Multiscale Methods of Data Assimilation and Feedback Optimal Control

Achi Brandt
The Weizmann Institute of Science, Rehovot 76100, Israel
Phone: +972-8-9342345
Fax: +972-8-9342945
Email: achi.brandt@weizmann.ac.il

Abstract

Following an introduction to nonlinear adaptible-discretization multigrid algorithms for steady-state and time-dependent partial differential equations (PDEs), their many potential benefits for solving inverse PDE problems are explained, focusing on the problems of atmospheric data assimilation and feedback optimal control. Using multigrid, the atmospheric flow equations with very stable and adaptable implicit time steps can be solved at a cost comparable to that of explicit steps. The multiscale computation allows the data assimilation to account for correlation at all scales, at a cost again comparable to solving the direct PDEs. Such computations can also provide full (not just initial-condition) control (which is more sensitive and accurate), yield flexible multiscale representations and fast inversion of full-matrix covariances, improve regularization (e.g., exploiting scale-dependent statistical theories), continuously fast-assimilate new observations, organize observational data in efficient hierarchical structures, and facilitate scale-dependent data types. To a fast multigrid solver of optimal control problems, feedback data can be incorporated at a cost just comparable to several explicit time steps.

Keywords: multigrid, inverse problems, data assimilation, feedback optimal control

1 Introduction

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 article we use the problems of atmospheric data assimilation and feedback optimal control to illustrate the many ways in which multiscale computation can benefit the solution of inverse PDE problems.

As will be explained below, the multiscale methods can often solve such problems for a cost comparable with the cost of solving the corresponding direct problem. Interestingly, for some ill-posed inverse problems, the overall multiscale inverse solver can cost even far less than the solver for the corresponding well-posed direct problem. This is because, whenever high-frequency solution components are ill-defined, the multiscale solver can use correspondingly lower resolution. In other words, an ill-posed problem may be less expensive to solve than a corresponding well-posed one because there is less meaningful information that can be extracted. An example is the impedance tomography multigrid solver described in [32], [17, §16.2], [33].

To understand the multiscale methods suggested below for inverse problems requires adequate familiarity with some basic concepts of nonlinear multigrid methods for direct problems. These concepts are introduced in Sections 2, 3 and 4 below. Section 5 describes the general approach to atmospheric data assimilation, followed in Section 6 by a detailed list of the many potential benefits of this approach. In Section 7 multigrid techniques for feedback optimal control are discussed.

2 Review of nonlinear multigrid solvers

The multigrid solvers for discretized PDE (see [7], [10], [36], [44], [31]) are based on two processes: (1) classical *relaxation* schemes, which are generally slow to converge but fast to smooth the error function; (2) approximating the smooth error on a *coarser grid* by solving there equations which are derived from the PDE and from the fine-grid residuals; the solution of these coarse-grid equations is obtained by using recursively the same two processes. As a result, large scale changes are effectively calculated on correspondingly coarse grids, based on information gathered from finer grids. Such multigrid solvers yield *linear complexity* (solution work proportional to the number of unknowns).

In many years of research, the range of applicability of these methods has steadily grown, to cover most major types of linear and nonlinear large systems of equations appearing in sciences and engineering. This has been accomplished by extending the concept of "smoothness" in various ways, finally to stand generally for any poorly locally determined solution component, and by correspondingly diversifying the types of coarse representations, to include for instance grid-free solvers, called *algebraic multigrid* (AMG; see [26], [27], [12], [41], [42]) non-deterministic statistical mechanics problems ([28], [15], [20], [34]) and multiple coarse-level representations ([8, §3.2], [23]).

This section reviews the main concepts of multigrid solvers. The described solver is typically used for nonlinear steady-state problems, e.g., in computational fluid dynamics (CFD). The ways of using it for time-dependent problems will be discussed in the next section.

Equations. It is assumed in this simplified review that a differential equation or a system of differential equations in the form

$$Au(x) = f(x),\ x \in \Omega \tag{1}$$

is given, where Ω is a domain in a low-dimensional (e.g., the physical 3D) space, $u(x)$ is the unknown function or a vector of unknown functions, $f(x)$ are given forcing terms, and A is a (generally nonlinear) partial differential operator. For simplicity of notation we assume that A already includes special equations describing the boundary and global conditions of the problem.

Discretization. It is further assumed that a discretization of the problem is given in the form

$$A^0 u^0(x^0) = f^0(x^0),\ x^0 \in \Omega^0\ , \tag{2}$$

where Ω^0 is a grid of points (or finite-element, or finite-volume locations) covering Ω, $u^0(x^0)$ are the discrete unknowns (e.g., $u^0(x^0)$ approximating $u(x^0)$), and A^0 is the discrete operator. For example, in the linear case A^0 is a sparse matrix. Generally, (2) will be a set of finite-difference, finite-element or finite-volume discrete nonlinear equations, approximating the differential equations and the boundary and global conditions. This is the set of equations for which a fast solution is sought.

Multigrid discretization. For the purpose of the multigrid solver, we assume that (1) is discretized not only on the given grid Ω^0, but also on a set of increasingly coarser grids $\Omega^1, \Omega^2, \ldots, \Omega^L$, where corresponding systems of equations

$$A^\ell u^\ell(x^\ell) = f^\ell(x^\ell),\ x^\ell \in \Omega^\ell,\ (1 \le \ell \le L) \tag{3}$$

have been constructed. Typically, the meshsize of Ω^ℓ is twice that of the next finer grid $\Omega^{\ell-1}$, but other situations may arise, where, for example, the meshsize is doubled in some but not all directions.

The coarsest grid Ω^L is assumed to be sufficiently coarse (typically including very few points) to allow very inexpensive direct solution of its equations (e.g., by Newton-Raphson iterations).

There are several ways in which the coarser discretizations (3) may be obtained. In many cases, the same computer routine that produces the basic discretization (2) can be re-used, with different parameters, to generate (3). Those coarse equations need not be accurate; in fact, they can be very crude, since they are later corrected (see below). There are also various *coarsening methods*, i.e., methods to derive the equations at level ℓ from those at the next-finer level $\ell-1$. These methods are usually similar to discretization (deriving (2) from (1)) methods; e.g., Galerkin-type coarsening, which is similar to Galerkin discretization.

More recently, a new type of coarsening methodology, called *Systematic Upscaling* (SU) has been advanced [18], which is more generally applicable, e.g., to strongly nonlinear and even non-deterministic problems. The SU coarse-level variables can often be some simple averages of the next-fine-level variables, but

in various problems much more sophisticated types of variables must be introduced. SU includes a systematic criterion for choosing these variables, and procedures to derive the (generally nonlinear) coarse equations.

Multigrid cycle. *A multigrid cycle for approximating u^ℓ, starting from an initial approximation u_0^ℓ*, is defined recursively by the following five steps.

1. *If $\ell = L$, solve (3) directly*, e.g., by Newton-Raphson iterations. Otherwise:

2. *Starting with u_0^ℓ, perform few (typically 1 or 2) relaxation sweeps*, producing a new approximation u_1^ℓ such that the error $v^\ell = u^\ell - u_1^\ell$ is smooth. In the case that A^ℓ is linear, this error satisfies the "residual equation"

$$A^\ell v^\ell = r^\ell \tag{4}$$

   where

$$r^\ell = F^\ell - A^\ell u_1^\ell \tag{5}$$

   is the "residual". More generally, including nonlinear problems, v^ℓ satisfies

$$A^\ell(u_1^\ell + v^\ell) - A^\ell u_1^\ell = r^\ell \ . \tag{6}$$

3. *Approximate the smooth error v^ℓ by a coarser-level function $v^{\ell+1}$.* To achieve this in the general nonlinear case, $v^{\ell+1}$ is required to satisfy, analogously to (6),

$$A^{\ell+1}(\uparrow_\ell^{\ell+1} u_1^\ell + v^{\ell+1}) - A^{\ell+1}(\uparrow_\ell^{\ell+1} u_1^\ell) = \overline{\uparrow}_\ell^{\ell+1} r^\ell \ , \tag{7}$$

   where $\uparrow_\ell^{\ell+1}$ and $\overline{\uparrow}_\ell^{\ell+1}$ are two fine-to-coarse transfer operators. For example, $(\overline{\uparrow}_\ell^{\ell+1} r^\ell)(x^{\ell+1})$ is typically an average of $r^\ell(x^\ell)$ over several points x^ℓ in the neighborhood of $x^{\ell+1}$. Actually, (7) is not solved exactly: An approximate solution $v_1^{\ell+1}$ is calculated by several (usually one or two) successive multigrid cycles (for level $\ell+1$), starting from the initial approximation $v_0^{\ell+1} = 0$.

4. Let $\uparrow_{\ell+1}^\ell v_1^{\ell+1}$ denote a function on Ω^ℓ obtained by interpolating $v_1^{\ell+1}$ from $\Omega^{\ell+1}$. This function approximates the error v^ℓ, hence an improved solution to (3) is

$$u_2^\ell = u_1^\ell + \uparrow_{\ell+1}^\ell v_1^{\ell+1} \ . \tag{8}$$

5. Further improve the solution u_2^ℓ by few (typically 1 or 2) relaxation sweeps.

Efficiency. Since the cost of calculation on increasingly coarser levels decreases geometrically, the total cost of the above multigrid cycle is just equivalent to the cost of some 4 to 6 relaxation sweeps at level ℓ. If the relaxation scheme and the intergrid transfers $\uparrow_\ell^{\ell+1}$, $\overline{\uparrow}_\ell^{\ell+1}$ and $\uparrow_{\ell+1}^\ell$ are properly chosen, each such

cycle can typically reduce the size of the algebraic error (the error in solving (3)) by at least an order of magnitude.

Full Approximation Scheme (FAS). To simplify Eq. (7) in nonlinear problems, instead of $v^{\ell+1}$ one introduces

$$u_*^{\ell+1} = \uparrow_{\ell+1}^{\ell} u_1^{\ell} + v^{\ell+1} \tag{9}$$

to serve as the coarse-level unknown function, yielding the equation

$$A^{\ell+1} u_*^{\ell+1} = f_*^{\ell+1} \ , \tag{10}$$

where

$$f_*^{\ell+1} = \overline{\uparrow}_{\ell}^{\ell+1} r^{\ell} + A^{\ell+1}(\uparrow_{\ell}^{\ell+1} u_1^{\ell}) \tag{11}$$

is known. Eq. (10) has the same form as the finer level equation (3), so the same computer routines can be used at all levels. It is thus Eq. (10) whose approximate solution is obtained at Step 3 of the algorithm, by several (1 or 2) level-$\ell+1$ multigrid cycles, starting from the initial approximation $u_{*0}^{\ell+1} = \uparrow_{\ell}^{\ell+1} u_1^{\ell}$. Having so obtained an improved approximation $u_{*1}^{\ell+1}$, one computes $v_1^{\ell+1} = u_{*1}^{\ell+1} - \uparrow_{\ell}^{\ell+1} u_1^{\ell}$ and proceeds to Step 4 as before.

Note that the new unknown $u_*^{\ell+1}$ represents on the coarse level the sum of the fine-level current approximation and its intended correction, hence the name FAS. By (9), at convergence (when $v^{\ell} = 0$ and hence $r^{\ell} = 0$ and hence also $v^{\ell+1} = 0$), $u_*^{\ell+1}$ represents the fine-level solution at the coarse level, a fact that has various applications (see below).

Assuming in (3) that the given forcing terms are discretized so that $f^{\ell+1} = \overline{\uparrow}_{\ell}^{\ell+1} f^{\ell}$, and using (5), one can rewrite (10) in the form

$$A^{\ell+1} u_*^{\ell+1} = f^{\ell+1} + \tau^{\ell+1} \ , \tag{12}$$

where

$$\tau^{\ell+1} = A^{\ell+1}(\uparrow_{\ell}^{\ell+1} u_1^{\ell}) - \overline{\uparrow}_{\ell}^{\ell+1}(A^{\ell} u_1^{\ell}) \ . \tag{13}$$

Note that Eq. (12) is the same as Eq. (3) for $u^{\ell+1}$, except for the additional term $\tau^{\ell+1}$. Adding this term replaces the original coarse-level solution $u^{\ell+1}$ by $u_*^{\ell+1}$, which has (at convergence) the accuracy of the finer level ℓ; hence $\tau^{\ell+1}$ is called *the fine-to-coarse defect correction.* Due to such corrections, the coarse operators need not be very accurate.

The $\tau^{\ell+1}$ correction has many applications: It approximates the local discretization error, hence can be used to decide where the discretization is not fine enough (see Sec. 3 below). In various situations where a problem needs to be re-solved many times, at all or at many parts of the problem domain $\tau^{\ell+1}$ need not be updated each time, allowing to avoid re-processing the fine level at those parts (see Secs. 4 and 7 below).

Full Multigrid (FMG) algorithm is defined recursively as the algorithm that solves Eq. (3) at level ℓ by n_c multigrid cycles, where the initial approximation is obtained by interpolation from a solution that has been previously

calculated, by the same FMG algorithm, at the next coarser level $(\ell+1)$. Usually, one cycle ($n_c = 1$) at each level is enough to reach an error (in solving Eq. (2)) smaller than the discretization error (the difference between the solutions of (2) and (1)). In the case of high-order discretizations, the number of cycles required per level to attain the high-order accuracy increases proportionately to that order, but is always independent of the number of points in Ω^0. The total cost of the algorithm, to attain second-order accuracy for example, is normally less than the cost of 10 relaxation sweeps at the finest ($\ell = 0$) level. Moreover, for essentially just the same cost, the FMG algorithm can incorporate processes such as continuation (see Item 5 in Sec. 6 below), inverse-problem solving (see Item 4 there) and grid adaptation (see Sec. 3 below), avoiding the need to re-solve a problem many times over.

Denote by u_f^ℓ the *final* level-ℓ solution in the FMG algorithm (before starting the level-$(\ell-1)$ cycles). A suitable cycle number n_c can inexpensively be chosen at the coarse FMG levels by checking that increasing it does not significantly reduce the differences $\|u_f^\ell - \uparrow_{\ell-1}^{\ell} u_f^{\ell-1}\|$. The rate of decrease of this full sequence of differences ($\ell = L, L-1, \ldots, 1$) gives a practical estimate for the effective discretization order, from which the discretization error in the target numerical solution u_f^0 can also be assessed.

A road map for CFD. Further techniques for efficient multigrid treatment of flow problems have been summarized in a detailed table called "Barriers to Achieving Textbook Multigrid Efficiency in CFD" [16]. 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.

Included in the table are staggered and nonstaggered, conservative and nonconservative 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 which are not visible on some of the coarse grids); large grid aspect ratios; boundary layer resolution; and grid adaptation.

3 Multigrid solver with grid adaptation

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 [7, §§7–9], [10, §9], [11, §9] or [3], so we only summarize the main points.

Local refinements are created by adding local *patches* of a finer grid 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 the *fine-to-coarse defect correction* $\tau^{\ell+1}$ to the equations of its parent (coarse) grid, thereby enforcing the fine-grid accuracy over that patch. This is a natural part of the FAS multigrid solver which therefore 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 effectively 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 some 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 $\tau^{\ell+1}$ 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 FMG solver: as the latter proceeds to increasingly finer levels, it can also decides (using these adaptation criteria) *where* those finer levels should be set, 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 (particularly fine cross-stream meshsizes). In geophysical calculation, while the overall coordinate system over the entire globe will necessarily harbor polar singularities, the local patches can each entertain its own nearly flat system.

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 brief "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.

PDE solvers on unbounded domains. As pointed out in [7, §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 in the grid adaptation described above. Using general grid optimization equations (see [7, §8.1] or [10, §9.5] or [11, §9.5]), one can calculate how far out one must use a certain meshsize to maintain a certain accuracy. Employing a suitable version of the λ-FMG

algorithm [11, §9.6], it has been shown for various problems 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.

4 Time-Dependent Problems

In the numerical solution of time-dependent problems, to allow large time steps and/or fully adaptable discretization, *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 [22].

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 $\tau^{\ell+1}$ is carried from each such visit to subsequent time steps [35], [37].

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 [9, §3.10], [13, §11] and [45]).

To achieve that, time is treated just as another space coordinate, and the whole problem is solved by the FMG algorithm, starting with a grid which is coarse *in both space and time*, proceeding to finer levels, with one to 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. 5 below).

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. 3.

5 Atmospheric data assimilation

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.

Multiscale methods can potentially avoid all these compromises, and assimilate the observational data at a cost just comparable to that of a fast multigrid solver of the direct flow equations.

This is considered in principle possible because: (1) The processes of data assimilation can be incorporated, level by level, into the multigrid solver of the direct problem. (2) Large scale averages can inexpensively be assimilated on the correspondingly coarse levels of the multigrid solver (coarse in both space and time). (3) Deviations from any large-scale average must be assimilated on some finer scale, but *their* correlation on that scale is local. (4) The measurements

(with their representativeness errors) are generally less accurate and in most regions less resolved than the numerical flow itself, hence their assimilation need not be done at the finest numerical level.

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 inverse problem (but possibly with the storage-saving windowing described below). This algorithm is like a usual FMG solver for the *direct* 4D atmospheric equations, except that at each stage, on each level, the relaxation of the solution (the flow) variables $u(x)$ is accompanied by a relaxation of the *control variables* $\sigma(x)$ at that level (for an example of the nature of $\sigma(x)$, see Sec. 6 below). The FAS representation (see Sec. 2) of both the solution and the control allows relaxing each of them at any desired level. As a result, in essence, large-scale averages of the solution are controlled on correspondingly coarse grids (coarse in both space and time), while a distributive relaxation scheme for the control (see below) ensures at each level local assimilation of local deviations.

For this purpose the observational data should be transferred to all levels of the algorithm. Using the FAS interpolation formulae, each observational relation at each level can directly be expressed at the next coarser level. Wherever there are too many such relations per coarse flow variable, their number can be reduced by suitable averaging. Note that this simple procedure would not exactly yield the "best" assimilation in any pre-defined norm, but such exact definitions of "best" are arbitrary anyway, and typically counterproductive (cf. [29, §13]).

Localized relaxation of the control variables $\sigma(x)$ (such as those described in Sec. 6 below) is obtained by *distributive relaxation*, i.e., each relaxation step simultaneously changes several neighboring values of σ. Namely, the k-th step has the form

$$\sigma(x) \leftarrow \sigma(x) + \rho_k(x) \cdot \delta_k \ , \tag{14}$$

where the support of each $\rho_k(x)$ is a small neighborhood that shifts from step to step. *The shape* $\rho_k(x)$ is designed so that the effect of the control change (14) is essentially local, i.e., it entails a solution change of the form

$$u(x) \leftarrow u(x) + v_k(x) \cdot \delta_k \ , \tag{15}$$

where to a good approximation $v_k(x)$ has also a local support, and can therefore be calculated by several local relaxation steps. Such approximate localness of the effect is usually obtained simply by designing each $\rho_k(x)$ to have several vanishing moments, e.g.,

$$\sum_x \rho_k(x) = 0 \ \text{ and } \ \sum_x x \cdot \rho_k(x) = 0 \ . \tag{16}$$

The amplitude δ_k of the control change (14) can then be calculated by requiring that the local solution change (15) best fits the observational data.

Note that due to restrictions like (15), the changes (14) do not span the entire space of possible changes; however, they are exactly complemented by changes to the control made at coarser levels. It is this kind of localized nature of relaxation at each level that allows good assessment of correction amplitudes and hence fast convergence of the multilevel inverse-problem solver.

The levels at which the control should be adjusted depends on the local density of the measurements, their accuracy and their distance from regions where details of the solution are of interest.

Time 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. Using FAS, the structure for this is similar to the local refinement techniques (see Sec. 3), but successively shifted in time. 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 more advanced window (shifted forward in time, but partly overlapping its predecessor) and so on. At the coarser levels, 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 current 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 fine-to-coarse (τ) corrections, thus still maintaining the fine-level accuracy of coarse-level features (see Sec. 2).

Spatial refinements. 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.).

6 Multiple benefits of multiscale techniques

There are *many* different ways in which multiscale computational methods can contribute to data assimilation problems (and similarly to other inverse problems). The following list of benefits may serve 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 nonlinear multigrid solvers make it possible to use implicit-time-step discretizations at full efficiency (see the general approach to time dependent problems in Sec. 4 above, and multigrid methods for shallow water and three-dimensional atmospheric models in [6], [5], [4], [46], [47], [38], [39] and [40]). 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) [4]. The unconditional stability is particularly 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. The multiscale environment simultaneously provides convenient flexible structures, refinement criteria and one-shot self-adaptive solvers; cf. Secs. 3 and 4.

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. 4 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, the reason being that, while the first derivatives of the data fitness are calculated, the second derivatives (the Jacobian) are not, hence the size of the needed corrections cannot be calculated. In multigrid solvers, by contrast, one can integrate the adjustment of the inverse parameters (the control) into the appropriate stages of only *one* direct-problem solver. This solver converges fast, since, due to the appropriate scale separation obtained by the distributive relaxation (see Sec. 5 above), a good approximate amplitude (δ_k) of the local correction can be calculated, locally at each scale. (This general approach has been described in [10, §13], with more details in [13, §8.2] and full development in [43], [1], [2], [32], [33]).

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 [10, §8.3.2]. 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 times 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 $\sigma(x,t)$ is a vector-valued grid function that at each point x and time t gives the deviations in satisfying the set of flow equations. The objective function (the error functional that should be minimized) has the general form

$$E = \boldsymbol{\sigma}^T S \boldsymbol{\sigma} + \mathbf{d}^T W \mathbf{d}, \tag{17}$$

where $\boldsymbol{\sigma} = \sigma(x,t)$ is the vector of all control parameters, $\mathbf{d} = \big(d(y)\big)$ 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,t) 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 at each time 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 above) 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 [32], [21].

So extensive control parameters can only be handled by a multiscale treatment, like the one described above. Moreover, using the described methods, the solution is expected not to be very expensive, especially since the control parameters $\sigma(x,t)$ need not be controlled at the finest computational levels; on such levels $\sigma(x,t)$ 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 inexpensive 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 (17), 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 [24], [14], [17, §10 and §14.3] and in [30] 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 some p-order derivatives of $G(x,y)$ decay with the distance from y to x). 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, for example, it may be enough to assume smoothness for variations in only some directions.

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 *high frequency 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. 5). 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 reasons, 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, called quadtrees in 2D or oct-trees in 3D, 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 multigrid methods for wave equations in [23] and [17, §7].)

7 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)

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

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 for optimality 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 $J(x,u) = \int (|\mathcal{C}x(t)|^2 + |u(t)|^2)dt$, 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*

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

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 multiscale 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 FAS-FMG solver for the nonlinear (non-LQR) problem. In fact, for various actual problems, it can share many of the potential benefits listed in Sec. 6 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, or the sense of non-localness, 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 (τ) corrections being frozen there (cf. [10, §15].) More precisely, at increasingly later times, any given multigrid level (a given timestep and, when relevant, spatial meshsize) needs to 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 currently 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).

Tests. 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 ($n_c = 2$) 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 indeed be frozen, so that re-solving can be restricted to coarse levels, as expected.

Acknowledgment. The research is supported by the Israel Science Foundation grant 295/01 and by the US Air Force Office of Scientific Research, contract F33615-03-D5408.

References

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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

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

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

[11] 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.

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

[13] 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.

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

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

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

[17] Brandt, A., Multiscale scientific computation: review 2001. In Barth, T.J., Chan, T.F. and Haimes, R. (eds.): *Multiscale and Multiresolution Methods: Theory and Applications*, Springer Verlag, Heidelberg, 2001, pp. 1–96. Available in www.wisdom.weizmann.ac.il/∼achi/review00.ps.

[18] Brandt, A., Multiscale solvers and systematic upscaling in computational physics, *Proc. Conf. on Computational Physics 2004*, September 2004, Genova, Italy. Computer Physics Communication, **169** (2005) 438–441.

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

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

[21] Brandt, A. and Gandlin, R., Multigrid for Atmospheric Data Assimilation Analysis. In: Hyperbolic Problems: Theory, Numerics, Applications (T.Y. Hou and E. Tadmor, eds.), Springer 2003, pp. 369–376.

[22] 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.

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

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Brandt, A. and Ron, D., Multigrid solvers and Multilevel Optimization Strategies. In: *Multilevel Optimization and VLSICAD*, (J. Cong and J.R. Shinnerl, eds.), Kluwer Academic Publishers, Boston, 2003, pp. 1–69.

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

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

[32] Gandlin, R., *Multigrid Solvers for Inverse Problems*, Ph.D. Thesis, The Weizmann Institute of Science, 2003.

[33] Gandlin, R. and Brandt, A., Two multigrid algorithms for an inverse problem in electrical impedence tomography, Proc. 2003 Copper Mountain Conf. Multigrid Methods, *Num. Lin. Alg. with Appl.*, submitted.

[34] Goodman, J. and Sokal, A.D., Multigrid Monte Carlo methods for lattice field theories. *Phys. Rev. Lett.*, 56 (1986) 1015–1018.

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

[36] Hackbusch, W., Multigrid Methods and Applications. Springer, Berlin, 1985.

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

[38] 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.

[39] 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.

[40] 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.

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

[42] Stüben, K., Algebraic multigrid (AMG): An introduction with applications, Guest appendix in Trottenberg, U., Oosterlee, C. W. and Schüller, A., *Multigrid*, Academic Press, London, 2000. A review of algebraic multigrid, *J. Comp. Appl. Math.* **128**(1–2), 2001.

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

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

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

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

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