Parallel Algorithms for PDE-Constrained Optimization

***Volkan Akçelik*[*], *George Biros*[†], *Omar Ghattas*[‡], *Judith Hill*[§], *David Keyes*[¶], *Bart van Bloemen Waanders*[‖]**

1 Introduction

PDE-constrained optimization refers to the optimization of systems governed by partial differential equations (PDEs). The *simulation problem* is to solve the PDEs for the *state variables* (e.g. displacement, velocity, temperature, electric field, magnetic field, species concentration), given appropriate data (e.g. geometry, coefficients, boundary conditions, initial conditions, source functions). The *optimization problem* seeks to determine some of these data—the *decision variables*—given performance goals in the form of an objective function and possibly inequality or equality constraints on the behavior of the system. Since the behavior of the system is modeled by the PDEs, they appear as (usually equality) constraints in the optimization problem. We will refer to these PDE constraints as the *state equations*.

Let $\mathbf{u}$ represent the state variables, $\mathbf{d}$ the decision variables, $\mathcal{J}$ the objective function, $\mathbf{c}$ the residual of the state equations, and $\mathbf{h}$ the residual of the inequality constraints. We

[*] Ultrascale Simulation Laboratory, Department of Civil & Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, 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)

[‡] Institute for Computational Engineering & Sciences, Departments of Geological Sciences, Mechanical Engineering, Computer Sciences, and Biomedical Engineering, and Institute for Geophysics, University of Texas at Austin, Austin, TX, USA (omar@ices.texas.edu)

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

[¶] Department of Applied Physics & Applied Mathematics, Columbia University, New York, NY, USA (kd2112@columbia.edu)

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

can then state the general form of a PDE-constrained optimization problem as:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{d}} \mathcal{J}(\mathbf{u},\mathbf{d}) \\ \text{subject to } \; &\mathbf{c}(\mathbf{u},\mathbf{d}) = \mathbf{0} \\ &\mathbf{h}(\mathbf{u},\mathbf{d}) \geq \mathbf{0} \end{aligned} \tag{1}$$

The PDE-constrained optimization problem (1) can represent an optimal design, optimal control, or inverse problem, depending on the nature of the objective function and decision variables. The decision variables correspondingly represent design, control, or inversion variables.

Many engineering and science problems—in such diverse areas as aerodynamics, atmospheric sciences, chemical process industry, environment, geosciences, homeland security, infrastructure, manufacturing, medicine, and physics—can be expressed in the form of a PDE-constrained optimization problem. The common difficulty is that PDE solution is just a subproblem associated with optimization. Moreover, while the simulation problem (given $\mathbf{d}$, find $\mathbf{u}$ from $\mathbf{c}(\mathbf{u},\mathbf{d}) = \mathbf{0}$) is usually well-posed, the optimization problem (1) can be ill-posed. Finally, when the state equations are evolutionary in nature, the optimality conditions for (1) are a boundary value problem in space–time. For these reasons, the optimization problem is often significantly more difficult to solve than the simulation problem.

The size, complexity, and infinite-dimensional nature of PDE-constrained optimization problems present significant challenges for general-purpose optimization algorithms. These features often require regularization, iterative solvers, preconditioning, globalization, inexactness, and parallel implementation that are tailored to the structure of the underlying operators. Continued advances in PDE solvers, and the rapid ascendance of parallel computing, have in recent years motivated the development of special-purpose optimization algorithms that exploit the structure of the PDE constraints and scale to large numbers of processors. These algorithms are the focus of this chapter.

To illustrate the main issues, let us consider a distributed optimal flow control problem for the steady-state Burgers equation:

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{d}} \mathcal{J}(\boldsymbol{u},\boldsymbol{d}) \overset{\text{def}}{=} \frac{1}{2}\int_{\Omega} \nabla\boldsymbol{u}\cdot\nabla\boldsymbol{u}\; d\boldsymbol{x} + \frac{\rho}{2}\int_{\Omega} \boldsymbol{d}\cdot\boldsymbol{d}\; d\boldsymbol{x} \\ \text{subject to } \; &-\nu\Delta\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{u} = \boldsymbol{d} \;\text{ in }\; \Omega \\ &\boldsymbol{u} = \boldsymbol{g} \;\text{ on }\; \partial\Omega \end{aligned} \tag{2}$$

Here, $\boldsymbol{u}(\boldsymbol{x})$ is the velocity field, $\boldsymbol{d}(\boldsymbol{x})$ is a domain source, $\boldsymbol{g}(\boldsymbol{s})$ is a boundary source, ν is the viscosity, Ω represents the domain and $\partial\Omega$ its boundary, and ρ is a parameter reflecting the cost of the controls. In the simulation problem we are given the data $\nu, \boldsymbol{d}, \Omega, \boldsymbol{g}$ and we seek the state $\boldsymbol{u}$. In the optimization problem the situation is reversed: we wish to determine a portion of the data, for example $\boldsymbol{g}$ (boundary control), $\boldsymbol{d}$ (distributed control), $\boldsymbol{\Omega}$ (shape or topology optimization), or ν (parameter estimation), so that the decision variable and resulting state $\boldsymbol{u}$ minimize some functional of these variables. In the particular example (2), the decision variable is just the distributed source $\boldsymbol{d}$, and ν, Ω, $\boldsymbol{g}$ are taken as knowns. The objective $\mathcal{J}$ represents a balance between the rate of energy dissipation and the L^2 cost of the controls.

A classical way to approach this problem is to introduce a Lagrange multiplier field, $\boldsymbol{\lambda}(\boldsymbol{x})$, known as the *adjoint state* or *costate* variable, and form a Lagrangian functional $\mathcal{L}$ that incorporates the PDE constraints via an "inner product" with $\boldsymbol{\lambda}$,

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{d}) \overset{\text{def}}{=} \mathcal{J}(\boldsymbol{u}, \boldsymbol{d}) + \int_{\Omega} [\nu \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{\lambda} + \boldsymbol{\lambda} \cdot (\nabla \boldsymbol{u})\boldsymbol{u} - \boldsymbol{d} \cdot \boldsymbol{\lambda}] \; \boldsymbol{dx} \tag{3}$$

One then requires stationarity of $\mathcal{L}$ with respect to the state ($\boldsymbol{u}$), decision ($\boldsymbol{d}$), and adjoint ($\boldsymbol{\lambda}$) variables. Taking variations and invoking the appropriate Green identities, we arrive at the following system of equations representing first-order necessary conditions for optimality:

$$\begin{aligned}
-\nu\Delta\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{u} &= \boldsymbol{d} \;\; \text{in} \;\; \Omega & & \textit{state equation} \quad (4)\\
\boldsymbol{u} &= \boldsymbol{g} \;\; \text{on} \;\; \partial\Omega & &\\
-\nu\Delta\boldsymbol{\lambda} + (\nabla\boldsymbol{u})^T\boldsymbol{\lambda} - (\nabla\boldsymbol{\lambda})\boldsymbol{u} - \boldsymbol{\lambda}\,\text{div}\,\boldsymbol{u} &= \Delta\boldsymbol{u} \;\; \text{in} \;\; \Omega & & \textit{adjoint equation} \quad (5)\\
\boldsymbol{\lambda} &= \boldsymbol{0} \;\; \text{on} \;\; \partial\Omega & &\\
\rho\,\boldsymbol{d} + \boldsymbol{\lambda} &= \boldsymbol{0} \;\; \text{in} \;\; \Omega & & \textit{decision equation} \quad (6)
\end{aligned}$$

The state equation (4) is just the original Burgers boundary value problem that appears as a constraint in the optimization problem (2). The *adjoint equation* (5), which results from stationarity with respect to the state, is a boundary value problem that is linear in the adjoint variable $\boldsymbol{\lambda}$, and involves the adjoint of the linearized state operator. With appropriate discretization, this adjoint operator is just the transpose of the Jacobian of the discretized state equation. Finally the *decision equation* (6) is in this case algebraic (it would have been differential had the cost of the controls been H^1 instead of L^2). The first-order optimality conditions (4)–(6) are a system of coupled, nonlinear PDEs, and are often known as the Karush-Kuhn-Tucker (KKT) conditions. For theory and analysis of PDE-constrained optimization problems such as (2), see for example [12], [32], [39], [66], [69]. For recent algorithmic trends and large-scale applications, see [19].

In this chapter we review efficient parallel algorithms for solution of PDE optimality systems such (4)–(6). Since the coupled optimality system can be formidable to solve simultaneously, a popular alternative is to eliminate state and adjoint variables, and, correspondingly, state and adjoint equations, thereby reducing the system to a manageable one in just the decision variable. Methods of this type are known as *reduced space* methods. For example, a *nonlinear elimination* variant of a reduced space method would proceed as follows for the KKT system (4)–(6). Given $\boldsymbol{d}$ at some iteration, solve the state equation (4) for the state variable $\boldsymbol{u}$. Knowing the state then permits solution of the adjoint equation (5) for the adjoint variable $\boldsymbol{\lambda}$. Finally, with the state and adjoint known, the decision variable $\boldsymbol{d}$ is updated via an appropriate linearization of the decision equation. This loop is then repeated until convergence. As an alternative to such nonlinear elimination, one often prefers to follow the Newton strategy of *first* linearizing the optimality system, and *then* eliminating the state and adjoint updates via block elimination on the linearized state and adjoint equations. The resulting Schur complement operator is known as the *reduced Hessian*, and the equation to which it corresponds can be solved to yield the decision variable update. Since the main components of reduced space method are (linearized) state and adjoint PDE solves, as well as dense decision space solves, parallelism for this reduced

Newton solution of the optimization problem is typically as straightforward to achieve as it is for the simulation problem. Algorithms of this class will be reviewed in Section 2.1.

Reduced space methods are attractive for several reasons. Solving the subsets of equations in sequence exploits the state/adjoint/decision structure of the optimality system, capitalizing on well-established methods and software for solving the state equation. Adjoint PDE solvers are becoming more popular, due to their role in goal-oriented error estimation and efficient sensitivity computation, so they can be exploited as well. Even in their absence, the strong similarities between the state and adjoint operators suggest that an existing PDE solver for the state equation can be modified with reasonable effort to handle the adjoint equation. Finally, exploiting the structure of the reduced Hessian is straightforward (at least for problems of moderate size), since it is a Schur complement of the linearized KKT conditions with respect to the decision variables and is therefore dense.

Another advantage of reduction is that the linearized KKT system is often very ill-conditioned (beyond, say, the usual h^{-2} ill-conditioning of second-order differential operators); the state and adjoint blocks on the other hand inherit the conditioning properties of the simulation problem. Moreover, the reduced Hessian often has favorable spectral structure (e.g. for many inverse problems its spectrum is similar to that of second kind integral operators) and Krylov solvers can converge in a mesh-independent number of iterations. However, as is the case for most exact Schur-type approaches, the major disadvantage of reduced methods is the need to solve the (linearized) state and adjoint equations exactly *at each reduced space iteration*, which is a direct consequence of the reduction onto the decision variable space.

In contrast to reduced space methods, *full space* methods solve for the state, decision, and adjoint variables simultaneously. For large-scale problems, this is typically effected via Newton-Krylov iteration. That is, the linear system arising at each Newton iteration on the KKT system is solved using a Krylov iterative method. The difficulty of this approach is the complex structure, indefiniteness, and ill-conditioning of the KKT system, which in turn requires effective preconditioning. Since the KKT optimality conditions are usually PDEs, it is natural to seek domain decomposition or multigrid preconditioners for this task. However, stationarity of the Lagrangian is a saddle-point problem, and existing domain decomposition and multilevel preconditioners for the resulting indefinite systems are not as robust as those for definite systems. Furthermore, constructing the correct smoothing, prolongation, restriction, and interface operators can be quite challenging. Despite these difficulties, there have been several successful algorithms based on overlapping and non-overlapping domain decomposition and multigrid preconditioners; these are reviewed in Section 2.3. Since these methods regard the entire optimality system as a system of coupled PDEs, parallelism follows naturally, as it does for PDE problems, i.e. in a domain-based way.

An alternative full-space approach to domain decomposition or multigrid is to retain the structure-exploiting, condition-improving advantages of a reduced space method, but use it as a preconditioner rather than a solver. That is, we solve in the full space using a Newton-Krylov method, but precondition with a reduced space method. Since the reduced space method is just a preconditioner, it can be applied approximately, requiring just inexact state and adjoint solves at each iteration. These inexact solves can simply be applications of appropriate domain decomposition or multigrid preconditioners for the state and adjoint operators. Depending on its spectral structure, one may also require preconditioners for the reduced Hessian operator. Substantial speedups can be achieved over reduced space methods due to the avoidance of exact solution of the state and adjoint equations at each decision iteration, as the three sets of variables are simultaneously converged. Since the main work per iteration is in the application of preconditioners for the state, adjoint, and decision equations, as well as carrying out PDE-like full space matrix-vector products, these reduced-space-preconditioned full-space methods can be made to parallelize as well as reduced space methods, i.e. as well as the simulation problem. Such methods will be discussed in Section 2.2.

Numerical evidence suggests that for steady-state PDE-constrained optimization problems, full-space methods can outperform reduced space methods by a wide margin. Typical multigrid efficiency has been obtained for some classes of problems. For optimization of systems governed by time-dependent PDEs, the answer is not as clear. The nonlinearities within each time step of a time-dependent PDE solve are usually much milder than for the corresponding stationary PDEs, so amortizing the nonlinear PDE solve over the optimization iterations is less advantageous. Moreover, time dependence results in large storage requirements for full-space methods, since the full space optimality system becomes a boundary value problem in the space–time cylinder. For such problems, reduced space methods are often preferable. Section 3 provides illustrative examples of optimization problems governed by both steady-state and time-dependent PDEs. The governing equations include convective-diffusive transport, Navier-Stokes flow, and acoustic wave propagation; the decision variables include those for control (for boundary sources), design-like (for PDE coefficients), and inversion (for initial conditions). Both reduced space and full space parallel KKT solvers are demonstrated and compared. Parallel implementation issues are discussed in the context of the acoustic inversion problem.

Notation in this chapter respects the following conventions. Scalars are in lowercase italics type, vectors are in lowercase boldface Roman type, and matrices and tensors are in uppercase boldface Roman type. Infinite dimensional quantities are in italics type, whereas finite dimensional quantities (usually discretizations) are upright. We will use d or $\boldsymbol{d}$ or $\mathbf{d}$ for decision variables, u or $\boldsymbol{u}$ or $\mathbf{u}$ for the states, and λ or $\boldsymbol{\lambda}$ for adjoint variables.

2 Algorithms

In this section we discuss algorithmic issues related to efficient parallel solution of first order optimality systems by Newton-like methods. Due to space limitations, we omit discussion of adaptivity and error estimation, regularization of ill-posed problems, inequality constraints on state and decision variables, globalization methods to ensure convergence from distant iterates, and checkpointing strategies for balancing work and memory in time-dependent adjoint computations. These issues must be carefully considered in order to obtain optimally scalable algorithms. The following are some representative references in the infinite-dimensional setting; no attempt is made to be comprehensive. Globalization in the context of PDE solvers is discussed in [63] and in the context of PDE optimization in [51], [77]. For a discussion of active set and interior point methods for inequality constraints in an optimal control setting, see [18], [76] and for primal-dual active set methods see [52]. For adaptive methods and error estimation in inverse problems see [11], [16], [17], [73]; for details on regularization see [36], [45], [79]. See [37], [54] for discussions of checkpointing strategies.

Our discussion of parallel algorithms in this section will be in the context of the *discrete form* of a typical PDE-constrained optimization problem; that is, we first discretize the objective and constraints, and then form the Lagrangian function and derive optimality conditions. Note that this is the reverse of the procedure that was employed in the optimal flow control example in the previous section, in which the infinite-dimensional Lagrangian functional was first formed and then infinite-dimensional optimality conditions were written. When these infinite-dimensional conditions are discretized, they may result in different discrete optimality conditions than those obtained by first discretizing and then differentiating to form optimality conditions. That is, differentiation and discretization do not necessarily commute. We refer the reader to [1], [31], [39], [53], [70] for details.

Let us represent the discretized PDE-constrained optimization problem by

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{d}} \mathcal{J}(\mathbf{u},\mathbf{d}) \\ &\text{subject to } \ \mathbf{c}(\mathbf{u},\mathbf{d}) = \mathbf{0} \end{aligned} \tag{7}$$

where $\mathbf{u} \in \mathbb{R}^n, \mathbf{d} \in \mathbb{R}^m$ are the state and decision variables, $\mathcal{J} \in \mathbb{R}$ is the objective function, and $\mathbf{c} \in \mathbb{R}^n$ are the discretized state equations. Using adjoint variables $\boldsymbol{\lambda} \in \mathbb{R}^n$, we can define the Lagrangian function by $\mathcal{L}(\mathbf{u},\mathbf{d},\boldsymbol{\lambda}) \overset{\text{def}}{=} \mathcal{J}(\mathbf{u},\mathbf{d}) + \boldsymbol{\lambda}^T\mathbf{c}(\mathbf{u},\mathbf{d})$. The first order optimality conditions require that the gradient of the Lagrangian vanish:

$$\left\{ \begin{array}{c} \partial_u \mathcal{L} \\ \partial_d \mathcal{L} \\ \partial_\lambda \mathcal{L} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{g}_u + \mathbf{J}_u^T \boldsymbol{\lambda} \\ \mathbf{g}_d + \mathbf{J}_d^T \boldsymbol{\lambda} \\ \mathbf{c} \end{array} \right\} = \mathbf{0} \tag{8}$$

Here, $\mathbf{g}_u \in \mathbb{R}^n$ and $\mathbf{g}_d \in \mathbb{R}^m$ are the gradients of $\mathcal{J}$ with respect to the states and decision variables respectively; $\mathbf{J}_u \in \mathbb{R}^{n\times n}$ is the Jacobian of the state equations with respect to the state variables; and $\mathbf{J}_d \in \mathbb{R}^{n\times m}$ is the Jacobian of the state equations with respect to the decision variables. A Newton step on the optimality conditions gives the linear system

$$\left[\begin{array}{ccc} \mathbf{W}_{uu} & \mathbf{W}_{ud} & \mathbf{J}_u^T \\ \mathbf{W}_{du} & \mathbf{W}_{dd} & \mathbf{J}_d^T \\ \mathbf{J}_u & \mathbf{J}_d & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \mathbf{p}_u \\ \mathbf{p}_d \\ \boldsymbol{\lambda}_+ \end{array} \right\} = - \left\{ \begin{array}{c} \mathbf{g}_u \\ \mathbf{g}_d \\ \mathbf{c} \end{array} \right\} \tag{9}$$

Here, $\mathbf{W} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the Hessian matrix of the Lagrangian (it involves second derivatives of both $\mathcal{J}$ and $\mathbf{c}$), and is block-partitioned according to state and decision variables; $\mathbf{p}_u \in \mathbb{R}^n$ is the search direction in the $\mathbf{u}$ variables; $\mathbf{p}_d \in \mathbb{R}^m$ is the search direction in the $\mathbf{d}$ variables; and $\boldsymbol{\lambda}_+ \in \mathbb{R}^n$ is the updated adjoint variable. This linear system is known as the Karush-Kuhn-Tucker (KKT) system, and its coefficient matrix as the *KKT matrix*. The KKT matrix is of dimension $(2n + m) \times (2n + m)$. For realistic 3D PDE problems, n and possibly m are very large, so LU factorization of the KKT matrix is not an option. Iterative methods applied to the full KKT system suffer from to ill-conditioning and non-positive-definiteness of the KKT matrix. On the other hand, it is desirable to capitalize on existing parallel algorithms (and perhaps software) for "inverting" the state Jacobian $\mathbf{J}_u$ (and its transpose). Since this is the kernel step in a Newton-based PDE solver, there is a large body of work to build on. For example, for elliptic or parabolic PDEs, optimal or nearly-optimal parallel algorithms are available that require algorithmic work that is linear or weakly superlinear in n, and scale to thousands of processors and billions of variables. The ill-conditioning and complex structure of the KKT matrix, and the desire to exploit (parallel) PDE solvers for the state equations, motivate the use of reduced space methods, as discussed below.

2.1 Reduced space methods

As mentioned in the introduction, one way to exploit existing PDE-solvers is to eliminate the state and adjoint equations and variables, and then solve the reduced Hessian system in the remaining decision space. We refer to this as a *reduced Newton* (RN) method. It can be derived by block elimination on the KKT system (9): eliminate $\mathbf{p}_u$ from the last block of equations (the state equations); then eliminate $\boldsymbol{\lambda}_+$ from the first block (the adjoint equations); and finally solve the middle block (the decision equations) for $\mathbf{p}_d$. This block elimination on (9) amounts to solving the following equations at each Newton step.

Reduced Newton (RN):

$$\begin{aligned} \mathbf{W}_z \mathbf{p}_d &= -\mathbf{g}_d - \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{g}_u + \mathbf{W}_{yz}^T \mathbf{J}_u^{-1} \mathbf{c} && \textit{decision step} \\ \mathbf{J}_u \mathbf{p}_u &= -\mathbf{J}_d \mathbf{p}_d - \mathbf{c} && \textit{state step} \\ \mathbf{J}_u^T \boldsymbol{\lambda}_+ &= -\left(\mathbf{g}_u + \mathbf{W}_{uu} \mathbf{p}_u + \mathbf{W}_{ud} \mathbf{p}_d\right) && \textit{adjoint step} \end{aligned} \qquad (10)$$

The right-hand side of the decision equation involves the *cross-Hessian* $\mathbf{W}_{yz}$, given by

$$\mathbf{W}_{yz} \overset{\text{def}}{=} \mathbf{W}_{ud} - \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d$$

The coefficient matrix of the decision step, which is the Schur complement of $\mathbf{W}_{dd}$, is given by

$$\mathbf{W}_z \overset{\text{def}}{=} \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d - \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{ud} - \mathbf{W}_{du} \mathbf{J}_u^{-1} \mathbf{J}_d + \mathbf{W}_{dd}$$

and is known as the *reduced Hessian* matrix. Because it contains the inverses of the state and adjoint operators, the reduced Hessian $\mathbf{W}_z$ is a dense matrix. Thus, applying a dense parallel factorization is straightforward. Moreover, since the reduced Hessian is of the dimension of the decision space, m, the dense factorization can be carried out on a single processor when the number of decision variables is substantially smaller than the number of states (as is the case when the decision variables represent discrete parameters that are independent of the mesh size). The remaining two linear systems that have to be solved at each Newton iteration—the state and adjoint updates—have as coefficient matrix either the state Jacobian $\mathbf{J}_u$ or its transpose $\mathbf{J}_u^T$. Since "inverting" the state Jacobian is at the heart of a Newton solver for the state equations, the state and adjoint updates in (10) are able to exploit available parallel algorithms and software for the simulation problem. It follows that the RN method can be implemented with parallel efficiency comparable to that of the simulation problem.

However, the difficulty with the RN method is the need for m solutions of the (linearized) state equations to construct the $\mathbf{J}_u^{-1} \mathbf{J}_d$ term within $\mathbf{W}_z$. This is particularly troublesome for large-scale 3D problems, where (linearized) PDE systems are usually solved iteratively, and solution costs cannot be amortized over multiple right hands as effectively as with direct solvers. When m is moderate or large (as will be the case when the decision space is mesh-dependent), RN with exact formation of the reduced Hessian becomes intractable. So while its parallel efficiency may be high, its algorithmic efficiency can be poor.

An alternative to forming the reduced Hessian is to solve the decision step in (10) by a Krylov method. Since the reduced Hessian is symmetric, and positive definite near a minimum, the Krylov method of choice is conjugate gradients (CG). The required action of the reduced Hessian $\mathbf{W}_z$ on a decision-space vector within the CG iteration is formed in a matrix-free manner. This can be achieved with the dominant cost of a single pair of linearized PDE solves (one state and one adjoint). Moreover, the CG iteration can be terminated early to prevent oversolving in early iterations and to maintain a direction of descent [33]. Finally, in many cases the spectrum of the reduced Hessian is favorable for CG and convergence can be obtained in a mesh-independent number of iterations. We refer to this method as a *reduced Newton-CG* (RNCG) method, and demonstrate it for a large-scale inverse wave propagation problem in Section 3.1.

While RNCG avoids explicit formation of the exact Hessian and the required m (linearized) PDE solves, it does still require a pair of linearized PDE solves per CG iteration. Moreover, the required second derivatives of the objective and state equations are often difficult to compute (although this difficulty may be mitigated by continuing advances in automatic differentiation tools [61]). A popular technique that addresses these two difficulties is a *reduced quasi-Newton* (RQN) method that replaces the reduced Hessian $\mathbf{W}_z$ with a quasi-Newton (often BFGS) approximation $\mathbf{B}_z$, and discards all other Hessian terms [20].

Reduced Quasi-Newton (RQN):

$$\begin{aligned}
\mathbf{B}_z \mathbf{p}_d &= -\mathbf{g}_d - \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{g}_u && \textit{decision step} \\
\mathbf{J}_u \mathbf{p}_u &= -\mathbf{J}_d \mathbf{p}_d - \mathbf{c} && \textit{state step} \\
\mathbf{J}_u^T \boldsymbol{\lambda}_+ &= -\mathbf{g}_u && \textit{adjoint step}
\end{aligned} \tag{11}$$

We see that RQN requires just *two* (linearized) PDE solves per Newton iteration (one a linearized state solve with $\mathbf{J}_u$ and one an adjoint solve with $\mathbf{J}_u^T$), as opposed to the m PDE solves required for RN (the adjoint step to compute the adjoint variable is superfluous in this algorithm). And with Hessian terms either approximated or dropped, no second derivatives are needed. When the number of decision variables m is small and the number of states n is large, the BFGS update (which involves updates of the Cholesky factors of the BFGS approximation) can be computed serially, and this can be done redundantly across all processors [67]. For problems in which m is intermediate in size, the BFGS update may become too expensive for a single processor, and updating the inverse of the dense BFGS approximation can be done efficiently in parallel. Finally, for large m (such as in distributed control or estimation of continuous fields), a limited-memory BFGS (in place of a full) update [68] becomes necessary. When implemented as an update for the inverse of the reduced Hessian, the required decision-space inner products and vector sums parallelize very well, and good overall parallel efficiency results [21]. A measure of the success of RQN is its application to numerous problems governed by PDEs from linear and nonlinear elasticity, incompressible and compressible flow, heat conduction and convection, phase changes, and flow through porous media. With RQN as described above, the asymptotic convergence rate drops from the quadratic rate associated with RN to 2-step superlinear. In addition, unlike the usual case for RN, the number of iterations taken by RQN will typically increase as the decision space is enlarged (i.e. as the mesh is refined), although this also depends on the spectrum of the reduced Hessian and on the difference between it and the initial BFGS approximation. See for example [59] for discussion of quasi-Newton methods for infinite-dimensional problems. Specialized quasi-Newton updates that take advantage of the "compact + differential" structure of reduced Hessians for many inverse problems have been developed [40].

As described in the Introduction, one final option for reduced space methods is a nonlinear elimination variant, which we term nonlinear reduced Newton (NLRN). This is similar to RN, except elimination is performed on the nonlinear optimality system (8). The state equations and state variables are eliminated at each iteration by nonlinear solution of $\mathbf{c}(\mathbf{u},\mathbf{d})=\mathbf{0}$. Similarly, the adjoint equations and adjoint variables are eliminated at each iteration by solution of the linear system $\mathbf{J}_u^T\boldsymbol{\lambda} = -\mathbf{g}_u$. This gives the following form at each Newton step.

Nonlinear Reduced Newton (NLRN):

$$\mathbf{W}_z\mathbf{p}_d = -\mathbf{g}_d - \mathbf{J}_d^T\mathbf{J}_u^{-T}\mathbf{g}_u \qquad \textit{decision step} \tag{12}$$

where $\mathbf{c}(\mathbf{u},\mathbf{d}) = \mathbf{0}$ is implicit in (12) and the adjoint solve contributes to the right-hand side. Alternatively, one may think of NLRN as solving the optimization problem (7) in the space of just the decision variables, by eliminating the state variables and constraints, to give the unconstrained optimization problem:

$$\min_{\mathbf{d}} \mathcal{J}(\mathbf{u}(\mathbf{d}),\mathbf{d})$$

Application of Newton's method to solve this problem, in conjunction with the implicit function theorem to generate the necessary derivatives, yields NLRN above [64]. NLRN can also be implemented in quasi-Newton and Newton-CG settings. These methods are particularly attractive for time-dependent PDE-constrained optimization problems, in particular those that require a large number of time steps or are time-integrated accurately and/or explicitly. In this case the need to carry along and update the current state and adjoint estimates (which are time-dependent) is onerous; on the other hand, there is little advantage to simultaneous solution of the state equations and the optimization problem (in the absence of inequality constraints on the states), if the state equations are weakly nonlinear (as they will be with accurate time-stepping) or explicitly-solved. NLRN permits estimates of just the decision variables to be maintained at each optimization iteration.

As successful as the reduced space methods of this section are in combining fast Newton-like convergence with a reduced number of PDE solves per iteration, they do still (formally) require the exact solution of linearized state and adjoint PDE problems at each iteration. In the next section, we see that a method can be constructed that avoids the exact solves while retaining the structure-exploiting advantages of reduced methods.

2.2 LNKS: Krylov full-space solution with approximate reduced space preconditioning

In this section we return to solution of the full-space Newton step (9). We consider use of a Krylov method, in particular symmetric QMR, applied directly to this system. QMR is attractive because it does not require a positive definite preconditioner. The indefiniteness and potential ill-conditioning of the KKT matrix demand a good preconditioner. It should be capable of exploiting the structure of the state constraints (specifically that good preconditioners for $\mathbf{J}_u$ are available), should be cheap to apply, should be effective in reducing the number of Krylov iterations, and should parallelize readily. The reduced space methods described in the previous section—in particular an approximate form of RQN—fulfill these requirements.

We begin by noting that the block elimination of (10) is equivalent to the following block factorization of the KKT matrix:

$$\begin{bmatrix} \mathbf{W}_{uu}\mathbf{J}_u^{-1} & \mathbf{0} & \mathbf{I} \\ \mathbf{W}_{du}\mathbf{J}_u^{-1} & \mathbf{I} & \mathbf{J}_d^T\mathbf{J}_u^{-T} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{J}_u & \mathbf{J}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{yz} & \mathbf{J}_u^T \end{bmatrix} \tag{13}$$

Note that these factors can be permuted to block triangular form, so we can think of (13) as a block LU factorization of the KKT matrix. To derive the preconditioner, we replace the reduced Hessian $\mathbf{W}_z$ in (13) by a (usually but not necessarily) limited memory BFGS approximation $\mathbf{B}_z$ (as in RQN), drop other second derivative terms (also as in RQN), and replace the exact (linearized) state and adjoint operators $\mathbf{J}_u$ and $\mathbf{J}_u^T$ with approximations $\tilde{\mathbf{J}}_u$ and $\tilde{\mathbf{J}}_u^T$. The resulting preconditioner then takes the form of the following approximate block-factorization of the KKT matrix:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{J}_d^T\tilde{\mathbf{J}}_u^{-T} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{J}}_u & \mathbf{J}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{J}}_u^T \end{bmatrix} \tag{14}$$

Applying the preconditioner by solving with the block factors (14) amounts to performing the RQN step in (11), but with approximate state and adjoint solves. A good choice for $\tilde{\mathbf{J}}_u$ is one of the available parallel preconditioners for $\mathbf{J}_u$—for many PDE operators, there exist near-spectrally-equivalent preconditioners that are both cheap to apply (their cost is typically linear or weakly superlinear in problem size) and effective (resulting in iteration counts that are independent of, or increase very slowly in, problem size). For examples of state-of-the-art parallel PDE preconditioners, see [2] for multigrid and [6] for domain decomposition.

With (14) used as a preconditioner, the preconditioned KKT matrix becomes:

$$\begin{bmatrix} \mathbf{I}_u & \mathcal{O}(\mathbf{E}_u) & \mathbf{0} \\ \tilde{\mathbf{W}}_{yz}^T\tilde{\mathbf{J}}_u^{-1} & \mathcal{O}(\mathbf{E}_u) + \mathbf{W}_z\mathbf{B}_z^{-1} & \mathcal{O}(\mathbf{E}_u) \\ \mathbf{W}_{uu}\tilde{\mathbf{J}}_u^{-1} & \tilde{\mathbf{W}}_{yz}\mathbf{B}_z^{-1} & \mathbf{I}_u \end{bmatrix}$$

where $\mathbf{E}_u \overset{\text{def}}{=} \mathbf{J}_u^{-1} - \tilde{\mathbf{J}}_u^{-1}$, $\mathbf{I}_u \overset{\text{def}}{=} \mathbf{J}_u\tilde{\mathbf{J}}_u^{-1}$, $\tilde{\mathbf{W}}_{yz} \overset{\text{def}}{=} \mathbf{W}_{ud} - \mathbf{W}_{uu}\tilde{\mathbf{J}}_u^{-1}\mathbf{J}_d$. For exact state equation solution, $\mathbf{E}_u = \mathbf{0}$ and $\mathbf{I}_u = \mathbf{I}$, and we see that the reduced space preconditioner clusters the spectrum of the KKT matrix, with all eigenvalues either unit or belonging to $\mathbf{W}_z\mathbf{B}_z^{-1}$. Therefore, when $\tilde{\mathbf{J}}_u$ is a good preconditioner for the state Jacobian, and when $\mathbf{B}_z$ is a good approximation of the reduced Hessian, we can expect the preconditioner (14) to be effective in reducing the number of Krylov iterations.

We refer to this method as Lagrange-Newton-Krylov-Schur (LNKS), since it amounts to a Newton-Krylov method applied to the Lagrangian stationarity conditions, preconditioned by a Schur complement (i.e. reduced Hessian) approximation. See [22], [23], [24],

[25] for further details, and [14], [15], [42], [43], [46], [61] for related methods that use reduced space preconditioning ideas for full-space KKT systems.

Since LNKS applies an approximate version of a reduced space method as a preconditioner (by replacing the PDE solve with a PDE preconditioner application), it inherits the parallel efficiency of RQN in the preconditioning step. The other major cost is the KKT matrix-vector product in the Krylov iteration. For many PDE-constrained optimization problems, the Hessian of the Lagrangian and the Jacobian of the constraints are sparse with structure dictated by the mesh (particularly when the decision variables are mesh-related). Thus, formation of the matrix-vector product at each Krylov iteration is linear in both state and decision variables, and it parallelizes well in the usual fine-grained, domain-decomposed manner characteristic of PDE problems. To achieve overall scalability, we require not just parallel efficiency of the components, but also algorithmic scalability in the sense of mesh-independence (or near-independence) of both Newton and Krylov iterations. Mesh-independence of Newton iterations is characteristic of a wide class of smooth nonlinear operator problems, and we have observed it for a variety of PDE-constrained optimization problems (see also [81]). Mesh-independence of LNKS's Krylov iterations depends on the efficacy of the state and adjoint PDE preconditioners and the limited memory BFGS (or other) approximation of the reduced Hessian. While the former are well-studied, the performance of the latter depends on the nature of the governing PDEs as well as the objective functional. In Section 3.2 we demonstrate parallel scalability and superiority of LNKS over limited memory RQN for a large-scale optimal flow control problem.

2.3 Domain decomposition and multigrid methods

As an alternative to the Schur-based method described in Section 2.2 for solution of the full-space Newton step (9), one may pursue domain decomposition or multigrid preconditioners for the KKT matrix. These methods are more recent than those of Section 2.1 and are undergoing rapid development. Here we give just a brief overview and cite relevant references.

In [71] an overlapping Krylov-Schwarz domain decomposition method was used to solve (9) related to the boundary control of an incompressible driven-cavity problem. This approach resulted in excellent algorithmic and parallel scalability on up to 64 processors for a velocity-vorticity formulation of the 2D steady-state Navier-Stokes equations. One key insight of the method is that the necessary overlap for a control problem is larger than that for the simulation problem. More recently a multi-level variant has been derived [72].

Domain-decomposition preconditioners for linear-quadratic elliptic optimal control problems are presented in [49] for the overlapping case and [50] for the non-overlapping case. Mesh-independent convergence for two-level variants is shown. These domain decomposition methods have been extended to advection-diffusion [13] and time-dependent parabolic [48] problems. Parallelism in the domain-decomposition methods described above can be achieved for the optimization problem in the same manner as it is for the simulation problem, i.e. based on spatial decomposition. Several new ideas in parallel time domain decomposition have emerged recently [41], [47], [78] and have been applied in the parabolic and electromagnetic settings. Although parallel efficiency is less than optimal, parallel speedups are still observed over non-time-decomposed algorithms, which may be crucial for real-time applications.

Multigrid methods are another class of preconditioners for the full-space Newton system (9). An overview can be found in [35]. There are three basic approaches: multigrid applied directly to the optimization problem; multigrid as a preconditioner for the reduced Hessian $\mathbf{W}_z$ in RNCG; and multigrid on the full space Newton system (9). In [65] multigrid is applied directly to the optimization problem to generate a sequence of optimization subproblems with increasingly coarser grids. It is demonstrated that multigrid may accelerate solution of the optimization problem even when it may not be an appropriate solver for the PDE problem. Multigrid for the reduced system (in the context of shape optimization of potential and steady-state incompressible Euler flows) has been studied in [7], [8] based on an analysis of the symbol of the reduced Hessian. For a large class of problems, especially with the presence of a regularization term in the objective functional, the reduced Hessian operator is spectrally equivalent to a second-kind Fredholm integral equation. Although this operator has a favorable spectrum leading to mesh-independent convergence, in practice preconditioning is still useful to reduce the number of iterations. It is essential that the smoother be tailored to the "compact + identity" structure of such operators [44], [57], [60], [62]. The use of appropriate smoothers of this type has resulted in successful multigrid methods for inverse problems for elliptic and parabolic PDEs [4], [34], [58].

Multigrid methods have also been developed for application to the full KKT optimality system for nonlinear inverse electromagnetic problems [9] and for distributed control of linear elliptic and parabolic problems [26], [27]. In such approaches, the state, adjoint, and decision equations are typically relaxed together in pointwise manner (or in the case of L^2 regularization, the decision variable can be eliminated and pointwise relaxation is applied to the coupled state–adjoint system). These multigrid methods have been extended to optimal control of nonlinear reaction-diffusion systems as well [28],[29]. Nonlinearities are addressed through either Newton-multigrid or the full approximation scheme (FAS). Just as with reduced space multigrid methods, careful design of the smoother is critical to the success of full space multigrid.

3 Numerical examples

In this section we present numerical results for parallel solution of three large-scale 3D PDE-constrained optimization problems. Section 3.1 presents an inverse acoustic scattering problem that can be formulated as a PDE-constrained optimization problem with hyperbolic constraints. The decision variables represent the PDE coefficient, in this case the squared velocity of the medium (and thus the structure is similar to a design problem). Because of the large number of time steps and the linearity of the forward problem, a full-space method is not warranted, and instead the reduced space methods of Section 2.1 are employed. In Section 3.2 we present an optimization problem for boundary control of steady Navier-Stokes flows. This problem is an example of an optimization problem constrained by a nonlinear PDE with dominant elliptic term. The decision variables are velocity sources on the boundary. The LNKS method of Section 2.2 delivers excellent performance for this problem. Finally, Section 3.3 presents results from an inverse problem with a parabolic PDE, the convection-diffusion equation. The problem is to estimate the initial condition of an atmospheric contaminant from sparse measurements of its transport. In these three examples, we encounter: elliptic, parabolic, and hyperbolic PDE constraints;

forward solvers that are explicit, linearly implicit, and nonlinearly implicit; optimization problems that are linear, nonlinear in the state, and nonlinear in the decision variable; decision variables that represent boundary condition, initial condition, and PDE coefficient fields; inverse, control, and design-like problems; reduced space and full space solvers; and domain decomposition and multigrid preconditioners. Thus, the examples provide a glimpse into a wide spectrum of problems and methods of current interest.

All of the examples presented in this section have been implemented on top of the parallel numerical PDE solver library PETSc [10]. The infinite-dimensional optimality conditions presented in the following sections are discretized in space with finite elements and (where applicable) in time with finite differences. For each example, we study the algorithmic scalability of the parallel method, i.e. the growth in iterations as the problem size and number of processors increase. In addition, for the acoustic scattering example, we provide a detailed discussion of parallel scalability and implementation issues. Due to space limitations, we restrict the discussion of parallel scalability to this first example; however, the other examples will have similar structure and similar behavior is expected. The key idea is that the algorithms of Section 2 can be implemented for PDE-constrained optimization problems in such a way that the core computations are those that are found in a parallel forward PDE solver, e.g. sparse (with grid structure) operator evaluations, sparse matvecs, vector sums and inner products, and parallel PDE preconditioning. (In fact, this is why we are able to use PETSc for our implementation.) Thus, the optimization solvers largely inherit the parallel efficiency of the forward PDE solver. Overall scalability then depends on the algorithmic efficiency of the particular method, which is studied in the following sections.

3.1 Inverse acoustic wave propagation

Here we study the performance of the reduced space methods of Section 2.1 on an inverse acoustic wave propagation problem [5], [75]. Consider an acoustic medium with domain Ω and boundary Γ. The medium is excited with a known acoustic energy source $f(\boldsymbol{x}, t)$ (for simplicity we assume a single source event), and the pressure $u^*(\boldsymbol{x}, t)$ is observed at N_r receivers, corresponding to points $\boldsymbol{x}_j$ on the boundary. Our objective is to infer from these measurements the squared acoustic velocity distribution $d(\boldsymbol{x})$, which is a property of the medium. Here d represents the decision variable and $u(\boldsymbol{x}, t)$ the state variable. We seek to minimize, over the period $t = 0$ to T, an L^2 norm difference between the observed state and that predicted by the PDE model of acoustic wave propagation, at the N_r receiver locations. The PDE-constrained optimization problem can be written as:

$$\min_{u,d} \mathcal{J}(u, d) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{j=1}^{N_r} \int_0^T \int_\Omega (u^* - u)^2 \, \delta(\boldsymbol{x} - \boldsymbol{x}_j) \, \boldsymbol{dx} \, dt + \rho \int_\Omega (\nabla d \cdot \nabla d + \varepsilon)^{\frac{1}{2}} \boldsymbol{dx} \, dt$$

$$\begin{aligned} \text{subject to} \quad \ddot{u} - \nabla \cdot d \nabla u = f \ &\text{ in } \ \Omega \times (0, T) \\ d \nabla u \cdot n = 0 \ &\text{ on } \ \Gamma \times (0, T) \\ u = \dot{u} = 0 \ &\text{ in } \ \Omega \times \{t = 0\} \end{aligned} \tag{15}$$

The first term in the objective function is the misfit between observed and predicted states, the second term is a *total variation* (TV) regularization functional with regularization parameter ρ, and the constraints are the initial–boundary value problem for acoustic wave propagation (assuming constant bulk modulus and variable density).

TV regularization preserves jump discontinuities in material interfaces, while smoothing along them. For a discussion of numerical issues and comparison with standard Tikhonov regularization, see [5], [80]. While regularization eliminates the null space of the inversion operator (i.e. the reduced Hessian), there remains the difficulty that the objective function can be highly oscillatory in the space of material model d, meaning that straightforward solution of the optimization problem (15) can fail by becoming trapped in a local minimum [74]. To overcome this problem, we use multilevel grid and frequency continuation to generate a sequence of solutions that remain in the basin of attraction of the global minimum; that is, we solve the optimization problem (15) for increasingly higher frequency components of the material model, on a sequence of increasingly finer grids with increasingly higher frequency sources [30]. For details see [5].

Figure 1 illustrates this multiscale approach and the effectiveness of the TV regularizer. The inverse problem is to reconstruct a piecewise-homogeneous velocity model (pictured at top left) that describes the geometry of a hemipelvic bone and surrounding volume from sparse synthetic pressure measurements on four faces of a cube that encloses the acoustic medium. The source consists of the simultaneous introduction of a Ricker wavelet at each measurement point. Two intermediate-grid models are shown (upper right, lower left). The fine-grid reconstructed model (lower right) is able to capture fine-scale features of the “ground truth” model with uncanny accuracy. The anisotropic behavior of the TV regularizer in revealed by its smoothing of ripple artifacts along the interface of the original model. The fine-scale problem has 2.1 million material parameters and 3.4 billion total space-time variables, and was solved in 3 hours on 256 AlphaServer processors at the Pittsburgh Supercomputing Center (PSC).

We next discuss how the optimization problem (15) is solved for a particular grid level in the multiscale continuation scheme. The first order optimality conditions for this problem take the following form:

$$
\begin{aligned}
\ddot{u} - \nabla\cdot d\nabla u &= f \ \text{ in } \ \Omega\times(0,T) \\
d\nabla u\cdot n &= 0 \ \text{ on } \ \Gamma\times(0,T) \qquad\qquad \textit{state equation} \\
u = \dot{u} &= 0 \ \text{ for } \ \Omega\times\{t=0\}
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
\ddot{\lambda} - \nabla\cdot d\nabla\lambda + \sum_{i=1}^{N_r} (u^* - u)\,\delta(\boldsymbol{x}-\boldsymbol{x}_j) &= 0 \ \text{ in } \ \Omega\times(0,T) \\
d\nabla\lambda\cdot n &= 0 \ \text{ on } \ \Gamma\times(0,T) \qquad \textit{adjoint equation} \\
\lambda = \dot{\lambda} &= 0 \ \text{ for } \ \Omega\times\{t=T\}
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
\int_0^T \nabla u\cdot\nabla\lambda\;dt - \beta\nabla\cdot(|\nabla d|_\varepsilon^{-1}\nabla d) &= 0 \ \text{ in } \ \Omega \qquad\qquad \textit{decision equation} \\
\nabla d\cdot n &= 0 \ \text{ on } \ d\Gamma
\end{aligned}
\tag{18}
$$

The state equation (16) is just the acoustic wave propagation initial-boundary value problem. Since the wave operator is self-adjoint in time and space, the adjoint equation (17)

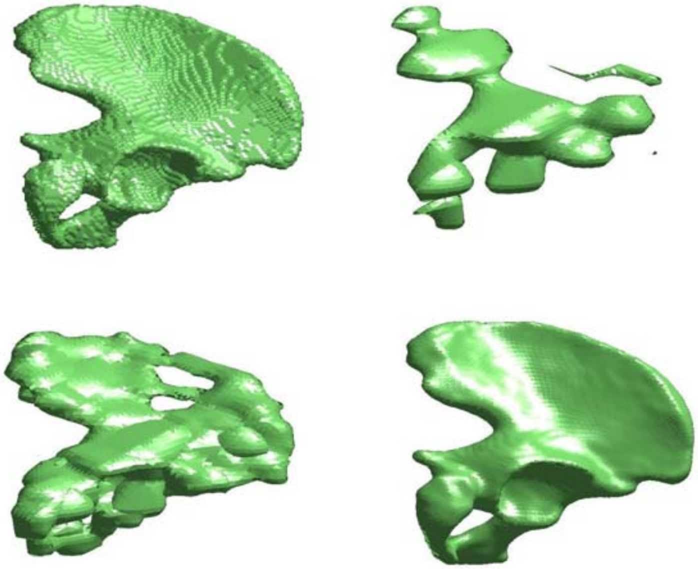

Figure 1. *Reconstruction of hemipelvic bony geometry via solution of an inverse wave propagation problem using a parallel multiscale reduced (Gauss) Newton conjugate gradient (RNCG) optimization algorithm with total variation (TV) regularization.*

has the same form as the state equation, i.e. it too is acoustic wave equation. However, the adjoint wave equation has *terminal*, as opposed to initial, conditions on λ in (17), and it has a different source, which depends on the state variable u. Finally, the decision equation (18) is integro-partial-differential and time-independent.

When appropriately discretized on the current grid level, the dimension of each of u and λ is equal to the number of grid points N_g multiplied by time steps N_t, d is of dimension N_g, and thus the system (16)–(18) is of dimension $2N_gN_t + N_g$. This can be very large for problems of interest—for example, in the largest problem presented in this section, the system contains 3.4×10^9 unknowns. The time dimension cannot be "hidden" with the usual time-stepping procedures, since (16)–(18) couples the initial and final value problems through the decision equation. The optimality system is thus a boundary value-problem in 4D space–time. Full space methods require storage of at least the entire time history of states and adjoints; moreover, because the state and adjoint equations are linear, there is no advantage in folding the state and adjoint solver iterations into the decision iteration, as would be done in a full space method. For this reason, the reduced space methods of Section 2.1 are preferable. Since the state equation is linear in the state u, there is no distinction between the linear and nonlinear variants RN and NLRN.

The numerical results in this section are based on the RNCG method with a limited memory BFGS variant of RQN as a preconditioner. Since this is a least-squares problem, we do not use exact Hessian information, instead using a Gauss-Newton approximation that neglects second derivative terms that involve λ. Spatial approximation is by Galerkin finite elements, in particular piecewise trilinear basis functions for the state u, adjoint λ, and decision d fields. For the class of wave propagation problems we are interested in, the Courant-limited time step size is on the order of that dictated by accuracy considerations, and therefore we choose to discretize in time via explicit central differences. Thus, the number of time steps is of the order of cube root of the number of grid points. Since we require time accurate resolution of wave propagation phenomena, the 4D "problem dimension" scales with the $\frac{4}{3}$ power of the number of grid points.

The overall work is dominated by the cost of the CG iteration, which, because the preconditioner is time-independent, is dominated by the Hessian-vector product. With the Gauss-Newton approximation, the CG matvec requires the same work as the reduced gradient computation: a forward wave propagation, an adjoint wave propagation, possible checkpointing recomputations based on available memory, and the reduction of the state and adjoint spatio-temporal fields onto the material model space via terms of the form $\int \nabla u \cdot \nabla \lambda \ dt$. These components are all "PDE-solver-like," and can be parallelized effectively in a fine-grained domain-based way, using many of the building blocks of sparse PDE-based parallel computation: sparse grid-based matrix-vector products, vector sums, scalings, and inner products.

We report results of fixed-size scaling on the Cray T3E at PSC. We expect the overhead due to communication, synchronization, and sequential bottlenecks to be very low, since one of the key features of the method is that it recasts the majority of the work in solving the inverse problem in terms of explicitly-solved wave propagation problems, both forward and backward in time, and local tensor reduction operations to form the reduced gradient and reduced Hessian-vector product. Because the communication patterns for these components are nearest-neighbor, and because there are no barriers to excellent load balance, the code should scale well. There are also some inner products and global reduction operations, associated with each iteration of CG and with the application of the preconditioner, that require global communication. In a standard Krylov-based forward PDE solver, such inner products can start to dominate as processor counts reach into the thousands. Here, however, it is the "PDE solver" that is on the inside, and the (inversion-related) Krylov iterations that are on the outside. Communication costs associated with inner products are thus negligible. Table 1 demonstrates the good parallel efficiency obtainable for an eightfold increase in number of processors on a Cray T3E, for a fixed problem size of 262,144 grid points (and thus material parameters), and the same number of state and adjoint unknowns per time step.

Table 1 shows a mild decrease in parallel efficiency with increasing problem size. Note the very coarse granularity (a few thousand grid points per processor) for the last few rows of the table. For many forward problems, one would prefer finer granularities, for greater computation-to-communication ratios. However, for most optimization problems, we are necessarily compute-bound, since a sequence of many forward-like problems has to be solved, and one needs as much parallelism as possible. We are therefore interested in appropriating as many processors as possible while keeping parallel efficiency reasonable.

We should point out that this particular inverse problem presents a very severe test of parallel scalability. For scalar wave propagation PDEs, discretized with low order finite elements on structured spatial grids (i.e. the grid stencils are very compact) and explicit central differences in time, there is little workload for each processor in each time step.

Table 1. *Fixed-size scalability on a Cray T3E-900 for a 262,144 grid point problem corresponding to a two-layered medium.*

processors	**grid pts/processor**	**time (s)**	**time/gridpts/proc (s)**	**efficiency**
16	16,384	6756	0.41	1.00
32	8192	3549	0.43	0.95
64	4096	1933	0.47	0.87
128	2048	1011	0.49	0.84

So while we can express the inverse method in terms of (a sequence of) forward-like PDE problems, and while this means we follow the usual “volume computation/surface communication” paradigm, it turns out for this particular inverse problem (involving acoustic wave propagation), the computation to communication ratio is about as low as it can get for a PDE problem (and this will be true whether we solve the forward or inverse problem). A nonlinear forward problem, vector unknowns per grid point, higher order spatial discretization, and unstructured meshes would all increase the computation/communication ratio and produce better parallel efficiencies.

By increasing the number of processors with a fixed grid size, we have studied the effect of communication and load balancing on parallel efficiency in isolation of algorithmic performance. We next turn our attention to algorithmic scalability. We characterize the increase in work as problem size increases (mesh size decreases) by the number of inner (linear) CG and outer (nonlinear) Gauss-Newton iterations. The work per CG iteration involves explicit forward and adjoint wave propagation solutions, and their cost scales with the $\frac{4}{3}$ power of the number of grid points; a CG iteration also requires the computation of the integral in (18), which is linear in the number of grid points. Ideally, the number of linear and nonlinear iterations will be independent of the problem size.

Table 2 shows the growth in iterations for a limited memory BFGS variant of reduced quasi-Newton (**LRQN**), unpreconditioned reduced (Gauss) Newton conjugate gradient (**RNCG**), and LRQN-preconditioned RNCG (**PRNCG**)) methods as a function of material model resolution. LRQN was not able to converge for the 4913 and $35,937$ parameter problems in any reasonable amount of time, and larger problems were not attempted with the method. The Newton methods showed mesh-independence of nonlinear iterations, until the finest grid, which exhibited a significant increase in iterations. This is most likely due to the TV regularizer, which results in an increasingly ill-conditioned reduced Hessian as the mesh is refined. On the other hand, the inner conjugate gradient iterations in PRNCG appear to remain relatively constant within each nonlinear iteration. To verify that the inner iteration would keep scaling, we ran one nonlinear iteration on a 257^3 grid (nearly 17 million inversion parameters) on 2048 AlphaServer processors at PSC. This required 27 CG iterations, which is comparable to the smaller grids, suggesting that the preconditioner is effective.

These results show that the quasi-Newton-preconditioned Newton-CG method seems to be scaling reasonably well for this highly nonlinear and ill-conditioned problem. Overall, the method is able to solve a problem with over 2 million unknown inversion parameters

Table 2. *Algorithmic scaling by limited memory BFGS reduced quasi-Newton (***LRQN***), unpreconditioned reduced (Gauss) Newton Conjugate Gradient (***RNCG***), and LRQN-preconditioned RNCG (***PRNCG***) methods as a function of material model resolution. For LRQN, the number of iterations is reported, and for both LRQN solver and preconditioner,* 200 *L-BFGS vectors are stored. For RNCG and PRNCG, the total number of CG iterations is reported, along with the number of Newton iterations in parentheses. On all material grids up to* 65^3*, the forward and adjoint wave propagation problems are posed on* 65^3 *grid × 400 time steps, and inversion is done on 64 PSC AlphaServer processors; for the* 129^3 *material grid, the wave equations are on* 129^3 *grids × 800 time steps, on 256 processors. In all cases, work per iteration reported is dominated by a reduced gradient (LRQN) or reduced-gradient-like (RNCG, PRNCG) calculation, so the reported iterations can be compared across the different methods. Convergence criterion is* 10^{-5} *relative norm of the reduced gradient. "*" indicates lack of convergence;* † *indicates number of iterations extrapolated from converging value after 6 hours of runtime.*

grid size	material parameters	LRQN its	RNCG its	PRNCG its
65^3	8	16	17 (5)	10 (5)
65^3	27	36	57 (6)	20 (6)
65^3	125	144	131 (7)	33 (6)
65^3	729	156	128 (5)	85 (4)
65^3	4913	*	144 (4)	161 (4)
65^3	$35,937$	*	177 (4)	159 (6)
65^3	$274,625$	—	350 (7)	197 (6)
129^3	$2,146,689$	—	1470† (22)	409 (16)

in just three hours on 256 AlphaServer processors. However, each CG iteration involves a forward/adjoint pair of wave propagation solutions, so that the cost of inversion is over 800 times the cost of the forward problem. Thus, the excellent reconstruction in Figure 1 has come at significant cost. This approach has also been applied to elastic wave equation inverse problems in the context of inverse earthquake modeling with success [3].

3.2 Optimal boundary control of Navier-Stokes flow

In this second example, we give sample results for an optimal boundary control problem for 3D steady Navier-Stokes flow. A survey and articles on this topic can be found in [38], [55], [56]. We use the velocity-pressure $(\boldsymbol{u}(\boldsymbol{x}), p(\boldsymbol{x}))$ form of the incompressible Navier-Stokes equations. The boundary control problem seeks to find an appropriate source $\boldsymbol{d}(\boldsymbol{s})$ on the control boundary $\partial\Omega_d$ so that the $\boldsymbol{H}^1$ seminorm of the velocity (i.e. the rate of dissipation of viscous energy) is minimized:

$$
\begin{aligned}
\min_{\boldsymbol{u},p,\boldsymbol{d}} \mathcal{J}(\boldsymbol{u},p,\boldsymbol{d}) \overset{\text{def}}{=} \frac{\nu}{2}\int_{\Omega} \nabla\boldsymbol{u}\cdot\nabla\boldsymbol{u}\; \boldsymbol{dx} + \frac{\rho}{2}\int_{\partial\Omega_d} |\boldsymbol{d}|^2\; \boldsymbol{ds} \\
\text{subject to} \quad -\nu\Delta\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{u} + \nabla p = \boldsymbol{0} \;\text{ in }\; \Omega \\
\nabla\cdot\boldsymbol{u} = 0 \;\text{ in }\; \Omega \\
\boldsymbol{u} = \boldsymbol{u}_g \;\text{ on }\; \partial\Omega_u \\
\boldsymbol{u} = \boldsymbol{d} \;\text{ on }\; \partial\Omega_d \\
-p\boldsymbol{n} + \nu(\nabla\boldsymbol{u})\boldsymbol{n} = \boldsymbol{0} \;\text{ on }\; \partial\Omega_N
\end{aligned} \tag{19}
$$

Here the decision variable is the control, i.e. the velocity vector $\boldsymbol{d}$ on $\partial\Omega_d$, and the objective reflects an $L^2(\partial\Omega_d)$ cost of the control. There are both Dirichlet (with source $\boldsymbol{u}_g$) and Neumann boundary conditions, and ν is the inverse Reynolds number. For the simulation problem we need not distinguish between $\partial\Omega_d$ and $\partial\Omega_u$ since both boundary subdomains are part of the Dirichlet boundary. In the optimization problem, however, $\boldsymbol{d}$ is not known. Figure 2 illustrates the effect of the optimal boundary control in eliminating the separated flow around a cylinder.

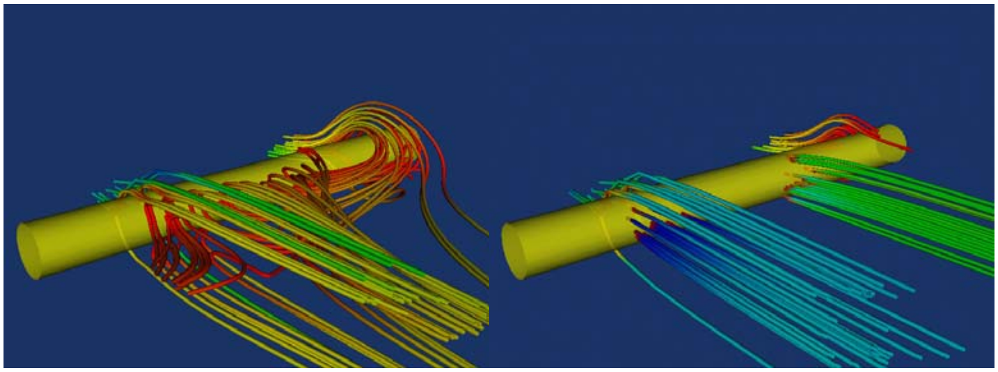

Figure 2. *An optimal boundary control problem to minimize the rate of energy dissipation (equivalent here to the drag) by applying suction or injection of a fluid on the downstream portion of a cylinder at Re=40. The left image depicts an uncontrolled flow; the right image depicts the optimally-controlled flow. Injecting fluid entirely eliminates recirculation and secondary flows in the wake of the cylinder, thus minimizing dissipation. The optimization problem has over 600,000 states and nearly 9000 controls, and was solved in 4.1 hours on 256 processors of a Cray T3E at PSC.*

To derive the optimality conditions, we introduce adjoint variables $\boldsymbol{\lambda}(\boldsymbol{x}), \mu(\boldsymbol{x})$ for the state variables $\boldsymbol{u}, p$, respectively. See [25] for details. The optimality system then takes the following form.

State equations:

$$
\begin{aligned}
-\nu\Delta\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{u} + \nabla p &= \boldsymbol{0} \;\text{ in }\; \Omega \\
\nabla\cdot\boldsymbol{u} &= 0 \;\text{ in }\; \Omega \\
\boldsymbol{u} &= \boldsymbol{u}_g \;\text{ on }\; \partial\Omega_u \\
\boldsymbol{u} &= \boldsymbol{d} \;\text{ on }\; \partial\Omega_d \\
-p\boldsymbol{n} + \nu(\nabla\boldsymbol{u})\boldsymbol{n} &= \boldsymbol{0} \;\text{ on }\; \partial\Omega_N
\end{aligned}
\tag{20}
$$

Adjoint equations:

$$
\begin{aligned}
-\nu\Delta\boldsymbol{\lambda} + (\nabla\boldsymbol{u})^T\boldsymbol{\lambda} - (\nabla\boldsymbol{\lambda})\boldsymbol{u} + \nabla\mu &= \nu\Delta\boldsymbol{u} \;\text{ in }\; \Omega \\
\nabla\cdot\boldsymbol{\lambda} &= \boldsymbol{0} \;\text{ in }\; \Omega \\
\boldsymbol{\lambda} &= \boldsymbol{0} \;\text{ on }\; \partial\Omega_u \\
\boldsymbol{\lambda} &= \boldsymbol{0} \;\text{ on }\; \partial\Omega_d \\
-\mu\boldsymbol{n} + \nu\nabla(\boldsymbol{\lambda})\boldsymbol{n} + (\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{\lambda} &= -\nu(\nabla\boldsymbol{u})\boldsymbol{n} \;\text{ on }\; \partial\Omega_N
\end{aligned}
\tag{21}
$$

Decision equations:

$$
\nu(\nabla\boldsymbol{\lambda} + \nabla\boldsymbol{u})\boldsymbol{n} - \rho\boldsymbol{d} = \boldsymbol{0} \;\text{ on }\; \partial\Omega_d
\tag{22}
$$

Since the flow equations are steady and highly nonlinear for the separated flow, there is significant benefit to integrating state solutions iterations with optimization iterations, and therefore we study the performance of the LNKS method of Section 2.2 in comparison with a limited memory BFGS variant of the reduced quasi-Newton method. We refer to the latter as LRQN. In Table 3 we quote a set of representative results from many we have obtained for up to 1.5 million state variables and 50,000 control variables on up to 256 processors. Approximation is by Taylor-Hood Galerkin finite elements, both for state and decision variables. The table provides results for 64 and 128 Cray T3E processors for a doubling (roughly) of problem size. LNKS-EX refers to exact solution of the linearized Navier-Stokes equation within the LRQN preconditioner, whereas LNKS-PR refers to application of a block-Jacobi (with local ILU(0)) approximation of the linearized Navier-Stokes forward and adjoint operators within the preconditioner. LNKS-PR-IN uses an inexact Newton method, which avoids fully converging the KKT linear system for iterates that are far from a solution.

The results in the table reflect the independence of Newton iterations on problem size, the mild dependence of KKT iterations on problem size, and the resulting reasonable scalability of the method. It is important to point out here that the Navier-Stokes discrete operator is very ill-conditioned, and there is room for improvement of its domain-decomposition preconditioner (single-level block Jacobi–ILU). The scalability of the LNKS methods would improve correspondingly. A dramatic acceleration of the LNKS algorithm is achieved by truncating the Krylov iterations. More detailed results are given in [22], [24], [25]. The important result is that LNKS solves the optimization problem in 4.1 hours, which is 5 times the cost of solving the equivalent simulation problem, and over an order of magnitude faster than a conventional reduced space method (LRQN).

Table 3. *Algorithmic scalability for Navier-Stokes optimal flow control problem on 64 and 128 processors of a Cray T3E for a doubling (roughly) of problem size.*

states **controls**	**method**	**Newton iter**	**average KKT iter**	**time (hours)**
389,440	LRQN	189	—	46.3
6,549	LNKS-EX	6	19	27.4
(64 procs)	LNKS-PR	6	2,153	15.7
	LNKS-PR-TR	13	238	3.8
615,981	LRQN	204	—	53.1
8,901	LNKS-EX	7	20	33.8
(128 procs)	LNKS-PR	6	3,583	16.8
	LNKS-PR-TR	12	379	4.1

3.3 Initial condition inversion of atmospheric contaminant transport

In this section we consider an inverse problem governed by a parabolic PDE. Given observations of the concentration of an airborne contaminant $\{u_j^*\}_{j=1}^{N_s}$ at N_s locations $\{\boldsymbol{x}_j\}_{j=1}^{N_s}$ inside a domain Ω, we wish to estimate the initial concentration $d(\boldsymbol{x})$ using a convection-diffusion transport PDE model. The inverse problem is formulated as a constrained, least squares optimization problem:

$$\begin{aligned} \min_{u,d} \mathcal{J}(u,d) &\overset{\text{def}}{=} \frac{1}{2}\sum_{j=1}^{N_s}\int_{\Omega}(u-u^*)^2\,\delta(\boldsymbol{x}-\boldsymbol{x}_j)\;\boldsymbol{dx}\;dt + \frac{\beta}{2}\int_{\Omega} d^2\;\boldsymbol{dx} \\ \text{subject to}\quad \dot{u}-\nu\Delta u+\boldsymbol{v}\cdot\nabla u &= 0 \;\text{ in }\; \Omega\times(0,T) \\ \nu\nabla u\cdot\boldsymbol{n} &= 0 \;\text{ on }\; \Gamma\times(0,T) \\ u &= d \;\text{ in }\; \Omega\times\{t=0\} \end{aligned} \tag{23}$$

The first term in the objective functional $\mathcal{J}$ represents the least-squares misfit of predicted concentrations $u(\boldsymbol{x}_j)$ with observed concentrations $u^*(\boldsymbol{x}_j)$ at sensor locations, over a time horizon $(0,T)$, and the second term provides L^2 regularization of the initial condition d, resulting in a well-posed problem. The constraint is the convection-diffusion initial-boundary value problem, where u is the contaminant concentration field, d is the initial concentration, $\boldsymbol{v}$ is the wind velocity field (assumed known), and ν is the diffusion coefficient. For simplicity, a steady laminar incompressible Navier-Stokes solver is used to generate wind velocity fields over a terrain of interest.

Optimality conditions for (23) can be stated as as follows.

State equation:

$$\begin{aligned} \dot{u}-\nu\Delta u+\boldsymbol{v}\cdot\nabla u &= 0 \;\text{ in }\; \Omega\times(0,T) \\ \nu\nabla u\cdot\boldsymbol{n} &= 0 \;\text{ on }\; \Gamma\times(0,T) \\ u &= d \;\text{ in }\; \Omega\times\{t=0\} \end{aligned} \tag{24}$$

Adjoint equation:

$$
\begin{aligned}
-\dot{\lambda} - \nu\Delta\lambda - \nabla\cdot(\lambda\boldsymbol{v}) &= -\sum_{j=1}^{N_s}(u-u^*)\delta(\boldsymbol{x}-\boldsymbol{x}_j) \;\text{ in }\; \Omega\times(0,T) \\
(\nu\,\nabla\lambda + \boldsymbol{v}\lambda)\cdot\boldsymbol{n} &= 0 \;\text{ on }\; \Gamma\times(0,T) \\
\lambda &= 0 \;\text{ in }\; \Omega\times\{t=T\}
\end{aligned}
\tag{25}
$$

Decision equation:

$$
\beta\, u_0 - \lambda|_{t=0} = 0 \;\text{ in }\; \Omega \tag{26}
$$

Equations (24) are just the original forward convection-diffusion transport problem for the contaminant field. The adjoint convection-diffusion problem (25) resembles the forward problem, but with some essential differences. First, it is a terminal value problem; that is, the adjoint λ 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 (26) is in the present case of L^2 regularization an algebraic equation. Together, (24), (25), and (26) furnish a coupled system of linear PDEs for (u, λ, d).

The principal difficulty in solving this system is that—while the forward and adjoint transport problems are evolution equations—the KKT optimality system is a *coupled boundary value problem in 4D space-time.* As in the acoustic inversion example, the 4D space-time nature of (24)–(26) presents prohibitive memory requirements for large scale problems, and thus we consider reduced space methods. In fact, the optimality system is a linear system, since the state equation is linear in the state, and the decision variable appears linearly. Block elimination produces a reduced Hessian that has condition number independent of the mesh size (it is spectrally equivalent to a compact perturbation of the identity). However, a preconditioner capable of reducing the number of iterations is still critical, since each CG iteration requires one state and one adjoint convection-diffusion solve. We are unable to employ the limited memory BFGS preconditioner that was used for the acoustic inverse problem, since for this linear problem there is no opportunity for the preconditioner to reuse built-up curvature information. Instead, we appeal to multigrid methods for second kind integral equations and compact operators [34], [44], [57], [58], [62] to precondition the reduced Hessian system. Standard multigrid smoothers (e.g. for elliptic PDEs) are inappropriate for inverse operators and instead a smoother that is tailored to the spectral structure of the reduced Hessian must be used; for details see [4].

The optimality system (24)–(26) is discretized by SUPG-stabilized finite elements in space and Crank-Nicolson in time. 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. Figure 3 illustrates solution of the inverse problem for a contaminant release scenario in the Greater Los Angeles Basin. As can be seen in the figure, the reconstruction of the initial condition is very accurate.

We next study the parallel and algorithmic scalability of the multigrid preconditioner. We take synthetic measurements on a 7 × 7 × 7 sensor array. CG is termi-

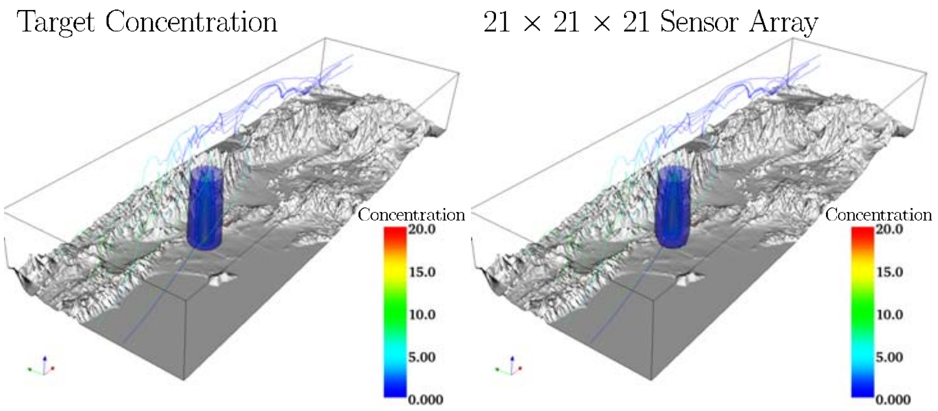

Figure 3. *Solution of a airborne contaminant inverse problem in the Greater Los Angeles Basin with onshore winds; Peclet number = 10. The target initial concentration is shown at left, and reconstructed initial condition on right. The measurements for the inverse problem were synthesized by solving the convection-diffusion equation using the target initial condition, and recording measurements on a* $21 \times 21 \times 21$ *uniform array of sensors. The mesh has 917,301 grid points; the problem has the same number of initial condition unknowns and 74 million total space-time unknowns. Inversion takes 2.5 hours on 64 AlphaServer processors at PSC. CG iterations are terminated when the norm of the residual of the reduced space equations is reduced by five orders of magnitude.*

nated when the residual of the reduced system has been reduced by six orders of magnitude. Table 4 presents fixed-size scalability results. The inverse problem is solved on a

Table 4. *Fixed size scalability of unpreconditioned and multigrid preconditioned inversion. Here the problem size is* $257 \times 257 \times 257 \times 257$ *for all cases. We use a three-level version of the multigrid preconditioner. 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, 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

$257 \times 257 \times 257 \times 257$ grid, i.e. there are 17×10^6 inversion parameters, and 4.3×10^9 total space-time unknowns in the optimality system (9). 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 5 presents isogranular scalability results. Here the problem size ranges from 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, 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 5 indicate substantially better multigrid performance. What accounts for this? First, the

Table 5. *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	
	d	(u, λ, d)		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

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; 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.

4 Conclusions

This chapter has given an overview of parallel algorithms for PDE-constrained optimization problems, focusing on reduced-space and full-space Newton-like methods. Examples illustrate application of the methods to elliptic, parabolic, and hyperbolic problems representing inverse, control, and design problems. A key conclusion is that an appropriate choice of optimization method can result in an algorithm that largely inherits the parallelism properties of the simulation problem. Moreover, under certain conditions, the combination of linear work per Krylov iteration, weak dependence of Krylov iterations on problem size, and independence of Newton iterations on problem size can result in a method that scales well with increasing problem size and number of processors. Thus, overall (parallel + algorithmic) efficiency follows.

There is no recipe for a general-purpose parallel PDE-constrained optimization method, just as there is no recipe for a general-purpose parallel PDE solver. The optimizer must be built around the best available numerical techniques for the state PDEs. The situation is actually more pronounced for optimization than it is for simulation, since new operators—the adjoint, the reduced Hessian, the KKT—appear that are not present in the simulation problem. PDE-constrained optimization requires special attention to preconditioning or approximation of these operators, a consideration that is usually not present in the design of general purpose optimization software.

However, some general themes do recur. For steady PDEs or whenever the state equations are highly nonlinear, a full-space method that simultaneously iterates on the state, adjoint, and decision equations can be significantly more effective than a reduced space method that entails satisfaction of (a linear approximation of) the state and adjoint equations at each optimization iteration. For example, in the optimal flow control example in Section 3.2, the LNKS method was able to compute the optimal control at high parallel efficiency and at a cost of just 5 simulations. LNKS preconditions the full space KKT matrix by an approximate factorization involving subpreconditioners for state, adjoint, and reduced Hessian operators, thereby capitalizing on available parallel preconditioners for the state equation. Alternatively, methods that seek to extend domain decomposition and multigrid preconditioners for direct application to the KKT matrix are being developed and show considerable promise in also solving the optimization problem in a small multiple of the cost of the simulation. Careful consideration of smoothing, intergrid transfer, and interface conditions is required for these methods. Like their counterparts for the PDE forward problem, parallelism comes naturally for these methods.

At the opposite end of the spectrum, for time-dependent PDEs that are explicit, linear, or weakly nonlinear at each time step, the benefit of full-space solution is less apparent, and reduced space methods may be required, if only for memory reasons. For small numbers of decision variables, quasi-Newton methods are likely sufficient, while for large (typically mesh-dependent) decision spaces, Newton methods with inexactly-terminated CG solution of the quadratic step are preferred. Preconditioning the reduced Hessian becomes essential, even when it is well-conditioned, since each CG iteration involves a pair of PDE solves (one state, one adjoint). For many large-scale inverse problems, the reduced Hessian has a "compact + identity" or "compact + differential" structure, which can be exploited to design effective preconditioners. Nevertheless, when the optimization problem is highly nonlinear in the decision space but weakly nonlinear or linear in the state space, such as for the inverse wave propagation problem described in Section 3.1, we can expect that the cost of solving the optimization problem will be many times that of the simulation problem.

A number of important and challenging issues were not mentioned. We assumed that the appropriate Jacobian, adjoint, and Hessian operators were available, which is rarely the case for legacy code. A key difficulty not discussed here is globalization, which must often take on a problem-specific nature (as in the grid/frequency continuation employed for the inverse wave propagation problem). Design of scalable parallel algorithms for mesh-dependent inequality constraints on decision and state variables remains a significant challenge. Parallel adaptivity for the full KKT system complicates matters considerably. Nonsmoothness and singularities in the governing PDEs, such as shocks, localization phenomena, contact, and bifurcation, can alter the convergence properties of the methods described here. Choosing the correct regularization is a crucial matter.

Nevertheless, parallel algorithms for certain classes of PDE-constrained optimization problems are sufficiently mature to warrant application to problems of exceedingly large scale and complexity, characteristic of the largest forward simulations performed today. For example, the inverse atmospheric transport problem described in Section 3.3 has been solved for 135 million initial condition parameters and 139 billion total space-time unknowns in less than 5 hours on 1024 AlphaServer processors at 86% overall efficiency. Such computations point to a future in which optimization for design, control, and inversion—and the decision-making enabled by it—become routine for the largest of today's terascale PDE simulations.

Acknowledgments

This work was supported in part by the U.S. Department of Energy through the SciDAC Terascale Optimal PDE Simulations (TOPS) Center and grants DE-FC02-01ER25477 and DE-FG02-04ER25646; the Computer Science Research Institute at Sandia National Laboratories; and the National Science Foundation under ITR grants ACI-0121667, EAR-0326449, and CCF-0427985 and DDDAS grant CNS-0540372. Computing resources at the Pittsburgh Supercomputing Center were provided under NSF PACI/TeraGrid awards ASC-990003P, ASC-010025P, ASC-010036P, MCA01S002P, BCS020001P, MCA04N026P. We thank the PETSc group at Argonne National Laboratory for their work in making PETSc available to the research community.

Bibliography

[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] M. F. Adams, H. Bayraktar, T. Keaveny, and P. Papadopoulos, *Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom*, in Proceedings of ACM/IEEE SC2004, Pittsburgh, 2004.

[3] V. Akçelik, J. Bielak, G. Biros, I. Epanomeritakis, A. Fernandez, O. Ghattas, E. Kim, D. O'Hallaron, and T. Tu, *High-resolution forward and inverse earthquake modeling on terascale computers*, in Proceedings of ACM/IEEE SC2003, Phoenix, November 2003.

[4] V. Akçelik, G. Biros, A. Drăgănescu, O. Ghattas, J. Hill, and B. van Bloeman Waanders, *Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants*, in Proceedings of ACM/IEEE SC2005, Seattle, November 2005.

[5] V. Akçelik, G. Biros, and O. Ghattas, *Parallel multiscale Gauss-Newton-Krylov methods for inverse wave propagation*, in Proceedings of ACM/IEEE SC2002, Baltimore, Maryland, November 2002.

[6] W. K. Anderson, W. D. Gropp, D. Kaushik, D. E. Keyes, and B. F. Smith, *Achieving high sustained performance in an unstructured mesh CFD application*, in Proceedings of ACM/IEEE SC99, Portland, November 1999.

[7] E. Arian and S. Ta'asan, *Multigrid one shot methods for optimal control problems: Infinite dimensional control*, Tech. Report ICASE 94-52, ICASE, NASA Langley Research Center, July 1994.

[8] ——, *Analysis of the Hessian for aerodynamic optimization*, Tech. Report 96-28, Institute for Computer Applications in Science and Engineering, 1996.

[9] U. Ascher and E. Haber, *A multigrid method for distributed parameter estimation problems*, ETNA, 15 (2003), pp. 1–12.

[10] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Web page*, 2001. http://www.mcs.anl.gov/petsc.

[11] W. BANGERTH, *Adaptive Finite Element Methods for the Identification of Distributed Parameters in Partial Differential Equations*, PhD thesis, University of Heidelberg, 2002.

[12] H. T. BANKS AND K. KUNISCH, *Estimation Techniques for Distributed Parameter Systems*, Birkhauser, 1989.

[13] R. BARTLETT, M. HEINKENSCHLOSS, D. RIDZAL, AND B. VAN BLOEMEN WAANDERS, *Domain decomposition methods for advection dominated linear-quadratic elliptic optimal control problems*, Tech. Report SAND 2005-2895, Sandia National Laboratories, April 2005.

[14] A. BATTERMANN AND M. HEINKENSCHLOSS, *Preconditioners for Karush-Kuhn-Tucker matrices arising in the optimal control of distributed systems*, in Optimal control of partial differential equations, W. Desch, F. Kappel, and K. Kunisch, eds., vol. 126 of International Series of Numerical Mathematics, Birkhäuser Verlag, 1998, pp. 15–32.

[15] A. BATTERMANN AND E. W. SACHS, *Block preconditioner for KKT systems in PDE-governed optimal control problems*, in Workshop on Fast Solutions of Discretized Optimization Problems, R. H. W. Hoppe, K.-H. Hoffmann, and V. Schulz, eds., Birkhäuser, 2001, pp. 1–18.

[16] R. BECKER AND B. VEXLER, *A posteriori error estimation for finite element discretization of parameter identification problems*, Numerische Mathematik, 96 (2004), pp. 435–459.

[17] L. BEILINA, *Adaptive hybrid FEM/FDM methods for inverse scattering problems*, Applied and Computational Mathematics, 2 (2003), pp. 119–134.

[18] M. BERGOUNIOUX, M. HADDOU, M. HINTERMÜLLER, AND K. KUNISCH, *A comparison of a Moreau-Yosida based active strategy and interior point methods for constrained optimal control problems*, SIAM Journal on Optimization, 11 (2000), pp. 495–521.

[19] L. BIEGLER, O. GHATTAS, M. HEINKENSCHLOSS, AND B. VAN BLOEMEN WAANDERS, eds., *Large-Scale PDE-Constrained Optimization*, vol. 30 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2003.

[20] L. T. BIEGLER, J. NOCEDAL, AND C. SCHMID, *A reduced Hessian method for large-scale constrained optimization*, SIAM Journal on Optimization, 5 (1995), pp. 314–347.

[21] G. BIROS, *Parallel Algorithms for PDE-Constrained Optimization and Application to Optimal Control of Viscous Flows*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 2000.

[22] G. BIROS AND O. GHATTAS, *Parallel Newton-Krylov algorithms for PDE-constrained optimization*, in Proceedings of ACM/IEEE SC99, Portland, November 1999.

[23] ——, *Parallel preconditioners for KKT systems arising in optimal control of viscous incompressible flows*, in Parallel Computational Fluid Dynamics 1999, D. E. Keyes, A. Ecer, J. Periaux, and N. Satofuka, eds., North-Holland, 1999.

[24] ——, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver*, SIAM Journal on Scientific Computing, 27 (2005), pp. 687–713.

[25] ——, *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, 27 (2005), pp. 714–739.

[26] A. E. BORZÌ, *Multigrid methods for optimality systems*, tech. report, University of Graz, Austria, 2003. Habilitation Thesis.

[27] ——, *Multigrid methods for parabolic distributed optimal control problems*, Journal of Computational and Applied Mathematics, 157 (2003), pp. 365–382.

[28] A. E. BORZÌ AND K. KUNISCH, *The numerical solution of the steady-state solid ignition model and its optimal control*, SIAM Journal on Scientific Computing, 22 (2000), pp. 263–284.

[29] A. E. BORZÌ, K. KUNISCH, AND M. VANMAELE, *A multigrid approach to the optimal control of solid fuel ignition problems*, Lecture Notes in Computational Science and Engineering, (2000), pp. 59–65.

[30] C. BUNKS, F. M. SALECK, S. ZALESKI, AND G. CHAVENT, *Multiscale seismic waveform inversion*, Geophysics, 50 (1995), pp. 1457–1473.

[31] S. S. COLLIS AND M. HEINKENSCHLOSS, *Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems*, Tech. Report CAAM TR02-01, Rice University, March 2002.

[32] M. DELFOUR AND J.-P. ZOLÉSIO, *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*, SIAM, 2001.

[33] R. DEMBO, S. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

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

[35] T. DREYER, B. MAAR, AND V. SCHULZ, *Multigrid optimization in applications*, Journal of Computational and Applied Mathematics, 120 (2000), pp. 67–84.

[36] H. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer, 1996.

[37] A. GRIEWANK, *Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation*, Optimization Methods and Software, 1 (1992).

[38] M. D. GUNZBURGER, ed., *Flow Control*, vol. 68 of IMA Math. Appl., Springer-Verlag, New York, 1995.

[39] M. D. GUNZBURGER, *Perspectives in Flow Control and Optimization*, SIAM, 2003.

[40] E. HABER, *Quasi-Newton methods for large scale electromagnetic inverse problems*, Inverse Problems, 21 (2004), pp. 305–317.

[41] ———, *A parallel method for large scale time domain electromagnetic inverse problems*. To appear, IMACS Journal, 2005.

[42] E. HABER, U. ASCHER, AND D. OLDENBURG, *Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach*, Geophysics, 69 (2004), pp. 1216–1228.

[43] E. HABER AND U. C. ASCHER, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001).

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

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

[46] S. B. HAZRA AND V. SCHULZ, *Simultaneous pseudo-timestepping for PDE-model based optimization problems*, BIT, 44 (2004), pp. 457–472.

[47] M. HEINKENSCHLOSS, *Time-domain decomposition iterative methods for the solution of distributed linear quadratic optimal control problems*, Journal of Computational and Applied Mathematics, 173 (2005), pp. 169–198.

[48] M. HEINKENSCHLOSS AND M. HERTY, *A spatial domain decomposition method for parabolic optimal control problems*, Tech. Report CAAM TR05-03, Rice University, May 2005.

[49] M. HEINKENSCHLOSS AND H. NGUYEN, *Domain decomposition preconditioners for linear-quadratic elliptic optimal control problems*, Tech. Report CAAM TR04-20, Rice University, November 2004.

[50] ———, *Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems*, Tech. Report CAAM TR04-01, Rice University, August 2004.

[51] M. HINTERMÜLLER AND M. HINZE, *Globalization of SQP-methods in control of the instationary Navier-Stokes equations*, Mathematical Modelling and Numerical Analysis, 36 (2002), pp. 725–746.

[52] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set method as a semi-smooth Newton method*, SIAM Journal on Optimization, 13 (2003), pp. 865–888.

[53] M. HINZE AND T. SLAWIG, *Adjoint gradients compared to gradients from algorithmic differentiation in instantaneous control of the Navier-Stokes equations*, Optimization Methods and Software, 18 (2003).

[54] M. HINZE, J. STERNBERG, AND A. WALTHER, *An optimal memory-reduced procedure for calculating adjoints of the instationary Navier-Stokes equations*, Optimal Control Applications and Methods, (2006). To appear.

[55] L. S. HOU, *Analysis and Finite Element Approximation of Some Optimal Control Problems Associated with the Navier-Stokes Equations*, PhD thesis, Carnegie Mellon University, Department of Mathematical Sciences, Pittsburgh, August 1989.

[56] L. S. HOU AND S. S. RAVINDRAN, *Numerical approximation of optimal flow control problems by a penalty method: Error estimates and numerical results*, SIAM Journal on Scientific Computing, 20 (1999), pp. 1753–1777.

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

[58] B. KALTENBACHER, M. KALTENBACHER, AND S. REITZINGER, *Identification of nonlinear $B-H$ curves based on magnetic field computations and multigrid methods for ill-posed problems*, European Journal of Applied Mathematics, 14 (2003), pp. 15–38.

[59] C. KELLEY AND E. SACHS, *Quasi-Newton methods and unconstrained optimal control problems*, SIAM Journal on Control and Optimization, 25 (1987), pp. 1503–1517.

[60] C. T. KELLEY AND E. W. SACHS, *Multilevel algorithms for constrained compact fixed point problems*, SIAM Journal on Scientific and Statistical Computing, 15 (1994), pp. 645–667.

[61] D. E. KEYES, P. D. HOVLAND, L. C. MCINNES, AND W. SAMYONO, *Using automatic differentiation for second-order matrix-free methods in PDE-constrained optimization*, in Automatic Differentiation of Algorithms: From Simulation to Optimization, Springer, 2002, pp. 35–50.

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

[63] D. A. KNOLL AND D. E. KEYES, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, Journal of Computational Physics, 193 (2004), pp. 357–397.

[64] R. M. LEWIS, *Practical aspects of variable reduction formulations and reduced basis algorithms in multidisciplinary optimization*, Tech. Report 95-76, Institute for Computer Applications in Science and Engineering, 1995.

[65] R. M. LEWIS AND S. G. NASH, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1811–1837.

[66] J.-L. LIONS, *Some Aspects of the Optimal Control of Distributed Parameter Systems*, SIAM, 1972.

[67] I. MALČEVIĆ, *Large-scale unstructured mesh shape optimization on parallel computers*, master's thesis, Carnegie Mellon University, 1997.

[68] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, 1999.

[69] O. PIRONNEAU, *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, 1983.

[70] E. POLAK, *Optimization: Algorithms and Consistent Approximations*, Springer, 1997.

[71] E. PRUDENCIO, R. BYRD, AND X.-C. CAI, *Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1305–1328.

[72] E. PRUDENCIO AND X.-C. CAI, *Parallel multi-level Lagrange-Newton-Krylov-Schwarz algorithms with pollution removing for PDE-constrained optimization*, 2006. Submitted.

[73] R. RANNACHER AND B. VEXLER, *A priori error estimates for the finite element discretization of elliptic parameter identification problems with pointwise measurements*, SIAM Journal on Control and Optimization, 44 (2005), pp. 1844–1863.

[74] W. W. SYMES AND J. J. CARAZZONE, *Velocity inversion by differential semblance optimization*, Geophysics, 56 (1991), pp. 654–663.

[75] A. TARANTOLA, *Inversion of seismic reflection data in the acoustic approximation*, Geophysics, 49 (1984), pp. 1259–1266.

[76] M. ULBRICH AND S. ULBRICH, *Superlinear convergence of affine-scaling interior-point Newton methods for infinite-dimensional nonlinear problems with pointwise bounds*, SIAM Journal on Control and Optimization, 6 (2000).

[77] M. ULBRICH, S. ULBRICH, AND M. HEINKENSCHLOSS, *Global convergence of trust-region interior-point algorithms for infinite-dimensional nonconvex minimization subject to pointwise bounds*, SIAM Journal on Control and Optimization, 37 (1999), pp. 731–764.

[78] S. ULBRICH, *Generalized SQP-methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization*. Submitted, 2004.

[79] C. VOGEL, *Computational Methods for Inverse Problems*, SIAM, 2002.

[80] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM Journal on Scientific Computing, 17 (1996), pp. 227–238.

[81] S. VOLKWEIN, *Mesh-independence for an augmented Lagrangian-SQP method in Hilbert spaces*, SIAM Journal on Optimization, 38 (2000), pp. 767–785.