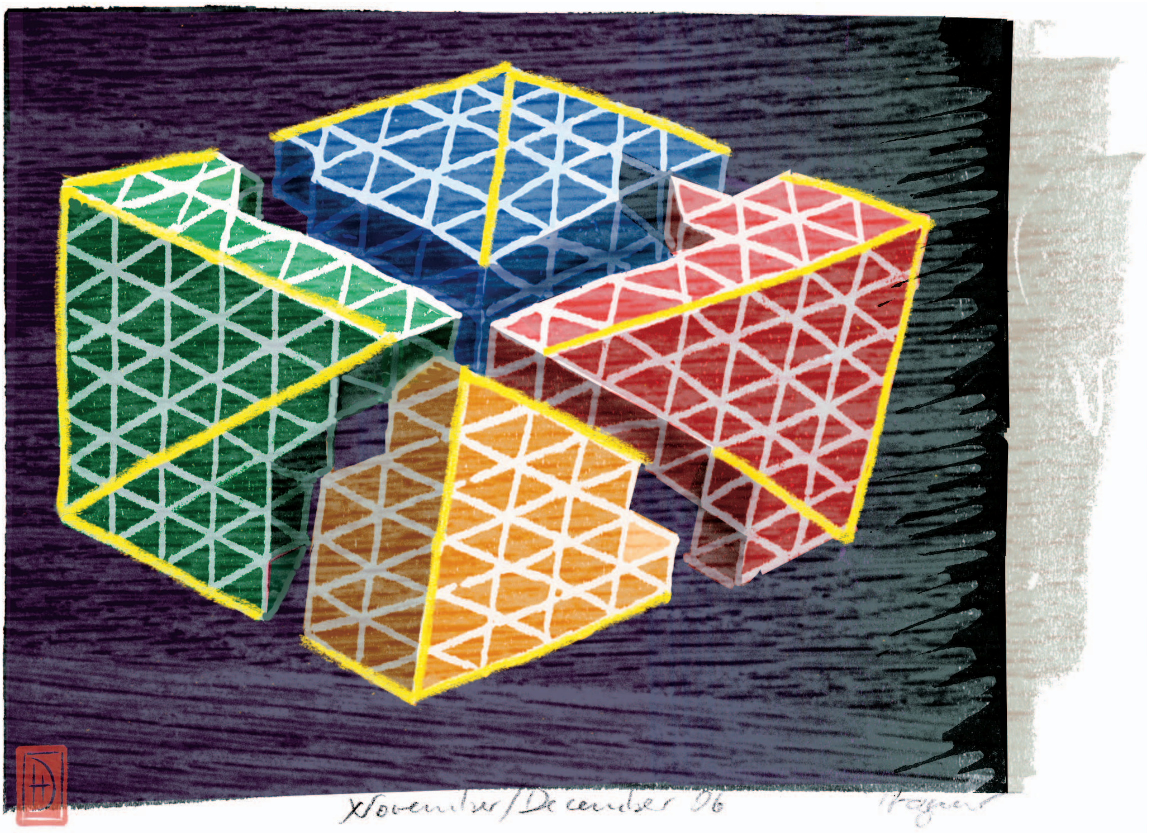
GUEST EDITORS' INTRODUCTION

Multigrid Computing



Multigrid methods are among the most important algorithms for computational scientists because they're the most efficient solvers for a wide range of problems. The modern era of multigrid methods began more than 30 years ago with the publication of Achi Brandt's seminal papers.[1,2]

Although it was originally successful for solving elliptic partial differential equations (PDEs) and the linear systems that arise when they're discretized, the basic multigrid principle of coupling multiple scales has much wider applicability.

Why a Special Issue?

At first glance, it might seem strange that such a fundamental task as the solution of linear systems is still worth a special issue in a computational science magazine in 2006. In fact, the solution algorithms for linear systems are usually hidden from the computational scientist—say, behind Matlab's backslash operator or within state-of-the art commercial finite element packages or computational fluid dynamics tools. Ideally, computational scientists shouldn't have

1521-9615/06/$20.00 © 2006 IEEE
Copublished by the IEEE CS and the AIP

Steve McCormick
University of Colorado, Boulder
Ulrich Rüde
Universität Erlangen-Nürnberg

to worry how a linear system is being solved, so why, then, did we bring together a special issue on it?

The answer is that the linear solver is still the computational bottleneck in terms of complexity for many mesh-based numerical simulations. Although we can usually perform data pre- and postprocessing linearly in the number of mesh points and then parallelize it with standard partitioning techniques, the complexity of most linear system solvers—direct or iterative—is worse than linear. Multigrid methods are among the few exceptions. In practice, the solver's complexity might not dominate small- or moderate-sized problems, but using multigrid methods can offer tremendous gains in efficiency for large-scale computing. One more aspect to the complexity argument concerns *asymptotically optimal* solvers—their overall cost is linear or close to linear in the number of unknowns. A classical example is a fast Fourier transform- (FFT-) based algorithm with $N\log N$ complexity.

An important issue is the constant in the complexity estimates because it ultimately decides which algorithm is best. Although much of the mathematical analysis involved doesn't exhibit these constants, we can get a solution for simple enough problems by using multigrid methods with fewer than 30 elementary arithmetic operations per unknown. A well-implemented multigrid algorithm is thus likely to be more competitive in time to solution, the ultimate metric for a computational scientist. Furthermore, unlike FFT-based methods, we can generalize multigrid methods to more complex domains, boundary conditions, differential operators, and so on.

But if multigrid methods are so good, why doesn't Matlab's backslash simply run a multigrid solver? Unfortunately, the multigrid algorithm draws its strength from exploiting the underlying differential equation's structure. Thus, we can't easily use it as a generic linear solver—rather, we often have to customize it for each individual problem.

Moreover, multigrid algorithms can't solve all linear systems efficiently (probably not even all sparse systems), so we can't easily connect them to standard linear algebra interfaces. Multigrid algorithms are unforgiving in the sense that optimality immediately disappears if the developer makes a single mistake, whether a coding error or a misconception (for example, how a fine-scale feature in the problem is transferred to the coarser resolutions). Developing a truly efficient multigrid method for a given problem is far from trivial.

The Articles

Irad Yavneh presents a basic introduction to multigrid methods that illustrates the fundamental principles in a very readable way. The article is well suited as basic material both for teaching multigrid methods and for nonspecialists in engineering or science classes that deal with numerical methods.

Robert Falgout describes the basics of algebraic multigrid (AMG) solvers, a variant of multigrid methods that attempts to reconstruct a grid-like hierarchy when given only the linear system. Because AMG tries to make multigrid usable via the usual linear algebra interfaces, this development has high practical relevance. Indeed, most multigrid users are likely to employ some form of an algebraic multigrid method eventually.

The three other articles in the issue explore common extensions of multigrid methods: a discussion of multigrid methods for systems by Cornelis Oosterlee and Francisco José Gaspar Lorenz; an excursion into multigrid-based adaptive finite element methods by Peter Bastian and Christian Wieners; and an article on very large-scale computing by Benjamin Bergen, Tobias Gradl, Frank Hülsemann, and Ulrich Rüde.

Computational scientists who must solve very large problems might have to understand multigrid methods; in some cases, they might even have to develop their own multigrid solvers. This special issue should help *CiSE* readers understand the potential of multigrid methods and start multigrid developments of their own.

References

1. A. Brandt, "A Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary Value Problems," *Proc. 3rd Int'l Conf. Numerical Methods in Fluid Mechanics*, LNP 18, H. Cabannes and R. Temam, eds., Springer-Verlag, 1973, pp. 82–89.
2. A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, vol. 31, 1977, pp. 333–390.

***Steve McCormick** is a professor at the University of Colorado, Boulder. His research interests include multigrid methods, adaptive mesh refinement, and least-squares finite element methods. McCormick has a PhD in mathematics from the University of Colorado. He is a member of SIAM. Contact him at stevem@colorado.edu.*

***Ulrich Rüde** is a professor at the Universität Erlangen-Nürnberg. His research interests include computational science and high-performance computing. Rüde has a Dr. habil. in computer science and applied mathematics from Technische Universität München. He is a member of SIAM and editor in chief of the* SIAM Journal on Scientific Computing*. Contact him at ruede@informatik.uni-erlangen.de.*