The perspective from PDE-constrained optimization: Fast algorithms and their potential for quantifying uncertainty in inverse problems

Omar Ghattas
Joint work with: Volkan Akçelik, George Biros, Pearl Flath, Judy Hill, Bart van Bloemen Waanders, Lucas Wilcox

Sandia CSRI/NSF/AFOSR Workshop on Large Scale Inverse Problems and Quantification of Uncertainty
Santa Fe, NM, September 10–12, 2007

Some background on this workshop

- Fourth in a series of Santa Fe workshops (2001, 2004, 2005, 2007)
- First three workshops focused on deterministic optimization, in particular for large-scale problems governed by PDEs
- This workshop shifts the focus to estimation of uncertainty in the solution of *large scale* inverse problems
- Brings together and merges the perspectives of several different viewpoints (frequentist, Bayesian, Kalman, optimization, ...)
- Begin with tutorial presentations on Day 1 to help cross-fertilize ideas
- We will attempt to capture these perspectives in a post-workshop published volume

Goals of this particular tutorial

- Review a family of contemporary, powerful methods for optimization of systems governed by PDEs
- Argue that the structure of many PDE optimization problems (in particular for ill-posed inverse problems) permits:
  - very fast solution of the optimization problem
  - cheap and accurate approximation of the Hessian

  where “fast” and “cheap” mean: in a constant number of forward solves, independent of mesh size
- Illustrate the role of fast algorithms for PDE optimization in estimation of uncertainty in the linear Gaussian case
- Pose the question of how such structure-exploiting techniques can be brought to bear on nonlinear/non-Gaussian inverse problems

Fast algorithms: Caveats

- Presentation generally follows V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders, *Parallel algorithms for PDE-constrained optimization*, in Frontiers of Parallel Computing, M. Heroux, P. Raghaven, and H. Simon, eds, SIAM, 2006.
- Will focus on a class of Newton-like methods that exploit forward sovlers (will not mention multigrid and domain decomposition methods for optimization problems)
- Focus on local convergence (will not mention globalization techniques)
- Will consider only equality-constrained optimization (no mention of inequalities)
- Algorithms stated in finite-dimensional form

General form of a (discretized) PDE-constrained optimization problem

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

where:

- $\mathbf{u}$: state variables
- $\mathbf{d}$: optimization variables (inversion, control, or design)
- $\mathcal{J}$: objective function
- $\mathbf{c}$: residual of state equations
- $\mathbf{h}$: the residual of parameter or state inequality constraints

Example: Optimal flow control: Distributed control for steady-state Burgers equation

$$\min_{\boldsymbol{u},\boldsymbol{d}} \mathcal{J}(\boldsymbol{u},\boldsymbol{d}) \stackrel{\text{def}}{=} \frac{1}{2}\int_\Omega \nabla\boldsymbol{u}\cdot\nabla\boldsymbol{u}\;\boldsymbol{dx} + \frac{\rho}{2}\int_\Omega \boldsymbol{d}\cdot\boldsymbol{d}\;\boldsymbol{dx}$$

$$\text{subject to} \quad -\nu\Delta\boldsymbol{u} + (\nabla\boldsymbol{u})\boldsymbol{u} = \boldsymbol{d} \;\text{ in }\; \Omega$$

$$\boldsymbol{u} = \boldsymbol{g} \;\text{ on }\; \partial\Omega$$

where:

- $\boldsymbol{u}(\boldsymbol{x})$: velocity field (state variable)
- $\boldsymbol{d}(\boldsymbol{x})$: distributed source (control variable)
- $\boldsymbol{g}(\boldsymbol{s})$: boundary source (known function)
- ν: fluid viscosity (known parameter)
- Ω: domain
- $\partial\Omega$: boundary
- ρ: cost of the controls (known parameter)

Example: Optimal flow control: Optimality conditions

Form Lagrangian, where $\boldsymbol{\lambda}$ is an adjoint varible:

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{d}) \stackrel{\text{def}}{=} \mathcal{J}(\boldsymbol{u}, \boldsymbol{d}) + \int_\Omega \left[\nu \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{\lambda} + \boldsymbol{\lambda} \cdot (\nabla \boldsymbol{u})\boldsymbol{u} - \boldsymbol{d} \cdot \boldsymbol{\lambda}\right] \, \boldsymbol{dx}$$

Stationarity of Lagrangian w.r.t. $\boldsymbol{\lambda}$, $\boldsymbol{u}$, and $\boldsymbol{d}$ yields first order optimality system:

$$\begin{aligned}
-\nu \Delta \boldsymbol{u} + (\nabla \boldsymbol{u})\boldsymbol{u} &= \boldsymbol{d} \text{ in } \Omega && \textit{state eqn} \\
\boldsymbol{u} &= \boldsymbol{g} \text{ on } \partial\Omega && \\
-\nu \Delta \boldsymbol{\lambda} + (\nabla \boldsymbol{u})^T \boldsymbol{\lambda} - (\nabla \boldsymbol{\lambda})\boldsymbol{u} - \boldsymbol{\lambda} \operatorname{div} \boldsymbol{u} &= \Delta \boldsymbol{u} \text{ in } \Omega && \textit{adjoint eqn} \\
\boldsymbol{\lambda} &= \boldsymbol{0} \text{ on } \partial\Omega && \\
\rho \, \boldsymbol{d} + \boldsymbol{\lambda} &= \boldsymbol{0} \text{ in } \Omega && \textit{control eqn}
\end{aligned}$$

PDE-constrained optimization, discrete form

$$\min_{\mathbf{u},\mathbf{d}} \mathcal{J}(\mathbf{u},\mathbf{d})$$
$$\text{subject to } \mathbf{c}(\mathbf{u},\mathbf{d}) = \mathbf{0}$$

Form Lagrangian:

$$\mathcal{L}(\mathbf{u},\mathbf{d},\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \mathcal{J}(\mathbf{u},\mathbf{d}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{u},\mathbf{d})$$

where

- $\mathbf{u} \in \mathbb{R}^n$: state variables
- $\boldsymbol{\lambda} \in \mathbb{R}^n$: adjoint variables
- $\mathbf{d} \in \mathbb{R}^m$: optimization variables
- $\mathcal{J} \in \mathbb{R}$: objective function
- $\mathbf{c} \in \mathbb{R}^n$: state equations

PDE-constrained optimization, discrete form

Stationarity of Lagrangian w.r.t. $\mathbf{u}$, $\boldsymbol{\lambda}$, and $\mathbf{d}$ yields first order optimality conditions:

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

where:

- $\mathbf{g}_u \in \mathbb{R}^n$: gradient of $\mathcal{J}$ w.r.t. $\mathbf{u}$
- $\mathbf{g}_d \in \mathbb{R}^m$: gradient of $\mathcal{J}$ w.r.t. $\mathbf{d}$
- $\mathbf{J}_u \in \mathbb{R}^{n \times n}$: Jacobian of $\mathbf{c}$ w.r.t. $\mathbf{u}$
- $\mathbf{J}_d \in \mathbb{R}^{n \times m}$: Jacobian of $\mathbf{c}$ w.r.t. $\mathbf{d}$

Newton solver framework

Newton step on the optimality conditions yields KKT system:

$$\begin{bmatrix} \mathbf{W}_{uu} & \mathbf{W}_{ud} & \mathbf{J}_u^T \\ \mathbf{W}_{du} & \mathbf{W}_{dd} & \mathbf{J}_d^T \\ \mathbf{J}_u & \mathbf{J}_d & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{p}_u \\ \mathbf{p}_d \\ \boldsymbol{\lambda}_+ \end{Bmatrix} = - \begin{Bmatrix} \mathbf{g}_u \\ \mathbf{g}_d \\ \mathbf{c} \end{Bmatrix}$$

where:

- $\mathbf{W} \in \mathbb{R}^{(n+m)\times(n+m)}$: Hessian matrix of the Lagrangian function, partitioned into $\mathbf{u}$ and $\mathbf{d}$ blocks
- $\mathbf{p}_u \in \mathbb{R}^n$: Newton direction in $\mathbf{u}$ variables
- $\mathbf{p}_d \in \mathbb{R}^m$: Newton direction in $\mathbf{d}$ variables
- $\boldsymbol{\lambda}_+ \in \mathbb{R}^n$: updated adjoint variable

Convergence rate is locally quadratic (often mesh-independent)

Solution of the KKT system: Computational issues

- KKT matrix is of dimension $(2n + m) \times (2n + m)$
- n is mesh-dependent, m often mesh dependent
- factorization of KKT matrix not an option for 3D problems
- iterative solvers fail unless structure of KKT matrix exploited
- "inverting" $\mathbf{J}_u$ is a kernel step in a Newton-based PDE solver; therefore desirable to capitalize on PDE solvers and libraries
- will focus on several related methods that exploit the connection with PDE solvers

Solution of the KKT system: Newton-like methods that bootstrap forward PDE solvers

- Reduced space methods
  - Reduced Newton (RN): linearize then block eliminate
  - Reduced Newton-CG (RNCG): linearize then block eliminate then iterate
  - Reduced Quasi-Newton (RQN): linearize then block eliminate then approximate
  - Nonlinear Reduced Newton (NRN): block eliminate then linearize
- Full space methods
  - Lagrange-Newton-Krylov-Schwarz (LNKS): linearize, then iterate, then (approximately) block eliminate

Reduced Newton (RN)

Elimination of $\mathbf{p}_u$ from (linearized) state equation, then $\boldsymbol{\lambda}_+$ from adjoint equation, yields reduced space optimization equation to be solved for $\mathbf{p}_d$ at each Newton iteration:

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

where:

- $\mathbf{W}_{yz} \stackrel{\text{def}}{=} \mathbf{W}_{ud} - \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d$ is the *cross-Hessian*
- $\mathbf{W}_z \stackrel{\text{def}}{=} \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d - \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{ud} - \mathbf{W}_{du} \mathbf{J}_u^{-1} \mathbf{J}_d + \mathbf{W}_{dd}$ is the *reduced Hessian* (i.e. the Schur complement of the state and adjoint blocks)

Construction of $\mathbf{J}_u^{-1} \mathbf{J}_d$ term within $\mathbf{W}_z$ requires m solves of the linearized state equations; becomes intractable for large m.

Reduced Gauss Newton Conjugate Gradients (RGNCG)

- Discard off-diagonal Hessian blocks $\mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{ud}$ and $\mathbf{W}_{du} \mathbf{J}_u^{-1} \mathbf{J}_d$
- Gauss Newton reduced Hessian then becomes $\mathbf{W}_z^{GN} \stackrel{\text{def}}{=} \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d + \mathbf{W}_{dd}$
- Solve reduced Newton equation by conjugate gradients
- Form on-the-fly matrix–vector products with $\mathbf{W}_z^{GN}$; just one pair of forward/adjoint (linearized) PDE solves required at each CG iteration
- Terminate CG iterations early to prevent oversolving
- For many ill-posed inverse problems, reduced Hessian has (or can be preconditioned to have) "compact + identity" structure; results in mesh-independent convergence of CG iterations

Reduced Gauss Newton Conjugate Gradients (RGNCG)

At each Newton iteration:

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

- Just one pair of forward/adjoint (linearized) PDE solves required at each CG iteration
- Convergence of Newton iterations is linear; depends on spectral radius of $(\mathbf{W}_z^{GN})^{-1}(\mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{ud} + \mathbf{W}_{du} \mathbf{J}_u^{-1} \mathbf{J}_d)$
- For least squares inverse problems, convergence can be very fast since $\mathbf{W}_{ud}$ and $\mathbf{W}_{du}$ are often small

Reduced Limited Memory Quasi Newton (RLMQN)

- Form limited-memory quasi-Newton approximation ($\mathbf{B}_z$) of reduced Hessian
- Drop all other Hessian terms
- Convergence rate reduces to linear
- But only one pair of forward/adjoint (linearized) PDE solves required at each (quasi-) Newton iteration:

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

- Limiting case equivalent to nonlinear conjugate gradients

Nonlinear Reduced Newton (NLRN)

Nonlinear elimination of state and adjoint variables: can be thought of as solving the unconstrained optimization problem

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

- At each Newton iteration,
  - solve (nonlinear) state equation $\mathbf{c}(\mathbf{u}, \mathbf{d}) = \mathbf{0}$
  - solve adjoint equation $\mathbf{J}_u^T \boldsymbol{\lambda} = -\mathbf{g}_u$
  - then solve optimization equation $\mathbf{W}_z \mathbf{p}_d = -\mathbf{g}_d - \mathbf{J}_d^T \boldsymbol{\lambda}$
- Conjugate gradient and limited memory quasi-Newton variants are possible
- Advantage for time-dependent PDE-constrained optimization problems: state and adjoint histories need not be carried along at each Newton iteration
- But full nonlinear solution of state equations required

Example: Inverse acoustic wave propagation

PDE-constrained optimization problem:

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

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

where:

- $u^*(\boldsymbol{x}, t)$: observed pressure
- $u(\boldsymbol{x}, t)$: predicted pressure
- $f(\boldsymbol{x}, t)$: acoustic energy source (assume single source)
- $\boldsymbol{x}_j$: locations of N_r receivers
- $d(\boldsymbol{x})$: squared acoustic velocity distribution
- observation window: $t = (0, T)$

Example: Inverse acoustic wave propagation

Optimality conditions:

State equation:

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

Adjoint equation:

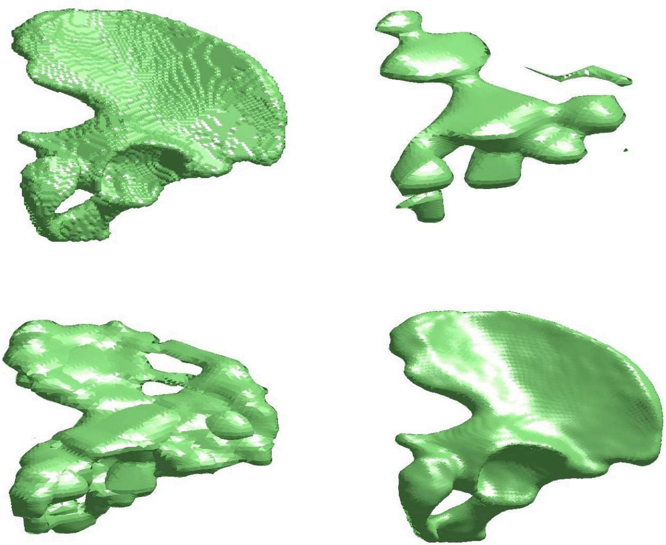
$$\begin{aligned} \ddot{\lambda} - \nabla \cdot d\nabla \lambda + \sum_{i=1}^{N_r} (u^* - u)\, \delta(\boldsymbol{x} - \boldsymbol{x}_j) &= 0 \ \text{ in } \ \Omega \times (0,T) \\ d\nabla\lambda \cdot n &= 0 \ \text{ on } \ \Gamma \times (0,T) \\ \lambda = \dot{\lambda} &= 0 \ \text{ for } \ \Omega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

Parameter equation:

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

Example: Inverse acoustic wave propagation

Reconstruction of hemipelvic bony geometry:



Example: Inverse acoustic wave propagation

Parallel scalability (strong) for 262,144 grid point problem:

procs	grid pts/proc	time (s)	time/gridpts/proc	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

Algorithmic scalability:

material grid	nonlinear iter	total linear iter	avg linear iter
125	17	144	8.5
729	12	249	21.0
4,913	12	396	33.0
35,937	25	439	17.6
274,625	19	370	19.5
2,146,689	22	436	19.8

Optimal boundary control of Navier-Stokes flow

PDE-constrained optimization problem:

$$\min_{\boldsymbol{u},p,\boldsymbol{d}} \mathcal{J}(\boldsymbol{u},p,\boldsymbol{d}) \stackrel{\text{def}}{=} \frac{\nu}{2}\int_{\Omega} \nabla\boldsymbol{u}\cdot\nabla\boldsymbol{u}\,\boldsymbol{dx} + \frac{\rho}{2}\int_{\partial\Omega_d} |\boldsymbol{d}|^2\,\boldsymbol{ds}$$

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

where:

- $\boldsymbol{u}(\boldsymbol{x})$: velocity field
- $p(\boldsymbol{x})$: pressure field
- $\boldsymbol{d}(\boldsymbol{s})$: velocity control on control boundary $\partial\Omega_d$
- $\boldsymbol{u}_g$: velocity source on Dirichlet boundary $\partial\Omega_u$

Optimal boundary control of Navier-Stokes flow

State equations:

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

Adjoint equations:

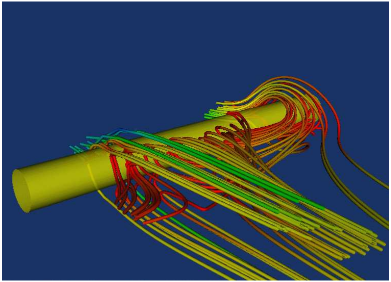
$$
\begin{aligned}
-\nu\Delta\boldsymbol{\lambda} + (\nabla\boldsymbol{u})^T\boldsymbol{\lambda} - (\nabla\boldsymbol{\lambda})\boldsymbol{u} + \nabla\mu &= \nu\Delta\boldsymbol{u} \ \text{ in } \ \Omega \\
\nabla\cdot\boldsymbol{\lambda} &= \boldsymbol{0} \ \text{ in } \ \Omega \\
\boldsymbol{\lambda}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

Control equations:

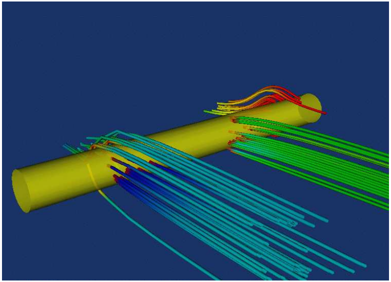
$$
\nu(\nabla\boldsymbol{\lambda} + \nabla\boldsymbol{u})\boldsymbol{n} - \rho\boldsymbol{d} = \boldsymbol{0} \ \text{ on } \ \partial\Omega_d
$$

Optimal boundary control of Navier-Stokes flow

no control:



control:



Optimal boundary control of Navier-Stokes flow

Algorithmic scalability on 64 and 128 procs for a doubling (roughly) of problem size:

states controls	**method**	**Newton its**	**avg KKT its**	**time (hr)**
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

Initial condition inversion of atmospheric contaminant transport

PDE-constrained optimizaton problem:

$$\min_{u,d} \frac{1}{2} \sum_{j=1}^{N_s} \int_\Omega \int_0^T (u - u^*)^2 \, \delta(\boldsymbol{x} - \boldsymbol{x}_j) \, \boldsymbol{dx} \, dt + \frac{\beta}{2} \int_\Omega d^2 \, \boldsymbol{dx}$$

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

where:

- $u^*(\boldsymbol{x}, t)$: contaminant observations
- $u(\boldsymbol{x}, t)$: contaminant predictions
- $\boldsymbol{x}_j$: N_s sensor locations
- $d(\boldsymbol{x})$ initial concentration
- $\boldsymbol{v}$: known velocity field
- ν: known diffusion coefficient

Initial condition inversion of atmospheric contaminant transport

State equation:

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

Adjoint equation:

$$-\dot{\lambda} - \nu\Delta\lambda - \nabla\cdot(\lambda\boldsymbol{v}) = -\sum_{j=1}^{N_s}(u-u^*)\delta(\boldsymbol{x}-\boldsymbol{x}_j) \ \text{ in } \ \Omega\times(0,T)$$

$$\begin{aligned} (\nu\,\nabla\lambda + \boldsymbol{v}\lambda)\cdot\boldsymbol{n} &= 0 \ \text{ on } \ \Gamma\times(0,T) \\ \lambda &= 0 \ \text{ in } \ \Omega\times\{t=T\} \end{aligned}$$

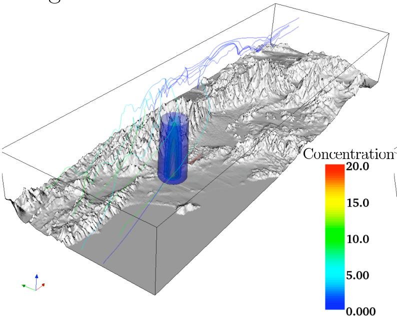
Decision equation:

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

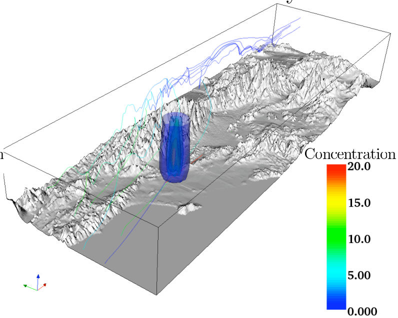
Initial condition inversion of atmospheric contaminant transport

Solution of a airborne contaminant inverse problem in the Greater Los Angeles Basin with onshore winds; Peclet = 10

Target Concentration



$21 \times 21 \times 21$ Sensor Array



Initial condition inversion of atmospheric contaminant transport

Fixed size scalability of unpreconditioned and multigrid preconditioned inversion; problem size is 257^4

CPUs	no preconditioner		multigrid	
	hours	efficiency	hours	efficiency
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

Isogranular scalability of unpreconditioned and multigrid preconditioned inversion:

grid	problem size		CPUs	no precond.		multigrid	
	d	(u, λ, d)		hours	iter	hours	iter
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

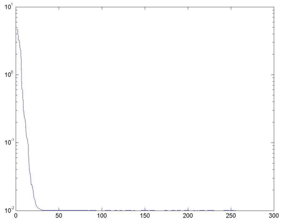
A peak into why CG is so effective for Hessians with "compact + identity" structure

At iteration k, CG solves the weighted least squares problem

$$\min_{P_k} ||e_k|| = \sum_i P_k\left[\lambda_i\right]^2 \xi_i^2 \lambda_i$$

where P_k is polynomial of order k and $e_0 = \sum_i \xi_i v_i, \quad W_z v_i = \lambda_i v_i$

Example spectrum of least squares portion of Hessian (from contaminant transport inverse problem):



Recall $\mathbf{W}_z^{GN} \stackrel{\text{def}}{=} \mathbf{J}_d^T \mathbf{J}_u^{-T} \mathbf{W}_{uu} \mathbf{J}_u^{-1} \mathbf{J}_d + \mathbf{W}_{dd}$

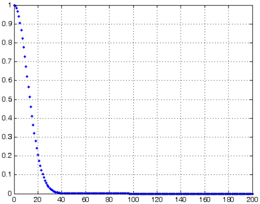
Analytical example of Hessian spectrum

1D convection-diffusion with periodic boundary conditions, inversion for initial condition with final time observations

$$\min_{u_0} \int_0^L (u - u^*(T))^2 dx + \frac{\beta}{2} \int_0^L u_0^2 dx$$

$$\begin{aligned} \text{where: } u_t - k u_{xx} + v u_x &= 0 \text{ in } (0, L) \times (0, T) \\ k u_x(0,t) &= k u_x(L,t) \text{ for } t \in (0,T) \\ u(0,t) &= u(L,t) \text{ for } t \in (0,T) \\ u &= u_0 \text{ in } (0,L) \times \{t = 0\} \end{aligned}$$

Hessian: jth eigenfunction: $e^{2\pi i j x/L}$, jth eigenvalue: $e^{-8j^2\pi^2 kT/L^2}$



Bayesian framework

(using Tarantola notation)

- Given:
  - a forward model $\mathbf{g}(\mathbf{m}) = \mathbf{d}$ relating model parameters $\mathbf{m}$ with observables $\mathbf{d}$, and its uncertainty
  - actual observations $\mathbf{d}_{\text{obs}}$ and their uncertainty
  - a "prior" estimate, of model parameters, $\mathbf{m}_{\text{prior}}$, and its uncertainty
- Seek a statistical characterization of model parameters consistent with observations, forward model, and prior model

Bayesian framework

Gaussian uncertainties, nonlinear forward model

If forward model uncertainty is Gaussian:

$$\theta(\mathbf{d}|\mathbf{m}) = \text{const.} \exp\left(-\frac{1}{2}\left(\mathbf{d}-\mathbf{g}(\mathbf{m})\right)^T \; \mathbf{C}_{\mathrm{T}}^{-1} \; \left(\mathbf{d}-\mathbf{g}(\mathbf{m})\right)\right)$$

forward model covariance

and observation uncertainty is Gaussian:

$$\rho_{\mathrm{D}}(\mathbf{d}) = \text{const.} \exp\left(-\frac{1}{2}\left(\mathbf{d}-\mathbf{d}_{\mathrm{obs}}\right)^T \; \mathbf{C}_{\mathrm{d}}^{-1} \; \left(\mathbf{d}-\mathbf{d}_{\mathrm{obs}}\right)\right)$$

observation covariance

and prior model parameter uncertainty is Gaussian:

$$\rho_{\mathrm{M}}(\mathbf{m}) = \text{const.} \exp\left(-\frac{1}{2}\left(\mathbf{m}-\mathbf{m}_{\mathrm{prior}}\right)^T \; \mathbf{C}_{\mathrm{M}}^{-1} \; \left(\mathbf{m}-\mathbf{m}_{\mathrm{prior}}\right)\right)$$

prior model parameter covariance

Then the posterior model parameter p.d.f. is given by:

$$\sigma_{\mathrm{M}}(\mathbf{m}) = k \exp\left(-S\left(\mathbf{m}\right)\right)$$

not Gaussian!

where the misfit function is:

$$S(\mathbf{m}) := \frac{1}{2}\left(\mathbf{g}(\mathbf{m})-\mathbf{d}_{\mathrm{obs}}\right)^T \; \mathbf{C}_{\mathrm{D}}^{-1} \; \left(\mathbf{g}(\mathbf{m})-\mathbf{d}_{\mathrm{obs}}\right) + \frac{1}{2}\left(\mathbf{m}-\mathbf{m}_{\mathrm{prior}}\right)^T \mathbf{C}_{\mathrm{M}}^{-1} \left(\mathbf{m}-\mathbf{m}_{\mathrm{prior}}\right)$$

$\mathbf{C}_{\mathrm{D}} = \mathbf{C}_{\mathrm{T}} + \mathbf{C}_{\mathrm{d}}$ forward model and measurement uncertainty combine

Bayesian framework

Gaussian uncertainties, linear forward problem

If modeling, measurement, and prior uncertainties are all Gaussian, and if in addition the forward problem is linear, i.e.,

$$\mathbf{G}\,\mathbf{m} = \mathbf{d}$$

Then the posterior p.d.f. for the model parameters is also Gaussian:

$$\sigma_{\mathrm{M}}(\mathbf{m}) = k\exp\left(-\,S\left(\mathbf{m}\right)\right)$$

where

$$\begin{aligned} 2\,S(\mathbf{m}) := &\left(\mathbf{G}\,\mathbf{m} - \mathbf{d}_{\mathrm{obs}}\right)^T \mathbf{C}_{\mathrm{D}}^{-1}\left(\mathbf{G}\,\mathbf{m} - \mathbf{d}_{\mathrm{obs}}\right) \\ &+ \left(\mathbf{m} - \mathbf{m}_{\mathrm{prior}}\right)^T \mathbf{C}_{\mathrm{M}}^{-1}\left(\mathbf{m} - \mathbf{m}_{\mathrm{prior}}\right) \end{aligned}$$

Bayesian framework

Gaussian uncertainties, linear forward problem (continued)

Since the posterior p.d.f. for the model parameters is Gaussian, its mean can be found by maximizing the p.d.f., which is equivalent to solving the weighted least squares optimization problem:

$$\tilde{\mathbf{m}} = \arg\min S(\mathbf{m}) := \|\mathbf{G}\,\mathbf{m} - \mathbf{d}_{\mathrm{obs}}\|^2_{\mathbf{C}_{\mathrm{D}}^{-1}} + \|\mathbf{m} - \mathbf{m}_{\mathrm{prior}}\|^2_{\mathbf{C}_{\mathrm{M}}^{-1}}$$

Note the connection with the regularization approach to inverse problems: $\mathbf{C}_{\mathrm{M}}^{-1}$ plays the role of the regularizer.

The posterior parameter covariance is given by the inverse of the Hessian:

$$\tilde{\mathbf{C}}_{\mathrm{M}} = \left(\mathbf{G}^T\,\mathbf{C}_{\mathrm{D}}^{-1}\,\mathbf{G} + \mathbf{C}_{\mathrm{M}}^{-1}\right)^{-1}$$

Note also the posterior p.d.f. for the data is also Gaussian, with mean and covariance given by:

$$\tilde{\mathbf{d}} = \mathbf{G}\,\tilde{\mathbf{m}} \qquad\qquad \tilde{\mathbf{C}}_{\mathrm{D}} = \mathbf{G}\,\tilde{\mathbf{C}}_{\mathrm{M}}\,\mathbf{G}^T$$

How to compute?

Even in the Gaussian/linear case, the problem can be intractable for high-dimensional parameter spaces: mean/covariance computation require "inverse" of Hessian matrix, which requires as many forward/adjoint solves as there are parameters.

However, least squares part of the Hessian is often a discretization of a compact operator; this suggests a low rank approximation using, e.g., a truncated spectral decomposition:

$$\tilde{\mathbf{C}}_{\mathrm{M}} = \overbrace{(\underbrace{\mathbf{G}^T \, \mathbf{C}_{\mathrm{D}}^{-1} \, \mathbf{G}}_{\text{truncated spectral decomposition}} + \mathbf{C}_{\mathrm{M}}^{-1})^{-1}}^{\text{Sherman-Morrison-Woodbury}}$$

The approximate Hessian can then be "inverted" in a constant number of forward solves.

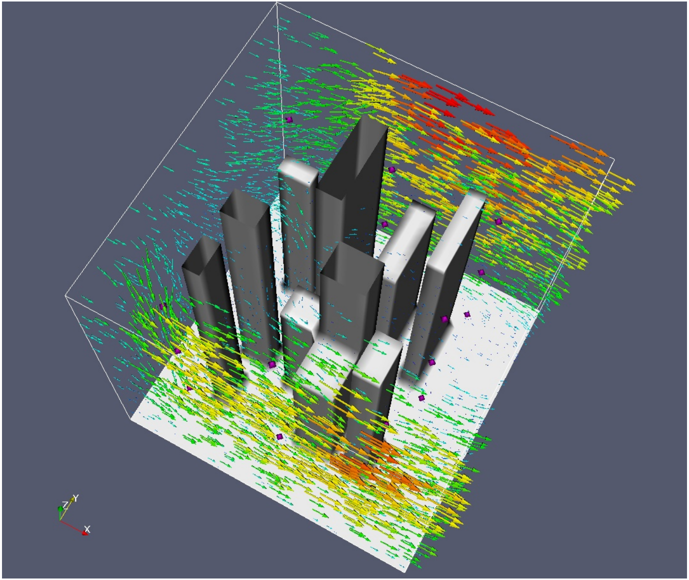
Example large-scale linear inverse problem

Estimation of initial condition of contaminants from pointwise observations of concentration and transport equation model

- Given:
  - Sparse sensor measurements of contaminant concentrations
  - Convection-diffusion PDE + BCs
  - Wind velocity field, contaminant diffusion constant
- We want to:
  - Estimate mean and covariance of initial concentration field
  - Predict evolution of concentration uncertainty

Example large-scale linear inverse problem

Flowfield and sensor locations in an urban canyon



Example large-scale linear inverse problem

Least squares optimization formulation

$$\min_{u,u_0} \sum_j \int_\Omega \int_0^T (u-u^*)^2 \delta(\mathbf{x}-\mathbf{x_j})\, d\mathbf{x}\, dt + \frac{\beta}{2}\int_\Omega u_0^2\, d\mathbf{x}$$

$$\begin{aligned} u_t - k\triangle u + \mathbf{v}\cdot\nabla u &= 0 \text{ in } \Omega \times (0,T) \\ u &= u_0 \text{ in } \Omega \times \{t=0\} \\ k\nabla u \cdot \mathbf{n} &= 0 \text{ in } \Gamma_N \times (0,T) \\ u &= 0 \text{ on } \Gamma_D \times (0,T) \end{aligned}$$

u	contaminant concentration	u_0	initial condition
$\mathbf{v}$	wind velocity	k	diffusion coefficient
T	length of time window	β	regularization constant
$\mathbf{x}_j$	jth sensor location		

Example large-scale linear inverse problem

Optimality conditions

State equation:

$$\begin{aligned} u_t - k\triangle u + \mathbf{v}\cdot\nabla u &= 0 \text{ in } \Omega\times(0,T) \\ u &= u_0 \text{ in } \Omega\times\{t=0\} \\ k\nabla u\cdot\mathbf{n} &= 0 \text{ on } \Gamma_N\times(0,T) \\ u &= 0 \text{ on } \Gamma_D\times(0,T) \end{aligned}$$

Adjoint equation:

$$\begin{aligned} -p_t - k\triangle p - \nabla\cdot(p\mathbf{v}) &= -\sum_j (u-u^*)\delta(\mathbf{x}-\mathbf{x_j}) \text{ in } \Omega\times(0,T) \\ p &= 0 \text{ in } \Omega\times\{t=T\} \\ (k\nabla p + \mathbf{v}p)\cdot\mathbf{n} &= 0 \text{ on } \Gamma_N\times(0,T) \\ p &= 0 \text{ on } \Gamma_D\times(0,T) \end{aligned}$$

Control equation:

$$-\beta u_0 - p\big|_{t=0} = 0 \text{ in } \Omega$$

Example large-scale linear inverse problem

Constructon of the Hessian

Discretized optimality conditions:

$$\begin{pmatrix} B^T B & 0 & A^T \\ 0 & \beta I & -T^T \\ A & -T & 0 \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ p \end{pmatrix} = \begin{pmatrix} B^T B u^* \\ 0 \\ 0 \end{pmatrix}$$

Elimination of u and p blocks yields the equation for u_0:

$$(\mathbf{G}^T \mathbf{G} + \beta I) u_0 = -\mathbf{G}^T B u^*$$

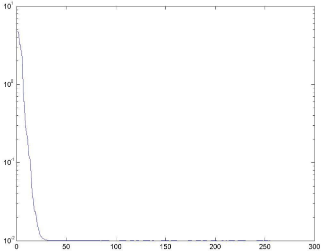
where

$$\begin{aligned} \mathbf{G} &= B A^{-1} T \text{ is the forward problem} \\ H &= \mathbf{G}^T \mathbf{G} + \beta I \text{ is the (reduced) Hessian} \\ &= \tilde{\mathbf{C}}_{\mathrm{M}}^{-1} \text{ (the inverse of the posterior parameter covariance)} \end{aligned}$$

Example large-scale linear inverse problem

Spectrum of the Hessian

Direct computation of H^{-1} intractable for large scale problems
However, make use of the exponential decay of eigenvalues of the compact part of the Hessian:



Retain only the dominant part of the spectrum

Example large-scale linear inverse problem

Approximating $\tilde{\mathbf{C}}_{\mathrm{M}}$ (aka H^{-1})

Low rank approximation of the compact part of H using a truncated spectral decomposition (e.g. via Lanczos):

$$H = (V\Lambda V^T + \beta I) \approx (V_r \Lambda_r V_r^T + \beta I)$$

Now use Sherman-Morrison-Woodbury formula to express posterior covariance of intial condition field:

$$\tilde{\mathbf{C}}_{\mathrm{M}} = H^{-1} \approx \frac{1}{\beta}(I - V_r D V_r^T)$$

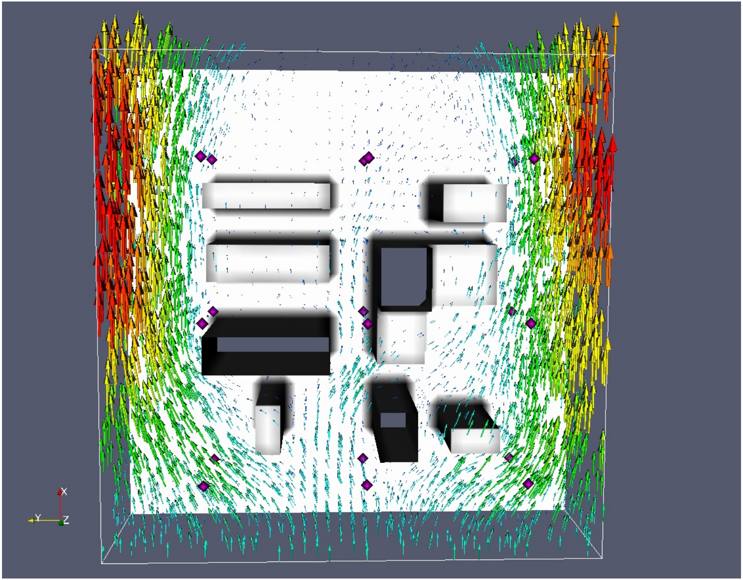
where

$$\begin{aligned} V, \Lambda &= \text{full eigenvectors and eigenvalues} \\ V_r, \Lambda_r &= \text{retained eigenvectors, eigenvalues} \\ D &= \text{diagonal matrix with } D_{ii} = \lambda_i / (\beta + \lambda_i) \end{aligned}$$

V_r and Λ_r can be computed at a cost of a constant number of forward/adjoint solves (proportional to dimension of range space of compact part of the Hessian).

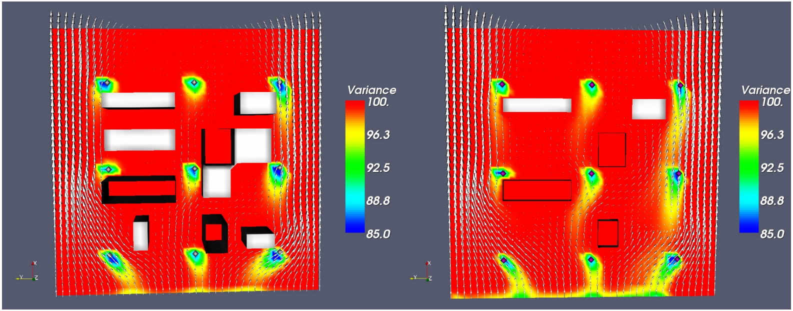
Example large-scale computation

Urban canyon with velocity field and sensors



Example large-scale linear inverse problem

Plot of variance field with superposed wind velocity field



Horizontal slice of variance field near ground (left) and near top of biuldings (right)

Conclusions and topics for further discussion

- Combination of Newton-like methods on the outside with Krylov methods on the inside often yield optimization methods that scale well, both algorithmically and w.r.t. number of processors
- Must be prepared to (approximately) solve adjoint PDEs and form action of Hessian on a vector
- When Hessian is a compact perturbation of the identity, convergence is mesh independent; with a good preconditioner, solution can often be found in a small multiple of cost of the forward problem
- As a result, fast low rank approximations of the compact portion of the Hessian are possible
- Numerical evidence has been given to illustrate the above, for problems with up to 130 million inversion parameters
- For linear/Gaussian inverse problems, low rank approximation of the Hessian serves as a *model reduction*
- For the general nonlinear/non-Gaussian statistical inverse problems, how can the structure embedded in the Hessian be exploited?