SAND2013-4975C

Solving massive-scale inve… using matrix-free sequential quadratic programming (SQP) methods

Denis Ridzal

Sandia National Laboratories

With M. Aguiló and J. Young (Sandia), and M. Heinkenschloss (Rice).

ASC™

Coupled Problems 2013, June-18-2013

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation,
a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's
National Nuclear Security Administration under contract DE-AC04-94AL85000.
Research sponsored by the NNSA Advanced Scientific Computing Program under CSAR.

Inverse problems illustrated

Formulations

Matrix-free full-space SQP

Preconditioning of optimality systems

Numerical results

Thermal inversion

If I give you:

- a metallic object;
- a hairdryer;
- a device to measure temperature;


Source: www.amazon.com


Source: www.wikipedia.com

Source: www.flir.com

can you figure out the object's composition?

You are not allowed to dissect the object, take material samples, etc.

Thermal inversion

We could:

- heat the object to steady state;
- measure its temperature on the surface;
- use a model of heat conduction, in which the object's material properties are *unknown*, to formulate and solve an inverse problem.

The output of the model – simulated temperature – must match the measured temperature.

Thermal inversion

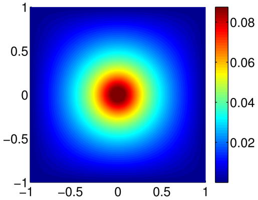
$$\underset{\{u,\kappa\}\in\, \mathcal{U}\times\mathcal{K}}{\text{Minimize}} \quad \frac{1}{2}\int_\Omega (u-\widehat{u})^2\,dx + \frac{\alpha}{2}\sqrt{|\nabla\kappa|^2+\beta}$$

subject to

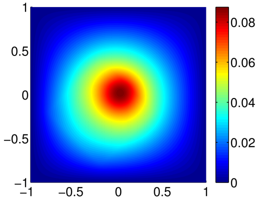
$$-\nabla\cdot(\kappa\nabla u) = f \text{ in } \Omega,$$

where $\mathcal{U} = \{u\colon\, u\in H^1(\Omega),\ u=0 \text{ on } \partial\Omega\}$, $\mathcal{K} = \{\kappa\colon\, \kappa\in H^1(\Omega),\ \kappa>0\}$ and $\alpha = 10^{-10}$, $\beta = 10^{-3}$, and f is a Gaussian heat source in the center of Ω, with amplitude 5 and width 0.1.

Temperature in uniform material



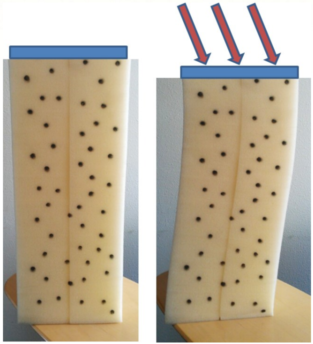
Our measured temperature $\widehat{u}$



Elastic inversion

If I give you:

- an elastic object;
- a device to apply force;
- a device to measure displacements;



can you figure out the object's composition?

You are not allowed to dissect the object, take material samples, etc.

Elastic inversion

We could:

- apply force to deform the object;
- measure particle displacement;
- use a model of elastic deformation, in which the object's material properties are *unknown*, to formulate and solve an inverse problem.

The output of the model – simulated displacement – must match the measured displacement.

Elastic inversion

$$\underset{\{u,\mu,\kappa\}\in\,\mathcal{U}\times\mathcal{G}\times\mathcal{K}}{\text{Minimize}} \quad \frac{1}{2}\int_{\Omega^0}(u_i-\widehat{u}_i)^2\,dx + R(\mu) + R(\kappa)$$

subject to

$$\begin{aligned} -(F_{ik}S_{kj})_{,j} &= 0 && \text{in} \quad \Omega^0, \\ (F_{ik}S_{kj})\,n_j &= \tau_i && \text{in} \quad \partial\Omega^0_\tau \end{aligned}$$

where $\mathcal{U}=\left\{u_i\colon u_i\in H^1\left(\Omega^0\right),\ u_i=0 \text{ on } \partial\Omega^0_u\right\}$, $\mathcal{G}=\{\mu\colon \mu\in L^2(\Omega^0),\ \mu>0\}$, and $\mathcal{B}=\{\kappa\colon \kappa\in L^2(\Omega^0),\ \kappa>0\}$.

— Here $\{u_i\}_{i=1,\dots,d}$ is the displacement in the i-th direction and τ_i is the surface traction in the i-th direction.

— The second Piola-Kirchhoff stress tensor for a Saint-Venant Kirchhoff material is given by $S_{ij}=\mathbf{C}_{ijkl}\,E_{kl}$, where the fourth-order tensor of elastic moduli is

$$\mathbf{C}_{ijkl} = (\kappa-\frac{2}{3})\,\delta_{ij}\,\delta_{kl} + \mu\,(\delta_{ik}\,\delta_{jl}+\delta_{il}\,\delta_{jk})\,.$$

The Green strain tensor is given by $E_{ij}=\frac{1}{2}\left(F_{ki}F_{kj}-\delta_{ij}\right)$, with the deformation gradient $F_{ij}=\frac{\partial u_i}{\partial x^0_j}+\delta_{ij}$, Kronecker delta δ_{ij}, and a reference material point x^0.

Inverse problems illustrated

Formulations

Matrix-free full-space SQP

Preconditioning of optimality systems

Numerical results

PDE-constrained optimization

— PDE-constrained optimization for *deterministic* inverse problems.

— Optimal design, optimal control and inverse problems at Sandia:

- heat conduction, acoustics, nonlinear elasticity
- coupled thermal/fluid flow, coupled elastic/acoustic
- magnetohydrodynamics, semiconductors, shock-hydrodynamics

— Computational regime: Often $\mathcal{O}(10,000)$ compute nodes ...

— Must use iterative linear system solvers ...

— **Scalable matrix-free optimization algorithms are needed.**

— **Scalability implies second-order, Newton-like algorithms.**

— **Since ca. 2006, matrix-free full-space SQP methods.**

Not covered in this talk: Statistical inverse problems.
(Bayesian methods, set-valued inverses of Butler/Estep 2011, etc.)

Formulations

Full-space formulation

$$\min_{u,g} \quad \frac{1}{2}\|u - \widehat{u}\|^2 + \frac{\alpha}{2}\|g\|^2$$

$$\text{s.t.} \quad A(g)u + Bg = f$$

- state u and parameter g
- gradually move to feasibility, for example a PDE solution, and optimality
- no A^{-1} in the formulation

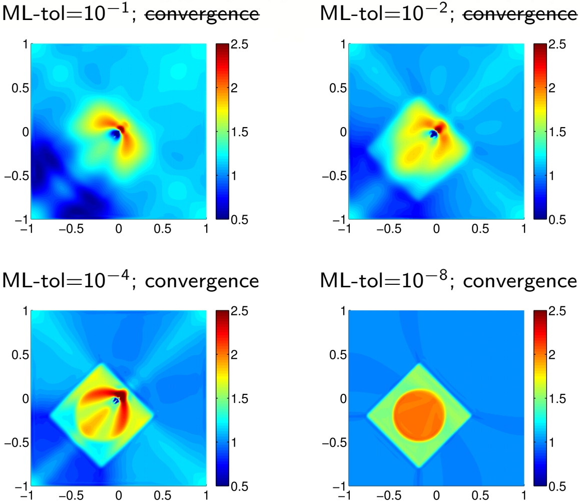
Reduced-space formulation

$$\min_{g} \quad \frac{1}{2}\|A(g)^{-1}(f - Bg) - \widehat{u}\|^2 + \frac{\alpha}{2}\|g\|^2$$

- parameter g only
- the constraint is eliminated at each optimization step, by solving $A(g)u = f - Bg$
- A^{-1} in the objective function!

An inadequate algorithm?

Solve reduced-space formulation of the thermal inversion problem, using Newton-CG method. Apply multigrid/ML to compute $A(g)^{-1}(f - Bg)$.



An inadequate formulation?

$$\min_{g} \quad \frac{1}{2}\|A(g)^{-1}(f - Bg) - \widehat{u}\|^2 + \frac{\alpha}{2}\|g\|^2$$

- Gradient cannot be computed accurately.
- Objective function cannot be computed accurately.
  - Difficult to evaluate progress.
  - Stagnation, inefficiency, false positives.

- Full-space SQP methods come to the rescue.

Inverse problems illustrated

Formulations

Matrix-free full-space SQP

Preconditioning of optimality systems

Numerical results

Sequential Quadratic Programming

Solve equality-constrained optimization problem:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

where $f : \mathcal{X} \to \mathbb{R}$ and $c : \mathcal{X} \to \mathcal{C}$, for some Hilbert spaces $\mathcal{X}$ and $\mathcal{C}$, and f and c are twice continuously Fréchet differentiable.

Define Lagrangian functional $\mathcal{L} : \mathcal{X} \times \mathcal{C} \to \mathbb{R}$:

$$\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, c(x) \rangle_{\mathcal{C}}$$

If *regular* point x_* is a local solution of the NLP, then there exists a $\lambda_* \in \mathcal{C}$ satisfying the *1st order necessary optimality conditions*:

$$\begin{aligned} \nabla_x f(x_*) + c_x(x_*)^* \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

Sequential Quadratic Programming

Newton's method applied to optimality conditions:

$$\begin{pmatrix} \nabla_{xx}\mathcal{L}(x_k,\lambda_k) & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} s \\ z \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x_k) + c_x(x_k)^*\lambda_k \\ c(x_k) \end{pmatrix}$$

If $\nabla_{xx}\mathcal{L}(x_k,\lambda_k)$ is positive definite on the null space of $c_x(x_k)$, the above KKT system is necessary and sufficient for solving the QP:

$$\begin{aligned} \min_{s\in\mathcal{X}} \quad & \frac{1}{2}\langle \nabla_{xx}\mathcal{L}(x_k,\lambda_k)s, s\rangle_{\mathcal{X}} + \langle \nabla_x\mathcal{L}(x_k,\lambda_k), s\rangle_{\mathcal{X}} + \mathcal{L}(x_k,\lambda_k) \\ \text{s.t.} \quad & c_x(x_k)s + c(x_k) = 0 \end{aligned}$$

Sequential Quadratic Programming

Newton's method applied to optimality conditions:

$$\begin{pmatrix} \nabla_{xx}\mathcal{L}(x_k,\lambda_k) & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} s \\ z \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x_k) + c_x(x_k)^*\lambda_k \\ c(x_k) \end{pmatrix}$$

If $\nabla_{xx}\mathcal{L}(x_k,\lambda_k)$ is positive definite on the null space of $c_x(x_k)$, the above KKT system is necessary and sufficient for solving the QP:

$$\begin{aligned} \min_{s\in\mathcal{X}} \quad & \frac{1}{2}\langle \nabla_{xx}\mathcal{L}(x_k,\lambda_k)s, s\rangle_{\mathcal{X}} + \langle \nabla_x\mathcal{L}(x_k,\lambda_k), s\rangle_{\mathcal{X}} + \mathcal{L}(x_k,\lambda_k) \\ \text{s.t.} \quad & c_x(x_k)s + c(x_k) = 0 \end{aligned}$$

Solve a sequence of *nonconvex* quadratic trust-region subproblems:

$$\begin{aligned} \min_{s\in\mathcal{X}} \quad & \frac{1}{2}\langle \nabla_{xx}\mathcal{L}(x_k,\lambda_k)s, s\rangle_{\mathcal{X}} + \langle \nabla_x\mathcal{L}(x_k,\lambda_k), s\rangle_{\mathcal{X}} + \mathcal{L}(x_k,\lambda_k) \\ \text{s.t.} \quad & c_x(x_k)s + c(x_k) = 0\,, \quad \|s\|_{\mathcal{X}} \le \Delta_k \end{aligned}$$

Possible incompatibility of constraints: composite-step approach.

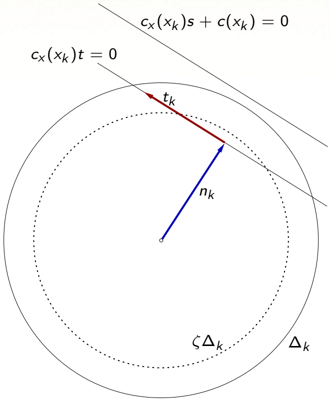
Composite-step approach for the solution of the trust-region subproblem

- **Trust-region step**:
  $s_k = n_k + t_k$

- **Quasi-normal step** n_k:
  reduces linear infeasibility

$$\min_{n \in \mathcal{X}} \quad \|c_x(x_k)n + c(x_k)\|_{\mathcal{C}}^2$$
$$\text{s.t.} \quad \|n\|_{\mathcal{X}} \le \zeta \Delta_k$$

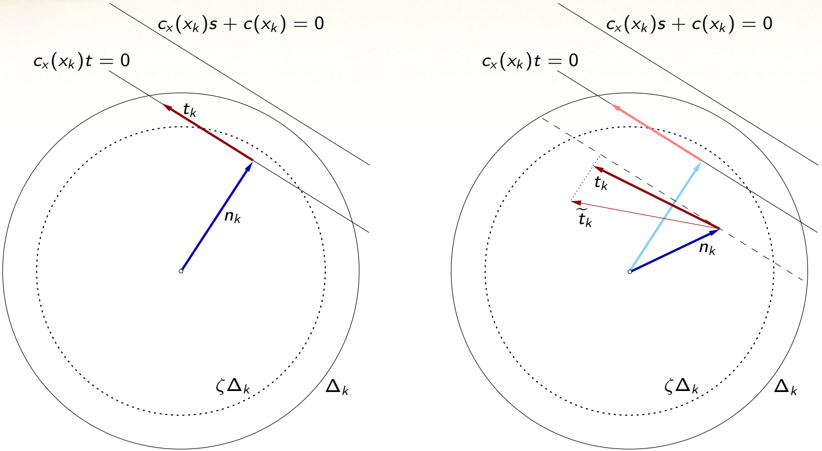
- **Tangential step** t_k:
  improves optimality while staying in the null space of the linearized constraints



$$\min_{t \in \mathcal{X}} \quad \frac{1}{2}\langle \nabla_{xx}\mathcal{L}(x_k,\lambda_k)(t+n_k), t+n_k\rangle_{\mathcal{X}} + \langle \nabla_x \mathcal{L}(x_k,\lambda_k), t+n_k\rangle_{\mathcal{X}} + \mathcal{L}(x_k,\lambda_k)$$
$$\text{s.t.} \quad c_x(x_k)t = 0\,, \quad \|t+n_k\|_{\mathcal{X}} \le \Delta_k$$

Omojokun (1989), Byrd, Hribar, Nocedal (1997), Dennis, El-Alem, Maciel (1997)

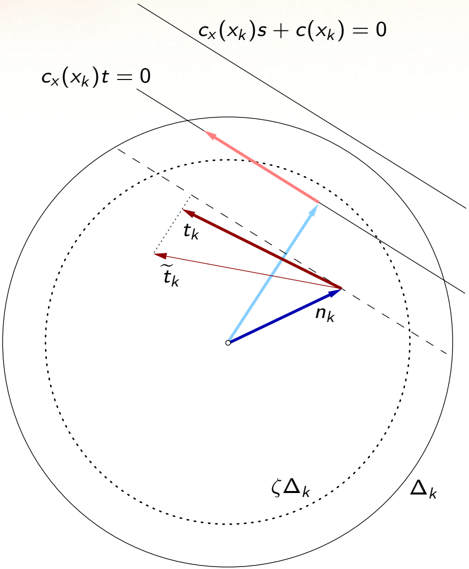
Composite step for matrix-free TR SQP



With matrix-free linear algebra — such as iterative linear solvers — the quasi-normal and tangential steps are computed **inexactly**.

Special care must be taken to guarantee convergence!

Sketch of the matrix-free TR SQP algorithm



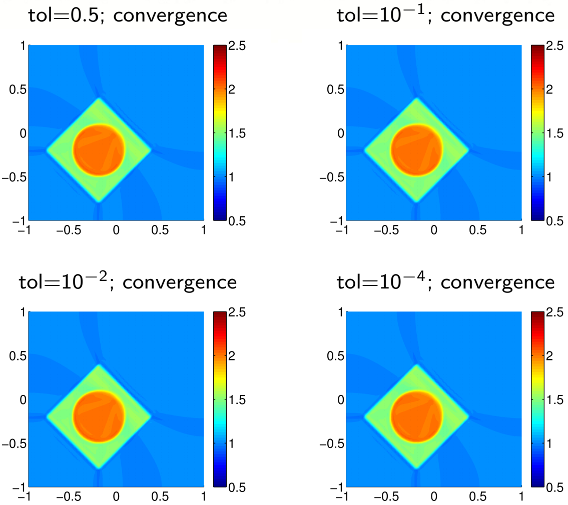
Composite step:

$s_k = n_k + t_k$

1. Compute quasi-normal step n_k using **inexact Powell dogleg**.
2. Solve tangential subproblem for $\widetilde{t}_k$ using **inexact projected Steihaug-Toint CG**.
3. Restore linearized feasibility, yielding tangential step t_k.
4. Update Lagrange multipliers λ_{k+1}.
5. Evaluate progress.

A robust algorithm

Use full-space matrix-free SQP to solve thermal inversion problem. Apply a single multigrid/ML cycle to accelerate solution of linear systems*.



Inverse problems illustrated

Formulations

Matrix-free full-space SQP

Preconditioning of optimality systems

Numerical results

Linear systems

I thought I saw a **Karush-Kuhn-Tucker (KKT) system**:

$$\begin{pmatrix} \nabla_{xx}\mathcal{L}(x_k, \lambda_k) & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix}$$

Difficult to solve in PDE-constrained optimization ...

Our SQP algorithm only solves simpler **augmented systems**:

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix}$$

- Newton direction for the quasi-normal step
- null-space projections in Steihaug-Toint CG
- feasibility-restoring null-space projection
- Lagrange multiplier computation

Control of stopping tolerances

Augmented systems are solved inexactly:

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} b^1 \\ b^2 \end{pmatrix} + \begin{pmatrix} r^1 \\ r^2 \end{pmatrix}.$$

The size of the nonzero residual $(r^1 \ r^2) \in \mathcal{X} \times \mathcal{C}$ is controlled via the general condition

$$\|r^1\|_{\mathcal{X}}^2 + \|r^2\|_{\mathcal{C}}^2 \leq \mathcal{R}(\|b^1\|_{\mathcal{X}}, \|b^2\|_{\mathcal{C}}, \|z\|_{\mathcal{X}}, \Delta_k, \xi)\,.$$

Here $\mathcal{R} : [\mathbb{R}^+]^4 \times (0,1) \to \mathbb{R}^+$ is tailored to each substep, and ξ is a prescribed *nominal stopping tolerance*.

The augmented system in PDE optimization

- Split variables x into state variables u and parameter variables g, i.e.:

$$x_k = (u_k, g_k)$$

- Write augmented system operator as a 3×3 block operator

$$\begin{pmatrix} I & 0 & c_u(x_k)^* \\ 0 & I & c_g(x_k)^* \\ c_u(x_k) & c_g(x_k) & 0 \end{pmatrix}$$

- Compress notation:

$$\begin{pmatrix} I & 0 & C_u^T \\ 0 & I & C_g^T \\ C_u & C_g & 0 \end{pmatrix}$$

The optimal preconditioner for augmented systems

$$\mathcal{A} = \begin{pmatrix} I & 0 & C_u^T \\ 0 & I & C_g^T \\ C_u & C_g & 0 \end{pmatrix} \qquad \mathcal{P}_{\mathcal{A}}^* = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (C_u C_u^T + C_g C_g^T)^{-1} \end{pmatrix}$$

- Independent of the objective function.
- Uses only the available PDE solvers and preconditioners:

$$(C_u C_u^T)^{-1} = C_u^{-T} C_u^{-1} = \text{'adjoint' solve } {\scriptstyle \text{following}} \text{ 'forward' solve}$$

- When the governing PDEs are elliptic, we prove a tight clustering of the eigenvalues of the preconditioned matrix $\mathcal{P}_{\mathcal{A}}\mathcal{A}$ around 3 values:

$$\frac{1}{2}(1-\sqrt{5}), \quad 1, \quad \frac{1}{2}(1+\sqrt{5})$$

- There is numerical evidence for extensions to other types of PDEs.
- Setup cost is amortized within each optimization iteration.

The optimal preconditioner for augmented systems

$$\mathcal{A} = \begin{pmatrix} I & 0 & C_u^T \\ 0 & I & C_g^T \\ C_u & C_g & 0 \end{pmatrix} \qquad \mathcal{P}_{\mathcal{A}} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & (C_u C_u^T)^{-1} \end{pmatrix}$$

- Independent of the objective function.
- Uses only the available PDE solvers and preconditioners:

$$(C_u C_u^T)^{-1} = C_u^{-T} C_u^{-1} = \text{‘adjoint’ solve}\ _{\text{following}}\ \text{‘forward’ solve}$$

- When the governing PDEs are elliptic, we prove a tight clustering of the eigenvalues of the preconditioned matrix $\mathcal{P}_{\mathcal{A}}\mathcal{A}$ around 3 values:

$$\frac{1}{2}(1-\sqrt{5}), \quad 1, \quad \frac{1}{2}(1+\sqrt{5})$$

- There is numerical evidence for extensions to other types of PDEs.
- Setup cost is amortized within each optimization iteration.

Inverse problems illustrated

Formulations

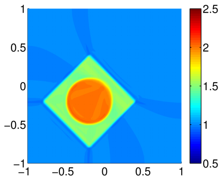
Matrix-free full-space SQP

Preconditioning of optimality systems

Numerical results

Thermal inversion in 2D

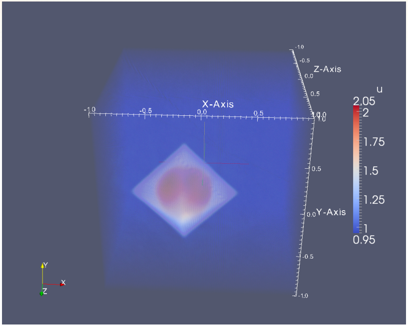
Mesh Size	Avg. GMRES Iters
64×64	4.7
128×128	4.6
256×256	4.7

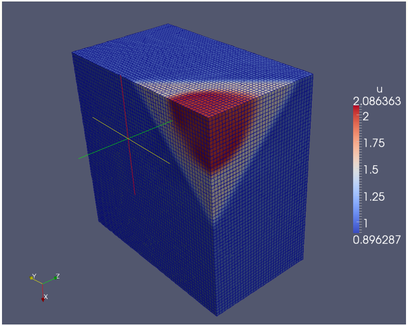


k	f(x_k)	|| c(x_k) ||	||L_x(x_k,lam_k)||	Delta_k	|| n_k ||	|| t_k ||	itcg
0	1.30240e-03	2.109265e-15	8.549061e-05	1.00e+02			
1	3.40849e-04	1.048282e-03	2.583983e-05	1.14e+02	4.03e-13	1.62e+01	8
2	7.39239e-05	8.268766e-04	7.580455e-06	1.45e+02	3.27e-01	2.07e+01	7
3	9.27937e-06	4.704437e-04	1.822981e-06	1.45e+02	1.89e-01	1.86e+01	6
4	7.53843e-07	7.155762e-04	2.777201e-07	1.45e+02	8.29e-02	1.78e+01	13
5	1.71122e-07	2.617447e-04	3.876637e-08	1.45e+02	2.22e-02	9.81e+00	22
6	1.24507e-07	5.506526e-05	3.219040e-09	1.45e+02	5.39e-03	4.31e+00	33
7	1.24282e-07	2.291248e-06	1.023034e-10	1.45e+02	1.06e-03	8.40e-01	35
8	1.24309e-07	3.396201e-09	1.979246e-12	1.45e+02	3.99e-05	2.93e-02	34

CG iters: 158

Thermal inversion in 3D





- Setup similar to the 2D example.
- One cycle of ML used as “inexact solver” for forward and adjoint problems within the augmented system preconditioner.
- One million elements, runs on my workstation.
- Parallelizes and scales as well as ML does:
  (1) 99% of compute time spent solving linear systems,
  (2) mesh-independent number of linear system solves.

Thermal inversion in 3D

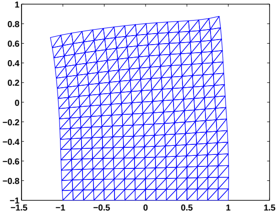
Avg. GMRES Iters = 4.8

k	f(x_k)	|| c(x_k) ||	||L_x(x_k,lam_k)||	Delta_k	|| n_k ||	|| t_k ||	itcg
0	7.73411e-07	1.101432e-15	3.318024e-08	1.00e+02			
1	1.33749e-07	1.218483e-04	1.770506e-08	1.00e+02	2.08e-13	1.00e+02	13
2	1.82025e-07	4.942232e-05	1.785741e-08	1.57e+02	1.00e-01	2.24e+01	14
3	2.96425e-08	2.296015e-04	6.570962e-09	1.96e+01	5.36e-03	1.96e+01	16
4	8.10717e-09	7.962145e-05	9.802347e-10	1.37e+02	1.50e-02	1.96e+01	10
5	6.10845e-09	6.779850e-05	1.724346e-10	1.37e+02	3.98e-03	2.16e+01	30
6	6.02571e-09	8.701002e-06	2.331547e-11	1.37e+02	1.03e-03	4.08e+00	27
7	6.02174e-09	1.318622e-07	2.630290e-12	1.37e+02	9.47e-05	2.47e+00	37
8	6.02176e-09	1.368711e-10	1.170072e-13	1.37e+02	2.10e-06	6.79e-02	33

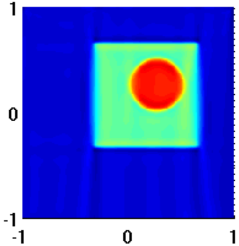
CG iters: 180

Nonlinear elastic inversion in 2D

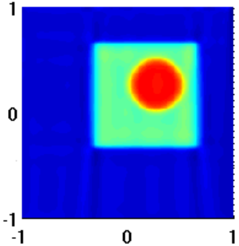
Nonlinear elastic deformation



Computed shear modulus μ



Computed bulk modulus κ



Nonlinear elastic inversion in 2D

Mesh Size	Avg. GMRES Iters
32×32	9.1
64×64	9.0
128×128	9.7

k	f(x_k)	|| c(x_k) ||	||L_x(x_k,lam_k)||	Delta_k	|| n_k ||	|| t_k ||	itcg
0	1.74452e-03	1.588574e-15	2.846439e-03	1.00e+02			
1	1.37411e-04	1.997871e-03	5.556572e-05	1.00e+02	3.16e-14	3.83e+00	56
2	4.73910e-06	4.318096e-03	2.443777e-06	1.00e+02	2.52e-01	6.38e+00	7
3	3.10439e-06	5.943678e-04	1.150126e-06	1.00e+02	1.13e-01	3.17e+00	7
4	2.14699e-06	6.900686e-04	9.066461e-08	1.00e+02	8.58e-03	3.52e+00	15
5	2.14079e-06	8.124125e-06	7.329380e-08	1.00e+02	6.31e-03	2.41e-01	8
6	2.13387e-06	1.199189e-05	4.643000e-09	1.00e+02	3.78e-05	4.29e-01	23
7	2.13393e-06	3.457890e-08	1.033532e-09	1.00e+02	8.60e-05	1.35e-02	11
8	2.13393e-06	2.451197e-09	4.289974e-11	1.00e+02	5.27e-07	6.11e-03	24

CG iters: 151

Summary

- Full-space matrix-free SQP enables the scalable solution of very large PDE-constrained optimization problems.
- Second-order convergence, iterative linear system solves.
- Demonstrated on inverse problems in heat transfer and elasticity.
- Recently implemented multi-frequency acoustic source inversion, with optimality systems approaching 3 billion variables; in collaboration with T. Walsh (Sandia) and W. Aquino (Duke).

- Applications to a variety of governing physics coming soon.
- Extensions to stochastic optimal design and inverse problems.

References

1. M. Heinkenschloss, D. Ridzal. "A Matrix-Free Trust-Region SQP Method for Equality Constrained Optimization", Preprint TR11-17, version May 2013, CAAM Tech. Report, Rice University, *Submitted to SIAM J. Optim.*
2. D. Ridzal, M. Aguiló and M. Heinkenschloss. "Numerical Study of a Matrix-Free Trust-Region SQP Method for Equality Constrained Optimization". Sandia report SAND2011-9346, Sandia National Laboratories, Dec. 2011. Available at `http://prod.sandia.gov/sand_doc/2011/119346.pdf`.
3. M. Heinkenschloss and D. Ridzal. "An Inexact Trust-Region SQP Method with Applications to PDE-Constrained Optimization". In *Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2007*, pages 613–620. Springer Verlag, 2008.
4. M. Heinkenschloss and D. Ridzal. "Integration of Sequential Quadratic Programming and Domain Decomposition Methods for Nonlinear Optimal Control Problems". In U. Langer, M. Discacciati, D. Keyes, O. Widlund, and W. Zulehner, editors, *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 69–80. Springer Verlag, 2008.
5. D. Ridzal. "Trust–Region SQP Methods with Inexact Linear System Solves for Large–Scale Optimization". PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 2006.

6. D. P. Kouri, M. Heinkenschloss, D. Ridzal, B. G. van Bloemen Waanders. "A Trust-Region Algorithm with Adaptive Stochastic Collocation for PDE Optimization Under Uncertainty", Preprint ANL/MCS-P3035-0912, September 2012. *Accepted at SIAM J. Sci. Comput.*
7. J. Young and D. Ridzal. "An Application of Random Projection to Parameter Estimation in Partial Differential Equations". *SIAM J. Sci. Comput.*, 34(4), A2344–A2365, 2012.