Multiscale Scientific Computation: Review 2000

Achi Brandt*
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract

Most of the fundamental problems in physics, chemistry and engineering involve computation too hard even for future supercomputers, if conventional mathematical approaches are used. The reason is always a product of several complexity factors associated with the wide range of space and time scales characteristic to such problems. Each of these complexity factors can in principle be removed by various multiscale algorithms, i.e., employing separate processing at each scale of the problem, combined with interscale iterative interactions. A wide range of multiscale computational methods is described, emphasizing main ideas and inter-relations between various fields. The reported areas include: top-efficiency multigrid methods in fluid dynamics; inverse PDE problems and data assimilation; feedback optimal control; PDE solvers on unbounded domains and on adaptable grids; wave/ray methods for highly indefinite equations; rigorous quantitative analysis of multigrid; many-eigenfunction problems and ab-initio quantum chemistry; fast evaluation of integral transforms on adaptive grids; multigrid Dirac solvers; fast inverse-matrix and determinant calculations and updates; multiscale Monte-Carlo methods in statistical physics, including the renormalization multigrid (RMG) methods; molecular mechanics (including fast force summation, fast macromolecular energy minimization, and Monte-Carlo methods at equilibrium, both for macromolecules and for large ensembles of small molecules); combination of small-scale equilibrium with large-scale dynamics; image processing (edge detection and picture segmentation); tomography (medical imaging and radar reconstruction); efficient, general and highly accurate algebraic multigrid (AMG) and numerical homogenization schemes; fast practical graph algorithms; data clustering; and multiscale approaches to global optimization.

*Research supported by AFOSR and the Materials and Manufacturing Directorate, AFRL, Wright-Patterson Base, contract No. F33615-97-D-5405, by the European Office of Aerospace Research and Development (EOARD) of the US Air Force, Contract F61775-00-WE067, by Israel Ministry of Science, Culture and Sport grant 9680, by Israel Absorption Ministry, Project No. 6682, by Israel Science Foundation grant No. 696/97, and by the Carl F. Gauss Minerva Center for Scientific Computation at the Weizmann Institute of Science.

Contents

1 Introduction

1.1 Multiscale computation

Despite their dizzying speed, modern supercomputers are still incapable of handling many most vital scientific problems. This is primarily due to *the scale gap*, which exists between the microscopic scale at which physical laws are given and the much larger scale of phenomena we wish to understand.

This gap implies, first of all, a huge number of *variables* (e.g., atoms or grid-points), and even a much larger number of *interactions* (e.g., one force between every pair of atoms). Moreover, computers simulate physical systems by moving *one variable at a time*; as a result, each such move must be extremely small, since a larger move would have to take into account all the motions that should in parallel be performed by all other variables. Such a computer simulation is particularly incapable of moving the system across large-scale *energy barriers*, which can each be crossed only by a large, and unknown, simultaneous motion of very many variables.

This type of computational obstacle makes it impossible, for example, to calculate the properties of nature's building blocks (elementary particles, atomic nuclei, etc.) from a certain known underlying theory — and thereby to confirm the theory itself. Likewise, such obstacles are the main bottleneck in the drive to *computerize chemistry*: to replace expensive experiments with computer simulations, yielding detailed understanding of molecular structures and interactions, creating the ability to design materials and processes, with enormous potential benefits for medicine, biotechnology, agriculture, material sciences, industrial processing, etc. Similar scale-born slowness factors and barriers, multiplying each other, plague many other engineering and scientific endeavors. All would be greatly facilitated if unlimited computing power were available — or if much better algorithms could be devised.

Just building ever faster machines will not do, in fact. With current computational methods the needed amount of computer processing often increases too steeply with the rise in problem size, so that no conceivable computer will be adequate. Completely new mathematical approaches are needed.

Past studies have demonstrated that all scale-born complexities can be effectively overcome, or drastically reduced, by *multiscale* ("multi-resolution", "multilevel", "multigrid", etc.) algorithms.

Indeed, any many-variable problem defined in the physical space can have an approximate description at any given length scale of that space: a continuum problem can be discretized at any given resolution; average motions of a many-particle system can be represented at any given characteristic length; etc. The multiscale algorithm recursively constructs a *sequence* of such descriptions at increasingly larger (coarser) scales, and combines local processing (relaxation of equations, simulation of statistical relations, etc.) at each scale with various inter-scale interactions. Typically, the evolving solution (or the simulated equilibrium) on each scale recursively dictates the *equations* (or the Hamiltonian) on coarser scales while supplying large-scale corrections to the *solutions* (or configurations) on finer scales. In this way large-scale changes are effectively calculated on coarse grids, based on information previously gathered from finer grids.

As a result of such multilevel interactions, the fine scales of the problem can be employed very sparingly, and sometimes only at special and/or representative small regions. Moreover, the inter-scale interactions can eliminate all kinds of scale-associated difficulties, such as: slow convergence (in minimization processes, PDE solvers, etc.); critical slowing down (in statistical physics); ill-posedness (e.g., of inverse problems); large-scale attraction basin traps (in global optimization and statistical simulations); conflicts between small-scale and large-scale representations (e.g., in wave problems); numerousness of long-range interactions (in many body problems or integral equations); numerousness of long-range (non-local) eigenfunctions (e.g., in quantum chemistry); the need to produce many fine-level solutions (e.g., in optimal control) or very many fine-level independent samples (in statistical physics); etc. Also, the evolving large-scale equations bring out the large-scale dynamics, or the macroscopic equations, of the physical system, which is often the very objective of the entire calculation.

Since the local processing (relaxation, etc.) in each scale can be done in parallel at all parts of the domain (e.g., at all cells of a given lattice), the multiscale algorithms, based on such processing, are ideal for implementation on massively parallel computers. Indeed, many problems cannot be efficiently solved by such computers without employing a multiscale procedure. For example, to fully parallelize a time-dependent calculation (i.e., to compute for earlier and later times simultaneously), a multiscale (multigrid) algorithm must be used (see Sec. 3.1).

Often, a combination of several multiscale approaches can benefit one particular problem in *many* different ways (see examples in Secs. 4.3, 9 and 14.2 below. Also the different multiscale algorithms discussed in Secs. 11, 12 and 13 are all parts of solving the same "grand challenge" problem of elementary particles).

Multilevel computation has evolved into a discipline by itself, having its own internal development, gradually increasing our understanding of the many types of multiscale interaction, their modes of operation and domains of application. Various underlying relations and algorithmic ideas are carried back and forth between widely varying types of problems.

1.2 Do you need multiscale algorithms?

A multiscale computation is usually considerably more complicated than the more common algorithms. Also, for small problems it is often more expensive. So when do you need to go into this extra trouble? The most obvious and important sign for such a need is that you have a *computational bottleneck* associated with the increasing "size" of the problem. If the computational cost rises more than linearly with the number of variables, or if the number of variables is so large that even linear-scaling algorithms would be too expensive, then a multiscale approach, or a combination of several multilevel procedures, may well be the answer.

To be sure, not *every* difficult computational task can be usefully multi-scaled. Intractable, undecidable and other impossible problems typical to theoretical computer science (see for example [101]) cannot be efficiently solved by *any* algorithm. Interestingly, however, these are mostly *man-made* problems. The computational tasks in natural sciences and engineering are not of this type. Their complexity usually results from a multitude of variables (particles, picture elements, a discretized function, etc.), most of which are usually positioned in some low dimensional spaces. The experience is that all problems of this type can benefit from multiscaling, which yields either low-complexity (normally linear-scaling) solvers or "macroscopic equations", i.e., the means for large-scale coarse simulations, derived from computations in just small fine-scale windows. The various sections of this review gives many examples of both these possibilities.

Note also that some problems can benefit from multiscaling because this is the best way to *formulate* the problem, or some parts of it (see Sec. 18.1).

1.3 The present survey

The present report has been written as a thorough updating and modification of [38], many parts of which had previously appeared in [35]. The first chapters summarize important recent techniques, and some less known older ones, in the field of multigrid PDE solvers, assuming a general familiarity with its basic elements. (For introductory books, see [67] and [153]; or at least read the "Elementary acquaintance with multigrid" in [24]; see also the basic insight described in Sec. 17 below.) Some algorithms and concepts are explained in more detail than others, mainly because they are more recent. Later chapters introduce a variety of other fields of multiscale computation, including fast matrix multiplication, integral and integrodifferential equations, statistical physics, chemistry, image processing and tomography.

In particular, the report surveys the main ideas, current developments and future perspectives in the following directions.

1. New top-efficiency multigrid methods for steady-state fluid dynamics at all Mach and Reynolds numbers, and other non-elliptic stationary PDE systems (see Sec. 2 below).

2. Multilevel approaches to time-dependent partial-differential equations, emphasizing a fast method for solving an implicit-time-step system of equations (sometimes faster than an explicit time step), and parallel processing and grid adaptation across both space and time (see Sec. 3).

3. *Grid adaptation* techniques exploiting multigrid structures and creating a one-shot solver-adaptor (Sec. 6.1). Similar techniques for treating prob-

lems in *unbounded domains*, costing essentially the same as in bounded domains.

4. Direct multigrid solvers for inverse problems, including system identification (e.g., impedance tomography; see in Sec. 16.2) and data assimilation (in atmospheric simulations — Sec. 4), showing multiple benefits of several kinds of multiscaling employed in one problem. The solution of an ill-posed problem can often cost far less than its well-posed counterpart.

5. Optimal control: Feedback control via very fast updating of open-loop solutions, based on their multiscale representations (Sec. 5).

6. Optimal location of singularities of PDE systems (e.g., finding the minimal-total-energy location of the nucleons in electronic structure calculations), integrated into a one-shot multigrid solver (Sec. 9.1).

7. Top-efficiency multigrid algorithms for highly indefinite (e.g., standing wave) problems, featuring ray equations (geometrical optics) at the limit of large scales with wave equations at small-scale regions where ray formulations break down (Sec. 7).

8. Multigrid solvers for the Dirac equations arising in quantum field theory (Sec. 11).

9. Compact multiresolution representation of the inverse matrix of a discretized differential operator; fast updating of the inverse matrix and of the value of the determinant upon changing an arbitrary term in the matrix itself; with application to the QCD fermionic interaction (Sec. 12).

10. Collective multiscale organization of eigenbases and $O(N \log N)$ calculation of N eigenfunctions of a differential operator, e.g., the Schrödinger operator in condensed-matter electronic-structure calculations (Sec. 9.2).

11. Calculation of the N roots of the secular equation in $O(N)$ operations.

12. Multiscale Monte-Carlo algorithms for eliminating both the critical slowing down and the volume factor in increasingly advanced models of statistical physics (Sec. 13).

13. Multigrid Monte-Carlo approaches for solving the high-dimensional (several-particle) Schrödinger equation by real-time path integrals (Sec. 18).

14. Introducing multiscale computations to many-particle (macromolecule or many-small-molecule) calculations, including fast evaluation of forces, fast convergence to ground states, fast Monte Carlo simulations and large time steps, with application to molecular mechanics (Sec. 14); a new approach to molecular dynamics, based on stochastic implicit time steps (Sec. 14.8).

15. Multigrid methods for fast dense-matrix multiplications, integral transforms and for integro-differential equations, on adaptable grids, with applications to tribology (Sec. 10).

16. Multiscale methods for the fast evaluation and inversion of the Radon transform and other line-integral transforms (Sec. 16.1); applications to medical tomography and radar reconstruction.
17. Multiscale algorithms for early vision tasks such as surface reconstruction, edge and fiber detection (Sec. 15.1) and segmentation (Sec. 15.2).
18. Multilevel clustering and other graph algorithms (Sec. 15.3).
19. Rigorous quantitative theory for predicting the performance of multigrid solvers (Sec. 8).
20. New efficient, general and accurate approaches for coarsening algebraic systems of equations, yielding very efficient and general algebraic-multigrid (AMG) solvers, as well as a general technique for numerical homogenization (Sec. 17).
21. Multilevel strategies for solving global optimization problems that harbor many local minima and nested multiscale attraction basins, including multilevel approaches for *formulating* fuzzy optimization problems (Sec. 18).
22. Some thoughts about wavelets (Sec. 19).

2 Steady-State Fluid Dynamics

2.1 Objective: textbook multigrid efficiency

An efficient multigrid algorithm for steady-state incompressible viscous flows in two dimensions appeared already in 1972 [18], a relatively efficient multigrid solver for a compressible inviscid transonic flow was demonstrated in 1975 [145], and a fully efficient solver for a *system* of *several* coupled differential equations, characteristic of computational fluid dynamics (CFD), was presented already in 1978 [40]. However, in the decades that followed, the development in this area has not been fully satisfactory. In particular, the efficiency of solvers for non-elliptic steady-state systems (such as Euler and high-Reynolds Navier-Stokes equations) has lagged several orders of magnitude behind the ideal efficiency that had been attained for general elliptic systems. Although the main reasons for this inefficiency have also been understood for a long time (see for example [22]), the recommended cures seemed complicated, and code developers opted for partial efficiency. The leading multigrid method has been based on multi-stage pseudo-time-stepping relaxation schemes [105], [104]. Although such schemes can be optimized to damp high-frequency errors [157], the resulting algorithms are still relatively slow, because some *intermediate* (neither high-frequency nor very smooth) "characteristic components" cannot adequately be reduced by coarse grids (cf. [22], [64]). Other multigrid solvers were based on incomplete LU decomposition (ILU) and related relaxation schemes [159], [155], [144]. While such schemes give excellent results in some cases, they cannot cure the aforementioned trouble of characteristic components in general transonic flows, especially in three dimensions. (Also, much of the efficiency of ILU schemes depends on their sequential marching, hence the performance on massively parallel machines will drastically diminish.) The same is true for other methods (e.g., based on defect corrections) which seem not even to identify that basic trouble.

More generally, all these attempted solution methods have failed to decompose the solution process into separate treatments of each factor of the PDE principal determinant, and therefore did not identify, let alone treat, the separate difficulties associated with each such factor. The fact is that, in a typical CFD problem, each of these factors may have different ellipticity measures (some are uniformly elliptic, others are non-elliptic at some or all of the relevant scales) and a different set of characteristic surfaces, requiring for top efficiency different relaxation/coarsening procedures.

Thus, the objective of the recent work has been to develop and demonstrate methods that solve non-elliptic *steady-state* problems in general, and high-Reynolds stationary flow problems in particular, at the same "*textbook multigrid efficiency*" attained for uniformly elliptic systems. This means, typically, to obtain an $O(h^2)$ approximation to the differential solution on a grid with meshsize h at a cost of just few (less than 10) "*minimal work units*", this unit being the amount of operations involved in the simplest discretization of the differential problem on the meshsize-h grid. The methods, again as in the elliptic case, will allow local refinements (cf. Sec. 6.1) and high degree of parallel processing. (For general remarks about *time-dependent* problems, see Sec. 3).

2.2 Problem decomposition

As shown in the past (see [24], [28] and [64]), to obtain that "textbook" multigrid efficiency for any discretized partial differential system of equations (PDE), it is necessary and usually (with proper boundary treatment) also sufficient to attain that efficiency for each factor of the PDE principal determinant. Each such factor is a scalar differential operator of first or second order, so its efficient solution is a vastly simplified task. The way for separating the factors is by a *distributed* (and possibly also *weighted*) relaxation scheme in which to each factor there corresponds a "*ghost*" discrete function. The latter can be directly relaxed for its corresponding factor, dictating a resulting pattern of changes to be distributed to the *actual* discrete functions (see details in [24, §3.7] and also in [163], and examples in Secs. 17–20 of [24]). To obtain the top efficiency, the relaxation of each ghost function should incorporate an essential part of an efficient multigrid solver for its corresponding operator: sometimes this is just the relaxation part of that solver, sometimes this may even be the entire solver (applied at some proper subdomain).

For the *incompressible* Euler and Navier-Stokes equations, the relevant factors are the Laplace and the convection (or convection-diffusion) operators. The former's multigrid solver is classical; the latter's can be based on downstream relaxation [64], with additional special procedures for recirculation flows [65],

[166]. Indeed, incorporating such procedures into the relaxation schemes for the appropriate ghost functions yields very efficient solvers for incompressible flows even at high Reynolds numbers and at second-order accuracy [64]. The same procedures will also yield efficient solvers for compressible flows at low Mach numbers, where the relevant factors are similar.

The most important remaining factor of flow systems for which no general adequate multigrid solver has been developed until recently is the "*full potential*" operator

$$(u\partial_x + v\partial_y + w\partial_z)^2 - a^2\Delta \ , \tag{2.1}$$

where (u, v, w) is the flow velocity vector and a is the speed of sound. This operator appears as a factor in the principal determinant of the 3-D *compressible* Euler equations. Its *Mach number* is the ratio $M = (u^2 + v^2 + w^2)^{1/2}/a$.

In the deep *subsonic* case ($M \leq .7$, say) the operator (2.1) is uniformly elliptic, hence a usual multigrid V-cycle, employing red/black Gauss-Seidel relaxation at all levels, yields top-efficiency solvers. When M approaches 1, however, the operator becomes increasingly anisotropic, and classical multigrid algorithms severely degrade, due to the above-mentioned difficulty with characteristic components. (An exception is the case where the anisotropy directions are aligned with grid directions. For example, if $u^2 + v^2 \ll w^2$, full efficiency can still be obtained by employing z-plane block relaxation).

In the deep *supersonic* case (e.g., $M \geq 1.3$) the full potential operator is uniformly hyperbolic (with the stream direction serving as the time-like direction), and an efficient solver can be obtained using downstream relaxation, marching in the time-like direction. If the equations are of higher-order and/or not strictly upstream, a *predictor-corrector* marching can provide the same approximation order, hence fast convergence of smooth components; this has been shown by detailed experiments and mode analyses [78]. This procedure no longer works as M drops toward 1, since the Courant number associated with this time-like marching approaches infinity.

Thus, the most difficult situation for solving the full potential operator is the *near sonic* regime ($.7 \leq M \leq 1.3$, say), especially in the (usual) case of *non-alignment* (e.g., when the grid is Cartesian and no velocity component is consistently much larger than the others). No "classical" multigrid approach would attain good efficiency in this case. A new approach has recently been developed, based on a piecewise semi-coarsening and some rules for adding artificial dissipation at the coarser levels. To understand this, note first that in the general scheme for solving, e.g., the Euler equations, the solution of (2.1) is only a *relaxation* step, and it is enough to confine this step to one subdomain at a time (whose size, however, is not $O(h)$ but $O(1)$). Without loss of generality we can therefore limit the discussion to the case that throughout this subdomain the velocity is, e.g., *vertically-inclined* (i.e., $w^2 \geq .3(u^2 + v^2)$, say). In this case, the multigrid solver of (2.1) will use *horizontal semi-coarsening* (coarsening only in the x and y direction), possibly together with *vertical* line relaxation. (This z-line relaxation is actually not needed on the finest levels, but may be required after several levels of semi-coarsening.) With this semi coarsening, the inherent cross-characteristic numerical dissipation at the coarse level is *smaller* than at the fine one (opposite to their relation upon *full* coarsening); we can therefore stably add artificial dissipation terms at the coarse level so that its total cross-characteristic dissipation matches the local fine-level average.

The resulting algorithm can fully exploit massively parallel processing. It can be extended to other non-elliptic operators, including the convection operator. (The aforementioned approach for the convection operator, based on downstream relaxation, is not fully efficient on massively parallel machines.)

Extensive numerical tests have been performed with the linear full-potential equation: first in 2D, then in 3D, starting with constant-coefficients, then variable. In 2D we have also carried out comprehensive half-space FMG mode analyses (cf. [24, §7.5]), achieving full agreement with the numerical tests. The results reported in [41], [42], [77] and [78] show that at any Mach number the algorithm can always attains the "textbook" efficiency.

2.2.1 Comment on semi-coarsening schemes

Instead of the *piecewise semi-coarsening* described above, another possibility is to use just one *global* semi-coarsening, but of one of the following two types (preferably the second).

A. *Total semi-coarsening*. By this we mean (e.g., in 2D) that each coarser grid is formed by omitting every other line from the next finer grid (every other vertical line *as well as* every other horizontal line), but on the remaining lines (the coarse-grid lines) leave *all* the fine-grid points (not just the intersections of the coarse-grid lines).

B. *Variable-direction semi-coarsening*. Here the coarser grid for each level is a *subset* of the total-semi-coarsening grid for that level. Simply omit from the latter all unnecessary points in regions where semi-coarsening at only one particular direction is needed (as in various anisotropic and non-elliptic cases, like those discussed above).

2.3 A road map

A group at NASA/Langley has launched a multi-year program aimed at achieving "textbook" multigrid efficiency for flows at all Mach and Reynolds numbers, using the general approach described above [151], [152].

A road map for further development has been assembled in the form of a detailed table called "Barriers to Achieving Textbook Multigrid Efficiency in CFD". It lists every foreseen kind of computational difficulty for achieving that goal, together with the possible ways for resolving the difficulty, their current state of development, and references [36].

Included in the table are staggered and nonstaggered, conservative and non-conservative discretizations of viscous and inviscid, incompressible and compressible flows at various Mach numbers, as well as a simple (algebraic) turbulence model and comments on chemically reacting flows. The listing of associated computational barriers involves: non-alignment of streamlines or sonic characteristics with the grids; recirculating flows; stagnation points; discretization and relaxation on and near shocks and boundaries; far-field artificial boundary conditions; small-scale singularities (meaning important features, such as the complete airplane, which are not visible on some of the coarse grids); large grid aspect ratios; boundary layer resolution; and grid adaptation.

3 Time-Dependent Differential Problems

In the numerical solution of time-dependent problems, to allow large time steps and/or fully adaptable discretization (cf. Sec. 3.2), *implicit* time steps must be used, hence a system of equations must be solved at each time step. Multigrid solvers for such systems are usually similar to but *simpler* than their steady-state counterparts, because these systems are easier than the steady-state equations, in various ways: they have better ellipticity measures (due to the time term); they do not involve the difficulties associated with recirculation (in flow problems); and they each come with a good first approximation (from the previous time step). A simple "F cycle" at each time step (effectively an FAS-FMG algorithm for the solution *increment*, i.e., its departure from the previous-time solution) should solve the equations much below the *incremental* discretization errors (the errors added in the current time step). Hence, the errors *accumulated* over time due to the solver are generally much below the *accumulated* discretization errors [49].

It is generally true that fully efficient multigrid methods for the steady-state equations directly yield also at-least-as-efficient methods for time-accurate integrations, where the work per implicit time step is just comparable to the work of an *explicit* time step. Moreover, in various cases (e.g., parabolic equations with steady or smoothly-varying-in-time forcing terms), the work can be substantially *smaller* than that of an explicit time step. This is due to the smoothness of solution *increments* (solution changes from a previous time or solution departures from a simple convection). Such smoothness is typically established away from the immediate neighborhood of oscillatory initial or boundary conditions. It implies that the high-frequency part of the solution changes slowly. Hence the multigrid solver applied at each time step needs to actually visit the finest levels only once per many time steps, provided that the fine-to-coarse correction τ_h^{2h} is carried from each such visit to subsequent time steps [95], [100].

3.1 Parallel processing in space-time

A unique feature of multigrid solvers is the possibility to apply parallel processing across space *and time*, i.e., to process simultaneously earlier and later time steps, whereas single-level solvers must proceed sequentially in time. (This unique feature is discussed in [21, §3.10], [27, §11] and elaborated in [156].)

To achieve that, time is treated just as another space coordinate, and the whole problem is solved by an FMG (full multigrid) algorithm, starting with a coarse grid *in both space and time*, proceeding to finer levels, with one (sometimes two) cycles at each level. At fine levels, where most of the computational work is spent, all the processes (relaxation and inter-grid transfers) can employ many processors in parallel, each one working in its own space-time subdomain. This of course makes it possible to use *efficiently* (i.e., at a given arithmetic to communication ratio) a larger number of parallel processors than can be used when parallelization is done only across space (marching sequentially in time).

Depending on the number of processors, available storage, etc., the above algorithm will often be applied not to the entire time evolution, but to one (large) time interval at a time. (A more sophisticated multilevel time windowing is described in Sec. 4.2 below).

3.2 Grid adaptation in space-time

Simultaneous space-time multigridding also yields a very efficient way for general *space-time grid adaptation*, where both the spatial meshsize and the time step can be adapted — locally in both space and time. Just as in the case of pure spatial (i.e., steady-state) problems, the multigrid environment can provide convenient flexible structures, where discrete equations need be derived only for uniform grids (facilitating economic high-order discretizations and parallelizable and vectorizable processing), while only negligible geometric information need be stored. The multigrid algorithm also provides local refinement criteria and one-shot self-adaptive solvers; see Sec. 6.1 below.

4 Inverse Problems. Atmospheric Data Assimilation

A *direct* partial differential problem involves an interior differential equation and a set of initial/boundary conditions which stably determines a unique solution. An *inverse* problem is one in which the differential equation and/or the initial/boundary conditioned are not fully given and instead the results of a set of solution observations (measurements) are known. The latter may contain errors, and even without errors the problem is usually ill-posed: the known data may be approximated by widely different solutions.

In this chapter we use the problem of atmospheric data assimilation to illustrate the many ways in which multiscale computation can benefit the solution of inverse PDE problems. Other problems of this type are discussed in Secs. 5 and 16.2 below.

4.1 Background and objectives

A major difficulty in weather prediction is the need to assimilate into the solution of the atmospheric flow equations a continuously incoming stream of data from measurements carried out around the globe by a variety of devices, with highly varying accuracy, frequency, and resolution. Current assimilation methods require much more computer resources than the direct solution of the atmospheric equations. The reason is the full 4-D coupling: Any measurement, at any place and time, should in principle affect the solution at any other place and time, thus creating a dense $N_sN_t \times N_sN_t$ matrix of influence, where N_s is the huge number of gridpoints representing the 3-D atmosphere and N_t is the large number of time steps spanning the full period over which large-scale atmospheric patterns are correlated. As a result, not only are current assimilation methods very slow, but they are also based on highly questionable compromises, such as: ignoring the all-important spatially or temporally *remote* correlations of *large-scale* averages; limiting control to only the initial value of the flow at some arbitrarily chosen initial time, instead of controlling the numerical equations at all times; and assimilating only the data from one time interval at a time, without fully correlating with other intervals.

The objective is to develop multiscale methods that can avoid all these compromises, and can assimilate the data into the multigrid solver of the direct flow equations at small extra cost, i.e., using extra computer time smaller than that required by the direct solver by itself.

This is considered possible because: (1) Large scale averages can inexpensively be assimilated on the correspondingly coarse levels of the multigrid solver (coarse in both space and time). (2) Deviations from any large-scale average must be assimilated on some finer scale, but their correlation on that scale is local. (3) The measurements (with their representativeness errors) are generally less accurate and in most regions less resolved than the numerical flow itself, hence their assimilation should not be done at the finest numerical level. (The overall solver of an ill-posed inverse problem can sometimes cost even far less than the solver of a corresponding well-posed problem, because ill-defined high-frequencies need not be calculated: see Sec. 16.2.)

4.2 Multiscale 4D assimilation

Since the atmospheric data assimilation problem involves full 4D couplings, both forward and backward in time, it is proposed to use one full-multigrid (FMG) algorithm for the entire 4D problem (but possibly with the storage-saving windowing described below). This algorithm would be like a usual FMG solver for the *direct* 4D atmospheric equations, except that at each stage, on each level excluding the finest ones, the relaxation of the solution variable will be accompanied by relaxation of the *control variables* $\sigma(x)$ at that level (see the nature of $\sigma(x)$ in Sec. 4.3 below). Thus, in essence, large-scale averages of the solution will be assimilated on correspondingly coarse grids (coarse in both space and time).

The levels at which $\sigma(x)$ will be adjusted will depend on the local density of the measurements, their accuracy and their distance from regions where details of the solution are of interest.

Windowing. Should the 4D solution require too much storage, it is possible to reorganize it in multiscale windows, marching in time, without much loss of efficiency. That is, only a certain window (time slice) of the finest grid need be kept in memory at a time. Having relaxed over it, residuals are then transferred from this window to the coarser grids. On returning from the coarser grids more relaxation is made on the finest grid, now in a somewhat advanced window (shifted forward in time, but partly overlapping its predecessor) and so on. At the coarser levels, on increasingly coarser grids, increasingly wider (in real time, but poorer in gridpoints) windows are kept and advanced in a similar manner. The domain covered by each coarse-grid window always strictly contains all the finer ones. The coarsest windows extend very far in time, especially into the past; as far indeed as there exist data whose large-scale averages are still correlated to the solution at the time of the current finest window. At times where a coarse window exists while the next finer one has already been removed, the coarse-level equations can still retain the FAS-multigrid fine-to-coarse (τ) corrections (static or modified), thus still maintaining the fine-level accuracy of coarse-level features (cf. the "frozen τ" technique in [23, §15] and in [95]).

Some of the finest windows may be local not only in time but also in space, effecting *local refinements* at regions of greater human interest and/or regions requiring higher resolution for mathematical and physical reasons (sea straits, islands, mountains, etc.).

4.3 Multiple benefits of multiple multiscale techniques

Our studies have uncovered *many* different ways in which multiscale computational methods can contribute to data assimilation problems (and similarly to other inverse problems). The following is the full list — brought as an example of what a "*full multiscaling*" of a problem may involve.

1. *Implicit nonlinear time steps.* At the level of the underlying *direct* CFD equations, fast multigrid solvers make it possible to use implicit-time-step discretizations at full efficiency (see the general approach to time dependent problems in Sec. 3 above, and multigrid methods for shallow water and three-dimensional atmospheric models in [11], [10], [9], [164], [165], [117], [118] and [133]). This entails not only unconditional linear stability, but also avoidance of bad effects associated with *linearized* time steps (in which one would use fully implicit equations, but based on linearization around the previous-time-step solution) [9]. The unconditional stability is important for the multiscale data assimilation processes, enabling work on various temporal and spatial scales, unconstrained by various Courant numbers.

2. *Local refinements* are well known to be greatly facilitated by the multigrid algorithm, as also hinted in the algorithm description above. The multiscale environment simultaneously provides convenient flexible structures, refinement criteria and one-shot self-adaptive solvers; cf. Secs. 3.2 and 6.1.

3. *Space + time parallel processing*. Still at the level of the direct CFD equations (but similarly also at the level of the inverse (data assimilation) problem), multiscaling is a necessary vehicle to obtain parallel processing not only across space at each time step, but also across time (see Sec. 3.1 above).

4. *One-shot solution of inverse problems.* Normally, inverse problems are solved by a *sequence* of direct solutions (e.g., direct multigrid solutions), through which an iterative adjustment is made to the control parameters (the inverse-

problem unknowns). For example, in the *adjoint method* for atmospheric data assimilation, a direct solver of the flow equations (marching forward in time) is followed by an adjoint solution (backward in time) that gauges the first derivatives of the data-fitness functional with respect to the initial values (the flow variables at the initial time). These derivatives then drive some adjustments of the initial values, from which another direct flow solution is next calculated, and so on. *Many* iterations are needed for this process to converge. In multigrid solvers, by contrast, one can integrate the adjustment of the inverse parameters into the appropriate stages of only *one* direct-problem solver (see Sec. 4.2 above and Secs. 5, 9.1 and 16.2 below. This general approach has been described in [23, §13] and [24, §13], with more details in [27, §8.2] and full development in [150], [2], [3]).

5. *One-shot continuation.* The assimilation problem is highly nonlinear, hence a good starting guess for the solution is important. A general way to obtain such an initial guess is by continuation (embedding), in which the problem is embedded in a sequence of problems, each requiring another application of the solver (using the previous-problem solution as the initial guess). In multigrid solvers, however, the continuation can often be integrated into just one FMG solver (see [23, S 8.3.2] or in [24]). For example, at the coarser stages of the FMG algorithm more artificial viscosity (and/or more regularization, and/or a smaller coefficient of D_t in the continuity equation) can be used, then gradually be taken out as the algorithm proceeds to finer levels. This makes the solution much easier in the first stages, from which it is then continuously dragged into the desired neighborhood. Such FMG continuation devices are often natural. For example, larger artificial viscosity would quite naturally be introduced on coarse grids, even without aiming at continuation. A natural continuation is also supplied by the inverse covariance matrix S (see below), which would be smaller on coarser FMG levels due to larger discretization-error estimates.

6. *Full flow control.* In most data assimilation approaches (such as the adjoint method described above), the control parameters (the parameters that can be changed to obtain fitness of solution to observations) are only the initial values of the solution. This makes it impossible to benefit from the details (the oscillating components) of the observations at time far removed from the initial time, because those details at those times are ill-determined by the initial values. Instead of controlling just initial values, one should really control the entire numerical solution. Namely, the control parameters $\sigma(x)$ is a vector-valued grid function that at each point x gives the deviations in satisfying the set of flow equations. The objective function (the error functional that should be minimized) has the general form

$$E = \boldsymbol{\sigma}^T S \boldsymbol{\sigma} + d^T W d, \tag{4.1}$$

where $\boldsymbol{\sigma} = \sigma(x)$ is the vector of all control parameters, $d = \big(d(y)\big)$ is the vector of deviations of the solution u from the observation u^0 (i.e., $d(y) = (P^0 u)(y) - u^0(y)$, where P^0 is a projection from the solution space (x) to the observation space (y)), and S and W are (positive-definite) weight matrices. In a crude approximation, one can take these matrices to be diagonal, where the diagonal inverse $S(x,x)^{-1}$ is (a very rough estimate of) the expected square error in the equation at x, which is the sum of the local discretization error (conveniently estimated by the "τ correction" of the FAS multigrid solver; see [24, §8.4]) and the local modeling errors (errors in the physical assumptions embodied in the equations). The diagonal inverse $W(y,y)^{-1}$ is (a very rough estimate of) the expected square error in the *measurement* $u^0(y)$, including in particular the "representativeness error" (accidental deviation at the point of measurement from the relevant local average). More precisely, S and W should be corresponding *general* (not necessarily diagonal) inverse covariance matrices (in which case the discussion at Item 8 below is relevant).

A detailed Fourier analysis by Rima Gandlin, comparing full-flow control with initial-value control in a model case of 1D + time wave equations, has demonstrated the great advantage of the former [90].

So extensive control parameters can only be handled by a multiscale treatment. Moreover, using the methods described above the solution is expected not to be expensive, especially since the control parameters $\sigma(x)$ need not be controlled at the finest computational levels; on such levels $\sigma(x)$ can simply be interpolated from the coarser levels and kept unchanged during the relaxation (cf. Item 9 below).

7. *Unlimited correlation range.* In conventional assimilation methods, each control value interacts with a limited range of measurements: measurements at a restricted (e.g., 6 hours) time interval and sometimes only at confined distances. However, it is clear that large-scale averages of the dynamic variables interact at much larger ranges. Multiscale data assimilation makes it possible to correlate solution and measurements at any desired distance in space and time, since correlations at increasingly larger distances are calculated on increasingly coarser grids.

8. *Efficient representation of direct and inverse covariance.* There are a number of ways to derive or estimate covariance matrices and various simplification assumptions are made. However, the real covariance matrices (especially the model error covariance) are actually dense (not sparse), and thus involve huge ($N_s^2 N_t^2$, in principle) amount of information. Even when the matrix is sparse, its inverse, used in (4.1), is certainly dense. The only efficient way of *representing*, let alone computing, such huge dense matrices and their inverses is a multiscale representation, based on their asymptotic smoothness. This would be similar to the methods described in Secs. 10 and 14.3 below and in [66] for calculating integral transforms, many-body interactions, solutions to integro-differential equations, and Kalmen filtering, all involving $n \times n$ dense matrices whose complexity (the amount of computer operations required to perform a multiplication by either the matrix *or its inverse*) is reduced to $O(n)$ by multiscale techniques.

To achieve such a low complexity it is of course necessary to assume the covariance matrices to be *reasonably smooth.* Namely, if the errors at two points, x and y, remote from each other, are correlated at all, their correlation is assumed to vary like $g_1(x)g_2(y)G(x,y)$, where $G(x,y)$ is *asymptotically smooth* (meaning that up to a certain order, p-order derivatives of $G(x, y)$ are not larger than $O(|x - y|^{-p+q})$, q being a fixed small integer). Such assumptions seem very reasonable in practice, and are certainly more accurate than neglecting distant error correlation altogether. They can also be weakened in various ways and still benefit from multiscale processing. (For example, it may be enough to assume at each point x smoothness for variations in only some directions, although the complexity may then rise to $O(n \log n)$. The processing in such cases would be akin to those in [43] and [55].)

9. *Improved regularization.* First, the multiscale solver described above is likely to require much less regularization than conventional solvers since the main ill-posedness in the problem is the long term and long range influence of fine-scale oscillations, while the multiscale large-scale interactions are mediated by coarse grids, omitting these oscillations. Secondly, attractive regularization devices are offered by the multiscale processing. For example, statistical theories of the atmospheric equations yield the relative expected energy at different scales. In a multiscale processing this can be used to properly penalize any excessive local energy at every scale, yielding an excellent regularization scheme (which could not even be *formulated* in uniscale processing). Generally, the multiscale data assimilation need not be governed by one all-embracing objective function, but can employ a collection of different directives at different scales. (Cf. Item 12 below).

10. *Fast assimilation of new data.* Normally, new observation data keep arriving and need to be assimilated into an already partly existing approximate solution; i.e., the new data should usually both modify the previous solution and extend it into a new time interval. The multiscale solver is particularly suitable for the task: The new data normally does not affect the *h–f details* of the solution in much older times; also, these details are normally no longer of interest. Hence, increasingly older times can participate in the new processing on increasingly coarser levels (still maintaining the fine-to-coarse τ corrections previously computed for them). This exactly fits into the windowing algorithm above (Sec. 4.2). The resulting ease of assimilating new pieces of data may well facilitate a *continuous assimilation policy*, with new data being assimilated much more often than today.

11. *Multiscale organization of observation data.* Either for the purposes of the multiscale assimilation procedure, or for a variety of other procedures, it is very useful to organize the observation data in a multiscale structure. This may simply mean pointers from a multiscale hierarchy of uniform grids into the set of data, with finer uniform levels introduced only where there are still more than a couple of observations per grid cell. Such data structures are commonly used to facilitate regional computations of all kinds. Beyond this, it is possible to replace many observations by their average at some larger scale, serving as a kind of *macro-observation*, its associated error estimate being of course reduced by standard rules of statistics. This can be repeated, to obtain still-larger-scale representations. Such structures may save much storage, and provide directly the needs of the multiscale assimilation algorithms.

12. *Scale-dependent data types.* Instead of simple averaging, the macro-

observations just mentioned can be formed from the fine-scale data in a variety of other, often more meaningful, ways. In particular, fine-scale waves should be represented on coarse scales by their slowly-varying *amplitude*. Indeed, at large distances the wave phase is ill-posed, while its amplitude is still meaningful. (Cf. the techniques in Sec. 7 below).

5 Feedback Optimal Control

We consider a dynamical system that involves a vector x of *state functions* and a vector u of *control functions*, both being functions of time (and possibly also of space), governed by the initial-value ODE (or PDE)

$$\frac{dx}{dt} = F(x,u) \ , \qquad B_0 x(t=0) = b_0 \ , \tag{5.1}$$

where F and B_0 are vectors of known functions (or spatial operators). The *optimal control* problem is to find the control u for which this dynamical system minimizes a given *objective functional* $J(x,u)$ under various constraints, such as target-time (T) conditions of the type $B_1 x(t=T) = b_1$. In the *feedback* optimal control problem it is assumed that new initial conditions $B_o x(t)$ are continuously fed from the controlled device at all times $0 \le t < T$, requiring continuous updating of the control u. Fast numerical updates are required for *real-time* control.

In a usual approach to the feedback problem, the dynamical system is approximated by a linear-quadratic regulator (LQR), in which $F(x,u) = \mathcal{A}x + u$ and $J(x,u) = \int |Cx(t)|^2(|\mathcal{C}x(t)|^2 + |u(t)|^2)dt$, where $\mathcal{A}$ and $\mathcal{C}$ are linear operators on a suitable Hilbert space $\mathcal{H}$, $x \in \mathcal{H}$ and $u \in \mathcal{U} \subset \mathcal{H}$. Provided the system is stabilizable and $\mathcal{C}$ renders it detectable, there exists a unique nonnegative solution $\mathcal{K}$ to the *Ricatti equation*

$$(\mathcal{A}^*\mathcal{K} + \mathcal{K}\mathcal{A} - \mathcal{K}^2 + \mathcal{C}^*\mathcal{C})x = 0 \ , \qquad \forall x \in \mathcal{H} \ , \tag{5.2}$$

and $u(t) = -\mathcal{K}x(t)$ yields the desired feedback. In most cases this approach is very inefficient, either because the LQR approximation should be iterated many times, and/or because of the non-sparseness of the (discretized) operator $\mathcal{K}$, and the resulting high dimensionality of the Ricatti equation.

Our approach, by contrast, is based on a fast multigrid solver for the *open-loop* (i.e., not feedback) optimal control problem, installed in a multiscale way that allows super-fast (essentially local) updates upon feedbacks.

The multigrid open-loop solver is very efficient by itself, a one-shot solver for the nonlinear (non-LQR) problem. In fact, for various actual problems, it can share many of the potential benefits listed in Sec. 4.3 above. The super-fast *updates* are based on the observation that, upon changing the initial conditions, the change in the solution is increasingly smoother at times increasingly far from the initial. (In various actual problems, the sense of this smoothness has to be carefully understood.) This makes it possible for the multigrid *re*-solving algorithm to re-process its *fine* grids only at the very early times, while at later times only coarse levels are re-processed, with FAS fine-to-coarse defect corrections being frozen there (cf. [23, §15].) More precisely, at increasingly later times, any given multigrid level (a given timestep and, when relevant, spatial meshsize) need be re-processed increasingly more rarely. As a result, the computational cost of re-resolving is equivalent to only *local* re-processing (essentially just few steps near the initial time) of the full solver. This will usually be far less expensive than applying $\mathcal{K}$ (even just once, and even assuming the Riccati equations has already been solved).

We have tested this approach on several toy F-16 maneuvering problems (given to us by Dr. Meir Pachter of the Air Force Institute of Technology at Wright Patterson Air Force Base.) The linear dynamics includes three state and one control functions. Both quadratic and non-quadratic objectives were tested, including the L_∞ norm (the maximum absolute value) of one of the state functions.

The open-loop optimal control problem, which in this case is a two-point boundary value ODE system with 7 unknown functions, has been discretized by second-order finite differences on a staggered grid and solved by an FMG algorithm. Just two $V(1,1)$ multigrid cycles per grid proved enough to produce a solution with algebraic errors much smaller than (only few percent of) the discretization errors. In the case of the L_∞ objective, a continuation process has been integrated into this FMG solver, approaching L_∞ by L_p, with $p = 2^m$ at the m-th FMG level. Each of the relaxation sweeps included one red/black pass for each of the 7 ODEs, some of the passes being of the Gauss-Seidel type, the others — Kacmarz type. For some of the toy problems the principal part of the ODE system was scale dependent, producing boundary layers and requiring two different discretization schemes, one at fine levels the other at coarse levels, each with its own corresponding relaxation scheme.

The feasibility of the super-fast solution updates upon feedbacks has been established in our tests by monitoring the fine-to-coarse (τ) corrections. When τ is appropriately scaled (divided by proper solution values available to the coarse-level re-solver) its values (excluding a couple of them near the initial time) turn out to change very little upon changing the initial conditions. This shows that τ can be frozen, so that re-solving can be restricted to coarse levels, as expected.

6 Adaptable Grids and PDE Solvers on Unbounded Domains

6.1 The multigrid solver + adaptor

A very substantial saving in the number of degrees of freedom needed by a discretization of a PDE to attain a given accuracy can be obtained by employing various forms of local grid adaptation. The multigrid solver, in its nonlinear FAS form, yields a particularly flexible and efficient framework for that purpose, with some additional benefits. The general description of this framework has been given elsewhere [19, §§7–9], [23, §9], [24, §9] or [6], so we only summarize the main points.

Local refinements are created by adding local *patches* of finer *uniform* grids over desired parts of a domain covered by a "parent" coarse grid. (The system is recursive: each of the "child" patches may itself contain smaller subdomains over which "grandchildren" patches of a further refinement are set). Each fine-patch solution supplies a certain *defect correction* to the equations of its parent (coarse) grid, thereby enforcing there the fine-grid accuracy. This is a natural part of the FAS multigrid solver (which introduces anyway coarser grids over finer ones to accelerate convergence, using in the process the same fine-to-coarse defect corrections). Hence, the multigrid solver solves the resulting composite discretization with the same efficiency (per degree of freedom) as solving uniform-grid equations. This composite structure is very flexible and can be highly non-uniform, while all its discrete equations are still written in terms of *uniform* grids. This makes it simple and inexpensive to use high-order approximations, while storing only a negligible amount of geometrical information. Also in this way no *unintended grid anisotropy* is introduced (in contrast to other grid generation or grid transformation methods, in which such anisotropies do enter, causing considerable complications for the multigrid solver).

The fine-to-coarse defect correction also yields, as a byproduct, precise *adaptation criteria*: a defect correction larger than a natural threshold indicates that a further local refinement is needed. Moreover, an automatic self-adaptation process can be integrated into the so-called "full multigrid" (FMG) solver: as the latter proceeds to increasingly finer levels, it can also decides (using these adaptation criteria) *where* those finer levels should be, thereby yielding a *one-shot solver/adaptor*.

It is also possible for each of the local grid patches to have its own *local coordinate system*. For example, in flow problems, the coordinate system may fit wall boundaries (or more generally: stream lines), facilitating the introduction of highly anisotropic grids in boundary layers (or particularly fine cross-stream meshsizes). In electronic structure calculations (cf. Sec. 9), a local patch at the atomic core can use spherical symmetry, as appropriate for core electrons.

An important feature of this adaptation is that often the calculation within the local-refinement patch can be done *once for all*: Although the solution in the patch changes when the parent-grid solution changes, the fine-to-coarse defect corrections usually change very little. At most one more short "visit" to the patch (e.g., one more relaxation sweep at the finer level) toward the end of the calculation will normally be needed to update the defect corrections. Alternatively, one can calculate apriori the approximately linear dependence of the defect corrections on the local parent-grid values.

6.2 PDE solvers on unbounded domains

As pointed out in [19, §7.1], problems in unbounded domains can be solved by a multigrid structure employing increasingly coarser grids on increasingly larger domains, using an FAS multigrid solver. The structure is essentially the same as described above (Sec. 6.1). We have embarked on a detailed study of how this should be done: At what rate should the domains increase with increased meshsize? What is the largest needed domain? What interpolation is needed at interior boundaries (boundaries of a grid h embedded in a larger domain covered by grid $2h$)? What multigrid algorithm should be applied?

For the Poisson equation $\Delta u = F$ we have developed, in collaboration with Jeffrey S. Danowitz, theoretical answers to these questions, then tested them numerically. Using general grid optimization equations (see [19, §8.1] or [24, §9.5] or [24, §9.3]) and the known smoothness properties of the solution, one can calculate how far out one must use a certain meshsize to maintain a certain accuracy. For example, one can find that if the domain of interest (outside which $F = 0$) has diameter d_0 and if the desired accuracy inside that domain would be obtained (had its boundary values been given) by a second-order discretization and a grid with meshsize h_0, then the diameter of each coarser grid h ($h = 2h_0, 4h_0, \ldots$) should only satisfy $d(h) \geq d_0(h/h_0)^{2/3}$ and $d(h) \geq d(h/2) + Ch \log h_0$. Without significantly departing from the desired accuracy one can cover a domain (the coarsest-grid domain) with diameter R, spending only $O(\log R)$ gridpoints, so R can easily be taken so large as to admit small enough boundary-condition errors. Employing a suitable version of the λ-FMG algorithm [24, §9.6], it has been shown that the accuracy-to-work relation typical to multigrid solvers of the *bounded*-domain problem can in this way be obtained for the *unbounded* domain, where accuracy is in terms of approaching the *differential* solution. The same can be obtained for higher-order discretizations (with another exponent in the first $d(h)$ inequality).

7 Wave/Ray Multigrid Methods

The aim is to develop advanced and general numerical tools for computing wave propagation on scales much larger than the wavelength, when there may also exist interactions with special smaller-scale inhomogeneities where ray representations (geometrical optics) would break down. Such tools can revolutionize important computations, such as: radar cross sections; wave propagation through dispersive media; seismic wave characteristics resulting from various types of explosion zones; generation and control of acoustic noise; electronic waves in condensed matter; etc.

We have developed two basic approaches relevant to the problem. One is a general multiscale solver for *integral* equations with oscillatory kernels [29], which is a very efficient way to solve wave propagation in *homogeneous* (and some *piecewise* homogeneous) media by replacing the differential equations with *boundary* integral equations. Multiscale ray representations first appeared in this work.

The other approach is a fast multigrid solver for the highly indefinite *differential* equations of stationary waves in a domain containing many wavelengths, with radiation boundary conditions. The basic idea of this work had been stated long ago (see, e.g., [20, §3.2], and more details in [30]), but important algorithmic aspects had still to be worked out.

The model equation we use is the Helmholtz equation

$$\Delta u(x) + k^2 u(x) = f(x) \ . \tag{7.1}$$

Traditional multigrid solvers are not effective for this problem, because some "characteristic" components (i.e., those with wavelength close to $2\pi/k$) are non-local (their size is determined by conditions many meshsizes away) exactly on all those grids which are fine enough to approximate such components.

On each of its levels, the new solver represents the solution as

$$u(x) = \sum_j A_j(x) \exp\left(i\varphi_j(x)\right) \ . \tag{7.2}$$

At the highest (finest) level this sum includes just one term and $\varphi_j(x) \equiv 0$, so the representation includes just one function — the desired solution — and the equation for it is the usual five-point finite-difference discretization of (7.1). Increasingly lower levels of the solver employ on the one hand increasingly *coarser* grids of x to discretize each amplitude $A_j(x)$ and each eikonal $\varphi_j(x)$, and, on the other hand, correspondingly *finer* sets of "momenta" (i.e., more terms j in the above sum). The interaction between these levels has been shown to yield a solver (for the discrete equations given at the highest level) which is as efficient as the best traditional multigrid solvers for definite elliptic systems. The radiation boundary conditions are naturally enforced at the lowest level, where the representation essentially coincides with geometrical optics (ray representation, appropriate for scales much larger than the wavelength).

Details of the one-dimensional solver and a preliminary version of the two-dimensional solver were given in [122]. The current version of the two-dimensional solver, together with numerical results, is described in detail in [51]; its accuracy is analyzed in [52].

An important feature of the solver is the alignment of the grid on which $A_j(x)$ is discretized with the propagation direction of the corresponding eikonal (the direction of $\nabla\varphi_j(x)$), its meshsize growing (upon coarsening) faster in that direction than in the perpendicular directions. Specifically, if J is the number of terms in the summation (7.2) at a given multigrid level, then the propagation-direction meshsize for that level is $O(J^2k^{-1})$, while the perpendicular-direction one is $O(Jk^{-1})$. Incidentally, such oriented grids should have also been employed in [25], reducing the order of complexity stated there to the same one as in the non-oscillatory case (with an additional $O(\log n)$ factor in the case of integral transforms or integral equations defined on a curved manifold of codimension 1, e.g., a boundary).

A finite-element representation akin to (7.2) appears in [5] and [124], but only on one level, and without the above-mentioned grid alignment. Unlike that representation, the present one can be used to bridge the entire gap between the wave discretization needed at small subdomains and the ray discretization needed at the large outer regions, thus producing fully efficient fast solver, as well as the basis for the development described next.

7.1 Future plans: variable coefficients, local refinements and diffraction

The plan is to develop the solver for the variable-coefficient case $k = k(x)$, and to advance a new setting where only geometrical optics is used in most of the domain, while the wave equations, as well as intermediate levels with representations of the type (7.2), are just introduced at special restricted subdomains where geometrical optics breaks down.

Geometrical optics can certainly be used throughout large regions where $k(x)$ is either a constant or has a small relative change per wavelength. Although in the latter case the rays are curved, they can still be followed by Snell's law, or more generally by marching solutions of the eikonal equation (see, e.g., [160]). Discontinuities in $k(x)$ can also be accommodated by geometrical optics, employing the usual rules of reflection and refraction, as long as the surfaces of discontinuity have curvature radii large compared with the wavelength (assuming the number of repeated reflections is not too large).

The pure geometrical optics approach will typically break down in smaller regions (e.g., neighborhood of fast changes in $k(x)$ or large-curvature surfaces of discontinuity). It is proposed to introduce, in such regions, nested local refinements structured in the usual FAS-multigrid manner (see Sec. 6.1). The finer levels will generally use representations of the type (7.2), the finer the level the smaller the number of terms in the summation, eventually yielding a direct discretization of (7.1) on sufficiently fine grids in small subdomains; see more details in [51, §10].

Effectively this will produce ray dynamics in the large, with relations between rays modified by the finer grids in the small special regions (around an aperture, corners, edges, a radar target, etc.), yielding a general numerical tool for computing diffraction (the rays produced by small-scale disturbances; cf. [111]).

8 Rigorous Quantitative Analysis of Multigrid

Since the early days of multigrid development, the "local mode analysis" (LMA), based on heuristic local Fourier decomposition of the error function, has been the chief tool for the practical design, precise quantitative understanding, and even debugging, of the various multigrid processes. Although rigorously justified in very special cases only, the easily computable predictions of that analysis have turned out to be precise for quite general PDE boundary value problems discretized on uniform grids with quite general domains and boundary conditions. In several important cases, however, the predicted LMA convergence factors were not obtained, presumably due to the influence of boundaries, which are usually not accounted for by the local analysis; domains with reentrant corners are a notorious example.

A general rigorous framework for the local mode analysis on one hand, and for the treatment of boundaries on the other hand, has appeared [28], [32].

For general linear elliptic PDE systems with piecewise smooth coefficients in general domains discretized by uniform grids, it has been proved that, in the limit of small meshsizes, the convergence factors predicted by LMA are indeed obtained (provided the multigrid cycle is supplemented with a proper processing at and near the boundaries). That processing, it is proved, costs negligible extra computer work. Apart from mode analysis, a Coarse Grid Approximation (CGA) condition has been introduced which is both necessary and sufficient for the multigrid algorithm to work properly.

Unlike most other multigrid theories, convergence factors thus proven are not just qualitative; they are *quantitatively sharp*: they are exactly obtained (or arbitrarily closely approached) by the worst local mode.

The assumptions made by the theory are shown to be, in a sense, the weakest possible. Except for its stability, no assumption is made about the relaxation scheme: whether it is good or bad is not *assumed*, but *calculated*.

The first part of this work [32] studies the L_2 convergence in one cycle, for equations with constant coefficients. In the second part, extensions are discussed (in the form of comments) to variable coefficients, to block relaxation schemes, to many cycles (asymptotic convergence), to more levels with arbitrary cycle types (V, W, etc.), and to FMG algorithms. Various error norms and their relations to the orders of the inter-grid transfer operators are analyzed. Global mode analysis, required to supplement the local analysis in various border cases, is developed, and practical implications of the analysis, including practical ways for constructing and debugging multigrid solvers, are generally reviewed. A major emphasis is on the importance and practicality of adding partial (local) relaxation passes to the multigrid algorithm (cf. [19, App. A.9]): Theory and practice show that multigrid efficiency is greatly enhanced by adding special relaxation steps at any local neighborhood exhibiting unusually large residuals.

9 Many Eigenfunction Problems: Electronic Structures

Some important scientific problems involve the computation of a large number of eigenfunctions of a partial differential operator. As an important example of such problems we consider here the *Kohn-Sham* equation, of the Density Functional Theory (DFT) [116], central to ab-initio condensed-matter and quantum-chemistry calculations of electronic structures. (For surveys of the DFT equations and their current multigrid solvers see [13], [17].) The Kohn-Sham N-eigenfunction equation is:

$$-\Delta\psi_n(r) + V(r)\psi_n(r) = \lambda_n\psi_n(r), \qquad r \in \mathbb{R}^3, (n = 1, \ldots, N), \tag{9.1}$$

where $2N$ is the number of electrons in the system (or their number per period, in the case that V is a periodic function) which can be very large. Actually, the "effective potential" V depends on the eigenfunctions ψ_n and the nuclear positions through the relation

$$V(r) = V_{nc} + V_{el} + V_{xc}, \quad V_{nc} = -\sum_{i=1}^{N_a} \frac{2Z_i}{|r - R_i|}, \quad V_{el} = \int \frac{2\rho(r')dr'}{|r - r'|}. \tag{9.2}$$

Here N_a is the number of atoms in the system, R_i and Z_i are the position and charge of the i-th nucleon, $\rho(r)$ is the electronic density defined by

$$\rho(r) = \sum_{n=1}^{N} |\psi_n(r)|^2, \tag{9.3}$$

and $V_{xc}(r)$ is the so-called exchange and correlation potential, describing the average effects of exchange and correlation derived from the theory of the electron gas. In the *local* density approximation, $V_{xc}(r)$ depends only on $\rho(r)$ and possibly on the gradient $(\nabla\rho)(r)$.

Fast multigrid eigenproblem solvers have been developed before [57], but the ab-initio problem includes new traits and difficulties that call for new multiscale techniques, such as in the following list:

(1) *Singularities.* The nuclear potential energy harbors a singularity at each atomic nucleus (if pseudo-potential is not used). The multigrid solver (unlike Fourier methods) allows local refinements that would remove the global inaccuracies associated with such singularities: see Sec. 6.1 above. Note that the local patches of finer grid levels can supply once-for-all (or seldom-updated) "defect corrections" that can act like, and indeed replace the use of, pseudopotentials.

Even with pseudopotentials, local refinements around nuclei, as well as *high-order discretization* everywhere, are necessary for efficiency, since high accuracies are required for predicting the energy *differences* that govern chemical processes. Because of the neighborhood of the singularity, *conservative discretization* is needed [6], which is especially tricky for high-order discretization at grid *interfaces* (the boundaries of any level of local refinement); see [12], where the FAS conservative discretization of [6] is extended to high-order schemes in three dimensions, and applications to quantum chemistry are discussed.

(2) *Unbounded or very-large-scale domains* can efficiently be treated by multigrid solvers which employ increasingly coarser grids at increasingly larger distances from the region(s) of interest (cf. Sec. 6.2 above). In particular, for electronic problems involving isolated molecules, the meshsize in the vacuum away from the molecules can grow; it can grow so fast that the total computational work is dominated by the work near (up to several inter-atomic distances away) from the nuclei.

Similarly, in problems with extended, essentially periodic structures, deviations from the periodic-structure solution due to defects or surfaces can be described on increasingly coarser grids at increasing distances from the defect or surface, while for extended *amorphous* material, increasingly coarser grids can be used at increasingly greater distances from the subdomain of interest. (This *cannot* be done without the new multiscale structure proposed in Sec. 9.2 below).

(3) *Self-consistency*. The dependence of the potential function V on the total electronic charge distribution ρ introduces a nonlinearity into the problems, which usually requires many iterative applications of a linear solver. FAS multigrid procedures can directly solve nonlinear problems, as efficiently as solving their linear counterparts [19], [23], [24]. The development of such one-shot solvers for the Schrödinger operator depends on the ability to update the self-consistent potential as the solution changes on the coarse grids. This is also related to the following issue.

(4) *Multi-integrations* are required in calculating the potential (e.g., the Hartree potential). This can be performed fast by solving auxiliary Poisson equations. Solving them by multigrid would facilitate the needed interaction between the coarse-level moves of this Poisson solver and the coarse-grid updates to the self-consistent potential in the eigenproblem solver (see #3 above).

(5) *External optimization*. In solving the electronic problem the nuclei are assumed fixed (the Born-Oppenheimer approximation), but one actually needs to find the nuclei positions for which the electronic-solution energy together with the inter-nucleus potential yield the minimal total energy. This external optimization would normally be done iteratively, requiring solving the electronic eigenproblem many times. Again, a one-shot multigrid solver + optimizer can and should be developed, incorporating suitable nucleus moves into each of the levels of the multigrid electronic solver. A model study reported below (Sec. 9.1) has shown the feasibility of this approach and the exact multigrid techniques required for its full efficiency.

(6) *Multitude of eigenfunctions*. Even with a multigrid solver, the cost of calculating a large number N of eigenfunctions (N being the number of electrons in the system) may grow proportionally to N^3 (employing discretizations with $O(N)$ degrees of freedom), since each eigenfunction is represented separately and may need to be orthogonalized with respect to all others to ensure their distinction. A theoretical study and tests with 1D model problems, reported in Sec. 9.2 below, indicate that for periodic problems (the usual type in condensed matter calculations), it may be possible to reduce the complexity to $O(N_g \log N \log \frac{1}{\epsilon})$, by employing a multiscale collective representation of the eigenmodes. Here ϵ is the desired accuracy and N_g is just the number of grid points per periodicity cell required for adequately resolving the various features of the potential function $V(x)$.

(7) *Highly oscillatory functions*, such as the orbitals of high-energy electrons, would normally require dense grids, and would not allow effective multigrid solvers, because of the corresponding high indefiniteness of their equation. The multiscale approach described in Sec. 9.2 below effectively deals also with this difficulty.

(8) Multiscale structures may offer improved non-local representations for the *exchange correlation* potential. This aspect is largely unexplored. Certain non-local exchange-correlation operators can be translated into a system of Poisson equations, which (like the Hartree potential in #4 above) augment the Kohn-Sham system, yielding a larger PDE system that is simultaneously solvable by a multigrid algorithm.

Of all the scaling difficulties listed above, several (those numbered 1, 2, 3, 4, and partly also #5) have been dealt with in other contexts (similar difficulties in other fields). So, once multigrid solvers are introduced, the technique for treating these difficulties will already be at hand. We have therefore focused our attention mainly on #5 (see Sec. 9.1) and #6 and #7 (see Sec. 9.2).

9.1 Model problem for external optimization

A simplified model problem for the external optimization is the minimization of the two-dimensional two-atom total energy

$$\min_{z=(z_1,z_2)\in D} \big[E(z)+\lambda(z)\big] \ , \tag{9.4}$$

where $E(z)$ models the ("external") repulsive energy between ions located at $(0,0)$ and at (z_1,z_2), and $\lambda(z)$ is the corresponding electronic energy, modeled by the eigenvalue of the equation

$$\big(-\Delta+V(x,z)\big)\psi(x)=\lambda\psi(x) \ , \qquad x=(x_1,x_2)\in D \ . \tag{9.5}$$

We chose $V(x,z)$ that models the Coloumbic potential at x of the two-ion system, $D=[0,1]\times[0,1]$, and ψ was required to satisfy periodic boundary conditions on D (having chosen V and E also with this periodicity).

The Euler equations for minimizing (9.4) under the constraint (9.5) can be simplified (since the Lagrange multiplier coincides with ψ) to the system of equations (9.5)–(9.7), where

$$\langle\psi,\psi\rangle=1 \ , \tag{9.6}$$

$$\frac{\partial E}{\partial z_i}+\left\langle\psi,\frac{\partial V}{\partial z_i}\psi\right\rangle=0 \ , \qquad (i=1,2) \ . \tag{9.7}$$

The eigenproblem (9.5)–(9.7) was solved by a classical FAS multigrid eigensolver [57]. The main point of the research was to find out how to include Eq. (9.7) and where to adjust z in the course of this solver. Since (9.7) is a global equation and z is a "global" unknown (unlike $\psi(x)$ it cannot be *smoothed*), it is enough to treat both of them at the coarsest level, where all the discrete equations can simply be solved simultaneously for all the unknowns, since their number is small. This would be fully efficient, provided a suitable "fine-to-coarse correction" for Eq. (9.7) is recursively calculated at each coarsening step, see [23, §5.6], except that in the FAS scheme the residual transfer is replaced by the τ_h^{2h} fine-to-coarse correction; see [23, §8.2].

The main finding of this research, done in collaboration with Ron Kaminsky, was that in the above situation (and for similarly "localized" global unknowns, whose movements may not be resolved on some of the coarse grids), a linear dependence on the global unknowns should be introduced into τ_h^{2h}; see details in [35, §6.1]. The linear terms are important in the cases where the functions $\partial V/\partial z_i$ are not resolved well enough on the coarse level to yield there the correct dependence of $\langle\psi,(\partial V/\partial z_i)\psi\rangle$ on variations in z. This generally happens when

V has a singularity (or a particularly large local variation on the scale of the grid h) which moves with z. Fortunately, exactly in such cases, it is enough to calculate τ_h^{2h} in a small neighborhood of the singularity.

With this simple change, the one-shot solver for the external optimization problem (9.2)–(9.4) has attained essentially the same convergence factors as in solving Poisson equation, costing only a fraction more.

This can straightforwardly be generalized to any number of unknown point locations (e.g., nuclei) in terms of which a PDE is formulated along with a corresponding number of additional "global" conditions, such as (9.7). At coarse levels where the separation between two such points becomes smaller than the meshsize, new equations can be formulated for their *collective* motion (keeping constant the position of one relative to the other).

9.2 $O(N \log N)$ calculation of N eigenfunctions

What is the amount of calculation needed to calculate the N lowest eigenfunctions of a differential operator discretized on N_g gridpoints? A usual multigrid eigensolver [57] would need $O(N^2 N_g)$ operations, since each eigenfunction needs to be orthogonalized with respect to each other. Under favorable conditions, these orthogonalizations can be performed on the coarse grids [73], [74], possibly reducing the cost to $O(N N_g)$. However, all of these methods are adequate only for N sufficiently small, $N \ll N_g$. For large N, with eigenfunctions featuring variations on the scale of the grid, coarser grids cannot be used in such "naive" ways. For large eigenvalues the eigenproblem is highly indefinite, hence methods akin to those in Sec. 7 above should be used for coarsening.

A new approach (pointed out in [30]) is being developed for calculating N eigenfunctions of a differential operator discretized on N_g gridpoints in $O(N_g \log N \log \frac{1}{\varepsilon})$ computer operations, ε being the accuracy. This approach is based on the observation that "neighboring" eigenfunctions are distinguishable from each other only at large enough scales, and hence, in suitable representations, one can use a common description of their details at finer scales, and progressively separate them out only on increasingly coarser grids. The core procedure is indeed similar in structure to multigrid algorithms developed for *wave* equations (see Sec. 7 above). (Recent "linear scaling" methods in electronic structure calculation, reviewed in [85] and [91], are based on a localization assumption, typically solving a localized problem in $O(N N_\ell^2)$ operations, where N_ℓ is the number of atoms in the localization radius. The new approach would solve such a problem in $O(N \log N_\ell)$ operations.)

The feasibility of obtaining the $O(N_g \log N)$ efficiency has first been demonstrated by Oren Livne for *one-dimensional* problems [119], [120]. Moreover, that work has also shown that the developed *multiscale eigenbasis* (MEB) structure can be used to expand a given function in terms of the N eigenfunctions, again at the cost of just $O(N_g \log N)$ operations. This has been extended to *general 1D linear differential operators*. It constitutes a vast generalization of the Fast Fourier Transform (FFT), whose basis functions are the eigenfunction of discretized differential operators with constant coefficients, periodic boundary conditions with 2^ℓ uniformly spaced gridpoints. The new $O(N \log N)$ expansion is in terms of the eigenfunctions of a general operator with general boundary conditions and a general number of gridpoints. Also, summations like (9.3), with N terms summed at each of N_g gridpoints, are performed at the same $O(N_g \log N)$ cost.

To be sure, these algorithms have been based on formulations unique to the one-dimensional case. In particular, in 1D it has been possible to avoid solving highly indefinite boundary-value problems, thus skipping the more involved mechanism required for coarsening such problems (see Sec. 7). The extension to higher dimensions (discussed in [120, §6]) is far from trivial, and intimately related to the extension of the wave/ray multigrid methods to variable coefficients (see Sec. 7.1) and to general matrices (see Sec. 17.2.2).

A work related to the computation of the eigenvalue of a matrix and the fast updating of the singular-value decomposition of matrices is reported in Sec. 10.2 below.

10 Matrix Multiplication, Integral Transforms and Integrodifferential Equations

Multilevel algorithms have been developed for the fast evaluation of integral transforms, such as

$$Gu(x) = \int_\Omega G(x,y)u(y)dy \ , \qquad \Omega \subseteq \mathbb{R}^n \ , \ x \in \Omega' \subseteq \mathbb{R}^n \ , \tag{10.1}$$

and for the solution of corresponding integral and integro-differential equations. They exploit in various ways the smoothness properties of the kernel $G(x,y)$. For an $M \times N$ evaluation (i.e., x being discretized with M gridpoints, y with N) of either the transform *or its inverse* to a certain accuracy ϵ, these algorithms require $O\big((M+N)\log\frac{1}{\epsilon}\big)$ operations in the case of the Gaussian $\big(G(x,y) = \exp(-(x-y)^2/r^2)\big)$ or the potential-type $\big(G(x,y) = |x-y|^{-\alpha}$ or $G(x,y) = \log|x-y|\big)$ transforms, and $O\big((M+N)\log(\min(M,N))\log\frac{1}{\epsilon}\big)$ operations in the case of the Fourier $\big(G(x,y) = \exp(ixy)\big)$ or the Laplace $\big(G(x,y) = \exp(-xy)\big)$ transforms, for example. More generally, the algorithms yield *fast matrix multiplication* rules for dense matrices that include large low-rank submatrices. (See [53], [29], [158] and references in [158]). The same algorithms can also be used for the fast $\big(O(N\log\frac{1}{\epsilon})\big)$ summation of all the forces that N particles exerts on each other (see Sec. 14.3 below).

10.1 Adaptive grids. Integrodifferential equations

In their original form, the fast algorithms for evaluating (10.1) for potential-type kernels relied for their efficiency on the (asymptotic) smoothness of the *discrete* kernel (the matrix) and thereby on *grid uniformity*. However, in actual applications, e.g., in contact mechanics [158], in many cases large solution gradients as well as singularities occur only locally, and consequently a substantial increase of efficiency can be obtained by using *nonuniform* grids.

A new discretization and evaluation algorithm has been developed more recently in collaboration with Kees Venner. It relies on the (asymptotic) smoothness of the *continuum* kernel only, independent of the grid configuration. (Asymptotic smoothness roughly means that $G(x, y)$ is smooth except possibly near $x = y$; cf. [29].) This will facilitate the introduction of local refinements, wherever needed. Also, the new algorithm is faster: for a d-dimensional problem only $O(s^{d+1})$ operations per gridpoint are needed, where s is the order of discretization and d is the dimension. See [62], [63]: Numerical results were obtained for a model problem in which u has a singularity where its derivative is unbounded. It has been demonstrated that with the new fast evaluation algorithm on a *non-uniform* grid one can restore the regular work to accuracy relation (where accuracy is measured in terms of approximating the *continuum* transform), i.e., obtain the same efficiency as for the case without a singularity.

In combination with a multigrid solver, the fast evaluation algorithm also yields a fast solver for *integral and integrodifferential equations* [53], [158]. The main special feature of this multigrid solver is the distributive relaxation: a combination of several local changes to the solution is introduced at a time, such that the effect of the changes on the integral equation at any far point is negligible (due to the asymptotic smoothness of the kernel). A full multigrid (FMG) solver can be organized so that all its evaluations of integrals of the form (10.1), except for one, use reduced accuracy, hence costing much less. The cost of such a solver is only a fraction above the cost of just *one* (fast) evaluation of the involved integral transform.

The plan is to develop a multigrid solver for integro-differential equations discretized on adaptive grids, based on the new discretization and evaluation algorithm. As previously developed for PDE systems (see Sec. 6.1), self-adaptation criteria based on the local fine-to-coarse defect corrections (τ) are planned, as well as full integration of the grid adaptation process into the solver.

10.2 Secular equations. Discontinuous softening

A special case that involves dense-matrix multiplications is the computational task of finding all the roots $\{\lambda_j\}_{j=1}^{N}$ of the *secular equation*

$$1 + \sigma v(\lambda) = 0 \ , \qquad v(\lambda) = \sum_{k=1}^{N} \frac{u_k}{d_k - \lambda} \tag{10.1}$$

where $d_1 < d_2 < \cdots < d_N$ are real, $\sigma > 0$ and $u_k > 0$ for all k. This problem has various applications in numerical linear algebra, such as subspace computations [80], [84], solving constrained least-squares type problems [83], [92], updating the singular value decomposition of matrices [68], and modifying the symmetric eigenvalue problem [75]; see survey of literature and solution methods in [125], [126]. The equations should often be solved many times as a subproblem of a larger one. All existing solution methods cost $O(N^2)$ operation, since just the direct evaluation of $\{v(\lambda_j)\}_{j=1}^{N}$ costs that much. In joint work with Oren Livne, we used our multilevel dense-matrix-multiplication algorithm to solve the problem in $O(CN)$ operations, where C depends logarithmically on the desired accuracy [121]. The kernel here is $G(d, \lambda) = 1/(d-\lambda)$.

The fast algorithms for matrix multiplication and integral transforms described above are based on the *smooth* softening of singular (but asymptotically smooth) kernels (such as $G(d,\lambda) = (d-\lambda)^{-1}$). For high-order approximations, the softener (i.e., the modified kernel in the "softened neighborhood" of the singularity) is a high-order polynomial. This polynomial has to be calculated for each pair of variables in the softened neighborhood (e.g., each pair (d_k, λ_j) such that $|d_k - \lambda_j|$ is less than the "softening radius"). In a d-dimensional problem, and with a target accuracy ε, the softened neighborhood of each variable (e.g., each d_k) should include at least $O\Big(\big(\log\frac{1}{\varepsilon}\big)^{d+\eta}\Big)$ neighbors (neighboring values of λ_j), where $\eta \geq 0$ depends on smoothness properties of G and the order of the polynomial is $O\big(\log\frac{1}{\varepsilon}\big)$; hence the total work turns out to be $O\Big(N\big(\log\frac{1}{\varepsilon}\big)^{d+\eta+1}\Big)$. As shown in [121], this work can be reduced to $O\Big(N\big(\log\frac{1}{\varepsilon}\big)^{d}\Big)$ by choosing a *simple* (e.g., zero) but discontinuous softener. This requires a substantially more complicated algorithm, and can be advantageous only for a low dimensional (in particular: one dimensional, such as (10.1)) problem and high prescribed accuracy. The discontinuous softening is particularly advantageous to 1D kernels such as $(d-\lambda)^{-1}$ that are harder to soften (compared with $|d-\lambda|^{-1}$, the more common type).

11 Dirac Solvers

A major part of lattice quantum field calculations is invested in the inversion of the discretized Dirac operator M^h appearing in the fermionic action. Solutions of systems of the form

$$M^h \phi^h = f^h \tag{11.1}$$

are many times called for, either for calculating propagators or for the fast update of $\det M^h$ (see Sec. 12).

In the Euclidean staggered lattice formulation [138], the discrete equation at the gridpoint z is defined by

$$(M^h\phi)(z) = \frac{1}{h}\sum_{\mu=1}^{d} \eta_\mu(z)\big[U(z+\tfrac{1}{2}e_\mu)\phi(z+e_\mu) - U^\dagger(z-\tfrac{1}{2}e_\mu)\phi(z-e_\mu)\big] + m_q\phi(z)\ ,$$

where h is the meshsize of the grid, $\phi = \phi^h$, d is the number of dimensions, m_q is the (given) quark mass, and e_μ is a vector of length h pointing in the μ-th coordinate direction. η_μ are complex numbers of modulus 1, and may be chosen as $\eta_1(z) = 1$, $\eta_2(z) = (-1)^{n_1}$, $\eta_3(z) = (-1)^{n_1+n_2}$ and $\eta_4(z) = (-1)^{n_1+n_2+n_3}$ for the gridpoint $z = h(n_1, \ldots, n_d)$, n_ν being integers. $U(z+\frac{1}{2}e_\mu)$ is the gauge field value defined on the directed link $(z, z+e_\mu)$. The inversely directed link

$(z, z - e_\mu)$ carries the gauge field $U^\dagger(z - \frac{1}{2}e_\mu)$, where $\dagger$ denotes the Hermitian conjugate of the matrix. Each $U(z + \frac{1}{2}e_\mu)$ is an element of the model's unitary gauge group.

In collaboration with Michael Rozantsev, we have investigated two such models: $U(1)$ and $SU(2)$. In the $U(1)$ model, the gauge group elements are complex numbers of modulus 1, and $\phi^h(z)$ and $f^h(z)$ are complex numbers. (In the case of a trivial gauge field ($U \equiv 1$) in 2D, the $U(1)$ operator M^h reduces to the well known Cauchy-Riemann system.) In the $SU(N_c)$ model the gauge group elements are unitary complex $N_c \times N_c$ matrices whose determinant is 1, and $\phi^h(z)$ and $f^h(z)$ are complex N_c-vectors. See more about these models in [162], [112], [113], [114], [147], and about a multigrid approach to related, simplified models in [108] and [109].

These systems, despite their linearity and good ellipticity measures, are very challenging, due to their topology-induced singular (or nearly singular) eigenmodes and their disordered and non-commutative coefficients (the gauge field). The disorder results from the probabilistic physical rules by which the gauge field is determined, and from the "gauge freedom", i.e., the fact that those rules determine the field only up to arbitrary "gauge transformations". The latter are arbitrary multiplication of each $\phi^h(z)$ by an element of the gauge group and corresponding changes of the gauge field U so that (11.1) is still satisfied. Such changes do not change the physical content of the field.

11.1 Geometric multigrid solvers

Our first approach, based on red/black Kacmarz relaxation (since all equations in the Dirac system are first order), on pre-coarsening gauge smoothing and on multiscale iterate recombination, had previously been applied to the two-dimensional ($d = 2$) $U(1)$ model (see general description in [31], and full account in [130]). More recently we have been working on the $U(1)$ and $SU(2)$ gauge models in 4D [131], [132].

For the 4D-$U(1)$ gauge model, general conditions have been formulated under which the gauge field can be smoothed *globally* by gauge transformations, hence a fully efficient multigrid solver can, and has been, constructed. An important concept in this multigrid solver (as in any other geometric multigrid Dirac solver, for any model in any dimension) is to distinguish between different *species of unknowns* and between different *species of equations*. They can best be distinguished at the limit of low-temperature (well-ordered) gauge fields, for which each *species of unknowns* forms a grid function that must be constant for the homogeneous equations to be satisfied, and each *species of equations* forms a grid function which changes smoothly upon a smooth change of any one species of unknowns. The multigrid fine-to-coarse transition must transfer each fine residual to a coarse equation of the same species. Similarly, the coarse-to-fine transition must interpolate a correction to a fine unknown from course unknowns of the same species. It is also important, in the fine-to-coarse gauge averaging, to distinguish different *species of gauge links*: Two links are in the same species if they join the same species of unknowns.

The conditions for global gauge smoothing are *not* satisfied, however, in two kinds of topological situations. In the first kind, the total topological charge over the domain does not vanish. In this case the field can still be smooth *semi-globally*, i.e., it can be smoothed everywhere except for a certain local neighborhood which can easily be shifted away to any other place by gauge transformations, so that good intergrid transfers can be formulated *locally*. This is enough for obtaining nearly top multigrid efficiency.

The second topological case is more severe, featuring a *local* topological object, i.e. gauge-field discontinuities which cannot be shifted away. In such cases, and in many other cases (e.g., non-abelian models), it has been found that global or semi global gauge smoothing is not feasible. A general procedure has then been developed by which just *local* gauge smoothing at a time (over just 5 gridpoints in each direction) allows local constructions on each grid of the fine-to-coarse residual transfers and the coarse-to-fine correction interpolations required for the multigrid cycle. The local smoothing, which can be applied in any model, is done in a unique way, so that the resulting inter-grid transfers come out gauge invariant. Also, a general gauge-invariant procedure for the fine-to-coarse averaging of the gauge field itself has been constructed, based on transport averaging similar to that of [16], [14], [15]. This averaging has the advantage of reproducing on the coarse level various local topological objects, facilitating good coarse-grid approximations.

The local topological objects in the 4D-$U(1)$ model are "*monopol loops*", and they can also appear in a $U(1)$ component of a 4D-$SU(2)$ configuration. In the $SU(2)$ case these loops would persist for only few Monte-Carlo passes in gauge simulations at the critical temperature, and it is believed that in both $U(1)$ and $SU(2)$ only short loops are physically probable at critical or lower temperatures. If not treated, these loops lead to critical slowing down (CSD) of the multigrid solver (i.e., the larger the grid the more computational work per unknown is required). The number of slowly converging components introduced by the loops is small, however, so they can be eliminated by recombining iterants (taking linear combinations of results of the latest multigrid cycles so as to minimize the residual L_2 norm; which can also be done on coarser levels of the multigrid hierarchy; see [130], [59]) together with local relaxation passes added around the local discontinuities. With these devices, and with the local-gauge intergrid transfers and transport gauge averaging mentioned above, the multigrid convergence is still slower than in the absence of loops, but it seems free of CSD (at least when applied to gauge fields which are physically probable at critical or lower temperatures) [131]. We suspect that with wider regions of local relaxation the better efficiency may be restored; unfortunately, our domains were not wide enough for testing this.

Indeed, a severe problem in our work on these 4D models was the huge amount of computer time needed to produce reasonably sized, well equilibrated gauge fields on which to test our solvers: the Monte Carlo processes for producing these fields were far too slow. A general method to overcome this problem has only recently been devised (see Sec. 13.2 below). We hope to obtain by such a method larger 4D gauge fields for testing our Dirac solvers.

11.2 Algebraic multigrid solvers

Increasing complexities have been accumulating in the geometric-multigrid approaches described above: the different species, the treatment of various topological structures, the need for local smoothing of very disordered fields, etc. In particular, it has been found that for full efficiency the *geometric* coarsening (in which a coarse-level gridlines (or grid hyperplanes) consist of every other gridline (hyperplane) of the next-finer level) must be supplemented with simultaneous relaxation of various gauge-dependent strongly-coupled local sets of fine-level variables. Such sets can be identified by *compatible* relaxation sweeps (see Sec. 17). However, it became increasingly clear that *algebraic* multigrid (AMG) methods (see again Sec. 17) may be more convenient for treating at once all these difficulties. So, as already suggested in [31], we have returned to the development of AMG Dirac solvers.

Our first AMG Dirac solver has been based on the highly-accurate coarsening techniques of [37] (see Sec. 17.1 below). This solver, briefly described in [37, App. C] and detailed in [132], has been tested for the Schwinger model (two-dimension, with $U(1)$ gauge). We use Kacmarz or least square relaxation and distributive coarsening (i.e., in the notation of Sec. 17.2 below: either $(P = I,\ M = A^T)$ or $(P = A^T,\ M = I)$, both in relaxation and in defining coarse variables), which is usually a very good distribution for first-order PDE systems. The coarse-level set of variables is first selected *geometrically* (taking every fourth fine-level ghost variable, in a certain fixed 2D pattern). Then this set is enhanced using the compatible relaxation tool (see Sec. 17), thereby adding another 10%–20% of the ghost variables to the coarse level. The coarse-level equations have been derived using either a 3×3 or 5×5 coarse-grid stencil, each including also all those extra coarse variables added (following the compatible relaxation test) at the corresponding region. The coarse-to-fine interpolation of corrections has been done by several passes of compatible relaxation. Recombination of iterants has also been employed.

The different tests we ran, on a 32×32 grid, proved that *all and each one* of the above devices is necessary for fast convergence in the more difficult cases. Very good asymptotic convergence rates have been obtained (e.g., a convergence factor of 0.2 to 0.3 per two-level cycle) with the 5×5 coarse stencil even for *hot* (practically random) gauge fields, provided some 15% extra points were added to the coarse level and upto 8 iterants were recombined. For *critical* gauge fields only a couple of iterants needed to be recombined.

Satisfactory as these results are, they can only serve to demonstrate the potential of the AMG approach. However, the coarsening method employed here, even though local in principle, is far too expensive, especially for the 5×5 stencil. Even with this size of stencil some iterant recombinations have proved necessary, showing that the lowest lying eigenmodes are not yet sufficiently well approximated. There is no hope to efficiently use this approach for large 4D models. A far less expensive coarsening, that can well approximate even the near-zero modes, is the improved "bootstrap" AMG approach described in Sec. 17.2 below.

12 Fast Inverse-Matrix and Determinant Updates

In parallel to the development of the multigrid fast Dirac solvers (Sec. 11), collaborating again with Michael Rozantsev, methods have been developed for using multigrid solvers for constructing an inexpensive structure of the *inverse* Dirac matrix, allowing fast self-updating upon each change in the matrix itself (each gauge update). This can be generalized to allow fast updating of the fermion propagators and the associated determinant (needed at each step of the Monte Carlo simulations of the unquenched gauge field). The approach was first described in [26, §12]; the substantially improved version presented here exploits the development described in Sec. 17 below.

For a large d-dimensional lattice with $N = L^d$ sites and meshsize $h = O(L^{-1})$, the storage of the Dirac inverse matrix $(M^h)^{-1}$ would require $O(N^2)$ memory and $O(N^2)$ calculations, even for fully efficient multigrid solvers. Using the following special multigrid structure, they can be reduced to $O\big(N(\log\varepsilon^{-1})^{q_1}\big)$ and $O\big(N(\log\varepsilon^{-1})^{q_2}\big)$, respectively, where ε is the relative error allowed in the calculations and q_1 and q_2 depend only on d. More important, the structure will allow very fast *self-updating*.

The implemented multigrid structure first calculates and stores $O(\log L)$ *accurate algebraic coarsening* levels of the operator M^H, from the given finest ($H = h$) to the coarsest possible ($H = O(1)$). Each M^H is constructed from the next finer one in the manner of [37] (see Secs. 11.2 above and 17.1 below): for accuracy ε, the stencil of M^H at each point involves some $n_c = O\big((\log\varepsilon^{-1})^{q_2}\big)$ neighbors (so each row of M^H has n_c non-zero terms). (With the much more efficient methods of Sec. 17.2, substantially smaller n_c will be needed.)

Secondly, the proposed structure calculates and stores at each level enough "central terms" of $(M^H)^{-1}$. Such central terms in each column of $(M^H)^{-1}$ are terms corresponding to variables neighboring to the equation associated with that column (e.g., the n_c neighbors participating in the equation will suffice). The central terms of $(M^H)^{-1}$ can easily be constructed from those of the next coarser level, since the latter are all one needs in a two-level multigrid cycle for solving the relevant systems (at the finest level, for example, each of these systems has the form (11.1), with f^h being the delta function corresponding to the calculated column of $(M^H)^{-1}$). See details in [132].

This structure can be immediately updated upon changes in the gauge field. Indeed, each local change in the gauge field, if done in a properly distributive manner (i.e., so that some moments of the fields are kept unchanged) has only local effect on the propagators. Since the calculation of the latter can be regarded as solving by multigrid the system (11.1) with $f^h = \delta_{x,y}$, the effect of each local change can be calculated just by local relaxation sweeps around the change on some of the finest levels. More global changes will similarly be introduced (in a distributive manner) at coarser levels of the gauge field Monte Carlo simulations. The cost per update is $O(1)$, i.e., independent of lattice size.

With $(M^h)^{-1}$ thus monitored, one can inexpensively calculate changes in

$\log detM^h$. For a *small* change δM^h in the gauge field

$$\delta \log detM^h = Tr\big((M^h)^{-1}\delta M^h\big) \ , \tag{12.1}$$

which can be computed locally, based on the central terms of $(M^h)^{-1}$. For *larger* changes one can locally integrate (12.1), since the local processing also gives the dependence of $(M^h)^{-1}$ on δM^h. Again, the amount of calculation per update does not depend on the lattice size.

Simplified model. The approach described above has first been developed for model matrices with a simplified structure: matrices M^h arising from discretizing on a lattice with meshsize h the random diffusion equations $Lu = f$, where

$$Lu(x,y) = \frac{\partial}{\partial x}\left[a(x,y)\frac{\partial}{\partial x}u(x,y)\right] + \frac{\partial}{\partial y}\left[b(x,y)\frac{\partial}{\partial y}u(x,y)\right] \ ,$$

and the discrete analogs of the diffusion coefficients $a(x,y)$ and $b(x,y)$ have random values, uniformly distributed in $(0,1)$. Excellent accuracies were obtained, but they required very expensive (though *local*) coarsening: 5×5 and even 7×7 stencils [132]. Far less expensive algorithms, based on the BAMG methods of Sec. 17.2, are possible.

13 Monte Carlo Methods in Statistical Physics

The general goal is the systematic development of advanced multigrid Monte-Carlo (MC) methods in statistical mechanics, molecular dynamics, quantum mechanics and quantum field theory.

A Monte Carlo simulation aimed at calculating an average of a certain observable is called "*statistically optimal*" if it achieves accuracy ϵ in $O(\sigma^2\epsilon^{-2})$ random number generations, where σ is the standard deviation of the observable. This is just the same order of complexity as needed to calculate, by statistical sampling, any simple "pointwise" average, such as the frequency of "heads" in coin tossing. The goal is to attain such an optimal performance in calculating much more complicated averages in statistical physics, including in particular thermodynamic limits, i.e., limits approached by the averages of system observables when the system size tends to infinity.

Two basic factors usually prevent naive Monte Carlo calculations of a thermodynamic limit from being optimal, even when $O(\sigma^2\epsilon^{-2})$ independent samples are indeed enough to average out their deviations down to $O(\epsilon)$ accuracy. First, to achieve an $O(\epsilon)$ approximation to the thermodynamic limit, each sample should be calculated on a system of sufficiently large *volume*, that is, a system whose linear size L grows with ϵ^{-1}; typically $L \sim \epsilon^{-\rho}$ for some $\rho > 0$. So in d physical dimensions, the required simulation volume for each sample is $L^d = O(\epsilon^{-\rho d})$. This factor is called the *volume factor*. The second factor is the *critical slowing down* (CSD), i.e., the increasing number n of MC passes needed when L grows in order to produce each new (essentially independent) sample; usually $n \sim L^z$, where z is typically (at least at the critical temperature) close to 2. As a result of these two factors, the cost of calculating the thermodynamic limit to accuracy ϵ rises as $O(\sigma^2 \epsilon^{-2-\rho d-\rho z})$. (Additional complexity factors, that multiply these two, arise in quantum field theory from propagator calculations and fermionic interaction; separate research for eliminating them is described in Secs. 11–12 above.)

Two different multiscale approaches have been developed for treating these two complexity factors. They are respectively described in Secs. 13.1 and 13.2 below, the latter being more generally applicable (e.g., for molecular-dynamics calculations: see Sec. 14). Each of these approaches generates a sequence of increasingly coarser descriptions ("levels") of the simulated system. The coarser levels basically perform three different tasks:

(i) *Acceleration* of the Monte Carlo simulations on the finer levels (to eliminate the CSD). This is in general similar to the multigrid *convergence* acceleration in PDE solvers.

(ii) *Gathering statistics*: large-scale fluctuations can be cheaply averaged out through coarse-level MC, by cycling enough between these levels (much more than usual in multigrid PDE solvers) before returning to finer levels. Indeed, averaging out *fine-scale* fluctuations does not require *many* returns to the fine levels, since such fluctuations are largely averaged out in any *one* fine-level configuration.

(iii) Increasingly larger computational domains can be simulated cheaply by using increasingly coarser levels: The finest level covers only a relatively small domain, or small "windows"; a coarse level created from it can then switch to a larger domain.

The elimination of both the volume factor and the CSD factor implies *ideal performance* (statistical optimality). It also implies that on sufficiently coarse level the derived description allows *true macroscopic simulation* of the system, i.e., such that does not require operations at finer levels.

Statistical optimality was first demonstrated in calculating various thermodynamic limits in Gaussian models with constant coefficients, and also in calculating the critical temperature of the Ising model [48], [86]. This led to several years of efforts to develop interpolation-based multigrid algorithms, with cycles similar to those of multigrid PDE solvers. These are reported in Sec. 13.1 below. The increasing complicated nature of the interpolation rules and the coarse-level Hamiltonians required to treat advanced non-linear models has later brought us to favor renormalization-type methods, which are described in Sec. 13.2.

13.1 Interpolation-based methods

These methods imitate multigrid solvers of discretized differential minimization problems, where the *Hamiltonian* (or the *energy* functional which should be minimized) is automatically defined on increasingly coarser grids by recursively specifying, level after level, coarse-to-fine interpolation rules. They also use the same type of cycles, except that the relaxation sweeps (each composed of a sequence of local minimization steps) are replaced by Monte Carlo sweeps (local steps of simulating the probability distributions induced by the Hamiltonian). The *cycle index* (specifying how many times one switches from a given multigrid level to the next coarser level per each switch to the next finer level) in statistical multigrid algorithms for computing large-scale observables will be larger than usual in multigrid solvers, to allow cheap averaging of large-scale fluctuations.

To obtain statistically optimal algorithms, as defined above, it is necessary to choose the coarse-to-fine interpolation so as to obtain full physical mobility at the coarse level: poor interpolation would not allow accessing mutually independent samples at the coarse level without in-between visits to the fine level for Hamiltonian updating. In addition, for statistical optimality the resulting coarse-level Hamiltonians need to be computable in a bounded complexity per coarse-level degree of freedom; they cannot be left just expressed in terms of finer-level variables.

Statistical optimality was first demonstrated for *Gaussian models with constant coefficients* [86], [48]. It was shown there, for the one-dimensional Gaussian model, that the susceptibility thermodynamic limit can be calculated to accuracy ϵ in about $4\sigma^2\epsilon^{-2}$ random number generations, while the average energy per degree of freedom requires $3\sigma^2\epsilon^{-2}$ generations for a similar accuracy. It was also found that the algorithmic flow (as determined by the multigrid *cycle index*) should generally depend on the observable being calculated. In the *two-dimensional* Gaussian model, the susceptibility limit can be measured to accuracy ϵ in about $20\sigma^2\epsilon^{-2}$ random number generations. In the one-dimensional *massive* Gaussian model, the susceptibility limit can be calculated in less than $8\sigma^2\epsilon^{-2}$ random generations, essentially independently of the mass size, although the algorithm flow may change with that size [45].

For the *variable-coupling Gaussian models*, it was shown that in order to reach ideal performance, the algorithm should employ during the multigrid cycle *weighted* interpolation and *variable* sampling (the Monte Carlo process should sample more frequently regions with smaller coupling values because such regions have larger contributions to observable fluctuations). Such algorithms have been implemented for *strongly* discontinuous cases in one and two dimensions. ("Strongly" means that the couplings may change by orders of magnitude between adjacent regions.) For the one dimensional variable-coupling Gaussian model, the susceptibility limit is calculated to accuracy ϵ in less than $8\sigma^2\epsilon^{-2}$ random number generations. In the two-dimensional variable-coupling Gaussian model, that limit can be measured in less than $20\sigma^2\epsilon^{-2}$ random generations [46]. These results are independent of the maximal ratio between strong and weak couplings, unlike the severe extra slowness that large such ratios can inflict on pointwise Monte Carlo.

The development of an optimal algorithm for the variable-coupling Gaussian model provides an important tool for general non-linear models, where non-constant couplings stochastically emerge at coarser levels of the multigrid Monte Carlo processing.

Doubts have however been raised whether ideal MC performance can really be obtained for non-linear models, where large-scale fluctuations are highly correlated with small-scale fluctuations. By applying the new analysis methods to the nonlinear *anharmonic crystal model* we have shown, and confirmed by actual simulations, that, down to a certain (small) ϵ, performance similar to that of the Gaussian models can still be obtained (although it requires careful choice of the multigrid cycling parameters [87], [47]). Such a performance is realizable because the large-scale fluctuations depend only on some *averages* of the small-scale fluctuations, and these averages are approximated well enough at any *single* fine-level configuration used at coarsening.

For a sufficiently small ϵ, however, and for models sufficiently dominated by the anharmonic term, both the analysis and the numerical tests show that ideal performance can no longer be obtained by a multigrid process which employs weighted linear interpolation. In fact, the analysis shows that no interpolation in the form of a linear operator can obtain ideal performance for all ϵ. We have therefore introduced another type of interpolation, the *minimization interpolation*.

This interpolation is best defined in terms of the Full Approximation Scheme (FAS; cf. [61, §7]), where the coarse-grid variables represent the *full* current configuration (i.e., the sum of a coarsened representation of the current fine-grid configuration and the current coarse-grid correction) instead of just the current coarse-grid correction. To define a value u_0 at a fine-grid point based on coarse-grid values $(u_1, u_2, \ldots)$, the minimization interpolation method is first to calculate $U_0(u_1, u_2, \ldots)$, defined as the value of u_0 that would be obtained by some, exact or approximate, local Hamiltonian minimization with the values of $(u_1, u_2, \ldots)$ being held fixed. Then, to retain statistical detailed balance, the FAS minimization-interpolation value is defined by

$$u_0 = U_0(u_1, u_2, \ldots) + \widetilde{u}_0 - U_0(\widetilde{u}_1, \widetilde{u}_2, \ldots) \ . \tag{13.1}$$

where the $\widetilde{u}_i$ are the values of the variables *at coarsening*, i.e., at the last transition from the fine level to the current coarse one.

Two-level unigrid experiments with the anharmonic crystal model have shown that the volume factor, along with the CSD, can be completely eliminated with an *exact* minimization interpolation. However, this interpolation creates a complicated coarse-level Hamiltonian, so we next designed simple *approximate* minimization interpolations, employing polynomial best fit. This yields a simple (fourth-order polynomial) coarse level, allowing the recursive construction of still coarser levels and application of complete *multi*-level cycles, which do indeed demonstrate the desired ideal MC performance [87], [47].

The situation is less convenient in more advanced physical models, where topological structures are present, because large-scale topologies may be correlated to specific fine-scale features, such as vortex centers. Also, linear-like interpolation of spinors is problematic.

A partial elimination of the volume factors in measuring susceptibility for Ising models was previously obtained by the three-spin coarsening technique [31], [48], as well as full elimination of that factor (namely, ideal MC performance) in determining that model's critical temperature [48].

Various attempts to attain ideal performance for two-dimensional non-linear σ *models* (several of which are described in [142]) have failed. Nevertheless, we have developed a variety of new *stochastic* coarsening procedures by which at least *partial* elimination of the volume factor can be achieved. These procedures include: a detailed-balance way to associate the introduction of linear (or linear-like) interpolation with a certain probability for reducing adjacent coupling strength; smart choice of the interpolation in a neighborhood depending on local features at coarsening; stochastic simplification of the derived coarse-grid Hamiltonian in ways which do not destroy the statistical detailed balance; and introduction of less restrictive stochastic interpolations [142]. Most of the developed schemes are applicable to specific cases of XY and Manton's models, while some are universal for any $O(N)$ model.

Specially devised two-grid numerical experiments have demonstrated that the designed techniques are capable of eliminating the volume factor almost *completely* at low temperatures of the XY and Manton's model, and *partially* in the $O(4)$ model as well as in the critical region of the XY model. The non-optimality of the latter results have been attributed to the insufficient accuracy in representing and sampling some of the statistically important features by means of currently employed interpolation and stochastic coarsening procedures.

This led us to an attempt to introduce the FAS minimization interpolation (13.1) also to the XY model. It yielded an improved, but not yet statistically optimal, performance. The reason for non-optimality has been shown to be the bias introduced by the FAS correction $\widetilde{u}_0 - U_0(\widetilde{u}_1, \widetilde{u}_2, \ldots)$. For example, if the coarse configuration at coarsening $(\widetilde{u}_1, \widetilde{u}_2, \ldots)$ happens to be locally *non-smooth*, then the corresponding FAS correction is likely to be large, preventing the coarse-level system from efficiently sampling *smooth* configurations. A way around this difficulty is to replace (13.1) by

$$u_0 = U_0(u_1, u_2, \ldots) + \frac{Q_0(u_1, u_2, \ldots)}{Q_0(\widetilde{u}_1, \widetilde{u}_2, \ldots)} \left[\widetilde{u}_0 - U_0(\widetilde{u}_1, \widetilde{u}_2, \ldots)\right] , \tag{13.2}$$

where $Q_o(u_1, u_2, \ldots)$ is a characteristic size of the likely fluctuations in u_0 given $(u_1, u_2, \ldots)$. More precisely, the interpolation (13.2), like (13.1), is suitable in case u_i are real variables; it has modified forms to suit other types of variables, such as XY.

13.2 Renormalization multigrid (RMG)

The increasing complexity of the coarse Hamiltonians produced by the interpolation-based techniques has led more recently to a new type of algorithms. They combine renormalization-like derivation of increasingly coarser "descriptions" of the system, with multigrid-like coarse-to-fine Monte-Carlo accelerations.

This RMG approach has already yielded optimal performances (eliminating both the CSD and the volume factors) for certain observables in the Ising model, and it can in principle be extended to arbitrary models, since it is not based on cluster-type (such as Swendsen-Wang and Wolff) algorithms. In fact, we have already preliminarily used this approach in several models of molecular dynamics: a model of polymers (see Sec. 14.6) and models of fluids (see Sec. 14.7).

Moreover, the approach has inspired a similar, equally-general procedure for the coarsening of *deterministic* equations (see Sec. 17).

For simplicity, the RMG techniques are surveyed here mainly in terms of the 2D Ising model, where they were first developed by Dorit Ron [60]. In this model, each fine level configuration U consists of an unbounded two-dimensional lattice of Ising spins u_i (variables taking either the value $+1$ or -1), with a probability distribution $P(U) \sim \exp\big(-\mathcal{H}(U)/k_\beta T\big)$, and Hamiltonian $\mathcal{H}(U) = -J\sum_{\langle i,j\rangle} u_i u_j$, where $\langle i,j\rangle$ runs over all distinct pairs of nearest-neighbor spins.

Coarsening. Generally, each coarse-level "description" in RMG consists of two items: The coarse-level *variables*, and the *statistical rules* that govern their probability distributions. The rules will generally be expressed in terms of *conditional probability* (*CP*) *tables* (whose far-reaching generality will be discussed in Sec. 13.2.2 below).

Each coarse-level *variable* will be defined in terms of a small local set of next-finer-level variables. For example, in 2D-Ising with *majority rule* blocking, the coarse level consists again of a 2D array of Ising spins (± 1 variables), each of which being a "block spin", i.e., its sign representing the sign of the sum of the four spins in a corresponding $b \times b$ block of the next finer level. (The sign of zero is taken to be $+$ or $-$, each with probability $1/2$. Usually $b = 2$.)

There is no unique way to choose the set of coarse-level variables. Indeed, given the full description (variables + CP table) of the next finer level, many different choices can be equally good. There exists however a general *criterion to gauge the adequacy of any candidate course set*. This criterion is the speed of equilibration of *compatible Monte Carlo* (CMC) runs. A CMC is a Monte Carlo process on the fine level which is restricted to the subset of fine-level configurations whose coarsening (e.g., $b \times b$ blocking) coincides with a given *fixed* coarse-level configuration. A *consistently fast* CMC equilibration (i.e., CMC with very short average decorrelation time, averaging being over an ensemble of the fixed coarse configuration) implies that the fine-level equilibrium can be produced from the coarse-level equilibrium just by local processing, which is indeed the main desired property of coarsening.

The fast CMC equilibration implies that the set of coarse variables enjoys the *near locality property*. This is the property that the *conditional* probability distribution of a coarse variable at a point A, *given fixed states of all other coarse variables*, depends mainly of the closest neighbors: the average dependence decays exponentially with the distance from A. (For example, in 2D-Ising if the neighborhood of A is changed only at points at distances larger than r from A, the conditional probability to have $+1$ at A given its entire neighborhood can change at most by $O\big(\exp(-c_\ell r)\big)$, with some constant c_ℓ.)

The strength of near locality (the rough size of c_ℓ) can be directly measured. Strong near locality has been measured, e.g., for the 2D-Ising with the majority coarsening. This yields the possibility to construct CP tables for coarse levels by the following quite general branching procedure.

The CP table for each coarse level is derived by running an efficient MC simulation at the next finer level, during which appropriate statistics are gathered. In 2D-Ising, statistics are gathered for estimating $P_+(u_1, u_2, \ldots, u_m)$, the probability for any block spin to be $+1$ given that its (ordered set of) neighboring block spins are having the values $(u_1, u_2, \ldots, u_m)$.

The size m of the considered neighborhood is *variable*: If only a small amount of statistics is gathered, only the four nearest neighbors (u_1, u_2, u_3, u_4) are considered. With more statistics, all *eight* closest (nearest and next-nearest) neighbors $(u_1, \ldots, u_8)$ are considered. Further on, when the amount of statistics *for a particular neighborhood* $(u_1, \ldots, u_8)$ is sufficiently large, that neighborhood is *split*, i.e., statistics is gathered for its "*child neighborhoods*": These are neighborhoods $(u_1, \ldots, u_8, u_9, \ldots, u_{12})$ with the same $(u_1, \ldots, u_8)$ as in the "parent" neighborhood, and with $(u_9, \ldots, u_{12})$ representing values of the four *subsequent* neighbors (those which are exactly two meshsizes away). Children with enough statistics may further be split into grand-children, and so on. The general rule is to split a neighborhood when (and only when) some of its children have enough statistics to make the difference between their P_+ values significant (i.e., larger than their standard deviations). Between several candidate splits of a neighborhood, the one with the largest *spread* (average child deviation) should be adopted.

The CP (e.g., P_+) table represents the coarse-level transition probabilities. Indeed, it is all one needs, and exactly what one needs, to run an MC simulation on that level (the level of blocks). Also, due to the adaptability in the size of the neighborhoods, this method for calculating the coarse transition probabilities is *statistically optimal*, in the sense that it will yield $O(\varepsilon)$ relative accuracy in calculating large-scale averages when the amount of statistics (the total number of random-number generations in producing the CP table) has been $O(\varepsilon^{-2})$. This claim has been confirmed in a sequence of numerical tests [60]. It may depend on having a fully efficient Monte-Carlo simulation at the fine level. Which is the next topic.

Monte-Carlo Acceleration. For a given finite lattice with a given CP table, suppose now that the CP tables for all its coarser levels (the level of blocks, the level of blocks of blocks, etc.) are also given. Then a new equilibrium configuration of the given action on the given lattice can efficiently be produced using the following "*coarse to fine equilibration*" (CFE) cycle.

First an equilibrium is easily obtained at the coarsest level of the finite lattice, by few MC passes with the corresponding CP table. From this, an equilibrium in the next level will be derived, and so on, until the target level (the given lattice) will be reached. To obtain an equilibrium in any level of spins given an equilibrium of its blocks, we use "*stochastic interpolation*", which is simply a small number of CMC passes. If the coarse-level (the block) CP table has not been fully accurate, the CMC passes should be followed by a small number of regular MC passes, a process we call "*post relaxation*".

A particular advantage of this equilibration process is the ability to cheaply produce very far regions of the same equilibrium configuration, without having to produce (at the fine levels) all the regions in between. This yields a very efficient way to calculate *far correlations*. (See also below about the role of "windows".)

Fast iterations. Since the derivation of a coarse CP table depends on efficient simulations at the finer levels, which in turn depends on employing the coarse CP table, iterating back and forth between these processes is in principle needed. However, these iterations converge very fast, since these processes only *weakly* depend on each other. Indeed, a very good first approximation to the P_+ block-spin tables is already obtained by just *local* equilibration at the spin level, produced by just a limited number (independent of the lattice size) of regular Monte Carlo passes (even starting from a completely random configuration). And, on the other hand, a very good approximation to critical-temperature equilibrium at the fine level can be obtained from a CFE cycle even with crude approximations to the coarse level CP tables (as long as they are kept critical: see below), provided a small number of post-relaxation sweeps is added.

In fact, an extremely simple way to obtain an approximate equilibrium on a given lattice with a critical action is by a CFE cycle employing *this same action* at all levels, plus post relaxation at each level. It can be shown that the required number of post relaxation sweeps is small whenever the convergence to a fixed point of the renormalization flow is fast. For the 2D-Ising model we have confirmed, in a sequence of numerical tests, the good approximation to equilibria obtained in this simple way [60].

Windows. Although just local equilibration is enough to produce good approximations to the CP table in the Ising model, in many other models it may suffer from *too low statistics* (or no statistics at all) for certain neighborhoods that are not *generally* rare, but that happen to be rare in the simulated regions (see example in Sec. 14.7.2). Hence, simulations at some coarse levels may run into regions whose simulation is inaccurate due to poor statistics in the CP table. In such a situation, and exactly at those regions, temporary *local* returns to finer levels should be made, in local *windows*, to accumulate more CP statistics relevant to the local conditions there. This can be done by first interpolating the window to the finer level: using CMC fine-level passes over the window, a correct equilibrium is generated in its *deep* interior (far enough from its borders); regular MC passes can then be done in that deep interior, to accumulate the desired statistics.

Generally then, the CFE technique would mostly be applied in windows rather than on the entire domain, its main role really being to supply rich samples of neighborhoods.

Errors. The principal sources of errors in the RMG processes are the finite statistics, the truncated size of the neighborhoods for which each CP table is calculated, and the finite size of the lattice employed at each level.

The latter type of error is easily removed: arbitrarily large lattices can be used due to the fast equilibration, while the P_+ calculations have nearly-local nature at all levels; and the simulation at any coarse level can be extended to its own desired domain size, since it is done directly, not through simulations at a finer level.

The finite-statistics errors are well controlled so as to keep all of them, at all levels, at the same optimal order ε, where the total amount of statistics is $O(\varepsilon^{-2})$. The truncation errors are also kept at $O(\varepsilon)$, by adjusting the neighborhood sizes; it is estimated that the size of the considered neighborhoods should grow proportionately to $\log(\varepsilon^{-1})$. The only trouble is the error enhancement from level to level, which is discussed next.

Back to criticality. In critical-temperature calculations of the CP tables, errors introduced at any level are magnified in the level derived from it (the next coarser level), and so on, due to the strong divergence of the renormalization flow away from the critical surface. To hold back this magnification, a mechanism should be added at each level to project the CP tables back onto the critical surface. Such a "criticalization" mechanism also facilitates calculating renormalization flows toward a fixed point when the critical temperature of the initial (finest-level) Hamiltonian is not known in advance.

The criticalization of a given CP table can be done by multiplying the temperature by a suitable factor $1/\theta$, i.e., by raising each probability to the power θ, then normalizing. (Normalization is not really needed for MC simulations with this CP tables.) The criticalization factor θ can be estimated in a number of ways. See details in [60]. Another, very accurate type of criticalization can be done near the fixed point (see next).

Fixed point algorithm. The fixed point of the renormalization group is approached by a sequence of coarsening steps, as described above, with a criticalization factor applied to each new CP table in the sequence. At each step the accuracy is raised in every respect: The total amount of statistics is much increased, accompanied by a (slower) increase in the lattice linear size and a (logarithmically slow, as dictated by the amount of statistics) increase in the size of the neighborhoods.

Critical exponents. The calculation is done in terms of the vector p of the entries of the CP table. Each stage of the fixed point algorithm can be regarded as a transformation $\mathcal{T}$, transforming p into $\mathcal{T}p$. The algorithm converges to the fixed point $p_* = \mathcal{T}p_*$. The correlation length critical exponent can immediately be derived from the largest eigenvalue λ_* of the equation

$$\mathcal{T}(p_* + q) = p_* + \lambda q \ , \qquad \| \, q \, \| \ll 1 \ .$$

Denoting by q_* the normalized eigenvector associated with λ_*, at each stage of the fixed point algorithm, increasingly more accurate values for λ_* and q_* are calculated (for the exact procedures, see [60]).

If p is near the fixed point p_*, further criticalizations and better approximations to p_* can be obtained by iterations, in each of which p is replaced with $(\lambda_1 \mathcal{T}_k p - \mathcal{T}_{k+1} p)/(\lambda_1 - 1)$, where λ_1 is the current approximation to λ_* and $\mathcal{T}_k p$ is the CP table calculated for $b^k \times b^k$ spin-blocks while running spin simulations with the CP table p; in particular $\mathcal{T}_1 = \mathcal{T}$. For convergence one should use $k > (\lambda - 1)^{-1}$; we used $k = 2$.

Once the CP table at the fixed point has been accurately calculated, the CFE cycle can be operated with the same p_* table being used at all levels, cheaply producing large equilibrated configurations (or very far regions of the same equilibrium configuration, as noted above). This can be used in a variety of ways for highly accurate calculations of various other critical exponents (a work in progress, by Ron and Swendsen).

13.2.1 Extension to continuous-state models

Initial steps of applying the above coarsening and acceleration techniques to the XY model are reported in [142]. Each 2×2 block spin is here defined to be the *average* of its four constituent spins, *without normalization* (whereby the original XY group of length-1 vectors is *not* preserved at the coarse levels). Compared with the ± 1 majority spins discussed above, each coarse spin here contains much more information; as a result, much smaller neighborhoods are needed in the probability tables to attain a given truncation accuracy. Still, these tables are more complicated than the Ising P_+ tables, since they should describe a *continuous* distribution, conditioned on *continuous* neighboring values.

To accumulate continuous-variable statistics, one partitions the range of this variable into *bins*: Counting the number of MC hits in each bin gives an estimate for the *integral* of the continuous variable over that bin. From those integrals, the value of the variable at any particular point can be *interpolated* (by a polynomial whose integrals over several adjacent bins fit the estimates). The same is true for a *vectorial* variable, such as the one representing the entire (truncated) neighborhood, whose bins may each be a tensor product of *elementary* bins, one elementary bin per each real variable participating in describing the neighborhood. More generally, the bins of the neighborhood are constructed *adaptively*, similar to the adaptively branching neighborhoods in the Ising case above, except that here a bin can be split into several bins in *two* ways: either by adding another variable to the description of that particular neighborhood, or by refining the current elementary binning of one or several of the existing variables.

The set of tests with the XY model reported in [142] clearly indicates that ideal MC performance free of the volume and CSD factors can be obtained in calculating various thermodynamic limits, such as the two-point correlation and the scaled susceptibility.

Future plans. The intention is to extend the RMG techniques to more advanced physical problems, possibly including gauge field models such as $U(1)$, $SU(2)$ and $SU(3)$. Together with the methods of Secs. 11–12 above, it is hoped ultimately to obtain ideal MC performance also with unquenched fermionic interactions.

13.2.2 Generalizations. Driven systems

As explained above, by a proper choice of coarse variables (checked by the CMC equilibration speed) the property of *near locality* is obtained for the next coarser level (the level of blocks), which allows the construction of that level's *CP table*.

Notice that the near locality property indirectly holds even in the case of long-range interactions, such as electrostatic or gravimetric interactions. Indeed, each such interaction can be decomposed into the sum of a *smooth* part and a *local* part (where "smooth" and "local" are meant relative to the particular scale of the next coarser level). All the smooth parts can be transferred (anterpolated) directly to the coarse level (cf. Secs. 14.3 and 14.6.1), hence it is only the local parts that remain to be expressed on the coarse level. For that expression the near-locality property still holds.

The CP table is a much more general representation of the "dynamics" (the transition probabilities) of models than the Hamiltonian representation, in the same way that, for deterministic models, systems of (differential or discrete) *equations* are much more general than variational (energy minimization) problems.

It has been found by Ron and Swendsen [129] that the CP representation of the renormalization-group transformation $\mathcal{T}$ described above provides an excellent test for the validity of the more common Hamiltonian representation. They have also developed a method based on the CP representation that allows them to produce a stable calculation of larger sets of renormalized coupling constants than either the Swendsen [148] or the Gupta-Cordery [97] methods, thus reducing the effects of truncation in renormalization-group calculations.

The CP table is particularly useful when even the given fine-level system is non-Hamiltonian. Such systems abound. An important case is that of *time-dependent* systems, such as driven diffusive systems [136]. The CP representation of such systems can be renormalized in both space and time, at various space/time coarsening ratios, yielding long-time and large-scale dynamics of the system.

A particular type of such renormalization leads to a fine-level CP table for the steady state of the driven system. The idea is to construct CP tables for the dependence of fine-scale fluxes on neighboring fluxes at the same time level t and on average densities at a previous time $t-\delta t$, where δt is doubled at each further renormalization transformation (together with a corresponding increase of the scale at which the densities are averaged). At the limit of such transformations, the steady-state CP tables should emerge.

13.2.3 Low temperature algorithms for frustrated system

Frustrated systems are those in which conflicting influences arise from different terms of the Hamiltonian; e.g., some terms tend to align neighboring spins with respect to each other, while others tend to anti-align them. For such systems, especially at low temperatures, simple blocking (such as $b \times b$ blocks with the majority rule) are inefficient, yielding slow CMC equilibration. In this situation, correct coarsening can gradually be identified, for increasingly larger scales, by a gradual decrease of the temperature. Cf. Sec. 14.7.3 below. See also Sec. 18 for the limit case of zero temperature (strict minimization).

14 Molecular Mechanics

14.1 Background and objectives

Molecular mechanics (or dynamics) is a major tool of theoretical chemistry, with immense practical potential in medicine, material design and biotechnology. The Born-Oppenheimer approximation to the potential energy $E(r)$ as function of the n atomic positions $r = (r_1, r_2, \ldots, r_n)$ can be imagined as the objective functional of these calculations, the electrons being implicit. Explicit approximations to $E(r)$ as a sum of various few-atom interactions are derived by accumulated computational experience, compared with finer-scale calculations (such as those discussed in Sec. 9 above) and with molecular measurement data (crystal structure geometries, vibrational spectroscopy, heats of formation, etc.). The most common few-atom interactions are of the following two kinds (see a typical example in Sec. 14.6): (1) *The bond interactions* between chemically-bonded atoms, including three types: length (distance) interaction between 2 atoms, angle interaction between 3 atoms and torsion interaction between 4 atoms. The first is much stronger than the second, which in turn is much stronger than the third. (2) *Non-bond interactions*, including the short-range Lennard-Jones and hydrogen-bond terms and the long-range Coloumbic potential.

The aim of the calculations is usually either *statics* (finding the configuration r which minimizes E), *dynamics* (calculating trajectories $r(t)$ which satisfy Newton's law $-\nabla E(r) = M\ddot{r}$, where M is the diagonal matrix of masses), or *equilibrium statistics* (average properties under the probability distribution $P(r) \sim \exp\big(-E(r)/k_B T\big)$), where k_B is the Boltzmann constant and T is the absolute temperature).

The computing cost of current molecular dynamics algorithms rises very steeply with problem size, restricting the modeling efforts to relatively small molecular ensembles and to time intervals many orders of magnitude smaller than needed. Preliminary model studies conducted by [33], [34] have indicated that this steep rise in cost can be radically reduced by combining several types of multiscale approaches.

Our research objective is to develop these approaches and demonstrate their ability to perform the above computational tasks in computing times that rise only *linearly* with the number n of atoms in the system. Also, the aim is to show the possibility to blend statistical approaches in the small (for the high-frequency molecular oscillations) with deterministic dynamics or statics in the large (see Sec. 14.8 below). The long term goal is to establish the computational tools for the development, scale by scale, of material "descriptions" at increasingly larger scales, each description being either in the form of a Hamiltonian, or more generally in the form of conditional probability tables for properly chosen coarse-level variables (cf. Secs. 13.2 above and 14.7 below). Such tools would facilitate and encourage an interactive, scale-by-scale development, by chemists and computational scientists, of computer libraries of ab-initio material descriptions.

14.2 Complexity factors and research strategy

The enormous complexity of molecular calculations is the product of several factors that multiply each other, including:

(1) A very large number of *long-range* (*electrostatic*) *interactions* that need to be summed up to calculate the energy difference associated with each move

of one atom.

(2) *Tiny time steps* (and similarly tiny steps in Monte Carlo simulations and in energy minimization) enforced by the strong chemical bonds.

(3) *Multiscale attraction basins*: The energy functional of the many particle problem includes a multitude of local minima. Moreover, each small-scale basin resides, together with similar basins, inside a larger-scale basin, which itself resides within a still-larger-scale basin, and so on. Conventional algorithms (in search for the global minimum, or in Monte-Carlo simulations at some finite temperature), even when capable of escaping some small-scale basins (e.g., by simulated annealing), remain practically trapped in larger-scale ones.

(4) *Thermal fluctuations*. In equilibrium statistics, to obtain accuracy ϵ in calculating a thermodynamics quantity, one has to average over $O(\epsilon^{-2})$ independent configurations. The computational cost of producing each such independent configuration by a Monte-Carlo process is very large due to the large number of degrees of freedom, *multiplied* by the *product* of the three aforementioned complexity factors.

To investigate multiscale techniques to deal with these obstacles, a systematic study has been undertaken of model problems which include only one or two obstacles at a time. Moreover, unlike the common methodology of starting a research on macromolecular algorithms with small molecules and advancing to increasingly larger ones, the development of multiscale techniques necessarily employs at each stage molecules of an *arbitrary* (large) size n, starting with very simple potential functionals and advancing to increasingly more complicated ones, progressing also from simple geometries (e.g., stretched homogeneous chains, then simple helices) to increasingly more realistic ones. At each stage just one new type of force is added, and the study objective is to still obtain the linear ($O(n)$) complexity. This research strategy is necessary since linear complexity and large-scale processing are indeed our ultimate aims, and since at small molecular systems the advantages of multiscaling cannot be observed.

14.3 Fast summation of electrostatic interactions

Direct summation of all the electrostatic interactions between n particles costs Cn^2 computer operations, where C is around 10. Instead, several methods exist to sum the forces in just $C_1 n$ operations (see, e.g., survey [94]), although note that in three dimensions $C_1 > 10^4$, so these methods become advantageous only for $n > 10^3$. A multiscale method for fast summation, suggested in [29] (based on an idea described earlier in [23, §8.6], [26, App. A] and [53], and related to the methods discussed in Sec. 10 above), is being used. It is based on a decomposition of the two-particle potential into a local part and a smooth part, the latter being evaluated at larger scales (interpolated from coarser grids), where a similar decomposition is being recursively used. An important advantage of this approach is that it gives the kind of multiscale description of the force fields which is needed for the efficient multiscaling of atomic *motions* — in statics, dynamics and equilibrium calculations (see for example the description of the electrostatic calculations in Secs. 14.6.1 and 14.7.4 below), or for solving *equations* (e.g., polarization equations).

Several important new developments by Bilha Sandak, reported in [135], include: (i) Generalization of the method to fields generated by *dipoles*, in addition to those created by charges. (ii) Substantially higher accuracy for negligible additional CPU time. This has been obtained by introducing enhanced interpolation orders and longer softening distances *at the coarser levels*, and by correcting for some false *self-interaction*, i.e., the residual interaction of a charge with itself, caused by the multiscale calculations. (iii) Efficient *software* for general use has been implemented.

14.4 Fast macromolecular energy minimization

Energy minimization may serve here two somewhat different objectives: one in statics, the other in dynamics. In statics, the objective is to calculate the lowest energy $E(r)$, yielding the most stable conformations of the molecular structure. In dynamics, the objective is the solution of the system of equations arising at each time step of *implicit* dynamics simulations. "Implicit" refers to the method which evaluates the forces $-\nabla E(r)$, at each time step, partly or wholly in terms of the particle *arrival* positions, i.e., positions r at the *end* of the step. This method ensures stability of very large time steps, but it does not yield the arrival positions explicitly. Instead, they should be calculated by solving a large system of equations. (Also, this method damps molecular vibrations at scales not resolved by the large time step; we return to this point in Sec. 14.8 below.) Solving the implicit system of equations is equivalent to minimizing an *augmented* energy functional, identical to $E(r)$ except for an additional quadratic *kinetic term* (cf., e.g., [128] and also the functional H in Sec. 14.8 below). For large time steps this additional term is locally very small, but its large-scale effect is still profound.

The macromolecular energy minimization problem is somewhat similar to the minimization problem encountered in structural mechanics, for which very efficient multigrid solvers have been developed. Of these, the closest to the ones needed in molecular mechanics are the *algebraic multigrid* (AMG) solvers (cf. Sec. 17 below), which do not assume that the problem arises from PDE or that the unknowns are really placed on a grid. The methods we have developed for molecular energy minimization follow the general AMG outline: coarser levels are constructed each by taking a suitable subset of the next-finer-level degrees of freedoms; a coarse-to-fine interpolation of displacements is defined based on the fine-level couplings and current configuration; the coarse-level set of equations (or rather, the coarse-level Hamiltonian) is derived based on this interpolation and on the current residual forces at the fine level; and the algorithm consists of relaxation (local minimization) sweeps at all levels with fine-to-coarse transfers of residual forces and coarse-to-fine interpolation of displacements. The molecular forces, however, are much more involved than those of structural mechanics (exhibiting severe nonlinearities and large variations in strength of the different types of interactions), so very systematic development of all these algorithmic components was required.

Our first stage of developing multiscale molecular energy minimizers, in collaboration with Dov Bai, was described in [33]. More advanced techniques for more advanced models are described in [35, §11.4] and [38, §14.4]. However, these studies inevitably led to the conclusion that macromolecular energy minimization is unnecessarily complicated. The energy barriers are much more easily traversed by multiscale methods equipped with the stochasticity introduced by the natural (e.g., room) temperature. Indeed, nearly all practical problems are actually given at finite temperatures (including dynamic problems; cf. Sec. 14.8). Moreover, it is unlikely that at finite temperatures the material stays exactly at the attraction basins of the minimal energy. For these reasons, our interest has shifted to finite temperature calculations, discussed in the following sections. Fortunately, some of the tools acquired in the study of minimization techniques, such as coarsening in terms of a combination of internal and cartesian coordinates, have proved very useful also for the finite-temperature algorithms.

Notice also that finite-temperature algorithms lead themselves to powerful minimization techniques: see Sec. 14.7.3.

14.5 Monte-Carlo methods at equilibrium: General

To calculate equilibrium statistics, an atom-by-atom Monte-Carlo process is usually performed. In this process, each atom in its turn changes position stochastically, according to the probability density distribution $P(r)$. Making repeated sweeps of this process, one can calculate the desired statistics on the sequence of produced configurations.

To calculate accurate averages of some observable, however, an extremely long sequence of configurations is needed. There are two basic causes for this complexity: (1) Due to the local nature of the Monte-Carlo process, only very slowly it affects large-scale conformational features, hence extremely many Monte-Carlo sweeps are needed to produce each new, statistically independent configuration. (2) Many such independent samples are needed to average out the deviation observed at each of them.

For some very simple lattice problems, multigrid Monte-Carlo algorithms were developed which overcome *both* these complexity causes (see Sec. 13 above, where these two causes, which multiply each other, are called the CSD factor and the volume factor, respectively). Two complementary types of multiscale Monte-Carlo methods for the fast simulation of atomistic systems have developed based on the RMG approach (Sec. 13.2): one type for macromolecules, the other for very large, flowing collections of small molecules (gases, liquids). The methods are described in the following two subsections.

In the future the intention is to combine those two types of methods to treat macromolecules in solutions.

14.6 Multiscale Monte Carlo for macromolecules

Monte Carlo simulation of long polymers (and generally all macromolecules) is one of the most computationally intensive tasks. This is due mostly to the large variation in time scales (10^{-15} seconds to several hours) and length scales (1Å–1000Å) involved in each problem and the many energy barriers and attraction basins found at all scales. While much of the interesting behavior occurs at longer time (or length) scales, the shorter scales, where the basic equations are given, constrain the size of steps in simulations. However, by applying multiscale methods these constraints can hopefully be removed as different physical scales are resolved on corresponding appropriate computational levels.

Together with Dov Bai, multiscale MC algorithms for the united-atom polymer model of [127] were studied. The details are reported in [7] and briefly reviewed below. For alternative coarsening schemes, see [8].

The simple polymer is a non-branching long chain of n repeat units (monomers; see the comprehensive survey [82]). In the united-atom model each monomer is considered as a single unit and details of its internal structure and interactions are ignored, so mathematically the polymer is represented as a chain of points in $\mathbb{R}^3$, located at positions $r_1, r_2, \ldots, r_n$. The internal coordinates employed to describe interactions are the distances $r_{i,j} = |r_i - r_j|$, the angles θ_i (angle between the lines $\overline{r_{i-1}r_i}$ and $\overline{r_i r_{i+1}}$) and the dihedral (or torsion) angles $\phi_{i+1/2}$ (angle between the planes $\overline{r_{i-1}r_i r_{i+1}}$ and $\overline{r_i r_{i+1} r_{i+2}}$). The overall Hamiltonian (energy) functional is

$$\begin{aligned}
H(r) = {} & \sum_{i=1}^{n-1} K_r (r_{i,i+1} - r_0)^2 && \text{bond-length potentials} \\
& + \sum_{i=2}^{n-1} K_\theta (\cos\theta_i - \cos\theta_0)^2 && \text{bond-angle potentials} \\
& + \sum_{i=2}^{n-2} F_\phi(\phi_{i+1/2}) && \text{bond-dihedral (torsion) potentials} \\
& + \sum_{|i-j|\geq 4} \epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6} \right] && \text{Lennard-Jones potentials}
\end{aligned}$$

where $r_0 = 1.52$ Å, $K_r = 250$ Kcal/mol/Å^2, $\theta_0 = 110°$, $K_\theta = 60$ Kcal/mol, $\epsilon = 0.09344$ Kcal/mol, $\sigma = 4.5$ Å and F_ϕ is a function featuring three local minima with 1.5 to 2 Kcal/mol energy barriers between them. The relative probability associated with each configuration r is $P(r) = \exp\big(-H(r)/k_B T\big)$, where T is the absolute temperature and k_B is Boltzmann's constant. One can see that the bond-length potentials are much stronger than the bond-angle potentials which in turn are much stronger than the torsion potentials. Therefore the dihedrals are the main active degrees of freedom. The trouble is that a conventional MC simulation is extremely slow in sampling the dihedrals, first because it is constrained by the stronger bond-length and angle potentials and, more important, because each MC switch of the local minimum at which one dihedral lives is only probable provided suitable (but unknown) similar switches are simultaneously performed at a substantial number of neighboring dihedrals.

In our multiscale approach, simulations are mainly performed at coarser levels, which already average over such local attraction basins. Each coarse level consists of a reduced number $N = n/m$ of points, or "atoms"; typically $2 \leq m \leq 4$. Each coarse-level "atom" stands for the average location R_I of m next-finer level atoms. The coarse-level Hamiltonian $H^c(R) = H^c(R_1, \ldots, R_N)$ is developed by extensive, but only local, Monte-Carlo simulations, iteratively fitting coarse-level distribution functions and correlations with those found in fine-level simulations.

The strategy is to calculate local terms of the coarse Hamiltonian by using simulations (to be described below) which involve only a *local set* (typically several dozens) of neighboring points (atoms). Bond interactions between points inside the local set and those outside are ignored: this does not significantly affect the accuracy of the coarse Hamiltonian terms located sufficiently deep inside (several atomic distances from the margins of) the local set. This approach is similar to the one used in Algebraic Multigrid (AMG) and in Renormalization Multigrid (RMG; cf. the *near locality property* in Sec. 13.2 and in Sec. 17). Such a derivation of coarse Hamiltonian terms needs to be done only once for all similarly-structured molecular neighborhoods. Due to the employment of internal coordinates, and with suitable coarsening ratios $1 : m$, it turns out that most inter-coordinate coarse-level correlations can be neglected, yielding quite simple coarse-level Hamiltonians.

The coarse-level Hamiltonian includes, first of all, modified Lennard-Jones-type interactions in terms of cartesian coordinates. The exact formula for this interaction has been derived once-for-all (for a given coarsening ratio), by averaging during MC simulations over all vectorial sums of fine-level Lennard-Jones forces involved in the interaction between two coarse atoms in a given distance bin.

The rest of the coarse Hamiltonian is in terms of local coordinates: distances, angles and dihedral angles (between the *coarse* atoms, of course, which also form a chain). So the general form of the coarse Hamiltonian is

$$H^c(R) = \sum_{|I-J| \geq m_0} H^c_{LJ}(|R_I - R_J|) + \sum_k A_k H_k(R)$$

where $H_{LJ}(\rho)$ is the Lennard-Jones interaction between two coarse atoms at distance ρ from each other, m_0 is a chosen small integer ($2 \leq m_0 \leq 4$), A_k are coefficients to be iteratively determined (as described below), and each H_k is either a *single-internal-coordinate interaction* or a *correlation* between two such coordinates. For each coarse internal coordinate U (either a coarse length $\overline{R_I R_{I+1}}$, or a coarse angle $\overline{R_I R_{I+1} R_{I+2}}$, or a coarse dihedral $\overline{R_I R_{I+1} R_{I+2} R_{I+3}}$) several single-coordinate interaction terms of the form

$$H_k = \begin{cases} 1 & \text{if } \alpha_k \leq U \leq \alpha_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{14.1}$$

are in principle included in H^c; except that one can include in one unified term all the interactions that are expected by symmetry to have the same coefficient

A_k (e.g., all terms of the form (14.1) associated with angles, excluding perhaps angles near the ends of the coarse polymer).

As a first approximation one can start with H^c that includes (besides H_{LJ}) only such single-coordinate terms, with coefficients A_k such that the distribution of each coarse local coordinate U is the one calculated at fine-level simulations (each fine-level configuration implying of course a value for each U). This approximation to H^c will generally fail to give the correct *correlations* between the coarse coordinates. So we next iteratively correct H^c, adding needed correlations terms and readjusting the coefficients $\{A_k\}$ to still yield the correct distributions. The first correlations to be added are those that are shown to be significant in the measurements conducted during the fine-level simulations; the most significant, it turns out, are the correlations between every angle $\overline{R_I R_{I+1} R_{I+2}}$ and each of the adjacent lengths $\overline{R_I R_{I+1}}$, $\overline{R_{I+1} R_{I+2}}$.

In each iteration a large number of MC steps over the local set are made with the coarse-level current Hamiltonian, calculating various observables and comparing them to corresponding values obtained by simulations of the local set at the fine level. The difference is then used to get a better coarse Hamiltonian, in a Newtonian iterative way which converges fast. That is, a set of corrections $\{\delta A_\ell\}$ to the set of coefficients $\{A_\ell\}$ is calculated by solving the linear system

$$\frac{1}{k_B T} \sum_\ell \left(\langle H_k \rangle \langle H_\ell \rangle - \langle H_k H_\ell \rangle \right) \delta A_\ell = \langle H_k \rangle_f - \langle H_k \rangle_c \quad \text{for all } k \,. \tag{14.2}$$

$\langle H_k \rangle_c$ is of course the average of the operator H_k calculated with the current (before correction) *coarse-level* Hamiltonian, while $\langle H_k \rangle_f$ is the corresponding average computed by *fine* level simulations. The averages on the left side of (14.2) can be calculated on either the fine or the coarse levels. (More generally, they can be just approximated, since they only serve as iteration coefficients. In fact, one can ignore most of these terms, only those should be calculated that correspond to neighboring coordinates which might be significantly correlated.) For H_k and H_ℓ that are more strongly correlated, $\langle H_k H_\ell \rangle$ is calculated *both* on the fine and on the coarse levels: In case one finds $\langle H_k H_\ell \rangle_c$ and $\langle H_k H_\ell \rangle_f$ to be too different, *in the next iteration a new correlation term $A_m H_m$ is added to H^c, where $H_m = H_k H_\ell$.*

In a small number of iterations, all the coefficients $\{A_k\}$ converge, with all needed added correlation terms, yielding H^c which very faithfully reproduces all the distribution and correlations exhibited at the fine level.

Numerical experiments have been conducted with two coarsening ratios, 1:3 and 1:4. The resulting coarse Hamiltonians did turn out to fulfill our expectations: With vastly-reduced number of degrees of freedom and allowing much larger simulation steps, it very accurately reproduced large-scale statistics: see [7] for details.

A separate study has shown that the coarsening ratio 1:3 is best, for the following reason. Unlike the 1:4 ratio, it yields fast CMC equilibration (see Sec. 14.7.2), since its coarse configurations fully determine the attraction basin in which each fine-level dihedral resides. This means that the coarsening statistics can be gathered in *windows of long chains*, not only in short local sets. (Cf. the concept of windows at Secs. 13.2 and 14.7.2). This is important since the short local sets may not give rich enough statistics, while long chains cannot be simulated efficiently at the fine level.

14.6.1 Electrostatic interactions

The next development is to test polymer models that also involve electrostatic interactions. To add such long-range interactions to this scheme it is proposed, similar to the approach described above (Sec. 14.3) to decompose each two-body electrostatic potential into the sum of two parts: a smooth potential and a local potential. (See details of such decompositions, for charge and for dipolar interactions, in [135].) The charges or dipoles will then be *anterpolated* to the coarse level. (Anterpolation is the adjoint of interpolation; see, e.g., [11, §3] or [135, §3.2].) This will give good coarse-level approximation to the *smooth* potentials.

Unlike the fast summation schemes based on the same principle, in which charges/dipoles are anterpolated to a *fixed* lattice (as in [29], [135]), here they will be anterpolated to points that *move* during the coarse-level simulations. As a result, the field produced by the coarse-level charges/dipoles will continue to approximate the fine-level *smooth* potentials even under large global movements of the (coarse) molecule.

The *local* parts of the electrostatic potentials remain only to be described at the coarse level. Being local, these interactions can be added to the local scheme described above, similar to the LJ interactions.

With this approach no explicit electrostatic summations are necessary, especially if the local MC simulations are made in a *distributive* manner (cf. [53]). This means, e.g., to move two particles at a time in such a way that their mass center remains unchanged. Such (and *higher-order*) distributive moves hardly affect, or are affected by, the smooth interactions. (Such distributive moves make sense only in a multilevel dynamics, where mass centers that remain fixed during the fine level motions are moved at the coarse level. This exactly is the motions of the coarse "atoms" described above.) Such distributive moves can also serve to reduce the cutoff distance for calculating LJ interactions.

14.7 Multiscale Monte Carlo for fluids

The efficient equilibrium simulation of gases and liquids at the atomic level, needed for the derivation of their large-scale behavior and macroscopic equations, is a central problem in scientific computation. Direct fine-scale MC simulations tend to be extremely inefficient due to the very slow change of various kinds of *clusters* at various scales. Especially difficult are the calculation at critical conditions, where clusters at all scales interact. Of particular interest is the simulation for *water*, by itself or as a solvent.

The main two kinds of clustering difficulties associated with water and other fluids are *positional* clustering and *electrostatic* (dipole) alignment. We will start out by studying in parallel the following two simple cases, in each of which only one kind of clustering is present. The first case will deal with the molecular motions (see Sec. 14.7.1), the second — with the molecular dipole rotations (Sec. 14.7.3). Later, these studies will lead to multiscale simulations of real water models, such as TIP4P [106], [107], in which both molecular translations and rotations, with both Lennard-Jones and dipole interactions, are considered.

14.7.1 Moving particles

Two models of single-atom molecules have been investigated in collaboration with Valery Ilyin [50]: a one-dimensional hard ball model, for which full exact thermodynamic description is known, and a two-dimensional Lennard-Jones fluid. In the latter, describing noble gases, the atoms move solely under their mutual two-atom Lennard-Jones interaction. Clusters of atoms that move together, and clusters of "*holes*" (i.e., absence of atoms inside a larger-scale atomic cluster) are stochastically formed, their likely scales depending on the physical conditions, such as temperature and density. At critical conditions all scales are likely. The larger the scale of clusters, the longer they persist in MC simulations, hence slower is their sampling.

Generally speaking, the multiscale approach here is again of the RMG type (cf. Sec. 13.2), but unlike the former cases (statistical mechanics in Sec. 13.2 and macromolecules in Sec. 14.6), here the coarse levels all *differ* from the finest one in their nature: at the finest level we have atoms at arbitrary locations, while each coarse level is defined on a uniform *lattice*. In simple cases, the value defined at each lattice point stands for the average fluid density in a lattice box around that point. The probability distribution of this density depends on neighboring-point densities, as specified in detail by a suitable *conditional-probability* (*CP*) *table*. The CP table for each level is calculated by extensive *local* simulations at the next finer level (the level of a *lattice* with half the meshsize or, eventually, the finest level of *atoms*).

The development of this multiscale structure along the lines described next (Sec. 14.7.2) is now in progress. It has been shown [50] that density fluctuations at all scales can be accurately calculated with only a bounded number of particles or gridpoints employed at each level.

14.7.2 General outline of the multiscale approach

Generally, the (scalar or vectorial) variable at each lattice point at each coarse level may represent various local averages: of density, or of electrostatic charge, or dipole moment, or energy density, etc. Also, at certain physical conditions, the locations of *moving* blobs, each carrying its own set of properties, may be added to the fixed lattice as additional degrees of freedom. As before (see Sec. 13.2), a general criterion in choosing the coarse level set of variables is the speed of equilibration of compatible Monte Carlo (CMC) runs (each such run being an MC simulation at the fine level, restricted to configurations that are compatible with a given coarse-level configuration); a fast-equilibrating CMC entails the locality property of the coarse variables and thus allows the construction of the CP tables.

The CP table "resolution" (i.e., the number of coarse neighbors on which the probability distribution of a coarse variable is conditioned, and the resolution at which each of these neighbors is tabulated) should in principle increase for, and only for, frequent neighborhoods (see the branching system in Sec. 13.2). The errors in the CP tables can be fully controlled by this resolution and by the amount of statistics gathered at the fine level in setting up the tables. These, together with the interpolation orders used in employing the tables at the coarse-level simulations, determine the accuracy of those simulations.

Because of the near-locality property, no *global* equilibration is needed; local equilibration is enough to provide the correct CP values for any neighborhood for which enough cases have appeared in the simulation. Thus, the fine-level simulation can be done in a relatively small periodicity cell. *The idea is to simulate increasingly larger volumes at increasingly coarser levels.*

However, since the fine-level canonical ensemble simulations use only a *small* periodicity cell, many types of neighborhoods that would be typical at some parts of a *large* volume (e.g., typical at parts with densities different than used in the periodicity cell) will not show up or will be too rare to have sufficiently accurate statistics. Hence, simulations at some coarse level may run into a situation in which the CP table being used has flags indicating that values one wants to extract from it start to have poor accuracy. In such a situation, a temporary *local* return to finer levels should be made, to accumulate more statistics that are relevant for the new local conditions.

To return from a coarse level to the next finer level one needs first to *interpolate*, i.e., to produce the fine level configurations represented by the current coarse level configuration, with correct relative probabilities. The interpolation is performed by CMC sweeps at the fine level; few sweeps are enough, due to the fast CMC equilibration. This fast equilibration also implies that the interpolation can be done just over a restricted *subdomain*, serving as a *window*: In the window *interior* fine-level equilibrium is reached. Additional passes can then be made of *regular* (not compatible) MC, to accumulate in the interior of the window the desired additional CP statistics, while keeping the window boundary frozen (i.e., compatible with the coarse level). The window can then be coarsened and returned to the coarse level, where simulations can now resume with the improved CP table.

Iterating back and forth between increasingly coarser levels and window processing at finer levels whenever missing CP statistics is encountered, one can quickly converge the required CP tables at all levels of the system, with only relatively small computational domains employed at each level. The size of those domains needs only be several times larger than the size of the neighborhoods being used (with a truncation error that than decreases exponentially with that size). However, somewhat larger domains may be better, since they provide sampling of a richer set of neighborhoods (diminishing the need for returning later to accumulate more statistics), and since the total amount of work at each level depends anyway only on the desired amount of statistics, not on the size of the computational domain.

Simulating at all levels in terms of such periodicity cells and windows can effectively eliminate both the volume factor and the slowing down which plague usual (one-level) MC simulations. Provided of course that the *coarsening ratios* (the ratio between a coarse meshsize and the next-finer meshsize), as well as the average number of original particles per mesh volume of the finest lattice, are all suitably low. The typical meshsize ratio is 2, typical number of particles per finest-lattice mesh is between 4 and 10. More aggressive coarsening ratios would require much longer simulations to accumulate accurate CP statistics.

The particle number density at the coarsest level is equal to some input value. Each finer-level window covers only part of the coarsest-level domain, so the particle number density may differ from the initial one. As a result of the multilevel process, the configurations produced at the coarsest level correspond to the canonical ensemble; at finer levels they yield direct accurate representation of the grand canonical ensemble [103].

At sufficiently coarse levels, this entire algorithm effectively produces *macroscopic "equations"* for the simulated system, in the form of numerical CP tables. This can yield a macroscopic numerical description for the fluid even for those (most frequent) cases where the traditional derivation of closed-form differential equations is inapplicable.

14.7.3 Low temperature procedures

The multilevel algorithm can efficiently get into equilibrium even at low temperatures by an *adaptive annealing* process. In this process the temperature is reduced step by step. At each step, upon reducing the temperature from a previous value T to a new one T', a first approximation to the CP tables of T' is obtained from those of T by raising each CP to the power T/T' (and renormalizing; actual renormalization is not necessary since only probability *quotients* are needed in using the CP table for MC simulations). Then, in just few multilevel cycles, the CP tables can easily be made more accurate, provided the quality of the set of coarse variables has not been deteriorated.

However, the type of coarse-level variables appropriate at low temperatures does generally differ from that at high temperatures. At high temperatrues the average density is an adequate coarse-level variable. At low temperatures, e.g., at the appearance of liquid drops in a gas or at the onset of piecewise crystallization, other coarse-level variables should be added, such as the average crystal direction, and/or the average density of holes, and/or the location of mass centers. Thus, in the annealing process one should monitor the quality of coarsening by occasionally checking the CMC equilibration speed. When this speed starts to deteriorate at some level, additional variables should be added at that level, with a corresponding extension of the CP table. Candidate new variables can be found by physical understanding and/or by suitably blocking highly-correlated variables at the next-finer level; then the new variables should be admitted provided they pass the CMC-equilibration-speed test. Some of the old variables may be removable, as judged again by CMC equilibration tests.

In fact, unlike the classical *simulated annealing* method (whose aim is the minimization of the energy, not the simulation of equilibrium), the chief purpose of annealing here is the gradual *identification* of the degrees of freedom that should be employed at increasingly coarser levels. At the zero-temperature limit these procedures can also yield powerful multiscale minimization procedures (see Sec. 18.2).

14.7.4 Rotating dipoles

The chosen model features a large set of electrostatic dipoles, of given strengths and *fixed* locations, rotating in their mutual fields in thermal equilibrium. Clusters of aligned dipoles tend to form, their sizes depending on the given temperature. Again, these clusters are very slow to change in ordinary MC simulations, making large-scale fluctuations extremely slow to average out.

As before (cf. Secs. 14.3 and 14.6.1), at any spatial scale the electrostatic interactions can be decomposed into the sum of a *smooth* part and a *local* part. In addition to using this decomposition for the fast summation of the dipole field, here it will also be used for accelerating the MC simulations and for cheap (coarse-level) averaging over many large-scale fluctuations.

Similar to the above (Secs. 14.7.2–3), each coarse level is defined on a grid, the mesh size being doubled at each coarsening. The vector defined at each lattice point stands for the dipole anterpolated from the next-fine-level dipoles. With this type of coarsening, the RMG methodology is again applied: a CP table at each level is derived by local MC simulations at the next finer level. Then this structure can be employed both for MC acceleration (e.g., by "half-V cycles"; see Sec. 13.2), and for calculation of the large-scale electrostatic properties.

This system is currently under development in collaboration with Bilha Sandak.

14.8 Small-scale statistics with large-scale dynamics

The multiscale structure may allow a natural combination of temperature-accurate statistical simulations at small scales with time-accurate dynamics at large scales. The following approach has been preliminarily studied.

Stochastic implicit time stepping. A first-order implicit discretization to Newtonian dynamics, leading from old positions $r^0 = r(t)$ and old velocities $v^0 = v(t)$ to new positions $r^1 = r(t+\delta t)$ and new velocities $v^1 = v(t+\delta t)$, is given by $v^1 = (r^1 - r^0)/\delta t$ and $M(v^1 - v^0)/\delta t = -\nabla E(r^1)$. This set of equations in r^1 and v^1 is equivalent to the minimization of the functional

$$H(r^1, v^1) = E(r^1) + w^T M w + \frac{1}{4}(v^1 - v^0)^T M (v^1 - v^0) \ ,$$

where $w = (v^1 + v^0)/2 - (r^1 - r^0)/\delta t$. In our *stochastic* dynamics, instead of minimizing H at each time step, we perform a multiscale Monte Carlo simulation with the probability density distribution

$$P(r^1, v^1) \sim e^{-\beta H(r^1, v^1)} ,$$

where $\beta = (k_B T)^{-1}$ and T is the real temperature of the system. The coarse-level moves we have used in the multiscale cycle are based on interpolation (see Secs. 13.1 and 14.4, and [35, §11.4]); an RMG approach to coarsening (cf. Secs. 13.2 and 14.6) may also be considered. At increasingly coarser scales β can be increased, to enforce practically deterministic large-scale dynamics.

This approach yields two benefits in performing very large time steps: first, it allows much easier handling of local minima. Secondly it avoids the killing of highly-oscillatory modes (those vibrations that are not resolved by the time step), which would occur if the implicit equations of a large time step were *imposed* at all scales. Instead, these modes assume stochastic amplitudes, nearly according to their equilibrium probability distribution. The desired temperature is introduced very directly in this way (with the fast atomic vibrations serving as a natural heat bath), thus getting around the need for fabricating Langevin stochastic forces.

Tests with this scheme on model problems with quadratic potential have shown the expected behavior, except that the stochastic treatment at fine levels gradually introduces deviation from deterministic evolution also at large scales. This deviation seems generally to be of the order of the discretization error. We have nevertheless learned how to control this deviation by "distributive Monte Carlo" (similar to distributive relaxation [24]), forcing fine-scale moves to be as nearly orthogonal to large-scale moves as desired.

15 Image Processing. Clustering. Graphs

15.1 Edge (or fiber) detection

Straight features. Fast multiscale approaches for some early vision tasks, such as edge detection and surface reconstruction from sparse, noisy or blurred data, have been developed in collaboration with Jonathan Dym [79]. In particular, fast multiscale methods for enhancing and detecting straight features (straight edges or straight fibers) have been demonstrated [43], [44]: They detect *all* such features, of all widths, lengths, positions and orientations, in just $O(N \log N)$ operations, where N is the number of pixels (picture elements) in the given picture. This has been achieved by constructing a hierarchical collection of numerical integrals of grey levels along straight segments of the pictures, with the lengths, positions and orientations of the segments in the collection chosen in such a way that:

(1) The collection is *rich enough*, in the sense that any other straight-segment integral over the picture can be readily interpolated from the collections's integrals; each interpolation is over a short distance, so that it is equivalent to interpolating grey levels only between nearest-neighbor pixels. Specifically, the collection includes segments of length $1, 2, 4, 8, \ldots$ (in pixel units), those of length ℓ

have locational resolution which is $O(\ell)$ in the direction of the segment and $O(1)$ in the perpendicular (width-wise) direction, while their orientational resolution is $O(\ell^{-1})$ (analogously to the *Heisenberg principle* in quantum mechanics).

(2) The collection is *fast to construct*, by using shorter segment integrals to calculate the next-longer ones, scale after scale, so that the construction of each integral requires only $O(1)$ operations.

Curved featuers. For detecting smooth curved features (edges or fibers), a variety of approaches have been proposed. One good example is the *completion fields*. In this approach, the picture is described in terms of "edgels" (edge elements), i.e., short pieces of a straight edge (or fiber), defined at $N_1 = O(N)$ locations in the picture, at m different orientations in each location. The *original* value of edgel i is the response u_i to an elementary edge detector at i; that is, u is the result of a local integral transform which yields a higher value if the local picture elements do indicate existence of an edge at that particular location and orientation (and at the chosen scales of length and width, typically being, respectively, 3 and 1 times the pixel size). The *completion field* value v_j of edge j can be built from the set of all elementary responses u_i in a variety of ways (see different approaches in [161] and [98]). As a representative example for our discussion here, we can take

$$v_j = \sum_{i=1}^{N_1 m} a_{ij} u_i \ , \qquad (j = 1, \dots, N_1 m) \ , \tag{15.1}$$

where a_{ij} expresses the "affinity" of edgels i and j: it is large if edgel j is a direct continuation of edgel i, and it falls off with their distance and orientation difference. For a given i, its "induction field" a_{ij} is qualitatively similar to the field of a magnetic dipole. It is shown in [161] that such completion fields are biologically plausible, and give eye-pleasing curves. They are particularly powerful in completing curves partly occluded by large objects. The original method however has several severe shortcomings, which can be overcome by multiscaling.

Multiscale methods can contribute to the process in two fundamental ways. First, the method as described in [161] would require $O(N_1^2 m^2)$ computer operations; multiscale methods, resembling those of Sec. 10 above, would do the same job in $O(N_1 m)$ operations, while retaining the same (very high) degree of computational parallelism.

Second, and more importantly, still with this low cost, the multiscale processing can produce much better completion fields.

Indeed, a fundamental flaw in the uniscale completion fields is their additivity, as in (15.1). In reality, the completion field of a long edge should be very *different* from (farther reaching and more orientation-specific than) the sum of the fields of several short edgels that compose it. In the multiscale approach, this flaw can be avoided, since completion fields can be constructed *separately* at each scale of length and width, with *scale-dependent* affinity parameters. The multi-resolution *input* of straight-edge responses required for such multiscale completion fields is exactly the kind resulting from the $O(N \log N)$ straight-feature algorithm mentioned above.

The multi-resolution of both the input (straight responses) and the output (completion fields) also involves further cost reductions. For example, as mentioned above, for *short* edgels only low orientational resolution need be used, while for long edgels a low locational resolution is needed (in the lengthwise direction). Thus, the value of $N_1 m$ mentioned above can itself be radically reduced. Moreover, the multiscale *output* of the algorithm is a very desirable structure to interact with the higher vision processes of labeling and segmentation (cf. Sec. 15.2), whether or not the latter are themselves multiscaled.

A detailed study of multiscale completion fields, their parameterization and fast implementation has been conducted with Eitan Sharon and Ronen Basri. It is summarized in [139].

Intriguing possibilities of combining the developed algorithms in a variety of ways should be investigated:

1. Iterating a multiscale algorithm, with the output of the first iteration (e.g., the set of v_j) being used in forming the input (e.g., the set of u_i) for the next iteration. This can be done in various manners: linear, nonlinear, with or without thresholding.

2. Using the output from one scale in forming the input for the next coarser scale.

3. Thresholding after the previous iteration, one can use in the next iteration several different and more complex algorithms, due to the smaller set of data. In particular, one can afford at this stage specialized algorithms, such as circle and corner detection. With further iterations, increasingly higher levels of recognition algorithms may enter.

4. Combining in various ways edge detection with picture segmentation (see next).

15.2 Picture segmentation

A basic task in pattern recognition is the decomposition of a given picture into meaningful segments. The criteria for blocking two picture elements into the same segment include similarity in color levels, absence of separating edges, etc. Quantitatively, these can be expressed in terms of *coupling* coefficients between neighboring pixels. It is not uniquely defined how to derive the segments once the coupling coefficients are given. Multiscale approaches can play *several* essential roles (somewhat analogous to their variety of roles in other areas; see for example Sec. 15.1 above).

Regarding the pixels as nodes of an electric network, and each coupling constant as the conductance (reciprocal of resistance) of a connecting wire, the approach to the segmentation problem is to define a picture segment as a block of nodes that will have approximately the same electric potential under whatever input currents applied to the network. The first possible role for a multiscale approach is in terms of a fast solver for such networks. Since the network is highly disordered, *algebraic multigrid* (AMG) solvers best fit the task (see Sec. 17).

As pointed out by Sorin Solomon, there is in fact no need to solve the electric-network problem for any particular input currents: Some of the coarse-level nodes defined by the AMG coarsening process can directly be identified with the desired picture segments. More precisely, if all the couplings of a node at any coarse level are weak (compared with its own couplings to finer-level nodes), the node can be recognized as a picture segment, containing all the pixels (finest-level nodes) which are coupled to it (through the AMG recursive coarse-to-fine interpolations).

The AMG hierarchical coarsening can indeed be viewed as a process of *iterated weighted aggregation*. In an iterated aggregation process, the elements (pixels) are blocked in small-scale aggregates, which are then blocked in larger-scale aggregates, then still larger, etc. In the *weighted* aggregation process, *fractions* of the same element can be sent into different small-scale aggregates, and similarly at all larger scales. This weighting is important in order to express the *likelihood* of elements to belong together; these likelihoods will then accumulate at the higher levels of the process, automatically reinforcing each other where appropriate.

Only *after* larger-scale aggregates have been formed, the *boundaries* for smaller-scale aggregates can be delineated more sharply, taking into account the larger-scale picture. Hence, into the bottom-up process of weighted aggregation, *up-bottom* procedures are added which at appropriate stages tighten and soften the couplings between pixels or between some fine-level aggregates, based on higher aggregation levels. (More specifically, the new couplings are based on values of local fine-level solutions, each obtained by an AMG-type coarse-to-fine interpolation of a coarser-level delta function followed by intermediate-level and fine-level local relaxation sweeps. The couplings to be strengthened are the couplings between those pixels that get values close to 1.) These up-bottom procedures serve to focus the created aggregates and sharpen the boundaries of the emerging segments.

This integrated multiscale process offers much more than simple segmentation. They in fact yield a *hierarchical segmentation*, where segments within segments can be recognized. They can also yield *scaled segmentation*, where the scale of the picture at which segmentation is desired can be specified.

More important, the multiscale weighted aggregation is free to apply new types of couplings at different levels. The coupling between larger-scale blocks (blocks which have been created by the smaller-scale aggregation, or alternatively, simple *geometric* blocks of $k \times k$ pixels), instead of (or in combination with) being induced by the fine-scale couplings (as in the AMG process), they can employ new criteria. Such criteria can include for example similarity in the *average* color levels of the blocks. More generally, all kinds of other intra-block "*observables*" can be used: the block's center of mass, its diameter, principal orientation, texture measures (being, e.g., statistics on average sizes and directions of smaller-scale sub-blocks), etc., with the number of observables per block increasing at coarser levels. For example, strong couplings can be assigned between two (not necessarily neighboring) aggregates whose principal orientations align with the direction of the line connecting their centers of mass. More important, strong couplings should be established between two neighboring aggregates whose boundaries (sharpened by the up-bottom procedures) seem to *complete* each other (using criteria akin to those used to form a_{ij} in (15.1)). These kinds of couplings may establish affinities even between quite *distant* aggregates, promoting the appearance of *disconnected* segments, presumably signifying partly occluded objects.

Another criterion for blocking at each level can be the absence of separating edges on the scale of that level. This will directly benefit from the *multiscale* edge-detection algorithms, as described above. Alternatively, it may be desired to detect the large-scale edges *from* the large-scale blocks by applying a suitable edge detector at that level (a suitable integral transform on a chosen block quantity, such as its average gray level or any other observable).

The multilevel aggregation and hierarchical segmentation algorithms are very fast. On coarse levels the number of variables is drastically reduced, so most of the work is at the initial, finest levels. On those levels the geometric ordering of the pixels and of the small aggregates still dominates and can be used to create very inexpensive processing, so that *the entire algorithm costs only several dozen computer operations per pixel*.

A detailed account of our current multiscale algorithm for image segmentation is given in [140], demonstrating its properties on several line drawings and real images.

The future strategy is to enhance the algorithm in several ways indicated above: adding more coarse-level observables, introducing various interactions with our multiscale edge detection processes, etc. Multiscale approaches to other aspects of image processing are also envisioned.

15.3 Clustering and graph algorithms

The problem of picture segmentation is a special case of the following *clustering problem*: Given a set of objects $\{1, 2, \ldots, n\}$ and "affinities" (or "couplings") $a_{ij} = a_{ji} \geq 0$ between the objects $(i, j = 1, \ldots, n;\ i \neq j)$, find "clusters", i.e., disjoint subsets of objects such that objects within the same subset are "strongly coupled", directly or indirectly, while objects in different subsets are generally weakly coupled. The strength of direct coupling between i and j may be defined by the size of $a_{ij}/(a_i a_j)^{1/2}$, where $a_i = \max_k a_{ik}$; indirect strong coupling between i and j is formed by a short chain $(i = i_0, i_1, i_2, \ldots, i_k = j)$ such that $i_{\ell-1}$ is strongly coupled to i_ℓ, $(\ell = 1, 2, \ldots, k)$. So defined, the clustering problem is of course fuzzy; it can be defined more precisely in various ways, although a direct precise definition in terms of the minimization of some functional can be coutnerproductive (cf. Sec. 18.1). In fact, as with the above special case of picture segmentation, the best definition can often only be done through the *multiscale clustering* process, where larger-scale affinities are defined or modified *at coarser levels*, depending on properties of intermediate aggregates.

In collaboration with Eitan Sharon and Ronen Basri, the segmentation algorithm described in Sec. 15.2 has been extended to a general clustering algorithm. The main feature distinguishing picture segmentation was its two-dimensional *locality*: affinities were only defined between *neighboring* pixels. In general clustering problems, *all* the affinities a_{ij} may be positive. To account for this situation, a general efficient way has been developed to involve increasing number of affinities at increasingly coarser levels of the algorithm.

Clustering algorithms are central to many areas of applications, including *bioinformatics* and *data mining*. In all these areas the multilevel clustering, and in particular the multilevel *definition* of affinities, have enormous potential, which we plan to demonstrate.

The clustering problem is a special case of *fuzzy graph problems*. Many other problems in this class can greatly benefit from multilevel algorithms, including such well-known problems as the *traveling salesman* (aiming at producing a close-to-optimal, not *the* optimal, route), the *transportation problem* (see [110] for an early multiscale approach), vertex ordering (see, e.g., in [4]), two-dimensional embedding or the problem of *drawing graphs nicely* (successfully multiscaled in [99]), *min-cut* or *max-flow* [72], *sparse spanners* [93], *dense subgraphs* [81], and others.

The general approach in these graph problems is that of *coarsening*: recursive transition to increasingly coarser graphs, each having only a fraction of the number of nodes and edges at the next finer graph. For each coarser graph a new problem is formed such that its solution would easily lead to an approximate solution at the next finer level. With various variations, the AMG coarsening (see Sec. 15.2, or the more general approach in Sec. 17.2) is the basic vehicle. Research along such lines has been initiated, in collaboration with Eitan Sharon and Evgeniy Bart. The emphasis is on *practical* algorithms, obtaining very good *approximate* solutions in very low *average* complexity (unlike the emphasis in theoretical comptuer science on exact solutions with low *worst-case* complexity). Linear average complexity should typically be expected from such multiscale algorithms.

16 Tomography: Medical Imaging

To develop multiscale computational methods for tomography, we have started by working on the two mathematically extreme cases: X-ray tomography, requiring the inversion of the sharp radon transform, and impedance tomography, requiring inversion of a very diffusive process.

16.1 Inverting the Radon transform and related problems

Reconstruction of a function of two or three variables from its Radon transform has proven vital in X-ray computed tomography (CT), emission computed tomography, nuclear magnetic resonance (NMR) imaging, astronomy, geophysics, and a number of other fields [76]. One of the best known reconstruction algorithms is the convolution backprojection method (CB), which is widely used in commercial medical CT devices [76] (with "rebinning" for divergent-beam projection [102]). It has also been applied to spotlight-mode synthetic aperture radar (SPSAR) image reconstruction [102]. While CB provides good reconstruction relatively efficiently, it is still too slow for some purposes, requiring large computational resources and limiting the ability of CT machines to produce real-time 3-D images or video. A faster technique sometimes used, based on direct Fourier method, yields images of much poorer quality.

For other medical imaging and radar problems which are non-uniform, existing Fourier-dependent methods (e.g., CB) are less applicable, resulting in worse performance. This includes the Positron Emission Tomography (PET), the Single Photon Emission Computed Tomography (SPECT), impedance tomography, ultrasound and similar medical imaging techniques (see, e.g., [75]), as well as non-uniform problems in CT, such as the limited-angle problem and the 3D cone-beam reconstruction.

A new multi-level approach to the inverse Radon transform (X-ray tomography) was developed by us several years ago. While the backprojection of the conventional CB raises the computational complexity of the method to $O(N^3)$ for an $N \times N$ images, we have developed a novel $O(N^2 \log N)$ multilevel backprojection algorithm and an accompanying, even less expensive, post-processing procedure [54], [55]. Tests for a number of phantoms, and measurements of point-spread functions, show that the combined method produces at least as good images as those produced by classical CB, in far less time. Further improvements, including an adjustment of the post-processing part to concrete CT machines and a stochastic device to obtain translation invariance in the multilevel backprojection, were introduced by Meirav Galun.

Fast algorithms for other fields with line-integral transforms are under development by Galun. Direct and inverse computation of line integrals of the two-dimensional SPECT is being done by multiscale computation in $O(N^2 \log N)$. (The direct algorithm is an extension of the line integral computation for the direct Radon transform [43].) Also being developed is a solver for the *limited-angle* problem, the case where the X-ray tomograph scans only part of the full 180-degree view. Based on the methods described in Sec. 10.1 above, we have developed a new type of multiscale transform which is applied in the process together with the backprojection, replacing the naive convolution which is less suitable in this case. The construction of the multiscale transform is done once for all, in off-line iterations. *Three dimensional Cone-Beam* reconstruction, used by the new generation of CT machines, can be achieved by fast multiscale solver in $O(N^3 \log N)$ complexity, using a similar off-line construction of a suitable multiscale transform together with a 3D version of our backprojection algorithm.

A more complicated problem is the *three dimensional* PET reconstruction. In this case, a typical situation is that the number of emission events is much smaller than the number of possible rays. Multiscale processes can be applied here to efficiently perform *three* types of tasks: the gathering and averaging of the event data; the backprojection; and, again, a multiscale transform to replace the convolution, constructed off-line.

16.2 Impedance tomography: inverse diffusion problem

An EIT (Electrical Impedance Tomography) device for medical use consists of a set of N electrodes attached to the chest of a patient. A small *known* current is passed between two driver electrodes. In each measurement the current is passed through a different electrode pair, while the voltage drops at *all* the electrodes is recorded. The collected data are used in order to calculate the conductivity distribution in a part of the patient's chest and then to display it on a screen, in order to detect anomalies, such as tumors.

The electrical potential satisfies the equation $\nabla(\sigma\nabla u) = 0$, where σ is the electrical conductivity. The set of measurements gives ideally (in the limit of many small electrodes and as many measurements) the *Neumann to Dirichlet* mapping: the Dirichlet (u) boundary condition resulting from any Neumann ($\partial u/\partial n$) condition. The inverse EIT problem is to calculate σ from this mapping.

The first description of the inverse EIT problem was given by Calderon [69]. Kohn and Vogelius [115] showed that under certain assumptions the conductivity of a medium is uniquely determined by the Neumann-to-Dirichlet mapping. Then Sylvester and Uhlmann [149] provided a general framework for proving uniqueness of the solution of the inverse problem. Alessandrini [1] gave a mathematical explanation for the blurriness of conductivity images and proved that the conductivity depends on the EIT data in a very weak way. Therefore the inverse problem of EIT is ill-posed, and a regularization is necessary if conductivity is to be obtained stably from data.

There exist some works on numerical methods for the relevant problems, but their number is rather sparse and even those papers do not consider the question of numerical efficiency, despite its importance for applications.

The main purpose of our work on this problem, together with Rima Gandlin, has been to demonstrate two general methodological points. First, an ill-posed problem is not necessarily difficult or expensive to solve. On the contrary: once the nature of the ill-posedness has been generally understood, the solution may even be much less expensive than solving the direct problem. For example, in the inverse EIT problem, employing local Fourier decompositions one can show that all components of wavelength λ are ill-posed at distances $r \gg \lambda$ from the boundary. Hence there is no need to use at such distances fine solution grids: all we can know about the solution can be calculated with grids whose meshsizes increase proportionality to r. Moreover, one can recombine the different measurements into N new ones, such that in the k-th measurement the electric current enters the k-th electrode and leaves *uniformly* through all other electrodes. Then it can be shown that in solving the k-th (direct) problem one needs a fine grid only near the k-th electrode, with increasingly coarser grids away from it.

The second general point is that such a careful choice of grids, in a suitable multigrid algorithm, can *replace the need for explicit regularization of the problem.*

Our first pass at the problem, employed the well known Tikhonov regularization method, reformulating the inverse problem as a variational minimization problem. The resulting Euler equations form a PDE system (for u, σ and a Lagrange-multiplier function), which make the problem suitable in principle for an effective numerical solution by multigrid methods. The FMG solvers were designed with large and then with progressively smaller regularization. Special attention has been paid to properly adapting many features of classical multigrid to the case of the problem under consideration (including intergrid communications, boundary condition treatment and coarse grid solution). In the case of large regularization, numerical experiments have demonstrated a good convergence of the developed solver, but the obtained solution is too smeared and doesn't approximate the real conductivity function too well.

At *small regularization values* the final approximation is much better, especially near the boundary. In this case, however, the system is no longer elliptic, and much more sophisticated relaxation methods are necessary, featuring a DGS scheme [24, §3.7], which effectively decomposes the system into its scalar factors. With this approach, although the multigrid cycles asymptotically slow down, the final approximation *to the conductivity* is practically obtained by just *one* multigrid cycle per grid refinement, even when approaching the smallest regularization for which solution still exists.

It took some effort [141], [88] to complete this part of the program. Then, the solution method *without regularization* was developed. Preliminary results [89] show it to give better approximations to σ than the regularized method with its many artificial parameters (the regularization coefficients, which should change over the domain), for less work (no Lagrange multipliers). However, it also turns out that without regularization the solver requires a control parameter p to be gradually decreased through the FMG algorithm, allowing in each stage only those σ changes whose "profit" (in terms of improving the approximations to the Dirichlet data) per unit change is at least p.

17 Algebraic Multigrid (AMG): New Approaches

Algebraic multigrid (AMG) algorithms are solvers of linear systems of equations which are based on multigrid principles but do not explicitly use the geometry of grids; see [23, §13.1], [56], [58], [25], [134], [146]. The emphasis in AMG is on automatic procedures for *coarsening* the set of equations, relying exclusively on its algebraic relations. AMG is widely employed for solving discretized partial differential equations (PDEs) on unstructured grids, or even on structured grids when the coarse grid can no longer be structured, or when the PDE has highly disordered coefficients. AMG can also be used (as in [56]) for many types of discrete systems not arising from differential equations.

Given any system of linear equations $Ax = b$, where A is any (possibly rectangular) matrix, the starting point for all multilevel (multigrid) fast solvers is the following insight. (For convenience we assume, without actually losing generality, that the matrix A is roughly *normalized*, i.e., the ℓ_2 norm of every row in A is roughly 1.)

For any approximate solution $\widetilde{x}$, denote by $e = x - \widetilde{x}$ the error vector, by $r = Ae = b - A\widetilde{x}$ the vector of residuals, and by $\| \cdot \|$ the ℓ_2 norm. The common feature of all local relaxation schemes is that at each step some corrections to $\widetilde{x}$ are calculated based on the values of a small number of residuals. As a result, convergence must be slow when the individual residuals do not show the true magnitude of the error, i.e., when $\| r \| \ll \| e \|$. The converse is also true (and proved in [25]): If the convergence of a *suitable* (e.g., Kacmarz) relaxation scheme is slow, then $\| r \| \ll \| e \|$ must hold. Since, for a normalized matrix A, the deeper the condition $\| Ae \| \ll \| e \|$ is satisified the more special must be the type of the error $\| e \|$, *a suitable relaxation can always efficiently reduce the information content of the error, and quickly make it approximable by far fewer variables*. (This is true even for general *nonlinear* systems.)

Thus, following a small number of relaxation sweeps, the remaining error can be approximated by a "coarser" (or "diluted") system, i.e., a system with only a much smaller number of variables (at most half the original number, for example). General approaches for first defining the *set* of coarse variables and then for deriving the *equations* they should satisfy are briefly described below. The coarse equations themselves are then (approximately) solved by a similar procedure: a small number of relaxation sweeps followed by approximating the remaining error with a still coarser system. This recursively defines the multilevel cycle, which, for a work comparable to that of just few relaxation sweeps over the finest level (the given system), can reduce the error to a small fraction (far less than .5, typically) of its pre-cycle size.

The set C of coarse variables is chosen as a *subset* of the set of fine (original) variables; or, more generally, each coarse variable is chosen to be a linear combination of a small number of fine variables (or fine *ghost* variables — a generalization explained in Sec. 17.2 below, and also in [37, App. A]). In classical AMG [56], [58], [25], [134], the set C is chosen so that each fine variable is "strongly coupled" to C. More generally, a criterion for gauging, and a practical method to control, the quality of this set can be based on sweeps of *compatible relaxation*. This is a modified fine-level relaxation scheme that keeps the coarse-level variables invariant (i.e., it keeps the fine-level configuration always compatible with the same coarse-level configuration). *The set C is guaranteed to be good when (and only to the extent that) the compatible relaxation exhibits uniformly fast convergence rates.* Where these rates are too slow, they point to variables part of which should be added to C (or, alternatively, they point to variables that should be relaxed *simultaneously*; see [37]). (An analogous criterion for coarsening *statistical* fields, involving fast equilibration of *compatible Monte Carlo*, is described in Secs. 13.2 and 14.7 above.)

The derivation of the coarse-level equations is described below for systems of *local* equations, i.e., systems $Ax = b$ whose variables $(x_1, x_2, \ldots)$ each has a location in a low-dimensional space, and whose equations each involves only few variables in a local neighborhood of that space. Generalizations exist to "sparsely positive definite" matrices, including positive-type matrices (see [25]), to "asymptotically smooth" and "asymptotically smoothly oscillatory" matrices, including electrostatic or gravimetric interactions (see [37, §11] and Secs. 13.2.2 and 14.6.1 above), and to some other types of systems. Also, the same procedures often work well for cases not belonging to any of these types.

The fast convergence of the compatible relaxation implies that the values of the coarse set of variables indeed determine, up to fast local processing, the values of the fine set. Moreover, it implies that the chosen coarse set satisfies the "*near locality*" property, i.e., the fine level solution at each point can be calculated *locally*, given just its coarse *neighborhood*, with very weak remnant dependence on coarse values outside that neighborhood: the remnant dependence decays *exponentially* (or even faster) as a function of the neighborhood radius. (Cf. the "near locality" for statistical problems, in Secs. 13.2 and 14.7 above.) For 2D discrete Poisson equations, for example, the remnant dependence tends (after enough coarsening levels) exactly to $\exp(-\pi r^2/2)$, where r is the neighborhood radius measured in meshsizes of the coarse level [137], [167].

Since each coarse variable is defined locally by few fine variables, it too depends only *nearly-locally* on all other coarse variables. Hence, an *equation* for each coarse variable in terms of other coarse variables can be derived locally, using only a *local set* of fine-level equations. The error in that coarse equation will decrease exponentially as a function of the size of that local set.

We describe below two approaches for deriving the coarse equations: In Sec. 17.1 the highly accurate derivations of [37] are mentioned, and examples for the use of very accurate coarsening are listed. A new, much more efficient general approach is detailed in Sec. 17.2.

17.1 Highly accurate coarsening

Several general methods for local derivation of highly accurate coarse equations are described and demonstrated in [37], including a method developed by Irad Yavneh. One approach is based on the tranditional *Galerkin coarsening*: the coarse-grid equation approximating $Ax = b$ is $A^c x^c = (I_c)^T b$, where $A^c = (I_c)^T A I_c$ and I_c is an accurate coarse-to-fine interpolation derived by solving a local optimization problem. (For highly non-symmetric A, see the more general form in Sec. 17.1 below.) Another approach, called *direct coarsening*, directly derives coarse equations by solving another local optimization problem. (In statistical physics, the Galerkin coarsening corresponds to the interpolation-based method (Sec. 13.1 above), while the direct coarsening is analogous to the RMG method (Sec. 13.2).)

In both these approaches one can control the coarsening *accuracy*, and the corresponding amount of computational work per coarse equation, by choosing the size of certain stencils. Although the work per equation is always in principle only $O(1)$ (i.e., it depends on the desired accuracy but not on the size of the matrix A), the actual constant can be very large, rising as some power of the size of the local set.

For the purpose of multi-level (multigrid) *cycles*, a low coarsening accuracy would usually suffice. For example, a coarse grid equation with at most 10% error for all "smooth" components (i.e., those slow to converge in relaxation) can yield a multilevel cycle with a convergence factor close to 0.1. By performing successively any number of such cycles, any desired solution accuracy can rapidly be obtained. This will usually be far more cost effective than deriving higher accuracy coarsening.

Such low coarsening accuracy can often be inexpensively obtained by the *classical AMG approach*, i.e., using the Galerkin coarsening, with x^c being a subset of x and the interpolation I_c having weights proportional to the size of the corresponding terms in A (or in A^2, or in A^TA). This approach is particularly effective for simple matrices, such as *positive-type* ones (matrix $A = \{a_{ij}\}$ such that $a_{ij} \le 0$ for all $i \ne j$ and $\sum_j a_{ij} \ge 0$ for all i).

In many other cases, however, higher degrees of coarsening accuracy, obtainable by the techniques of [37], or those of Sec. 17.2 below, are really needed. Usually in such cases, the system involves a high degree of repetetiveness, so the high cost of deriving very accurate coarsening can be afforded. Examples:

(i) Once-for-all coarsening, for the purpose of deriving the *macroscopic equations* of the given system, or *homogenizing* it.

(ii) Cases in which one needs to solve many linear systems of the form $Ax = b$, where large parts of A and b do not change from one system to the next, so re-computing those parts at fine levels can be avoided by having accurately coarsened them before. One important such case is the calculation of many desired terms (e.g., the main diagonal) of A^{-1}; this requires solving many times the system $Ax = b$, each time b being another unit vector. Moreover, in important cases (e.g., see Sec. 12 above), those desired terms of A^{-1} must be recalculated upon each change in A occuring during certain Monte-Carlo simulations.

(iii) Problems with a large number of almost-zero modes (AZMs), i.e., eigenvectors with unusually close to zero eigenvalues. Such modes often reflect some ill defined global moves, such as rigid-body motions of the entire system in problems of elasticity, or a gliding motion of two rigid bodies along their contact surface. Such AZMs also plague various disordered problems, such as Dirac equations on critical gauge fields (cf. Sec. 11). For problems with *many* AZMs, a general cure is to increase the coarsening accuracy. A small number m of AZMs (such as those associated with global rigid body motions) may still persist even at higher accuracies, but they can be eliminated by recombining $m + 1$ iterants (each being, for example, the approximate solution obtained after another multi-level cycle) so as to minimize the ℓ_2 residual norm; see, e.g., [59].

(iv) A computing environment which makes it preferable to use as few multigrid cycles as possible, such as massive parallel processing with poor interprocessor communications, or a computer with a very high-speed cache memory.

17.2 Bootstrap AMG (BAMG)

The methods described above offer highly accurate coarse equations. They are however very expensive, being practical only for highly repetetive systems. The main flaw in these and other AMG methods is the completely *local* derivation of the equations. This cannot yield *efficient* approximation to the lowest eigen modes. More generally practical coarsening methods, developed in collaboration with Irad Yavneh, are described next. They develop the AMG structure *iteratively*, using the evolving AMG solver itself to improve its interpolation rules.

The proposed coarse equations depend on the properties of the matrix A, which can be described in terms of the relaxation scheme that goes with it. A general relaxation scheme for the system $Ax = b$ is *weighted distributed Gauss-Seidel* (WDGS), which is a Gauss-Seidel (or SOR) relaxation of the system $A'x' = b'$, where $A' = PAM$, $x = Mx'$ and $b' = Pb$. The "weighting matrix" P and the "distribution matrix" M are chosen in various ways. If for example A is symmetric and semi-definite, or even non-symmetric but with enough diagonal dominance, then one can choose $A' = A$ and $P = M = I$, the identity matrix. If the system is a discretization of a PDE system, P and M are usually determined at the differential level, or at the level of the first differential approximation to the discrete operator, based on the operator matrix (cf. Sec. 2.2 above; see [36]). If nothing better is available, one usually chooses either $P = I$, $M = A^T$ (Kacmarz relaxation) or $P = A^T$, $M = I$ (least-square relaxation). This indeed guarantees convergence of the WDGS relaxation, but that may be insufficient. What is needed is to have good "smoothing" in the generalized sense, that each value in a relaxed vector essentially depends only on its neighboring values (except possibly for a deviation that decays exponentially with the size of the neighborhood), where the neighborhood is defined either geometrically or algebraically (in terms of strongest couplings). Such a smoothing condition would not generally happen for discretization of integral or integro-differential equations. To obtain good smoothing one can then multiply either P or M by a matrix that corresponds to *differencing* (i.e., taking differences of neighboring values), raised to sufficiently high power.

The variable x' are called "*ghost variables*" because they need not be known: the relaxation calculates changes $\delta x'$ for these variables, but those can be directly expressed as changes $\delta x = M(\delta x')$ introduced to the explicit variables x.

For simplicity we will assume below that the given system $Ax = b$ can be relaxed by Gauss-Seidel ($P = M = I$); otherwise, A, x, and b in the discussion below can be replaced by A', x' and b', respectively.

A comment is passing: in rare situations the matrix A' may have few eigenvalues with magnitudes much larger than all other eigenvalues. In this situation (and *only* in this situation) the relaxation process should use iterant recombinations (e.g., conjugate gradient or GMRES) to reduce the corresponding outlying error components, so that the relaxation parameters can be fitted to treat efficiently the majority of eigenmodes.

The coarse-level equations $A^c x^c = b^c$ proposed here are of the Galerkin type: $A^c = I^c A I_c$ and $b^c = I^c b$. Here I_c is the coarse-to-fine interpolation; i.e., if the equation $Ax = b$ is already relaxed, a good approximation to its solution is expected to satisfy $x \approx I_c x^c$. The issue treated below is how to construct I_c and the fine-to-coarse transfer I^c. They will generally be constructed in several iterations.

It can easily be shown that I_c should interpolate the low eigenvectors of A well; i.e., a vector $\widetilde{x}$ which is a combination of low-eigenvalue eigenvectors should have a vector $\widetilde{x}^c$ such that $\| \widetilde{x} - I_c \widetilde{x}^c \| \ll \| \widetilde{x} \|$. It can also be shown that $(I^c)^T$ should well interpolate low eigenvectors of A^T. We describe here the derivation of the interpolation I_c; if A (or actually A') is symmetric or nearly symmetric, $I^c = (I_c)^T$ can be used; otherwise I^c will be derived by a similar procedure, applied to A^T.

A general form of the interpolation I_c is

$$(I_c x^c)_i = \sum_{j=1}^{n_i} w_{ij} x^c_{I_{i,j}} \quad . \tag{17.1}$$

The sequence $\{I_{i,j}\}_{j=1}^{n_i}$ is the ordered set of the n_i indices of coarse-level variables from which interpolation to the i-th fine-level variable is made. They are chosen in the "neighborhood" of x_i, defined either geometrically or in terms of algebraic couplings. A necessary lower bound for their number n_i is often known in advance. For example, in solving discretized PDEs, the orders m_c and m^c of I_c and I^c, respectively, should satisfy well-known rules (see [23] or [24] or [32] or [153]), so for d-dimensional problems $n_i \geq m_c + d$. Generally, one should start with a small reasonable value for each n_i, since the iterative procedure described below will indicate when n_i should increase, or when the set $\{I_{i,j}\}_{j=1}^{n_i}$ needs to be modified.

As the example of PDE systems show, to keep all the n_i's small, it is beneficial, when possible, to divide the set of variables into disjoint "*species*", both on the fine and on the coarse levels, such that the coarse variables of each species are defined in terms of the fine-level variables of the same species, and the interpolation too is defined within each species. For example, in discretized PDE systems each species corresponds to the discretization of one function.

First approximation. Let $n_c = \max(n_i)$. A first approximation to I_c will be derived from $\overline{n}_c$ relaxed solutions $x^{(k)}$, $(k = 1, \ldots, \overline{n}_c)$, where typically $n_c < \overline{n}_c < 2n_c$. Namely, each $x^{(k)}$ is a result of several fine-level relaxation sweeps on the homogeneous equation $Ax = 0$, each starting from another *random* approximation. The number of sweeps for each $x^{(k)}$ should be small (typically less than 5) since it is enough to start with a crude approximation to I_c. A first approximation to the set of interpolation coefficients $\{w_{ij}\}_{j=1}^{n_i}$ for each i is determined so that it satisfies best, in the least-square sense, the over-determined set of equations

$$x_i^{(k)} = \sum_{j=1}^{n_i} w_{ij} x^{(k)c}_{I_{i,j}} \; , \qquad (k = 1, \ldots, \overline{n}_c) \tag{17.2}$$

where $x^{(k)c}$ is the coarse vector corresponding to $x^{(k)}$ (see above: the coarse variables are defined in terms of the fine ones, e.g., as a subset). If the least-square procedure for a particular i does not satisfy (17.2) well enough (the least-square error is larger than a threshold), then n_i is increased and the procedure for that particular i is repeated until satisfaction is obtained. (The threshold should be chosen comparable to the size of the current local normalized residuals of the homogeneous equations.) In these iterations for a particular i, one can also try to cancel any interpolation point $x^c_{I_{i,j}}$ which turns out to have a small interpolation weight w_{ij} or which exhibits near-dependence on others (a fact naturally detected by the least-square solver).

This procedure already gives a reasonable approximation to I_c, in the sense that it well interpolates most low-eigenvector eigenvalues, except that it is not likely to be good enough for many eigenvectors with *too low* eigenvalues, because the interpolation error should be small compared with the corresponding (normalized) eigenvalue. A similar first approximation is obtained for I^c. (Another way to obtain these first approximations is of course by the traditional AMG coarsening, *when applicable*.) This yields the first approximation to the coarse-level matrix $A^c = I^c A I_c$, which can then be used in a similar way to obtain a first approximation for the next, still-coarser-level matrix. There is, however, no point usually in proceeding this way too far: It is useless to access very coarse levels, whose role is to approximate very-low-eigenvalue eigenvectors, when the latter are ill approximated already in A^c.

Improved approximations. Once several coarse levels have been so defined, they can be used to obtain much better approximations to I_c and I^c. These are defined similarly to the first approximation described above, but instead of the *relaxed* vectors $x^{(k)}$, one obtains each of these vectors by a short *multilevel* procedure: Starting from a random configuration at the coarsest currently-available level, one relaxes the homogeneous equation on that level, then interpolates it to the next-finer level, where the result is again relaxed with the (finer-level) homogeneous equation, and so on to the finest level. Each interpolation is a two-stage procedure: First one uses the already-available I_c, then the result is relaxed by *compatible* relaxation (before it will next be relaxed by a usual relaxation). Each relaxation (compatible or usual) employs just a couple of sweeps.

Having obtained in this way improved approximations to A^c and similarly to coarser matrices, one can use them to similarly obtain such matrices on more levels. Then one obtains still better approximations by repeating the above procedures once more, now with more levels and with much better accuracy. This better accuracy is achieved by adding to the above short multilevel procedure a multilevel *correction cycle* to get better approximate solution to $Ax = 0$ (but keeping at the coarsest employed level still the same, relaxed but not converged, random configuration).

Accuracy and cost. The overall cost *per unknown* of this accurate coarsening procedure is $O\big(\overline{n}_c n_c^2 \log \frac{1}{\varepsilon}\big)$, where $\overline{n}_c n_c^2$ is the work needed to set up each of the least-square systems, and ε is the desired accuracy in approximating *the* *lowest eigenvectors*. For producing a good multigrid solver, it is enough to have ε which is small compared with the lowest normalized eigenvalues (except perhaps for few of them, whose corresponding error can be expelled by recombining iterants of the multigrid cycles).

An important advantage of the above procedure is that it keeps all n_i (and hence also n_c, $\overline{n}_c$) almost as small as possible, hence producing A^c almost as sparse as possible, saving much work in its calculation, and also in the actual operation of the multigrid solver. The latter is often the most important consideration, as the solver is re-used many times (cf. Sec. 7.2.1).

For some purposes (see item (i)–(iv) in Sec. 17.1) one may want to have a certain accuracy ε_1 in approximating also the *other, perhaps even all, eigenvectors*. For that purpose one has to increase n_c (and accordingly also $\overline{n}_c$). the likely relation is $n_c = O\Big(\Big(\log\frac{1}{\varepsilon_1}\Big)^q\Big)$; for 2D-Poisson equations, for instance, $q = 1$. The algorithm to derive accurate I_c in this case is actually simpler than the above: The relaxed vectors $\{x^{(k)}\}_{k=1}^{\overline{n}_c}$ should each be obtained by a long enough sequence of *compatible* relaxation sweeps, starting from a random $x^{(k)c}$; the sequence needs not be really long, just $O\Big(\log\frac{1}{\varepsilon_1}\Big)$, due to the fast convergence of such a relaxation. Despite the simplicity, this approach is of course considerably more expensive: it similarly costs $O\Big(\overline{n}_c n_c^2 \log\frac{1}{\varepsilon_1}\Big)$, but n_c here is usually much larger. Also, the produced multigrid solver is less efficient, since it involves heavier I_c, I^c and A^c at all levels.

Various combinations of the two approaches are also conceivable, depending on the nature of the desired accuracy. In such a combination, each $x^{(k)}$ may be produced by interpolations (including compatible relaxation) from a different level, and larger weights in the least square calculation may be attached to "smoother" $x^{(k)}s$, i.e., $x^{(k)}$ produced from coarser levels. Working with very smooth $x^{(k)}s$ and high accuracy (smaller ε) would yield increased values of $\{n_i\}$, effectively producing *higher order* interpolations and coarsening.

17.2.1 Nonlinear and repeated problems

Most problems that require very fast solvers need to be solved again and again many times over with small variations. This includes nonlinear problems, in which one repeatedly solves a linearized version, and the kind of problems listed at the end of Sec. 17.1. Most of the coarsening work described above need not be repeated each time the problem is modified. The coarsening should not be repeated at all if only the right-hand side changes (as in the cases of calculating propagators and determinants, described in Sec. 12 above). When the operator A changes only in some neighborhood (as in the case of *updating* the determinant value in Sec. 12), the coarsening computations need be repeated only at that neighborhood (plus at most several meshsizes around it, at each level); only the last iteration may sometimes have to be repeated globally.

Quasilinearity. Nonlinear problems can often usefully be written in the *algebraic quasilinear* form $A(x) \cdot x = b$, where the dependence of $A(x)$ on x is *non-principal*, by which we mean that $\| A(x+\delta)\cdot(x+\delta) - A(x)\cdot(x+\delta) \| \ll \|$

$A(x) \cdot \delta \parallel$ for any small δ. For example, most nonlinear PDE systems in mathematical physics are *differentially quasilinear*, meaning that each term in the system is linear in the highest derivative included in it; in the discretization, only the dependence on the highest derivative (in each such term) is principal, so the algebraic quasilinear form comes here naturally. Unlike Newton linearizations, this quasilinear discretization is *autonomous* (independent of external information, such as an approximate solution) wherever the PDE is autonomous.

In a quasilinear system, to a very good approximation the interpolation I_c depends only on $A(x)$ and, furthermore, I_c need seldom be changed when x changes. Also the form of $A(x)$ is often simple and explicit; e.g., in CFD and other areas, each term in A depends on x *linearly*. It is then possible to transfer this form of dependence also to the coarse level, enabling the employment of an FAS-like algorithm (see [19] or [23]), where the nonlinear problem is solved *directly*, without linearizations.

17.2.2 Indefinite and eigen problems

Indefinite-like systems. For some systems, at some or all regions, from a certain level of coarsening on, the numbers $\{n_i\}$ of required interpolation points (as indicated by the inaccuracies of the least square solutions) will start to snowball, calling for multiplying $\overline{n}_c$ by a certain factor for each further coarsening level, causing swelled complexity. The typical exmaples are highly indefinite systems, although some definite systems exhibit similar traits (e.g., definite systems with indefinite factors, such as $A = B^T B$, where B is highly indefinite). To check this complexity, algebraic devices generalizing the wave/ray algorithms (see Sec. 7) should be developed. Namely, the vectors $\{x^{(k)}\}$ should be recombined to extract from them a small, locally nearly orthonormal set of smooth "*basic vectors*". Any relaxed error approximates a linear combination of those basic vectors, similar to (7.2) in Sec. 7 above. As pointed out in [20, §3.2], the coarse-level correction should then actually be the sum of several such corrections, each prolongated by another "shape function". So instead of deriving one interpolation I_c, several such shape functions should be identified from the basic vectors. A generalized procedure is under development, closely related to the work on many-eigenfunction solvers (cf. Sec. 9.2).

Eigenfunction calculation. Note that the above coarsening scheme can directly yield very inexpensive calculations of many eigenfunctions of A. All the *lowest* eigenfunctions, for example, and quite many of them, would accurately be interpolated by the same interpolation I_c (especially with the *higher order* interpolations mentioned above). Hence, for all of them, the generalized eigenproblem $(A - \lambda B)x = 0$ can simultaneously be coarsened to the eigenproblem $(A^c - \lambda B^c)x^c = 0$, with $A^c = I^c A I_c$, $B^c = I^c B I_c$. This *joint* coarsening can be continued for several levels (using increasingly higher order interpolations, if necessary). Only on some coarse level the eigenfunctions are *separately* calculated. Similarly, to calculate all the eigenfunctions with eigenvalues close to a certain λ_0, the same process can be repeated for the matrix $A - \lambda_0 B$ instead of A, except that now the procedure described above for dealing with indefiniteness may have to be invoked. One can proceed this way to increasingly coarser levels by progressively narrowing the set of approximated eigenfunctions. This will naturally lead to the construction of a multiscale eigenbasis (MEB) for the matrix (cf. Sec. 9.2).

18 Global Optimization: Multilevel Strategies

An optimization problem is the task of minimizing (or maximizing — for definiteness we discuss minimization) a certain real-valued "objective functional" (or "cost", or "energy", or "performance index", etc.) $E(u)$, possibly under a set of constraints of the form $A(u) = 0$ and/or $B(u) \leq 0$, where $u = (u_1, u_2, \ldots, u_n)$ is a *vector* (often the discretization of one or several *functions*) of unknown variables (real or complex numbers, and/or integers, and/or Ising spins, etc.). A general process for solving such problems is the *point-by-point minimization*, in which one changes only one variable u_j (or few of them) at a time, lowering E as much as possible in each such step. More generally, the process accepts any candidate change of one or few variables if it causes a drop in energy ($\delta E < 0$). This process would usually suffer from the following two types of difficulties:

(i) *Slow convergence*: due to the localness of the process, large-scale features (e.g., smooth components) in u are slow to converge. Acceleration by multiscale (e.g., multigrid) methods is the general cure to this trouble, since it supplement the local processing with increasingly larger scale processing, based on information suitably gathered from the fine scale. This in fact is the topic of many chapters above; a fairly general efficient approach is presented in Sec. 17.2.

(ii) *False convergence*: instead of converging to the true *global* minimum of E, the process converges to the minimum of E in a certain restricted "*attraction basin*", in which the process is trapped. The basin is a set of configurations from which the *employed* process cannot proceed to configurations with lower E, although such configurations do exist. The emphasis in *global* optimization methods is the treatment of this type of trouble. In this chapter we do not attempt to fully cover this very extensive topic. We only outline some basic multilevel strategies that deal with it.

18.1 Multilevel formulations

In many, perhaps most, global optimization problems, the objective functional E is not uniquely determined by direct physical laws, but is man-constructed, somewhat arbitrarily, to give a precise meaning to a practical problem, whose original form is more fuzzy.

This, for example, is the formulation of ill-posed problems, like *inverse PDE problems* (system identification, as in Sec. 16.2 above, or data assimilation, as in Sec. 4, etc.). The solution of such problems is often uniquely and stably fixed with the aid of *regularization*, which recasts the problem into a minimization task. The same is true in formulating *optimal control* problems (see Sec. 5

above). In all these cases, the objective, or the sense in which one solution is considered to be better than another, is not exactly apriori given; it is *chosen*, with somewhat arbitrary form and parameters.

Another typical example is the problem of reconstructing pictures from blurred or noised data. It is often recast as the problem of minimizing an energy functional which is the sum of penalty terms, penalizing the reconstruction for various unwanted features, such as (i) its distance from the data; (ii) non-smoothness, except across lines recognized as "edges"; (iii) proliferation of such edges; (iv) non-smoothness of edges; etc. This combination of penalty terms creates a monstrous minimization problem, with many nested attraction basins at all scales. It is extremely difficult to solve — *and unnecessarily so*: The difficulty largely arises from taking too seriously a set of arbitrary choices. Indeed, the form and the numerical coefficients of the various terms are quite arbitrarily chosen; a picture which is slightly better than another according to one choice may well be worse according to many other, equally reasonable choices.

More generally, unnecessary computational difficulties often arise from our tradition to cast fuzzy tasks into "stationary" formulations, that is, to define as a solution a configuration which satisfies (exactly or approximately) one well-defined criterion, such as minimizing a certain functional under specified constraints. A more universal, and often far easier way is to admit a solution which is just the end product of a suitable *numerical process*, not necessarily designed to satisfy, even approximately, any one governing criterion. In reconstructing pictures, for example, features like edges and segments can be captured very satisfactorily by very inexpensive (multiscale) processes (few dozen operations per picture element; see Sec. 15 above); the results may well fit our perception even *better* than the true or approximate minimizer of the objective functional mentioned above. Similarly, for many other fuzzy problems, a numerical process can yield excellent solutions, whose only "fault" is our inability to say what stationary objective functional they (at least approximately) optimize.

While this may be fairly obvious, one can argue that the objective-functional formulation is still in principle the "true" one: if fully carefully chosen, it would precisely reflect what one would *want* to obtain, complicated or impractical as it may be. However, even this is often not the case: a numerical process can incorporate a host of driving directives that are impossible to include in one stationary criterion. *Examples*: (i) The process for detecting curved edges can employ different completion-field parameters at different scales (see Sec. 15.1). (ii) The process for detecting picture segments can introduce new affinities between emerging intermediate aggregates, based on their internal statistics (see Sec. 15.2). The same is true in more general clustering problems and various other fuzzy graph problems (see Sec. 15.3 and the example of graph drawing [99]). (iii) In solving inverse PDE problems one can apply *multiscale* regularizations, which use different penalty terms at different scales (see for example Item 9 in Sec. 4.3).

It can be seen from these examples that an important tool in *formulating* various problems is to have different, sometimes even conflicting, objectives at different scales of the problem. The multiscale processing is thus not just a method to accelerate convergence and escape false attraction basins (as discussed below), but can often also be essential for an improved *definition* of the problem.

Incidentally, even for *linear* problems multi-scale *formulations* are sometimes needed. An example is the case of wave equations with radiation boundary conditions: such conditions are most appropriately formulated at the coarsest levels of the wave/ray algorithm (see Sec. 7 above), while the differential equations themselves are discretized at the finest level.

18.2 Multilevel annealing

A general method to escape false attraction basins is to modify the strict point-by-point minimization by a process that still accepts each candidate change which lowers the energy ($\delta E < 0$), but also assigns a positive probability, proportional for example to $\exp(-\beta \cdot \delta E)$, for accepting a candidate step that *increases* the energy ($\delta E > 0$). This is similar to a Monte Carlo simulation of the system at a finite temperature T, where $\beta = (k_B T)^{-1}$ and k_B is the Boltzmann constant. This is indeed the very way by which natural materials escape various attraction basins and advance toward lower energies.

To have a reasonable chance to escape *wide* attraction basins or basins within *high* energy barriers in a tolerable computational time, a low value of β, or a high temperature, must of course be applied. This however makes it improbable to hit the true minimum. A general approach therefore is the *gradual* decrease of temperature, hoping first to escape false high-energy attraction basins, than lower-energy ones, etc. This process is called *simulated annealing*, since it simulates the common industrial process of "annealing" — obtaining low-energy materials (such as less brittle glass) by carefully gradual cooling. Variations on the theme include various procedures of alternate heating and cooling.

The simulated annealing algorithms are extremely inefficient for many physical problems, requiring exponentially slow temperature decrease to approach the true minimum. This is usually due to the multiscale structure of the attraction basins: small-scale basins reside within larger-scale ones, which reside within still-larger-scale ones etc. The small-scale basins correspond to *local* structures in the physical space; larger-scale basins correspond to larger physical structures. When the temperature is high enough to enable transition between large-scale attraction basins it would completely randomize finer-scale basins, even when they have already settled into low-energy local structures (by a previous cooling).

Clearly, the transitions between basins at various scales should be better coordinated. It should employ much lower temperatures in switching between large-scale basins, which can be achieved only if well orchestrated large-scale moves are constructed. This is done by what we will generally call "*multi-level annealing*", whose main features are described below. Its first, incomplete version appeared in [61].

18.2.1 Identifying multiscale variables or moves

In multilevel optimization, the main role of annealing is to *identify* increasingly larger-scale degrees of freedom that are acceptable to simulation at progressively lower temperatures. We describe two approaches to go about it.

One approach is to work in term of coarse-level **variables** that are coupled to each other through temperature-dependent conditional probability (CP) tables, as in the RMG method (cf. Secs. 13.2 and 14.7.2). Gradually, as the temperature is lowered, new coarse-level variables are generally introduced, checked by the CMC-equilibration test. The procedure is like that of Monte Carlo simulation at low temperatures (see Secs. 13.2.3 and 14.7.3), except that it can be executed without strict adherence to statistical fidelity ("detailed balance"). In many cases a low-temperature-like simulation is actually more realistic than strict minimization, either because the minimization task is fuzzy anyway (see Sec. 18.1), or simply because the material whose minimal energy is sought has in reality a *finite* temperature.

Note the similarity of this procedure to the BAMG approach in Sec. 17.2, in which increasingly coarser (large-scale) variables and interpolation rules associated with increasingly lower eigenvalues (corresponding to lower temperatures here) are gradually revealed, through a process that uses coarser levels already accessible by the current interpolation rules to accelerate relaxation (or the Monte Carlo simulation here) at finer levels.

In this approach each coarse level configuration corresponds to the *equilibrium* of all fine-level configurations that are compatible with it. When the temperature is lowered, the equilibrium narrows down to the vicinity of few specific fine-level configurations. Another approach then is to work explicitly with the fine level, and to identify on it increasingly larger-scale **moves** that can be done with progressively lower temperatures. If an efficient simulation has already been obtained at some temperature T, it can be employed to identify suitable moves for a lower temperature T', assuming $T - T' \ll T'$. Indeed, the moves already identified for T are at a scale close to those required for T', hence each suitable T'-move is approximately a linear combination of just a small number of T-moves. Such combinations can be identified by calculating correlations between neighboring T-moves during Monte Carlo simulations with the temperature T. Each combination can then be "reshaped" into more precise T'-move by optimizing around it (see Sec. 18.2.2).

The work in terms of large-scale *variables* is perhaps preferable whenever the system is *highly repetitive*, so that the same coarse-level variables and CP tables can be used at all (or many) subdomains, as in the case of fluids (Sec. 14.7). The tables then can be derived in just representative small *windows* of the fine-scale system (see the description of windows in Secs. 13.2 and 14.7.2). On the other hand, the identification of *explicit large-scale moves* is perhaps more practical for systems that have different specific structures at different neighborhoods, making it too expensive to derive place-dependent CP tables. However, the explicit moves are not flexible enough, requiring the device discussed next.

18.2.2 Reshaping large scale moves

Any preassigned large-scale move is likely to bring about a substantial energy increase since its fine details would not generally quite fit the fine details produced by other large scale moves. In other words, in switching to a new large-scale attraction basins one does not generally immediately hit the lowest-energy configurations of that basin; since in the *previous* basin a process of minimization *has* already taken place, the new configuration is likely to exhibit much higher energy. Thus, only rarely the large-scale move will be accepted in a low-temperature simulation, even if the new attraction basin does harbor lower energy configurations. Therefore, before applying the acceptance test to a large-scale move, one should "*reshape*" it, or "optmize around it", by employing in the neighborhood around it a Monte Carlo simulation of smaller-scale moves. Each of these smaller-scale moves may itself need "reshaping" by local simulations around it at still finer scales. And so on. Such nested reshaping processes are needed when the energy landscape has nested attraction basins. Each of these processes can itself employ a kind of annealing (see details in [61]).

Working with the difficult discrete optimization problem of spin glasses, it was shown already in [61] that such multiscale nested optimization techniques (together with the technique of Sec. 18.2.3 below) work reasonably well ever *without* any prior identification of specialized moves at all scales (cf. Sec. 18.2.1). However, the amount of work in that case turned out to increase at least quadratically as a function of the number of spins in the system, due to the excessive nested reshaping processes that were required. Much shorter reshaping procedures will suffice with more specialized moves. (Also, as mentioned in Sec. 18.2.1, the reshaping procedure can be used to optimize the specialized moves themselves, prior to their use in the T' simulations.)

Note that the reshaping procedure (unless confined only to the prior identification of moves) does not satisfy the statistical detailed balance. It is very efficient in the search for a minimum, but cannot be used for obtaining accurate finite-temperature statistics.

18.2.3 Taming local fluctuations and genetic-type algorithms

In any sufficiently large-scale problem with local couplings (i.e., its objective functional is the sum of terms each of which depends only on a *local* set of variables, in some space), there is a large accumulation of likelihood that any stochastic simulation, even with a low temperature, will create some small-scale local fluctuations, frustrating the chance to identify *the* global minimum. Since these fluctuations are indeed likely to be local, one can eliminate them by the following simple procedure.

Keep in memory one or several of the best-so-far (BSF) configurations. Once in a while (e.g., whenever the stochastically-evolving current configuration yields a particularly low energy) compare the current configuration with each of the BSF configurations. The two compared configurations will generally have spots of just *local* disagreement, i.e., *disconnected* subsets where the values of the two configurations differ, but outside which the configurations coincide. Hence, for each such subset, separately from all other subsets, one can decide whether or not to replace the BSF values by those of the current configuration, depending which option would yield at that spot the lower energy. In this way all the BSF configurations can be replaced by better ones. The current configuration should continue its evolution from its previous value, in search for new optima. At the end, the BSF configurations can be compared to choose the best among them.

This device should apply not only to the main optimization process, but also to each of the auxiliary "reshaping" processes defined above (Sec. 18.2.2), as successfully demonstrated in [61].

Analogous devices can be used even for more general problems (not just locally coupled). The general approach can be described as a combination of multilevel annealing with genetic-type algorithms. Instead of one minimization process, a population of such processes evolve in parallel. Once in a while one of the evolving configurations (a "parent") chooses another (a "partner"), from which it borrows a combination of large-scale moves, reshaping them using its own finer multiscale moves (see Sec. 18.2.2), then (and only then) deciding whether to adopt the resulting configuration (accept it as an addition to the population or as a replacement). Each of the reshaping processes can itself be done in terms of several evolving children, and so on recursively. "Fitness" parameters can be defined in terms of the low-energy levels attained by the evolving configuration and its relatives. The choice of "partner" can be based on its fitness and criteria of compatibility with the choosing "parent".

In short, one can marry the ideas of multiscale optimization with those of genetic algorithms and study the (fuzzy) fitness of their evolving offsprings. The success is likely to be especially high for problems dominated by a multitude of local couplings.

19 What About Wavelets?

Wavelets, perhaps the currently most popular form of multiscale representation, have not been mentioned in any of the above sections. An explanation is due.

First, in all areas described above, either wavelets are not at all applicable (as in Secs. 13, 14, 15, 16, 17), or they are less developed than multigrid-type methods. Part of the reason for that is historical: Multigrid methods, in either finite-difference or finite-element formulations, have appeared long before wavelets. The excitement about wavelets is probably due not only to its mathematical elegance, but also to the fact that this has been the first form of multiscale representation encountered by several communities, hence the form through which they first discovered the great computational benefits of multiscaling. There are indeed many very effective algorithms developed with wavelets. But multiscaling had existed before, and the question is how wavelets compare with other forms of multiscale representation from the point of view of computational efficiency. The answer is that wavelets are less efficient, at least for the kind of problems surveyed in this article.

To explain, note that there are mainly two (related) differences in emphasis distinguishing wavelets from more general formulations. The first, and less essential difference is the *incremental* representation used by wavelets: while multigrid methods represent at the fine level the full function, wavelets separately represent its high-resolution (e.g., high-frequency) part. All other scales of resolution are similarly separated. This separation degrades the efficiency in treating *nonlinear* problems. (In FAS multigrid the full function is represented at all levels; as a result the solution of nonlinear problems is usually as fast and easy as solving linear problems: no linearizations, with their vast extra storage and iterations, are needed [19], [23].)

The separate representation of the high-resolution part is considered by some to be advantageous from the point of view of self-adaptive discretization: wherever that part is larger than some threshold, the need for a still-finer resolution is indicated. However, there is no special advantage here, because, in any other multilevel representation, that high resolution part can straightforwardly be estimated from the difference between the solutions at the finest level and at the next level. (The true *local* criterion for grid adaptation is solving PDE systems is the local error in the *equation*, not in the solution. In FAS multigrid methods this error is automatically given by the fine-to-coarse correction τ_h^{2h} [19], [23].)

The second, and more important, feature that distinguish wavelets is the *orthogonality* of the levels: Each level of wavelet resolution is exactly orthogonal to all other levels. This is very pleasing mathematically, but is also a source of a certain computational inefficiency: The representation is substantially more complicated and costly than a simple multigrid representation at the corresponding order. Although the latter does not enjoy orthogonality of levels, it does always have *near orthogonality*: even though the processing (e.g., relaxation) at each level does slightly affect other levels, this slight influence for most purposes is insignificant. For example, it does not harm the multigrid convergence rates, which fit the "smoothing rates" of the separate levels [19], [23], [28]. Moreover, in special cases where this is needed, one can make inter-level influence as small as desired by employing, e.g., *distributive* moves at each level (see for example Secs. 10 and 14.6.1 above).

Thus, the natural near-orthogonality of *any* multiscale representation makes it unnecessary, and usually more expensive, to use the exact orthogonality offered by wavelets.

References

[1] Alessandrini, G., Stable determination of conductivity by boundary measurements, *Appl. Anal.* **27** (1988) 153–172.

[2] Arian, A. and Ta'asan, S., Multigrid one shot methods for optimal control problems: Infinite dimensional control, ICASE Report, 1994, pp. 94–52.

[3] Arian, A. and Ta'asan, S., Shape optimization in one shot, in: *Optimal Design and Control* (Boggard, J., Burkardt, J., Gunzburger, M. and Peterson, J., eds.), Birkhauser Boston Inc., 1995.

[4] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A. and Protasi, M., *Complexity and Approsimation*, Springer Verlag, 1999.

[5] Babuška, I. and Melenk, J. M., The partition of unity method, *Int. J. Num. Methods in Engineering* **40** (1997) 727–758.

[6] Bai, D. and Brandt, A., Local mesh refinement multilevel techniques, *SIAM J. Sci. Stat. Comput.* **8** (1987) 109–134.

[7] Bai, D. and Brandt, A., Multiscale computation of polymer models, in [39], pp. 250–266.

[8] Baschnagel, J., Binder, K., Doruker, P., Gusev, A. A., Hahn, O., Kremer, K., Mattice, W. L., Müller-Plathe, F., Murat, M., Paul, W., Santos, S., Suter, U. W. and Tries, V., Bridging the gap between atomistic and coarse-grained models of polymers: Status and perspectives, *Advances in Polymer Science*, Springer Verlag **152** (2000) 41–156.

[9] Bates, J. R., Li, Y., Brandt, A., McCormick, S. F. and Ruge, J., A global shallow water numerical model based on the semi-Lagrangian advection of potential vorticity, *Quart. J. Roy. Met. Soc.* **121** (1995) 1981–2005.

[10] Bates, J. R., McCormick, S. F., Ruge, J., Sholl, D. S. and Yavneh, I., A semi-Lagrangian approach to the shallow water equations, 6^{th} *Copper Mountain Conf. on Multigrid Methods*, NASA Conference Publication 3224, Part 2, 1993, pp. 593–604.

[11] Bates, J. R., Semazzi, F. H. M., Higgins, R. W. and Barros, S. R. M., Integration of the shallow water equations on the sphere using a vector semi-Lagrangian scheme with a multigrid solver, *Mon. Wea. Rev.* **18** (1990) 1615–1627.

[12] Beck, T. L., Multigrid high order mesh refinement technique, *J. Comput. Chem.* **20** (1999) 1731.

[13] Beck, T. L., Multiscale methods for self-consistent electronic structure in real space, in [39], pp. 90–103.

[14] Ben-Av, R., *Acceleration Methods in MC Simulations*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, 1991.

[15] Ben-Av, R., Brandt, A., Harmatz, M., Katznelson, E., Lauwers, P. G., Solomon, S. and Wolowesky, K., Fermion simulation using parallel transported multigrid, *Phys. Lett.* **B253** (1991) 185.

[16] Ben-Av, R., Brandt, A. and Solomon, S., The Fermionic matrix, instantons, zero modes and multigrid, *Nucl. Phys.* **B329** (1990) 193.

[17] Bernholc, J., Briggs, E. L., Buongiorno Nardelli, M., Fattebert, J.-L., Ramamoorthy, M., Schmidt, W. G. and Sullivan, D. J., Large-scale, multilevel solutions of Kohn-Sham equations: methodology and applications, in [39], pp. 65–89.

[18] Brandt, A., Multi-level adaptive technique (MLAT) for fast numerical solutions to boundary value problems, in: *Proc. 3^{rd} Int. Conf. on Numerical Methods in Fluid Mechanics* (Cabannes, H. and Temam, R., eds.), Lecture Notes in Physics **18**, Springer-Verlag, 1973, pp. 82–89.

[19] Brandt, A., Multi-level adaptive solutions to boundary value problems, *Math. Comp.* **31** (1977) 333–390.

[20] Brandt, A., Stages in developing multigrid solutions, in: *Proc. 2^{nd} Int. Congr. on Numerical Methods for Engineers* (Absi, E., Glowinski, R., Lascaux, P. and Veysseyre, H., eds.), Dunod, Paris, 1980, pp. 23–43.

[21] Brandt, A., Multi-grid solvers on parallel computers, in: *Elliptic Problem Solvers* (Schultz, M., ed.), Academic Press, New York, 1981, pp. 39–84.

[22] Brandt, A., *Multi-Grid Solvers for Non-Elliptic and Singular-Perturbation Steady-State Problems*, Weizmann Institute of Science, Rehovot, November 1981.

[23] Brandt, A., Guide to multigrid development, in: *Multigrid Methods* (Hackbusch, W. and Trottenberg, U., eds.), Springer-Verlag, 1982, pp. 220–312.

[24] Brandt, A., *Multigrid Techniques: 1984 Guide, with Applications to Fluid Dynamics*, 191 pages, 1984, ISBN-3-88457-081-1, GMD Studien Nr. 85. Available from GMD-AIW, Postfach 1316, D-53731, St. Augustin 1, Germany, 1984.

[25] Brandt, A., Algebraic multigrid theory: The symmetric case, *Appl. Math. Comput.* **19** (1986) 23–56.

[26] Brandt, A., Multilevel computations: Review and recent developments, in: *Multigrid Methods: Theory, Applications and Supercomputing* (McCormick, S. F., ed.), Marcel-Dekker, 1988, pp. 35–62.

[27] Brandt, A., The Weizmann Institute Research in Multilevel Computation: 1988 Report, in: *Proc. 4^{th} Copper Mountain Conf. on Multigrid Methods* (Mandel, J. *et al.*, eds.), SIAM, 1989, pp. 13–53.

[28] Brandt, A., Rigorous local mode analysis of multigrid, in: *Preliminary Proc. 4^{th} Copper Mountain Conf. on Multigrid Methods*, Copper Mountain, Colorado, April 1989. An updated version appeared as Weizmann Institute Report, 1991. The first part has appeared as [32].

[29] Brandt, A., Multilevel computations of integral transforms and particle interactions with oscillatory kernels, *Comput. Phys. Comm.* **65** (1991) 24–38.

[30] Brandt, A., Multiscale computational methods: research activities, in: *Proc. 1991 Hang Zhou International Conf. on Scientific Computation*, (Chan, T. and Shi, Z.-C., eds.), World Scientific Publishing Co., Singapore, 1992.

[31] Brandt, A., Multigrid methods in lattice field computations, *Nuclear Phys. B Proc. Suppl.* **26** (1992) 137–180.

[32] Brandt, A., Rigorous quantitative analysis of multigrid: I. Constant coefficients two level cycle with L_2 norm, *SIAM J. Numer. Anal.* **31** (1994) 1695–1730.

[33] Brandt, A., Multiscale research in molecular dynamics, Gauss Center Report WI/GC–3, April 1995.

[34] Brandt, A., Proceedings of the multigrid tutorial, with applications to molecular dynamics, Weizmann Institute of Science, October 10–12, 1995.

[35] Brandt, A., The Gauss Center research in multiscale scientific computation, *Electr. Trans. Numer. Anal.* **6** (1997) 1–34.

[36] Brandt, A., *Barriers to Achieving Textbook Multigrid Efficiency in CFD*, ICASE Interim Report No. 32, NASA/CR-198-207647. Gauss Center Report WI/GC–10, June 1998; updated December, 1998. Appears as Appendix C in [153].

[37] Brandt, A., General highly accurate algebraic coarsening schemes, *Electr. Trans. Num. Anal.* **10** (2000) 1–20.

[38] Brandt, A., Multiscale scientific computation: six year summary, Gauss Center Report WI/GC–12, May 1999. Also in: http://www.wisdom.weizmann.ac.il/∼achi, and in MGNET.

[39] Brandt, A., Bernholc, J. and Binder, K. (Eds.), *Multiscale Computational Methods in Chemistry*, NATO Science Series, Computer and System Sciences, Vol. 177, IOS Press, Amsterdam (2000).

[40] Brandt, A. and Dinar, N., Multi-grid solutions to elliptic flow problems, in: *Numerical Methods for Partial Differential Equations* (Parter, S., ed.), Academic Press, New York, 1979, pp. 53–147.

[41] Brandt, A. and Diskin, B., Multigrid solvers for the non-aligned sonic flow: the constant coefficient case, Gauss Center Report WI/GC–8, 1997. *Computer and Fluids* **28** (1999) 511–549.

[42] Brandt, A. and Diskin, B., Multigrid solvers for non-aligned sonic flows, *SIAM J. Sci. Comput.* **21** (1999) 473–501.

[43] Brandt, A. and Dym, J., Fast computation of multiple line integrals, *SIAM J. Sci. Comput.* **20** (1999) 1417–1429.

[44] Brandt, A. and Dym, J., Fast robust discontinuity detection using multiple scales, preprint.

[45] Brandt, A. and Galun, M., Optimal multigrid algorithms for the massive Gaussian model and path integrals, *J. Statist. Phys.* **82** (1996) 1503–1518.

[46] Brandt, A. and Galun, M., Optimal multigrid algorithms for variable-coupling isotropic Gaussian models, *J. Statist. Phys.* **88** (1997) 637–664.

[47] Brandt, A. and Galun, M., Statistically optimal multigrid algorithm for the anharmonic crystal model, Gauss Center Report WI/GC–9, March 1998.

[48] Brandt, A., Galun, M. and Ron, D., Optimal multigrid algorithms for calculating thermodynamic limits, *J. Statist. Phys.* **74** (1994) 313–348.

[49] Brandt, A. and Greenwald, J., Parabolic multigrid revisited, in: *Multigrid Methods III* (Hackbusch, W. and Trottenberg, U., eds.), Birkhäuser Verlag, Basel, 1991, pp. 143–154.

[50] Brandt, A. and Iliyn, V., Multilevel approach in statistical physics of liquids, in [39], pp. 187–197.

[51] Brandt, A. and Livshits, I., Wave-ray multigrid methods for standing wave equations, *Electr. Trans. Numer. Anal.* **6** (1997) 162–181.

[52] Brandt, A. and Livshitz, I., Accuracy properties of the wave-ray multigrid algorithm for Helmholtz equation, in preparation.

[53] Brandt, A. and Lubrecht, A. A., Multilevel matrix multiplication and the fast solution of integral equations, *J. Comput. Phys.* **90**(2) (1990) 348–370.

[54] Brandt, A., Mann, J. and Brodski, M., *A Fast and Accurate Radon Transform Inversion Scheme*, Patent assigned to Yeda Research and Development Co. Ltd., August 1995. U.S. Patent and Trademark Office Application No. 08/659,595, filed 06/06/96. U.S.

Patent No. 5,778,038, granted July 7, 1998 under the title "Computerized Tomography Scanner and Method of Performing Computerized Tomography". European Patent Office Application No. 97l08722.6-2305, filed 05/30/97.

[55] Brandt, A., Mann, J., Brodski, M. and Galun, M., A fast and accurate multilevel inversion of the Radon transform, *SIAM J. Appl. Math.* **60** (1999) 437–462.

[56] Brandt, A., McCormick, S. and Ruge, J., Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations, Institute for Computational Studies, POB 1852, Fort Collins, Colorado, 1982.

[57] Brandt, A., McCormick, S. and Ruge, J., Multi-grid methods for differential eigenproblems, *SIAM J. Sci. Statist. Comput.* **4** (1983) 244–260.

[58] Brandt, A., McCormick, S. and Ruge, J., Algebraic multigrid (AMG) for sparse matrix equations, in *Sparsity and its Applications* (Evans, D. J., ed.), Cambridge University Press, Cambridge, 1984, pp. 257–284.

[59] Brandt, A. and Mikulinsky, V., Recombining iterants in multigrid algorithms and problems with small islands, *SIAM J. Sci. Comput.* **16** (1995) 20–28.

[60] Brandt, A. and Ron, D., Renormalization Multigrid (RMG): Statistically optimal renormalization group flow and coarse-to-fine Monte Carlo acceleration, *J. Stat. Phys.* **102** (2001) 231–257. Also appeared as Gauss Center Report WI/GC–11, 1999, and with modification in [39], pp. 163–186.

[61] Brandt, A., Ron, D. and Amit, D. J., Multi-level approaches to discrete-state and stochastic problems, in: *Multigrid Methods, II* (Hackbusch, W. and Trottenberg, U., eds.), Springer-Verlag, 1986, pp. 66–99.

[62] Brandt, A. and Venner, C. H., Fast evaluation of integral transforms with asymptotically smooth kernels, Gauss Center Report WI/GC–2, 1995. *SIAM J. of Sci. Comput.* **19** (1998) 468–492.

[63] Brandt, A. and Venner, C. H., Multilevel evaluation of integral transforms on adaptive grids, Gauss Center Report WI/GC–5, 1996. In: *Multigrid Methods V*, Lecture Notes in Computational Science and Engineering **3** (Hackbusch, W. and Wittum, G., eds.), Sprinter Verlag, Berlin, 1999, pp. 20–44.

[64] Brandt, A. and Yavneh, I., On multigrid solution of high-Reynolds incompressible entering flows, *J. Comput. Phys.* **101** (1992) 151–164.

[65] Brandt, A. and Yavneh, I., Accelerated multigrid convergence and high-Reynolds recirculating flows, *SIAM J. Sci. Comput.* **14** (1993) 607–626.

[66] Brandt, A. and Zaslavsky, L. Yu., Multiscale algorithm for atmospheric data assimilation, *SIAM J. Sci. Comput.* **18** (1997) 949–956.

[67] Briggs, W. L., Henson, V. E. and McCormick, S. F., *A Multigrid Tutorial*, 2nd Ed., SIAM, 2000.

[68] Bunch, J. R. and Nielsen, C. P., Updating the singular value decomposition, *Numer. Math.* **31** (1978) 111–129.

[69] Calderon, A., On an inverse boundary value problem, in: *Seminar on Numerical Analysis and its Applications* (Meyer, W. and Raupp, M., eds.), Brazilian Mathematical Society, Rio de Janeiro, 1980, pp. 1–7.

[70] Chan, T. F. and Keller, H. B., Arc-length continuation and multigrid techniques for nonlinear elliptic eigenvalue problems, *SIAM J. Sci. Stat. Comput.* **3** (1982) 173–194.

[71] Chan, T. F. and Bank, R. E., PLTMGC: A multi-grid continuation program for parametrized nonlinear elliptic systems, *SIAM J. Sci. Stat. Comput.* **7** (1986) 540–559.

[72] Cormen, Leiserson and Rivest, Introduction to algorithms, *J. ACM* (2000).

[73] Costiner, S., *Simultaneous Multigrid Solvers for the Nonlinear Schrödinger Eigenvalue Problem*, M.Sc. Thesis, Weizmann Institute of Science, 1990.

[74] Costiner, S. and Ta'asan, S., Adaptive multigrid techniques for large-scale eigenvalue problems: solutions of the Schrödinger problem in two and three dimensions, *Phys. Rev.* **E51** (1995) 3704–3717.

[75] Cuppen, J. J. M., A divide and conquer method for the tridiagonal eigenproblem, *Numer. Math.* **36** (1981) 177–195.

[76] Deans, S. R., *The Radon Transform and Some of its Applications*, John Wiley and Sons, New York, 1983.

[77] Diskin, B., Multigrid algorithm with conditional coarsening for the non-aligned sonic flow, *Electr. Trans. Numer. Anal.* **6** (1997) 106–119.

[78] Diskin, B., *Efficient Multigrid Solvers for the Linearized Transonic Full Potential Equation*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, June 1998.

[79] Dym, J., *Multilevel Methods for Early Vision*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, 1994.

[80] Faddeeva, V. N., *Computational Methods of Linear Algebra*, Dover Publications, New York, 1959.

[81] Feige, U., Kortsarz and Peleg, D., The dense k-subgraph problem, *Algorithmica*, March 2001.

[82] Flory, P. J., *Statistical Mechanics of Chain Molecules*, January 1989, ISBN 1569900191.

[83] Forsythe, G. E. and Golub, G. H., On the stationary values of a second-degree polynomial on the unit sphere, *SIAM J. Appl. Math.* **13** (1965) 1050–1068.

[84] Fuhrmann, D. R., An algorithm for subspace computation with applications in signal processing, *SIAM J. Matrix Anal. Appl.* **9** (1988) 213–220.

[85] Galli, G., Linear scaling method for electronic-structure calculations and quantum molecular dynamics simulations, *Current Opinion in Solid State and Material Science* **1**(6) (1996) 864–874.

[86] Galun, M., *Optimal Multigrid Algorithms for Model Problems in Statistical Mechanics*, M.Sc. Thesis, Weizmann Institute of Science, Rehovot, 1992.

[87] Galun, M., *Multigrid Algorithms for Optimal Computations in Statistical Physics*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, April 1998.

[88] Gandlin, R., *Multigrid Algorithms for an Inverse Problem in Impedance Tomography*, M.Sc. Thesis, Weizmann Institute of Science, Rehovot, June 1998.

[89] Gandlin, R., Proposal for Ph.D. Thesis, Weizmann Institute of Science, Rehovot, 1999.

[90] Gandlin, R., *Multigrid Algorithms for Inverse Problems*, Progress Report, Feinberg Graduate School, Weizmann Institute of Science, Rehovot, 2000.

[91] Gödecker, S., Linear scaling electronic structure methods, *Rev. Modern Phys.* **71** (1999) 1085–1123.

[92] Golub, G. H. and von Matt, U., Quadratically constrained least squares and quadratic problems, *Numer. Math.* **59** (1991) 561–580.

[93] Graph spanners, *J. Graph Theory* **13**(1) (1989) 99–116.

[94] Greengard, L., Fast algorithms for classical physics, *Science* **265** (1994) 909–914.

[95] Greenwald, J. C., *Multigrid Techniques for Parabolic Problems*, Ph.D. Thesis, Weizmann Institute of Science, June 1992.

[96] Grünbaum, F. A., Diffuse tomography: the isotropic case, *Inverse Problems* **8** (1992) 409–419.

[97] Gupta, R. and Cordery, R., Monte Carlo renormalized Hamiltonian, *Phys. Lett.* **105A** (1984) 415.

[98] Guy, G. and Medioni, G., Inferring global perceptual contours from local features, *Proc. DARPA Image Understanding Workshop*, 1993, pp. 881–893.

[99] Hadany, R. and Harel, D., A multi-scale method for drawing graphs nicely, to appear in *Discrete Appl. Math.*. Also *Proc. 25th Int. Workshop on Graph-Theoretic Concepts in Computer Science* (*WG'99V*), Lecture Notes in Computer Science, Vol. 1665, Springer-Verlag, June 1999, pp. 262–277.

[100] Haras, Z. and Ta'asan, S., Finite difference schemes for long time integration, *J. Comp. Phys.* **114**(2) (1994).

[101] Harel, D., *Computer Ltd.*, Oxford University Press, 2000.

[102] Herman, G., *Image Reconstruction from Projections*: *The Fundamentals of Computerized Tomography*, Academic Press, New York, 1980.

[103] Hill, T. L., *Statistical Mechanics*, McGraw-Hill Book Company Inc., New York-Toronto-London, 1956.

[104] Jameson, A., Solution of the Euler equations by a multigrid method, *Appl. Math. Comput.* **13** (1983) 327–356.

[105] Jameson, A., Schmidt, W. and Turkel, E., Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes, AIAA Paper 81-1259, 1981.

[106] Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W. and Klein, M. L., Comparison of potential functions for simulating liquid water, *J. Chem. Phys.* **79** (1983) 926–935.

[107] Jorgensen, W. L. and Jenson, C., Temperature dependence of TIP3P, SPC and TIP4P water from NPT Monte Carlo simulations: seeking temperatures of maximum density, *J. Comp. Chem.* **19** (1998) 1179–1186.

[108] Kalkreuter, T., *Multigrid Methods for the Computation of Propagators in Gauge Fields*, Ph.D. Thesis, Hamburg University, Report DESY92–158, 1992.

[109] Kalkreuter, T., Idealized multigrid algorithm for staggered fermions, *Phys. Rev.* **D48** (1993) 1926–1930.

[110] Kaminsky, R., *Multilevel Solution of the Long Transportation Problem*, M.Sc. Thesis, Weizmann Institute of Science, 1989.

[111] Keller, J. B., Geometrical theory of diffraction, *J. Optical Soc. Am.* **52** (1962).

[112] Kogut, J. B., An introduction to lattice gauge theory and spin systems, *Rev. Modern Phys.* **51** (1979) 659–713.

[113] Kogut, J. B., The lattice gauge theory approach to quantum chromodynamics, *Rev. Modern Phys.* **55** (1983) 775–836.

[114] Kogut, J. B. and Susskind, L., Hamiltonian formulation of Wilson's lattice gauge theory, *Phys. Rev.* **D11** (1975) 395–408.

[115] Kohn, P. and Vogelius, M., Determining conductivity by boundary measurements, *Comm. Pure Appl. Math.* **37** (1984) 289–298.

[116] Kohn, W. and Sham, L. J., *Phys. Rev.* **140** (1965) A1133.

[117] Li, Y. and Bates, J. R., A study of the behavior of semi-Lagrangian models in the presence of orography, *Quart. J. R. Met. Soc.* **122** (1996) 1675–1700.

[118] Li, Y., Ruge, J., Bates, J. R. and Brandt, A., A proposed adiabatic formulation of three-dimensional global atmospheric models based on potential vorticity, *Tellus* **52A** (2000) 129–139.

[119] Livne, O. E., *Multiscale Eigenbasis Algorithms*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, 2000.

[120] Livne, O. E. and Brandt, A., $O(N \log N)$ multilevel calculation of N eigenfunctions, in [39], pp. 112–136. Also in the present volume.

[121] Livne, O. and Brandt, A., N roots of secular equation in $O(N)$ operations, submitted to *SIAM J. Matrix Anal. Appl.*

[122] Livshits, I., *Multigrid Solvers for Wave Equations*, Ph.D. Thesis (Brandt, A., supervisor), Bar-Ilan University, November 1995.

[123] McWilliams, J. C., Weiss, J. B. and Yavneh, I., Anisotropy and coherent vortex structures in planetary turbulence, *Science* **264**(5157) (1994) 410–413.

[124] Melenk, J. M., *On generalized finite element method*, Ph.D. Thesis, University of Maryland, 1995.

[125] Melman, A., Numerical solution of a secular equation, *Numer. Math.* **69** (1995) 483–493.

[126] Melman, A., A numerical comparison of methods for solving secular equations, *J. Comp. Appl. Math.* **86** (1997) 237–249.

[127] Paul, W., Yoon, D. Y. and Smith, G. D., An optimized united atom model for simulations of polymethylene melts, *J. Chem. Phys.* **103** (1995) 1702–1709.

[128] Peskin, C. S. and Schlick, T., Molecular dynamics by backward-Euler method, *Comm. Pure Appl. Math.* **42** (1989) 1001.

[129] Ron, D. and Swendsen, R. H., Calculation of effective Hamiltonians for renormalized or non-Hamiltonian systems, *Phys. Rev. E*, to appear.

[130] Rozantsev, M., *Methods for Multigrid Solution of Quantum Electrodynamics Equations*, M.Sc. Thesis, Weizmann Institute of Science, Rehovot, 1993.

[131] Rozantsev, M., *Multigrid Solver for 4D Quantum Chromodynamics Equations*, Final Report, Feinberg Graduate School, Weizmann Institute of Science, Rehovot, January 1999.

[132] Rozantsev, M., *Multigrid Solver for 4D Quantum Chromodynamics Equations*, Ph.D. Thesis, submitted to Feinberg Graduate School, Weizmann Institute of Science, Rehovot, February 2001.

[133] Ruge, J. W., Li, Y., McCormick, S., Brandt, A. and Bates, J. R., A nonlinear multigrid solver for a semi-Lagrangian potential vorticity based shallow water model on the sphere, *SIAM J. Sci. Comput.* **21** (2000) 2381–2399.

[134] Ruge, J. and Stüben, K., Algebraic multigrid, in: *Multigrid Methods* (McCormick, S. F., ed.), SIAM, Philadelphia, 1987, pp. 73–130.

[135] Sandak, B. and Brandt, A., Multiscale fast summation of long range charge and dipolar interactions, in [39], pp. 6–31. *J. Comp. Chem.*, in press.

[136] Schmittmann, B. and Zia, R. K. P., *Statistical Mechanics of Drivan Diffusive Systems*, Academic Press, 1995.

[137] Schröder, J., Trottenberg, U. and Reuterssberg, H., Reuktionsverfahren fuer differenzengleichungen bei randwertaufgaben II, *Num. Math.* **26** (1976) 429–459.

[138] Sharatchandra, H., Thun, H. and Weisz, P., Susskind fermions on a Euclidean lattice, *Nuc. Phys.* **B192** (1981) 205.

[139] Sharon, E., Brandt, A. and Basri, R., Completion Energies and Scale, *IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR–97), Puerto Rico, 1997. Report CS97–19, Weizmann Institute of Science, Rehovot. *IEEE Trans. on Pattern Anal. and Machine Intelligence* **22** (2000) 1117–1131.

[140] Sharon, E., Brandt, A. and Basri, R., Fast multiscale image segmentation, *IEEE Conf. on Computer Vision and pattern recognition*, South Carolina, 2000, pp. 70–77.

[141] Shmulyian, F., *Multigrid Algorithms for an Inverse Problem in Impedance Tomography*, M.Sc. Thesis, Weizmann Institute of Science, Rehovot, August 1995.

[142] Shmulyian, S., *Toward Optimal multigrid Monte Carlo Computations in Two-Dimensional $O(N)$ Non-Linear σ-models*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, March 1999.

[143] Smith, G. D., Paul, W., Yoon, D. Y. and Richter, D., Microscopic Dynamics in a Long-chain Alkane Melt: Computer Simulation and Neutron Scattering, Report, Department of Chemical Engineering, University of Missouri at Columbia, July 1996.

[144] Sonneveld, P., Wesseling, P. and de Zeeuw, P. M., Multigrid and conjugate gradient methods as convergence acceleration techniques, *Proc. Multigrid Conf.* (Holstein, H. and Paddon, D., eds.), Bristol, 1983.

[145] South, J. C. and Brandt, A., Application of a multi-level grid method to transonic flow calculations, in: *Transonic Flow Problems in Turbo Machinery* (Adam, T. C. and Platzer, M. F., eds.), Hemisphere, Washington, 1977, pp. 180–207.

[146] Stüben, K., Algebraic multigrid (AMG): An introduction with applications, Guest appendix in [153]. A review of algebraic multigrid, *J. Comp. Appl. Math.* **128**(1–2), 2001.

[147] Susskind, L., Lattice fermions, *Phys. Rev.* **D16** (1977) 3031–3039.

[148] Swendsen, R. H., Monte Carlo calculation of renormalized coupling parameters, *Phys. Rev. Lett.* **52** (1984) 1165.

[149] Sylvester, J. and Uhlmann, G., A global uniqueness theorem for an inverse boundary value problem, *Ann. Math.* **125** (1987) 153–169.

[150] Ta'asan, S., One-shot methods for optimal control of distributed parameter systems, I. Finite dimensional control, ICASE Report 91–2, 1991.

[151] Thomas, J. L., Diskin, B. and Brandt, A., Textbook multigrid efficiency for the incompressible Navier-Stokes equations: High Reynolds number wakes and boundary layers, ICASE Report No. 99–51, 1999. *Computers and Fluids*, to appear.

[152] Thomas, J. L., Diskin, B., Brandt, A. and South, J. C. Jr., General framework for achieving textbook multigrid efficiency: quasi-1-D Euler example, in: *Frontiers of Computational Fluid Dynamics — 2000* (Caughey, D. A. and Hafez, M. M., eds.), World Scientific Publishing Company, Singapore.

[153] Trottenberg, U., Oosterlee, C. W. and Schüller, A., *Multigrid*, Academic Press, London, 2000.

[154] Tziperman, E., Yavneh, I. and Ta'asan, S., Multilevel turbulence simulations, *Europhysics Lett.* **24**(4) (1993) 239.

[155] Van der Wees, A. J., Van der Vooren, J. and Meelker, J. H., Robust calculation of 3D transonic potential flow based on the nonlinear FAS multigrid method and incomplete LU-decomposition, AIAA Paper 83-1950, 1983.

[156] Vandewalle, S. and Horton, G., Fourier mode analysis of the waveform relaxation and time-parallel multigrid methods, *SIAM J. Sci. Comput.* (1995).

[157] Van Leer, B., Tai, C. H. and Powell, K. G., Design of optimally-smoothing multi-stage schemes for the Euler equations, AIAA Paper 89-1933, June 1989.

[158] Venner, C. H. and Lubrecht, A. A., *Multilevel Methods in Lubrication*, Elsevier (Tribology Series, 37), Amsterdam, 2000.

[159] Wesseling, P. and Sonneveld, P., Numerical experiments with a multiple grid and a preconditioned Lanczos type method, in: *Approximation Methods for Navier-Stokes Problems* (Rautmann, R., ed.), Lecture Notes in Math. **771**, Springer-Verlag, 1980, pp. 543–562.

[160] Whitham, G. B., *Linear and Nonlinear Waves*, Wiley-Interscience, New York, 1974.

[161] Williams, L. R. and Jacobs, D. W., Stochastic completion fields: A neural model of illusory contour shape and salience, *ICCV-95*, 1995, pp. 408–415.

[162] Wilson, K. G., Confinement of quarks, *Phys. Rev.* **D10** (1974) 2445–2459.

[163] Yavneh, I., A method for devising efficient multigrid smoothers for complicated PDE systems, *SIAM J. Sci. Comput.* **14** (1993), 1437–1463.

[164] Yavneh, I. and McWilliams, J. C., Breakdown of the slow manifold in the shallow-water equations, *Geo. Astr. Fluid Dyn.* **75** (1994) 131–161.

[165] Yavneh, I. and McWilliams, J. C., Robust multigrid solution of the shallow-water balance equations, *J. Comput. Phys.* **119** (1995) 1–25.

[166] Yavneh, I., Venner, C. H. and Brandt, A., Fast multigrid solution of the advection problem with closed characteristics, *SIAM J. Sci. Comput.* **19** (1998) 111–125.

[167] Zimare, H., *Beweis der sternenkonvergenz und untersuchung der stabilitaet beim verfahren der totalen reduktion*, Ph.D. Thesis, University of Cologne, 1980.