DYNAMIC DATA-DRIVEN INVERSION FOR TERASCALE SIMULATIONS: REAL-TIME IDENTIFICATION OF AIRBORNE CONTAMINANTS*

VOLKAN AKÇELIK†, GEORGE BIROS‡, ANDREI DRĂGĂNESCU§
JUDITH HILL¶, OMAR GHATTAS‖, AND BART VAN BLOEMEN WAANDERS**

Abstract. In contrast to traditional terascale simulations that have known, fixed data inputs, *dynamic data-driven* (DDD) applications are characterized by unknown data and informed by dynamic observations. DDD simulations give rise to inverse problems of determining unknown data from sparse observations. The main difficulty is that the optimality system is a boundary value problem in 4D space-time, even though the forward simulation is an initial value problem. We construct special-purpose parallel multigrid algorithms that exploit the spectral structure of the inverse operator. Experiments on problems of localizing airborne contaminant release from sparse observations in a regional atmospheric transport model demonstrate that 17-million-parameter inversion can be effected at a cost of just 18 forward simulations with high parallel efficiency. On 1024 Alphaserver EV68 processors, the turnaround time is just 29 minutes. Moreover, inverse problems with 135 million parameters — corresponding to 139 billion total space-time unknowns — are solved in less than 5 hours on the same number of processors. These results suggest that ultra-high resolution data-driven inversion can be carried out sufficiently rapidly for simulation-based "real-time" hazard assessment.

1. Introduction. Traditionally, terascale supercomputers have been employed for simulations of complex physical systems that are based on static, known data. Typically, the behavior of the physical system is modeled by partial differential equations (PDEs), and the data comprise boundary conditions, initial conditions, sources, geometry, and material coefficients. Simulations are carried out to study the behavior of the system for the given data. In this realm of *static data-driven* simulations, absolute turn-around time is often subordinate to considerations of desired accuracy and resolution.

Recently, a new class of *dynamic data-driven* (DDD) applications — requiring the highest levels of supercomputing performance — has emerged [9, 10]. These applications are characterized by uncertain or unknown data, and are informed by observations or measurements that become available dynamically. DDD simulations appear in such scenarios as hazard assessment, emergency response, treaty verification, structural health monitoring, image-driven surgery, weather forecasting, geophysical exploration, and closed-loop process control, to name just a few. All of these applications share the challenge of reconstructing unknown input data from sparse, dynamically-obtained measurements — and the need to issue

*Supported in part by the Computer Science Research Institute at Sandia National Laboratories, the U.S. Department of Energy under the SciDAC Terascale Optimal PDE Simulations (TOPS) Center and grant DE-FG02-04ER25646, and the National Science Foundation under ITR grants ACI-0121667 and EAR-0326449. Computing resources at the Pittsburgh Supercomputing Center provided under NSF TeraGrid award MCA04N026P.

†Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX USA (volkan@cs.cmu.edu)

‡Departments of Mechanical Engineering & Applied Mechanics and Computer & Information Science, University of Pennsylvania, Philadelphia, PA USA (biros@seas.upenn.edu)

§Optimization & Uncertainty Estimation Department, Sandia National Laboratories, Albuquerque, NM USA (aidraga@sandia.gov)

¶Optimization & Uncertainty Estimation Department, Sandia National Laboratories, Albuquerque, NM USA (jhill@andrew.cmu.edu)

‖Institute for Computational Engineering and Sciences, and Departments of Geological Sciences, Mechanical Engineering, Computer Science, and Biomedical Engineering, University of Texas at Austin, Austin, TX USA (omar@ices.utexas.edu)

**Optimization & Uncertainty Estimation Department, Sandia National Laboratories, Albuquerque, NM USA (bartv@cs.sandia.gov)

©2005 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government t purposes only.

SC|05 November 12–18, 2005, Seattle, Washington, USA ©2005 ACM 1-59593-061-2/05/0011 $5.00

predictions rapidly.

In general, DDD applications can be cast as *inverse problems*, in which the goal is to reconstruct the missing, uncertain, unknown, or errant data from sparse observations and measurements over a finite time interval. The reconstruction must be *model-based*: that is, the reconstructed input data must be mapped to the observed measurements in a manner that is consistent with the PDE *forward* simulation model. Once an estimate of the input data has been constructed, it can be used to initialize forward simulations that predict future system behavior over an appropriate time horizon. The "observe–invert–predict" cycle is then repeated for the next time interval of observations, and so on.

Rapid turn-around time is paramount for many DDD simulations. Historically, this meant that simulation accuracy and resolution were sacrificed for speed. Where rapid prediction and response are mandated, the supporting simulations have had to revert from high-resolution, high-fidelity three-dimensional PDE models back to simplified models such as lookup tables, algebraic models, or one- or two-dimensional PDEs.

However, in recent years it has become meaningful — and in many cases imperative — to contemplate DDD simulations of complex physical systems that are *both* rapid and highly-resolved. This has been motivated by advances in sensing technologies, deployment of very high bandwidth networks, and the availability of terascale supercomputers. To capitalize on this emerging infrastructure, a central challenge facing computational scientists is the construction of robust scalable parallel algorithms for near-real time solution of the underlying data-driven inverse problems. Unfortunately, inverse problems are often much more difficult to solve than corresponding forward simulations, because inverse problems

- usually require *numerous* repeated forward simulations;
- are usually *ill-posed* despite the well-posedness of the forward problem; and
- are *boundary value problems in four-dimensional space-time*, despite the initial-value, time-marching character of the forward problem.

Nevertheless, there is a pervasive need in many application areas for *near-real time, high-fidelity, dynamic data-driven inversion*. The development of scalable parallel algorithms for this task is the goal of our paper.

Although our approach is general and widely applicable, we have chosen a specific driving application to instantiate and evaluate our algorithms and implementation. We focus on the localization of airborne contaminant releases in regional atmospheric transport models from sparse observations [2], in time scales short enough for predictions to be useful for hazard assessment, mitigation, and evacuation procedures. In particular, our goal is model-based rapid reconstruction — via solution of a large-scale inverse problem — of the unknown initial concentration of the airborne contaminant in a convection-diffusion transport model, from limited-time spatially-discrete measurements of the contaminant concentration, and from a velocity field as predicted, for example, by a mesoscopic weather model. Mathematically, transport of the contaminant is described by the convection-diffusion equation

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u + \boldsymbol{v} \cdot \nabla u = 0 \quad &\text{in} \quad \Omega \times (0,T), \\ \nu \nabla u \cdot \boldsymbol{n} = 0 \quad &\text{on} \quad \Gamma \times (0,T), \\ u = u_0 \quad &\text{in} \quad \Omega \times \{t = 0\}, \end{aligned} \tag{1.1}$$

where $u(\boldsymbol{x}, t)$ is the contaminant concentration field, $u_0(\boldsymbol{x})$ is the initial concentration that, together with the velocity field $\boldsymbol{v}(\boldsymbol{x}, t)$, drives the system, and ν is the diffusion coefficient. We seek to reconstruct the initial concentration u_0 from measurements of the concentration u over a short time horizon, taken at a small number of sensor locations throughout the domain.

Then — using the just-reconstructed initial concentration — we can issue predictions of the longer-time transport of the contaminant plume throughout the region.

Inverse problems for convective-diffusive transport of this type arise in several settings: characterization of pollutants in the atmosphere, unintentional catastrophic accidents involving (for example) chemical plants, or intentional releases of hazardous chemical or biological agents. Although studies have been conducted of the sensitivity of chemical concentration with respect to source terms or observer placement, [8, 16, 18, 19], very little work has been done on reconstruction of initial concentrations via solution of an inverse problem.

In Section 2 we formulate the inverse problem as an output least squares optimization problem with a convection-diffusion PDE constraint. First order optimality conditions produce a coupled system of partial differential-algebraic equations, which includes the initial value convection-diffusion PDE, the *terminal-value* adjoint convection-diffusion PDE, and an algebraic equation for the initial concentration. As mentioned above, this system is an ill-posed boundary value problem in 4D space-time, and typical problem sizes of interest present a significant challenge for rapid solution. To overcome the four-dimensionality of this system, we invoke a block elimination that reduces the system to one in just the (3D) spatially-discretized initial concentration variable u_0. Unfortunately, the operator for this reduced system is non-local and cannot even be formed for the problems we target, even with petascale computing resources. Fortunately, the action of the reduced operator on a vector can be formed by solving a pair of forward/adjoint convection-diffusion PDEs, and the spectral character of the reduced operator (as for many inverse problems) guarantees that a Krylov method applied to this system converges in a mesh-independent number of iterations.

However, a constant number of iterations independent of mesh size is by itself not sufficient for real-time dynamic data-driven applications: *the constant itself must be reduced* so that no more than a few iterations — and hence forward/adjoint simulations — are needed to solve the inverse problem. This requires a scalable and effective preconditioner, which is particularly challenging because the inverse operator is never formed. In Section 3 we present a parallel multigrid preconditioner designed to reduce the number of Krylov iterations for DDD inverse problems. Unlike PDE operators, inverse operators are compact, and their spectral properties are different from those of differential operators. As a result, standard multigrid smoothers are not applicable for inverse operators. Instead, special-purpose smoothers that are tailored to their spectral properties must be employed, and these are presented in Section 3. Section 4 provides a prototype inversion scenario: localization of the release of a contaminant in the Los Angeles harbor from short-term measurements of its transport by onshore winds, followed by longer-term prediction of the transport of the contaminant throughout the Greater LA Basin. We also provide results on the performance and scalability of our inversion algorithm. Our results demonstrate that due to high parallel and algorithmic efficiency, inverse problems with 17 million initial concentration unknowns, and 8.7 billion total space-time unknowns, can be solved in less than 30 minutes on 1024 processors of an Alphaserver EV68-based system. The time taken is just 18 times that of a *single* forward transport simulation. Moreover, inverse problems with 135 million initial concentration parameters — and 139 billion total space-time unknowns — are solved in less than 5 hours on the same number of processors.

Ultimately, our results demonstrate that — with careful attention to the design of scalable parallel algorithms — high-resolution inverse transport problems can be solved in "real time," i.e. in time scales feasible for simulation-based hazard assessment and response. More generally, for DDD inverse problems characterized by other classes of forward simulations, the turn-around time will, of course, depend on the complexity of the forward simulation. However, *our results indicate that model-based reconstruction of incomplete initial condi-*

tions — which is necessary for data-driven prediction — can be achieved in a small multiple of the cost of the forward simulation, even when millions of uncertain parameters must be estimated.

2. Formulation and optimality conditions. In this section we give details on the mathematical formulation of the inverse problem, its discretization, and the overall strategy for its numerical solution.

Given observations of the concentration $\{u_j^*\}_{j=1}^{N_s}$ at N_s locations $\{x_j\}_{j=1}^{N_s}$ inside a domain Ω, we wish to estimate the initial concentration $u_0(\boldsymbol{x})$ that leads to the closest reproduction of the observed concentrations using the forward transport PDE. The inverse problem is formulated as a constrained, least squares optimization problem:

$$\begin{aligned} \min_{u,u_0} \mathcal{J}(u,u_0) &\stackrel{\text{def}}{=} \frac{1}{2}\sum_{j=1}^{N_s}\int_0^T\int_\Omega (u-u^*)^2\,\delta(\boldsymbol{x}-\boldsymbol{x}_j)\,\boldsymbol{dx}\,dt + \frac{\beta}{2}\int_\Omega u_0^2\,\boldsymbol{dx}, \\ \text{subject to} \quad & \frac{\partial u}{\partial t} - \nu\Delta u + \boldsymbol{v}\cdot\nabla u = 0 \quad \text{in} \quad \Omega\times(0,T), \\ & \nu\nabla u\cdot\boldsymbol{n} = 0 \quad \text{on} \quad \Gamma\times(0,T), \\ & u = u_0 \quad \text{in} \quad \Omega\times\{t=0\}. \end{aligned} \tag{2.1}$$

The first term in the objective functional $\mathcal{J}$ represents a least-squares misfit of predicted concentrations $u(\boldsymbol{x}_j)$ with observed concentrations $u^*(\boldsymbol{x}_j)$ (the delta function localizes the u and u^* fields to the sensor locations). The second term in $\mathcal{J}$, scaled by the constant $\beta/2$, is a regularization term that is needed to yield a well-posed problem. Inverting for the initial concentration u_0 would be well-posed — and the regularization term would not be needed — if observations were available everywhere in space and time, that is if u^* were known for all $(x,t)\in\Omega\times(0,T)$.[1] In practice, however, we can expect to have just limited measurements in space and perhaps also in time. This sparsity of measurements induces the ill-posedness of the inverse problem. In this case, we cannot hope to recover components of the initial concentration that are much more oscillatory than dictated by the spacing of the sensors. Therefore, oscillatory components of u_0 lie in the null space of the inverse operator, and — in the absence of regularization — will appear as arbitrary noise in the reconstructed initial concentration field. To address this source of ill-posedness, we employ $L^2(\Omega)$ regularization, which penalizes the L^2 norm of u_0.

The constraints in the optimization problem (2.1) are just the contaminant transport convection-diffusion equation, boundary condition, and initial condition. The transport of the pollutant is driven by the initial conditions, the diffusion, and the velocity field. In practice, the velocity field would be provided by a regional numerical weather prediction model such as MM5 [20]. For our present purposes, however, we are interested in assessing the real-time viability and algorithmic and parallel scalability of our inversion method. For simplicity we employ a steady laminar incompressible Navier-Stokes solver to generate wind velocity fields over a terrain of interest.

The inverse problem then is to determine the initial concentration field $u_0(\boldsymbol{x})$, and the resulting space-time evolution of the concentration $u(\boldsymbol{x},t)$, by solving the optimization problem (2.1). First-order necessary conditions for optimality — the so-called *KKT conditions* —

[1] As opposed to inverse problems for which just final time observations are available; any noise in the data is thus exponentially amplified since it is convolved with the inverse of the heat kernel.

may be derived by introducing a Lagrangian functional:

$$\begin{aligned}\mathcal{L}(u,u_0,p) &\stackrel{\text{def}}{=} \frac{1}{2}\sum_{j=1}^{N_s}\int_0^T\int_\Omega (u-u^*)^2\,\delta(\boldsymbol{x}-\boldsymbol{x}_j)\,\boldsymbol{dx}\,dt + \frac{\beta}{2}\int_\Omega u_0^2\,\boldsymbol{dx} \\ &+ \int_0^T\int_\Omega \left(p\frac{\partial u}{\partial t} + \nu\nabla u\cdot\nabla p + p\boldsymbol{v}\cdot\nabla u\right)\boldsymbol{dx}\,dt + \int_\Omega p(u-u_0)\,\boldsymbol{dx},\end{aligned} \tag{2.2}$$

in which the *adjoint concentration* $p(\boldsymbol{x},t)$ is used to enforce the convection-diffusion equation and initial condition. Requiring stationarity of the Lagrangian $\mathcal{L}$ with respect to p, u, and u_0 (respectively) yields the KKT conditions, which consist of:

The forward convection-diffusion problem

$$\begin{aligned}\frac{\partial u}{\partial t} - \nu\Delta u + \boldsymbol{v}\cdot\nabla u = 0 \quad &\text{in} \quad \Omega\times(0,T), \\ \nu\nabla u\cdot\boldsymbol{n} = 0 \quad &\text{on} \quad \Gamma\times(0,T), \\ u = u_0 \quad &\text{in} \quad \Omega\times\{t=0\}.\end{aligned} \tag{2.3}$$

The adjoint convection-diffusion problem

$$\begin{aligned}-\frac{\partial p}{\partial t} - \nu\Delta p - \nabla\cdot(p\boldsymbol{v}) = -\sum_{j=1}^{N_s}(u-u^*)\delta(\boldsymbol{x}-\boldsymbol{x}_j), \quad &\text{in} \quad \Omega\times(0,T) \\ (\nu\,\nabla p + \boldsymbol{v}p)\cdot\boldsymbol{n} = 0, \quad &\text{on} \quad \Gamma\times(0,T), \\ p = 0 \quad &\text{in} \quad \Omega\times\{t=T\}.\end{aligned} \tag{2.4}$$

The initial concentration equation

$$\beta\,u_0 - p|_{t=0} = 0 \quad \text{in} \quad \Omega. \tag{2.5}$$

Equations (2.3) are just the original forward convection-diffusion transport problem for the contaminant field. The adjoint convection-diffusion problem (2.4) resembles the forward problem, but with some essential differences. First, it is a terminal value problem; that is, the adjoint p is specified at the final time $t = T$. Second, convection is directed backward along the streamlines. Third, it is driven by a source term given by the negative of the misfit between predicted and measured concentrations at sensor locations. Finally, the initial concentration equation (2.5) is in the present case of L^2 regularization an algebraic equation. Together, (2.3), (2.4), and (2.5) furnish a coupled system of linear PDEs for (u,p,u_0). The principal difficulty in solving this system is that — while the forward and adjoint transport problems are parabolic-hyperbolic problems — the KKT optimality system is a *coupled boundary value problem in 4D space-time.*

To simplify discussion of solution approaches, we introduce operators A, T, B and R. Here, A denotes the forward transport operator and A^{-1} its inverse; T extends a spatial field at initial time into space-time; B is an observation operator that localizes space-time to points at which sensors are placed; R is the regularization operator (in the present case the identity); A^* is the adjoint transport operator and A^{-*} its inverse; and T^* restricts a space-time field to a spatial field at $t = 0$. With these definitions, we can write the KKT conditions in operator

form:

$$\begin{bmatrix} B & 0 & A^* \\ 0 & \beta R & -T^* \\ A & -T & 0 \end{bmatrix} \begin{bmatrix} u \\ u_0 \\ p \end{bmatrix} = \begin{bmatrix} Bu^* \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

Special-purpose Krylov solvers and parallel preconditioners can be very effective at solving discretized versions of optimality systems such as (2.6) for optimization problems constrained by *steady-state* PDEs [4, 5]. Here, however, the 4D space-time nature of (2.6) presents prohibitive memory requirements for large scale problems. Solution of (2.6) in its *full-space* form is essentially intractable for such problems using present computing resources. Instead, we pursue a *reduced-space* method that amounts to a block elimination combined with a matrix-free Schur complement solver.

Eliminating the concentration u and forward transport equation (third row of (2.6)) and adjoint concentration p and adjoint transport equation (first row of (2.6)) from the KKT optimality system, we obtain the Schur complement system for the initial concentration u_0:

$$Hu_0 = g, \tag{2.7}$$

where the *reduced Hessian* (or inverse) operator H is defined by

$$H \stackrel{\text{def}}{=} T^* A^{-*} B A^{-1} T + \beta R, \tag{2.8}$$

and the *reduced gradient* g is defined by

$$g \stackrel{\text{def}}{=} T^* A^{-*} B u^*.$$

It is immediately clear that H is a symmetric and strictly positive definite operator. Thus, the optimization problem has a unique solution for non-vanishing β.

Notice that H is a non-local operator (when discretized it will be a full matrix) and its explicit construction is completely out of the question. For example, for the problem with 139 billion space-time unknowns solved in Section 4, H is of dimension 135×10^6 by 135×10^6. Thus, storing H would require about 10^{23} bytes of memory. Moreover, forming H would require 135 million solutions of forward convection-diffusion transport equations; on the 1024 processor Alpha system we used for the numerical experiments of Section 4, this would require over 400 years of computing time. While explicit formation of H and its singular value decomposition constitutes an attractive and popular approach for small-scale inverse problems [14], alternative approaches are essential for large-scale problems, and in particular those for which near real-time response is mandated.

Therefore, we opt to solve (the discretized form of) (2.7) using the preconditioned Conjugate Gradient (CG) method. H is never formed explicitly; instead, we compute $w = Hv$, the action of the reduced Hessian on a given spatial field v, in matrix-free fashion as follows. (i) Set $u_0 = v$ and solve the forward transport equation (2.3) to obtain the concentration evolution u. (ii) Compute the misfit between measurements u^* and predicted concentration u at the sensor locations, and use this misfit as a source to solve the adjoint transport equation (2.4) backward in time to obtain $p|_{t=0}$, the adjoint at $t = 0$. (iii) Set $w = \beta v - p|_{t=0}$. Therefore, each application of the reduced Hessian requires two transport equation solutions, one forward in time and one backward. Besides u_0, we need to store the entire time history of u (if we have measurements over the entire time interval $(0, T)$) but *only* at the sensor locations. In contrast with full-space methods, we avoid storing the forward and adjoint concentration time histories. Thus, memory requirements for the inverse problem (2.1) are similar to those of the forward problem (1.1).

The overall computational cost of the (unpreconditioned) CG method is the work per iteration — dominated by the two transport equations solutions — multiplied by the number of CG iterations. The latter depends on the condition number of the reduced Hessian. One can show that, for fixed β, H is a compact perturbation of the identity and thus has a bounded and hence mesh-independent, condition number [11]. Furthermore, its spectrum has a small number of clusters; it collapses exponentially onto β. Therefore, if we use a Krylov method, such as CG, to solve (2.7), the number of iterations for a specific relative residual reduction will also be mesh-independent. This is an optimal number of iterations, and we have verified it numerically (see Section 4). Thus, mesh-independent convergence comes for free (but the constant deteriorates with β). The constant also depends on the Peclet number, the length of the time horizon, the complexity of the velocity field, and the topography.

However, mesh-independence of CG iterations is by itself *not sufficient for DDD problems requiring real-time inversion*. Although algorithmically optimal, the number of unpreconditioned CG iterations is often so large that the cost of solving the inverse problem is equivalent to many tens to hundreds of forward transport solutions, which precludes the use of high-resolution models in the real-time setting. Our goal therefore is to reduce the absolute number of iterations so that the cost is equivalent to a handful of forward transport solves. To achieve this goal, we must *reduce the constant in the complexity estimate.* The immediate idea is to precondition the reduced Hessian system to further decrease the number of CG iterations and, most importantly, reduce the overall wall-clock time. One challenge in constructing suitable preconditioners for the reduced Hessian is the impossibility of explicitly forming this operator for reasons stated above. For this reason — and due to their demonstrated success as preconditioners for second-kind integral operators [12, 17], we pursue multigrid preconditioners. Details are given in the next section.

3. Multigrid Preconditioner. Multigrid methods have revolutionized scientific computing, especially for linear systems related to elliptic and parabolic partial differential equations. Such multigrid schemes, however, are not directly applicable to reduced Hessian operators for inverse problems. For this reason, there has been recent interest in developing specific multigrid-like methods for inverse problems (for example see [11, 13, 15, 17, 21]). The main difficulty lies in constructing a proper smoothing operator for the reduced Hessian operator H.

For our problem, the continuous reduced Hessian operator is spectrally equivalent to a Fredholm integral operator of the second kind. Multigrid solvers for such problems have been very successful [12]. The overall algorithm follows the standard multigrid hierarchy: pre-smoothing, restriction to a coarser grid, solution, prolongation back to the fine grid, correction, and post-smoothing. The key aspect is the smoother, which must address the spectrum of the reduced Hessian.

Classical solvers such as Jacobi and Gauss-Seidel work well as smoothers for elliptic PDE operators, where the large eigenvalues of the differential operator correspond to high frequency eigenvectors (see [6]). These methods rapidly eliminate oscillatory components of the numerical error, but are notoriously slow at eliminating the smooth components. The multigrid method can be used to effectively eliminate the smooth error components by iterating on coarser scales.

Unlike elliptic operators, however, the continuous reduced Hessian is a strongly smoothing, compact, and nonlocal operator (hence its discrete version H is represented by a dense matrix). Its eigenvector–eigenvalue correspondence is reversed, with large eigenvalues associated with smooth eigenvectors, and small eigenvalues associated with oscillatory eigenvectors. Neither Krylov-subspace methods, nor stationary methods such as Jacobi and Gauss-Seidel, act as smoothers for H; in fact, in addition to being expensive to apply, they act more

as roughers, since the "high energy" (large eigenvalue) components, which correspond to smooth eigenvectors, are typically resolved first, leaving oscillatory components in the error.

The smoothing properties of the continuous reduced Hessian, combined with its approximation of the discrete counterpart, imply that by decreasing the mesh size, both H and the "high energy" eigenvectors are increasingly well represented on the immediately coarser grid. For clarity we denote by H_h the discrete reduced Hessian at resolution h. If we assume that the "coarse" space V_{2h} is embedded into the "fine" space V_h (as is often the case with finite element discretizations), and we denote by $P_h : V_h \to V_{2h}$ the L^2-orthogonal projection, then the action of the reduced Hessian H_h on the "coarse" space is well approximated by H_{2h}. If in addition we regard the orthogonal complement $W_{2h} = (I - P_h)V_h$ of V_{2h} in V_h as the space of high frequency functions (this is only approximately true; in fact W_{2h} actually has both smooth and oscillatory components [6]), then the strong smoothing properties of the continuous reduced Hessian imply that

$$H_h \approx \beta(I - P_h) + H_{2h}P_h \ , \tag{3.1}$$

which combined with orthogonality between P_h and $I - P_h$ suggests [2] the following two-level preconditioner M_h [17]:

$$H_h^{-1} \approx M_h \stackrel{\text{def}}{=} \beta^{-1}(I - P_h) + (H_{2h})^{-1}P_h \ . \tag{3.2}$$

One can generalize this procedure recursively to obtain a multigrid preconditioner. Note that the first part of the preconditioner, $\beta^{-1}(I - P_h)$, acts as a smoother, since it removes high frequency components from the residual. If V_h are finite element spaces, and we invert for the initial value given the entire final-time state, it has been shown in [11] that for h small enough

$$1 - Ch^p/\beta \leq \langle M_h u, u\rangle \left\langle (H_h)^{-1}u, u\right\rangle^{-1} \leq 1 + Ch^p/\beta, \ \text{ for all } u \in V_h, \ u \neq 0 \ , \tag{3.3}$$

where $\langle \cdot, \cdot \rangle$ is the L^2-inner product, p is the convergence order of the forward method ($p = 2$ for piecewise linear polynomials) and C is independent of h. A similar result holds for the multigrid preconditioner when using a $\mathcal{W}$-cycle. The statement (3.3) shows that the two-level preconditioner becomes increasingly effective at high resolution.

There are a number of implementation issues to consider. At the coarsest level, the approximate inverse is replaced by an "exact" solve. The mesh size for the coarse level cannot be chosen arbitrarily, and is a function of the regularization parameter β and the compact part of H (see [11]). Since the reduced Hessian is not available explicitly (even at the coarsest scale) the exact coarse solve is performed by the CG solver. The orthogonal decomposition $(I - P_h)$ can be replaced by less expensive projection-like operators, such as the standard interpolation-restriction operators from classical multigrid theory. Our preconditioner then becomes

$$H_h^{-1} \approx M_h = \beta^{-1}(I - I_{2h}^h I_h^{2h}) + I_{2h}^h M_{2h} I_h^{2h} \ , \tag{3.4}$$

where $I_{2h}^h : V_{2h} \to V_h$ is the natural interpolation operator, and $I_h^{2h} : V_h \to V_{2h}$ is the full-weighting restriction operator defined by $I_h^{2h} = c\left(I_{2h}^h\right)^T$ with c chosen so that a constant function is restricted to itself. The operator $\beta^{-1}(I - I_{2h}^h I_h^{2h})$ acts as a smoother, replacing

[2] Assume that $\beta = 0$. If $v \in V_h$ is decomposed into a smooth $v_s \in V_{2h}$ and an oscillatory $w_o \in W_{2h}$ then $H_h v = H_h v_s + H_h w_0 \approx H_{2h} v_s = H_{2h} P_h v$. Furthermore, we can write $H_h = (1 - P_h)H_h(1 - P_h) + P_h H_h (1 - P_h) + (1 - P_h)H_h P_h + P_h H_h P_h \approx P_h H_h P_h$. Then for $\beta \neq 0$, (3.1) follows easily.

the more expensive $\beta^{-1}(I - P_h)$. Note that the smoothing operator is explicit, symmetric, and sparse; it acts only on initial concentrations, and therefore its application has negligible computational cost compared to the pair of forward/adjoint convective-diffusive transport solves at each CG iteration. Since we coarsen in both temporal and spatial dimensions at the same rate (in our implementation the time-stepping and spatial discretization methods have the same approximation order), the cost of one reduced Hessian–vector multiplication on level l is 2^4 times more expensive than that at the immediately coarser level $l-1$. Therefore, by using just three levels, the cost of a reduced Hessian-vector product on the finest grid is 256 times the cost of a similar operation at the coarsest level. By using a simple $\mathcal{V}$-cycle strategy at the finest level we avoid computing the finest-level residual inside the preconditioner; however, we use a $\mathcal{W}$-cycle (as in King's original algorithm) at the middle level, in order to make up for a possible loss of quality of the preconditioner at coarser resolution, as would follow from (3.3). As stated before, CG is used as a direct solver on the coarsest level. We employ the full multigrid framework (i.e. grid sequencing) to compute initial guesses for each level. In the next section, we discuss numerical results.

4. Implementation and Numerical Results. We demonstrate our dynamic data-driven inversion framework on a hypothetical atmospheric contamination event in the Greater Los Angeles Basin (GLAB) region. Using real topographical data and synthesized velocity fields, we conduct numerical experiments in which sparse observations are extracted from forward simulations and subsequently used in the inverse problem. Our implementation builds on PETSc [3] to manage parallel data structures, coordinate different grid resolutions in our multigrid preconditioner, interface with linear solvers and domain decomposition preconditioners, and utilize a range of software services. We first discuss discretization and geometry details and problem setup. We then briefly present numerical results for initial concentration inversions in the GLAB. Finally, we provide parallel and algorithmic scalability results on structured grids without topography.

In our actual implementation, we first discretize the optimization problem (2.1), and then write optimality conditions (as opposed to the writing the infinite-dimensional optimality conditions (2.6) and then discretizing; in the present case the two are not identical [1]). We employ Streamline Upwind Petrov-Galerkin (SUPG) finite elements [7] in space and Crank-Nicolson in time. For problems with high Peclet number, stabilized methods such as SUPG are more accurate than standard Galerkin on coarse meshes. We use a logically-rectangular topography-conforming isoparametric hexahedral finite element mesh on which piecewise-trilinear basis functions are defined. Since the Crank-Nicolson method is implicit, we "invert" the time-stepping operator using a restarted GMRES method, accelerated by an additive Schwarz domain decomposition preconditioner, both from the PETSc library.

Contaminant transport is modeled over a 360 km $\times$ 120 km $\times$ 5 km in the GLAB. Land surface elevations are obtained at 1 km spacing from the USGS Land Processes Distributed Active Archive Center (GTOPO30 digital elevation model).[3] The three-dimensional mesh is created from the surface elevations by inserting equally spaced grid points vertically from the surface grid to the top of the domain at 5 km.

To simulate a contamination event, an initial contaminant plume with a Gaussian concentration given by $20\exp(-0.04|\boldsymbol{x}-\boldsymbol{x}_c|)$ is centered at $x_c =$(120 km, 60 km, 0 km). The plume is transported over the GLAB region by solving the convection-diffusion equation (1.1) with specified velocity field over a time horizon of 120 minutes. Sensor measurements are taken every 3 minutes to develop a time history from which to invert. For this contaminant, the diffusion coefficient is taken as $\nu = 0.05$. The regularization parameter is fixed at $\beta = 0.01$.

[3] `http://edcdaac.usgs.gov/gtopo30/dem_img.asp`

The forward and inverse problems are solved on a mesh with 361 $\times$ 121 $\times$ 21 grid points, representing 917,301 concentration unknowns at each time step. The time step is the same as the sensor recording rate, i.e. 3 minutes, for a total of 40 time steps. Therefore, there are about 74×10^6 total space-time variables in the KKT optimality system (2.6).

The velocity field $\boldsymbol{v}$ in the convection-diffusion equation is synthesized by solving the steady-state incompressible Navier-Stokes equations, $\rho\left(\boldsymbol{v}\cdot\nabla\right)\boldsymbol{v}+\nabla p-\mu\Delta\boldsymbol{v}=\mathbf{0}$, $\nabla\cdot\boldsymbol{v}=0$, where p is the fluid pressure and (ρ,μ) are its density and viscosity. To simulate an onshore wind, an inflow Dirichlet boundary condition with $v_x = v_{\max}\left(z/(5.0-z_{\text{surface}})\right)^{0.1}$ and zero for the other components is applied to the $x = 0$ plane, where v_{max} is specified as 30 km/hr. Traction-free boundary conditions are applied to the outflow plane at $x = 360$ km. Traction-free tangential and no-slip normal boundary conditions are applied to the remaining portions of the boundary. An SUPG-stabilized finite element method is employed to solve the Navier-Stokes equations using linear tetrahedral elements derived by subdividing the convection-diffusion hexahedral mesh.

We sample the concentrations from the forward transport simulations on a uniformly-spaced array of sensors, and use them as synthetic observations to drive the inverse problem. Figure 4.1 depicts inversion results for different sensor array densities, along with the actual initial concentration (labeled *Target* in the figure). One of the critical issues is to determine the number of sensors required to resolve the initial concentration. As the number of sensors in each direction increases, the error between the actual initial concentration and predicted initial concentration is reduced. The relative L^2 norm error for the $21 \times 21 \times 21$ sensor array is 34%, but as can be observed in the final image, the initial concentration is localized very accurately. Recall that due to the non-vanishing regularization parameter and the fixed mesh size, we cannot expect to recover the initial concentration exactly.

The run-time on a modest number of Alphaserver processors (64) is 2.5 hours for the finest sensor array. As the sensor array becomes denser, the number of CG iterations increases, causing an increase in wall-clock time. With new information provided by the additional more finely-spaced sensors, one might expect a quicker inversion due to a less-poorly-posed problem. In fact, the opposite is true: richer information provided by more the finely-spaced sensors provides more energy to the oscillatory components of the residual; the CG solver thus must work harder to recover these solution components. (Note that in this set of experiments we have not used the multigrid preconditioner.) In addition to the influence of the number of sensors, we have also studied the sensitivity of the inversion to the regularization parameter β, the Peclet number, and the added noise level in the measurements. For brevity we defer presenting these results to a separate article.

What is of ultimate interest is how successful the reconstructed initial field is in predicting the actual transport of the contaminant. Figure 4.2 compares the actual evolution (left) and predicted evolution (right) of the contaminant plume in time. It is evident from this figure that although the reconstructed concentration does not match the actual concentration exactly at $t = 0$, the difference between the two diminishes over time, due to the dissipative nature of the forward convection-diffusion problem.

We next study the parallel and algorithmic scalability of the multigrid preconditioner. In all experiments, we use a regular grid with a constant unidirectional velocity field. This is an important simplification of the problem. Further tests are necessary for the case of more complex and time-dependent velocity fields. The corresponding Peclet number is 3. We take synthetic measurements on a $7 \times 7 \times 7$ sensor array. CG is terminated when the residual of (2.7) has been reduced by six orders of magnitude.

Table 4.1 presents fixed-size scalability results. The inverse problem is solved on a $257 \times 257 \times 257 \times 257$ grid, i.e. there are 17×10^6 inversion parameters in (2.7) and 4.3×10^9

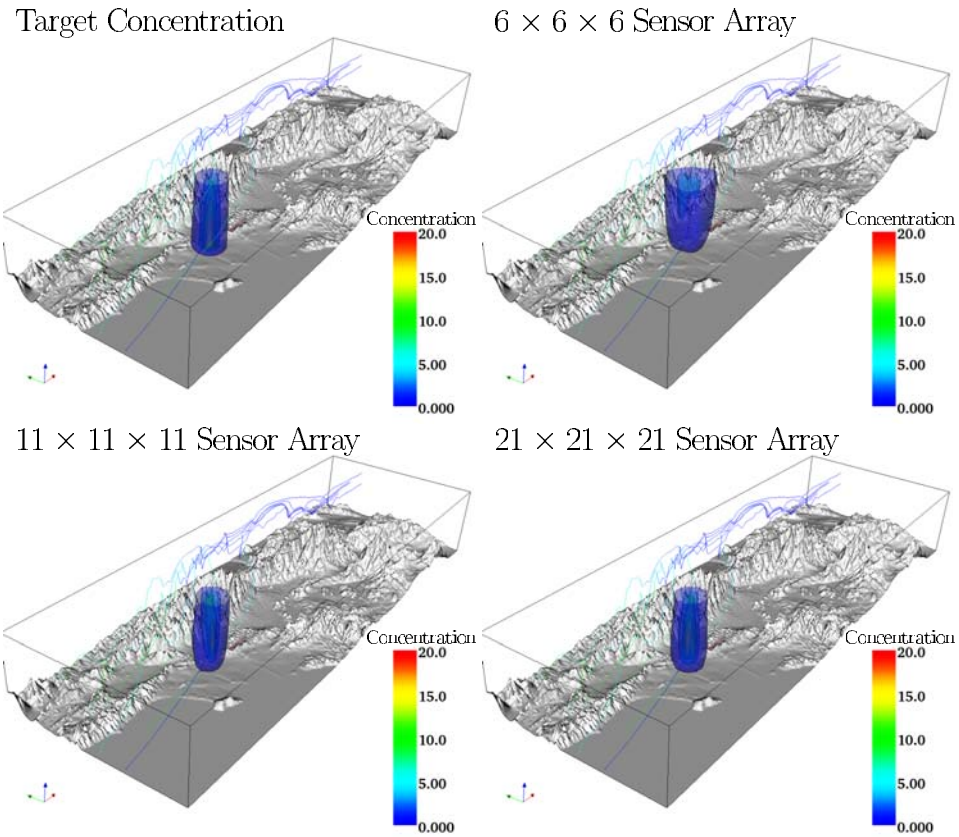

FIG. 4.1. *Sensitivity of the inversion result to the sensor array density. The target initial concentration is shown in the upper-left corner, and inversion results using successively-finer sensor arrays are shown in the subsequent images. As the number of sensors in each direction increases, the quality of the reconstruction of the initial concentration plume improves.* $L^2(\Omega)$ *norm relative errors are* 0.79, 0.49, *and* 0.34 *for the* $6\times6\times6$, $11\times11\times11$, *and* $21\times21\times21$ *sensor arrays, respectively. Inversion using the* $21\times21\times21$ *sensor array takes 2.5 hours on 64 processors of the Alphaserver EV68 system at the Pittsburgh Supercomputing Center. CG iterations are terminated when the norm of the reduced gradient is reduced by five orders of magnitude. Topographical elevation has been exaggerated for visualization purposes.*

TABLE 4.1

Fixed size scalability of unpreconditioned and multigrid preconditioned inversion. *Here the problem size is* $257\times257\times257\times257$ *for all cases. We use a three-level version of the multigrid preconditioner described in Section 3. The variables are distributed across the processors in space, whereas they are stored sequentially in time (as in a multicomponent PDE). Here* hours *is the wall-clock time in hours, and* η *is the parallel efficiency inferred from the runtime. The unpreconditioned code scales extremely well since there is little overhead associated with its single-grid simulations. The multigrid preconditioner also scales reasonably well, but its performance deteriorates since the problem granularity at the coarser levels is significantly reduced. Nevertheless, wall-clock time is significantly reduced over the unpreconditioned case.*

CPUs	no preconditioner		multigrid	
	hours	η	hours	η
128	5.65	1.00	2.22	1.00
512	1.41	1.00	0.76	0.73
1024	0.74	0.95	0.48	0.58

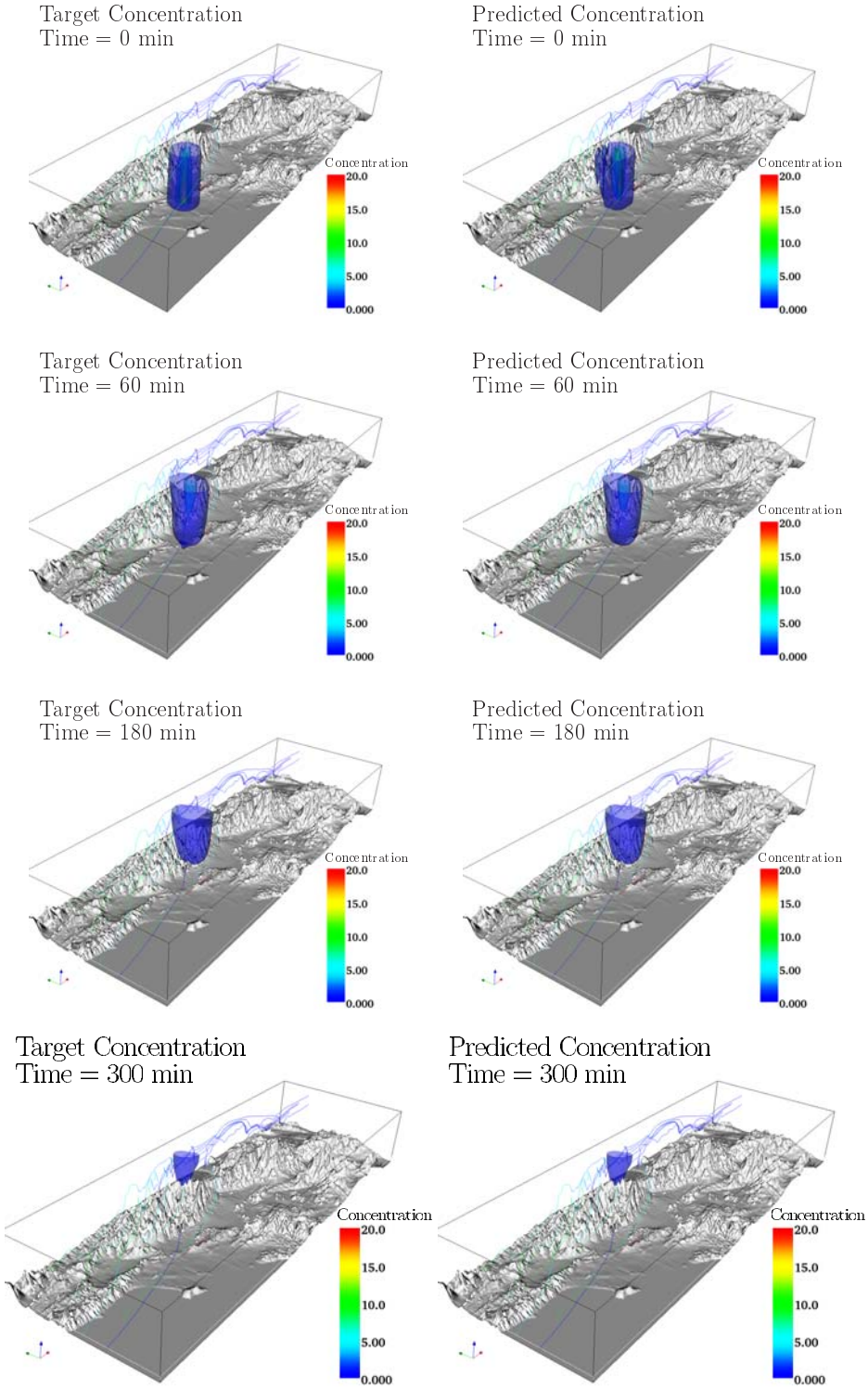

FIG. 4.2. *Illustration of the predictive capabilities of our inversion algorithm, using an* $11 \times 11 \times 11$ *sensor array. Forward transport of the actual initial concentration is compared with forward transport of the reconstructed initial concentration plume. The trajectories are close to each other, and the comparison improves with time.*

total space-time unknowns in (2.6). Note that while the CG iterations are insensitive to the number of processors, the forward and adjoint transport simulations at each iteration rely on a single-level Schwarz domain decomposition preconditioner, whose effectiveness deteriorates with increasing number of processors. Thus, the efficiencies reported in the table reflect parallel as well as (forward) algorithmic scalability. The multigrid preconditioner incurs non-negligible overhead as the number of processors increases for fixed problem size, since the coarse subproblems are solved on ever larger numbers of processors. For example, on 1024 processors, the $65 \times 65 \times 65$ coarse grid solve has just 270 grid points per processor, which is far too few for a favorable computation-to-communication ratio.

On the other hand, the unpreconditioned CG iterations exhibit excellent parallel scalability since the forward and adjoint problems are solved on just the fine grids. Nevertheless, the multigrid preconditioner achieves a net speedup in wall-clock time, varying from a factor of 2.5 for 128 processors to 1.5 for 1024 processors. Most important, the inverse problem is solved in less than 29 minutes on 1024 processors. This is about 18 times the wall-clock time for solving a single forward transport problem.

Table 4.2 presents isogranular scalability results. Here the problem size ranges from

TABLE 4.2

Isogranular scalability of unpreconditioned and multigrid preconditioned inversion. *The spatial problem size per processor is fixed (stride of 8). Ideal speedup should result in doubling of wall-clock time. The multigrid preconditioner scales very well due to improving algorithmic efficiency (decreasing CG iterations) with increasing problem size. Unpreconditioned CG is not able to solve the largest problem in reasonable time.*

grid size	problem size		CPUs	no preconditioner		multigrid	
	u_0	(u, p, u_0)		hours	iterations	hours	iterations
129^4	2.15E+6	5.56E+8	16	2.13	23	1.05	8
257^4	1.70E+7	8.75E+9	128	5.65	23	2.22	6
513^4	1.35E+8	1.39E+11	1024	—	—	4.89	5

5.56×10^8 to 1.39×10^{11} total space-time unknowns, while the number of processors ranges from 16 to 1024. Because we refine in time as well as in space, and because the number of processors increases by a factor of 8 with each refinement of the grid, the total number of space-time unknowns is not constant from row to row of the table; in fact it doubles. However, the number of grid points per processor does remain constant, and this is the number that dictates the computation to communication ratio. For ideal overall (i.e. algorithmic + parallel) scalability, we would thus expect wall-clock time to double with each refinement of the grid. Unpreconditioned CG becomes too expensive for the larger problems, and is unable to solve the largest problem in reasonable time. The multigrid preconditioned solver, on the other hand, exhibits very good overall scalability, with overall efficiency dropping to 95% on 128 processors and 86% on 1024 processors, compared to the 16 processor base case. From the fixed-size scalability studies in Table 4.1, we know that the parallel efficiency of the multigrid preconditioner drops on large numbers of processors due to the need to solve coarse problems. However, the isogranular scalability results of Table 4.2 indicate substantially better multigrid performance. What accounts for this? First, the constant number of grid points per processor keeps the processors relatively well-populated for the coarse problems. Second, the algorithmic efficacy of the multigrid preconditioner improves with decreasing mesh size (as predicted by (3.3)); the number of iterations drops from 8 to 5 over two successive doublings of mesh resolution. The largest problem exhibits a factor of 4.6 reduction in CG iterations relative to the unpreconditioned case (5 vs. 23). This improvement in algorithmic efficiency helps keep the overall efficiency high.

5. Conclusions. We have presented a methodology for solving terascale dynamic data-driven inverse problems of determining unknown initial condition data from sparse observations in a model-consistent manner. The methodology has been instantiated in the context of inverse convection-diffusion transport problems. The main difficulty is that the optimality system is a boundary value problem in 4D space-time, even though the forward simulation problem is an initial value parabolic-hyperbolic problem. We have presented special-purpose parallel multigrid algorithms that exploit the spectral structure of the inverse operator. Experiments on problems of localizing airborne contaminant release from sparse observations in a regional atmospheric transport model demonstrate that:

- 17-million-parameter inversion can be effected at a cost of just 18 forward simulations;
- wall-clock time on 1024 Alphaserver EV68 processors for the 17-million parameter inversion case is just 29 minutes;
- the multigrid preconditioner reduces the number of iterations by as much as a factor of 4.6; and
- inverse problems with 135 million initial condition parameters and 139 billion total space-time unknowns are solved in less than 5 hours on 1024 processors at 86% overall (parallel + algorithmic) efficiency.

These results suggest that ultra-high resolution data-driven inversion can be carried out sufficiently rapidly for linear transport problems to enable simulation-based "real-time" hazard assessment. The next step is to test the scalability of the algorithm on complex wind fields. Our long-term goal is to integrate the current framework with more sophisticated transport models that include, for example, deposition, regional weather models, and more accurate terrain information.

REFERENCES

[1] F. ABRAHAM, M. BEHR, AND M. HEINKENSCHLOSS, *The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations*, Finite Elements in Analysis and Design, 41 (2004), pp. 229–251.

[2] V. AKCELIK, G. BIROS, O. GHATTAS, K. R. LONG, AND B. VAN BLOEMEN WAANDERS, *A variational finite element method for source inversion for convective-diffusive transport*, Finite Elements in Analysis and Design, 39 (2003), pp. 683–705.

[3] S. BALAY, K. BUSCHELMAN, W. D. GROPP, D. KAUSHIK, L. C. MCINNES, AND B. F. SMITH, *PETSc home page*. http://www.mcs.anl.gov/petsc, 2001.

[4] G. BIROS AND O. GHATTAS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, SIAM Journal on Scientific Computing, (2005). In press.

[5] ———, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange Newton solver, and its application to optimal control of steady viscous flows*, SIAM Journal on Scientific Computing, (2005). In press.

[6] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A multigrid tutorial*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., 2000.

[7] A. N. BROOKS AND T. J. R. HUGHES, *Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199–259.

[8] D. N. DAESCU AND G. R. CARMICHAEL, *An adjoint sensitivity method for the adaptive location of the observations in air quality modeling*, Journal of Atmospheric Sciences, 60 (2003), pp. 434–450.

[9] *DDDAS 2000 Workshop Report*. `http://www.nsf.gov/cise/cns/darema/dd_das/dd_das_work_shop_rprt.pdf`, March 2000.

[10] C. DOUGLAS, *DDDAS.org: Dynamic Data-Driven Application Simulation*. `http://www.dddas.org`.

[11] A. DRĂGĂNESCU, *Two investigations in numerical analysis: Monotonicity preserving finite element methods, and multigrid methods for inverse parabolic problems*, PhD thesis, University of Chicago, August 2004.

[12] W. HACKBUSCH, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[13] M. HANKE AND C. R. VOGEL, *Two-level preconditioners for regularized inverse problems. I. Theory*, Numerische Mathematik, 83 (1999), pp. 385–402.

[14] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, 1997.

[15] B. KALTENBACHER, *V-cycle convergence of some multigrid methods for ill-posed problems*, Mathematics of Computation, 72 (2003), pp. 1711–1730.

[16] T. KAMINSKI AND M. HEIMANN, *A coarse grid three-dimensional global inverse model of the atmospheric transport i. Adjoint model and Jacobian matrix*, Journal on Geophysics, 104 (1999), pp. 18535–18663.

[17] J. T. KING, *Multilevel algorithms for ill-posed problems*, Numerische Mathematik, 61 (1992), pp. 311–334.

[18] G. KONTAREV, *Adjoint equation technique applied to meteorological problems*, Tech. Report 21, European Center for Medium Range Weather Forecasts, Reading, England, 1980.

[19] G. MARCHUK AND V. PENENKO, *Study of the sensitivity of discrete models of atmospheric and oceanic dynamics*, Atmospheric and Oceanic Physics, 15 (1980), pp. 785–789.

[20] *The MM5 community mesoscale model*. `http://www.mmm.ucar.edu/mm5`.

[21] A. RIEDER, *A wavelet multilevel method for ill-posed problems stabilized by Tikhonov regularization*, Numerische Mathematik, 75 (1997), pp. 501–522.