The Scope of Multiresolution Iterative Computations

By Achi Brandt

Some of the major bottlenecks in science and technology today are computational in nature. To calculate the mass of elementary particles from first principles, for example, would require millions of computing *years* on a modern supercomputer with state-of-the-art algorithms. Any refinement of the theory could increase computational demands by orders of magnitude. Or consider the determination of the three-dimensional form into which given proteins fold, which is needed for an understanding of their biological interactions and would possibly pave the way to the development of precise methods for designing new proteins. In principle this is just a computational task—but in practice current computational capabilities fall short. Likewise, many central engineering endeavors, from the design of fusion reactors and airplane maneuvers to the search for oil in the ground or tumors in a body, would be greatly facilitated if infinite computer power were available—or if much better algorithms could be devised.

Why do we need zillions upon zillions of computer operations to solve such problems? Are there some general mathematical approaches that would allow us to compute fast the processes that nature brings about so readily and surely?

Spatial Calculations: Origins of Computational Complexity

Most computational superproblems have one feature in common: They are *spatial* problems, that is, their variables are functions defined on the physical continuum of space and/or time. Alternatively, the problems can have a host of discrete variables—*particles*—each attached to a position in space, possibly changing in time. Closer examination reveals that the computational complexity of these problems results directly from this spatial nature, in several general ways that come up again and again, in different guises, with a tendency to compound each other many times over.

For a spatial problem to be treatable by computers, it must first be *discretized*: The continuum is replaced by a discrete set of points—a *grid*—and each continuum function is replaced by a *grid function* (or *"configuration"*), represented by a set of discrete variables, each attached to a certain point of the grid. The physical laws of the continuum are replaced by a system of computable algebraic relations between these discrete variables: a system of algebraic equations in some cases, of statistical relations in others. (This discretization is of course more trivial when the original problem is already stated in terms of particles.)

The first, and most readily understood, source of the enormous computational complexity of spatial problems is the sheer size of this algebraic system: To properly approximate the continuum and resolve the interesting phenomena, the grid must be sufficiently fine. The number, n, of discrete variables thus tends to be huge, especially in full four-dimensional space/time problems (not to mention the problems of much higher dimension that quite often arise in physics). The number of algebraic relations between the variables must of course be equally large.

The complexity is greatly increased, sometimes even squared, by the procedures for processing these algebraic relations in order to extract from them the desired information. This is easy to see when the interactions are *global*, i.e., derived from long-range physical forces such as gravity; in such cases each algebraic relation couples all, or $O(n)$, variables, so that the total number of terms in the system is $O(n^2)$. The number of computer operations needed to *solve* such a system of global equations seems to be at least $O(n^2)$, and is often much larger.

In many cases the basic physical laws can be cast as *local* relations, such as partial differential equations, or local statistical rules (a Hamiltonian composed of local interactions). When discretized, the resulting algebraic system will be local too; each algebraic relation will involve only a small number of neighboring variables. The total number of terms in the system will then be only $O(n)$. If, for example, each algebraic relation is actually a linear equation, the system will be described by a sparse $n \times n$ matrix containing only $O(n)$ nonzero entries.

In this case, however, the complexity is increased in another way. To preserve the sparsity of the relations, the numerical processing is usually local too. For example, each step of the processing may consist of satisfying one discrete equation (or simulating one statistical relation) by changing one unknown; at most, it may consist of treating several neighboring relations by changing several neighboring variables. Passing with such local processing over the entire grid is called a *relaxation sweep* (or a *Monte-Carlo sweep*, in the statistical case). The purpose of iterating such sweeps is to drive the system toward a solution (or to produce a statistically representative sample). The difficulty is that this goal will necessarily be attained very slowly. Many sweeps will be needed to reduce an initial error substantially (or to produce a new representative sample, essentially independent of the initial one). This malady, called "slow convergence" by numerical analysts and "critical slowing down" by statistical physicists, is a direct corollary of the local character of the processing. Indeed, it is the nonlocal features—such as smooth components of the grid functions—that are slow to converge (or to change, in the statistical case).

Each grid supplies corrections to the equations of the next coarser grid, and to the approximate solution of the next finer one.

A further source of increased complexity is the need to repeat the calculations many times over. In statistical simulations, for example, it is not enough to produce one representative sample; each such sample is likely to exhibit a large statistical deviation from any average quantity to be extracted. To average out these deviations sufficiently, many statistically independent samples should be produced; their numbers often run into the millions. The solution of a system of equations is also routinely repeated over and over, for a variety of reasons—to optimize and/or identify some parameters on which the system depends; to solve an inverse problem; to follow the bifurcation diagram or the time evolution of the solution, etc.

Remedy: Multiresolution

Other major maladies of spatial calculations are mentioned below, but for readers who are becoming impatient, a discussion of some remedies is in order. The cornerstone of all such remedies is the archvillain itself: the spatial origin of the problem. Any problem with that origin can be discretized and treated not only on one grid, but on a hierarchy of increasingly fine grids with, for example, a scale ratio of 1:2. With such a system of grids (or "levels"), it is possible to design numerical procedures that greatly benefit from intergrid (interscale) iterative interactions. This is true even when the original problem is given in terms of particles; increasingly coarse grids that describe increasingly smooth collective motions of the particle ensemble can then be constructed.

Perhaps the best known example of such multiscale iterative processing is the multigrid solver for steady-state differential equations. As mentioned above, the usual local processing of the equations on some given fine grid is inefficient because it only very slowly affects nonlocal features of the solution. The starting point of the multigrid solver is the observation that these nonlocal features that cause the trouble on the fine grid are exactly the ones that could be resolved, and inexpensively processed, on suitable coarser grids. The solver thus involves relaxation sweeps not only on the given grid, but also on all coarser ones, with intergrid transfer of information. Typically, each grid supplies corrections to the *equations* of the next coarser grid, and to the approximate *solution* of the next finer one.

This process leads to a solution of a system of n local equations (the equations of the finest employed grid—whether linear or nonlinear) in just $O(n)$ computer operations. Such a solver is also ideally suited to parallel processing machines: Using $O(n)$ processors, it will typically require only $O((\log n)^2)$ parallel steps.

An analogous algorithm can be used in statistical simulations. Employing Monte-Carlo sweeps on grids covering all the scales of the problem, the algorithm will require only $O(n)$ operations to produce a new, statistically independent configuration.

What about the case of *global* equations, including discretized integral equations, or particle problems with long-range forces, all of which seem, by definition, to be of $O(n^2)$ complexity? Interestingly, multigrid solvers have been designed that, again, need only

Continued on page 9

Multiresolution Computations,

continued from page 8

$O(n)$—sometimes $O(n \log n)$—operations to produce a solution. They are based on the natural smoothness of interactions between distant variables: the greater the distance, the coarser the grid needed to represent their interactions with enough accuracy.

Surprisingly, a multiresolution approach can also eliminate most of the above-mentioned work related to the *repetition* of calculations many times over. In the statistical case, for instance, it is the less local features (e.g., the smoother components) that actually require more sampling; local features are already abundantly sampled in every single configuration, since they recur spatially, in each small subdomain. Hence, it is only at the coarser levels of a (properly structured) multigrid Monte-Carlo simulation that many repetitions are needed, and there they are very inexpensive.

In evolution problems, e.g., in fluid dynamics, fine-scale flow structures are, to a first approximation, just convected by large-scale streams; hence, over long periods, calculations can be confined to coarse levels, provided certain terms of fine-to-coarse defect correction (fine-grid corrections to the coarse-grid equations) are convected with the flow, representing the effect of the finer structures. Only once in a great while, and only locally in space, should calculations at finer levels be performed to find the deformations in those structures and to update the defect-correction terms correspondingly.

More generally, large problem domains should almost always be treated only by suitably coarse computational grids, with rare and localized activation of finer grids, in a careful hierarchical manner. In this way the number of active discrete variables can be vastly reduced. What makes it possible to activate finer levels very sparingly is the basic idea that their task is to correct the coarse-grid *equations* rather than the coarse-grid *solution*.

Typical of autonomous systems, the coarse-grid equations produced by such a process in one region can serve equally well in others, if the size of the region increases appropriately with the level of coarsening. In particular, this leads to a computational method for *deriving macroscopic equations from microscopic physics*. Unlike group renormalization methods, which attempt to derive such equations once and for all, in the multigrid approach a certain, suitably bounded amount of iterativeness will usually remain. Namely, at certain extreme conditions the coarse-level processing will locally transfer control to recursively finer levels, which in turn will update the coarse-level equations. It can be shown by examples that this iterativeness is essential for keeping the coarse equations *simple*: They need not once and for all foretell *all* possible relations. The amount of work required to activate finer levels can be kept modest compared with the normal coarse-level work, because it can be done sufficiently rarely. This can also be viewed as a computational tool for overcoming the problem of *closure* of large-scale equations.

Several generic types of relations between fine and coarse emerge. A finest level consisting of particles may give rise, after several coarsening steps, to equations equivalent to those obtained by discretizing differential equations. Conversely, a discretized differential equation (e.g., a wave equation) at larger scales may produce particles (or rays). A finest-level stochastic system may become deterministic after several coarsenings. Conversely, negligible microscopic stochasticity may be quickly amplified by coarsening, yielding macroscopically a stochastic system. This latter process may prove to be an effective approach for computing chaotic spatial problems.

Some additional important tasks for multiresolution processing stem from other major factors of inefficiency that plague many spatial calculations. One of them arises typically in optimization, or energy-minimization, problems, such as protein folding and discrete optimization. At each step in the iterative solution of such problems, some local unknowns are changed so as to lower the energy. Much like the relaxation sweeps mentioned above, this local processing is doomed to converge slowly. But this is often not the main difficulty. Far more troublesome is the fact that the process is likely to be attracted to a false basin, which does not contain the desired minimum at all. Such false attraction basins can sometimes be escaped through the addition of a certain, gradually diminishing amount of stochasticity to the process, a technique called "simulated annealing." But spatial problems are likely to exhibit *multiscale* attraction basins: small ones within larger ones within still larger ones—corresponding in fact to scales of moves in the physical space. Simulated annealing is then ineffective and should be replaced by an interesting multiresolution iterative process called "multilevel annealing."

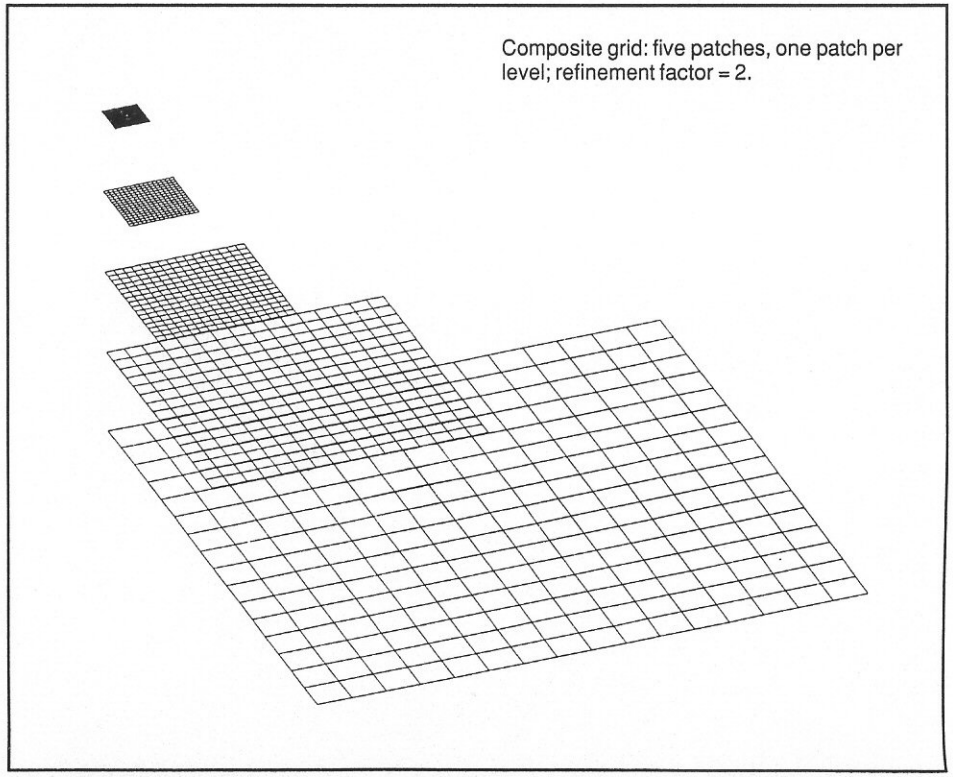
A multiresolution algorithm has been designed that can calculate in $O(n \log n)$ operations the *determinant* of an $n \times n$ matrix of grid equations, a crucial task that needs to be repeated at each step of elementary particle calculations. Other very effective types of multiresolution procedures are being developed in such diverse fields as image processing, tomography, and mathematical programming (for problems of a spatial nature).

More challenging problems await further research in the evolving field of multiscale computations. The research, on the one hand, exhibits the usual interdisciplinary and cross-fertilizing role of applied mathematics, in that various underlying relations and algorithmic ideas are carried back and forth between widely varying areas of applications. On the other hand, it has its own internal systematic development, gradually increasing our understanding of the various types and operational modes of interscale interactions, through analytical and computational studies of carefully selected model problems.

For further reading and references, see the author's articles "Multilevel Computations: Review and Recent Developments" (in Multigrid Methods: Theory, Applications and Supercomputing, *edited by S.F. McCormick; Marcel-Dekker, 1988) and "The Weizmann Institute Research in Multilevel Computations: 1988 Report" (in* Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods, *edited by Mandel et al.; SIAM, 1989). A good perspective on the basics of the multigrid solver can be obtained from the bestseller* A Multigrid Tutorial *by William L. Briggs (SIAM, 1987).*

The upcoming Multigrid Short Course, to be held at the University of Colorado, Denver, May 14-18, 1990, will feature in-depth coverage of the topics discussed in this article (see the advertisement in SIAM News, *January 1990, page 10; for further information, contact (303) 556-4807, (303) 556-2341, or cliu@copper.colorado.edu).*

Achi Brandt is the Elaine and Bram Goldsmith Professor of Applied Mathematics at the Weizmann Institute of Science, Rehovot, Israel. He is currently visiting the Department of Mathematics at the University of Colorado, Denver.



Composite grid: five patches, one patch per level; refinement factor = 2.