Solving the Euler Lagrange equations

$$
\begin{aligned}
\text{adjoint} \qquad & \mathcal{J}_y + c_y^\top \lambda = 0 \\
\text{state} \qquad & \mathcal{J}_u + c_u^\top \lambda = 0 \\
\text{constraint} \qquad & c(y,u) = 0
\end{aligned}
$$

Approximate the Hessian and solve at each iteration the KKT system

$$
\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}
$$

Solving the Euler Lagrange equations

In many applications approximate the Hessian by

$$\begin{pmatrix} \mathcal{J}_{yy} & \mathbf{O} & c_y^\top \\ \mathbf{O} & \mathcal{J}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Gauss-Newton SQP [Bock 89]

If $\mathcal{J}_{yy}$ and $\mathcal{J}_{uu}$ are positive semidefinite then the reduced Hessian is likely to be SPD.

Solving the KKT system

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & O \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

- Direct methods are (almost) out of the question!
- Multigrid methods for the KKT system
- The reduced Hessian
- Preconditioners

Solving the KKT system - multigrid

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & O \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

- Multigrid is a good tool to study the problem
- May use other techniques at the end
- Learn about the discretization/solver

Solving the KKT system - multigrid

Ascher & H. 2000, Kunish & Borzi 2003

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & O \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

- Check ellipticity of the continuous problem
- Check h-ellipticity of the discrete problem

Multigrid h-ellipticity

Look at the symbol Ta'asan

$$\widehat{H}(\theta) = \begin{pmatrix} \widehat{\mathcal{L}}_{yy} & & \widehat{c}_y^* \\ & \widehat{\mathcal{L}}_{uu} & \widehat{c}_u^* \\ \widehat{c}_y & \widehat{c}_u & 0 \end{pmatrix}$$

Compute the determinant

$$|\det(H)(\theta)| = \widehat{\mathcal{L}}_{yy}\widehat{c}_u^*\widehat{c}_u + \widehat{\mathcal{L}}_{uu}\widehat{c}_y^*\widehat{c}_y$$

Look at high frequencies

Example

Load problem

$$\min \ \frac{1}{2}\|y-d\|^2 + \frac{\alpha}{2}\|Lu\|^2 \quad \text{s.t } \Delta y - u = 0$$

$$\widehat{H}(\theta) = \begin{pmatrix} 1 & & \widehat{\Delta}_h \\ & \alpha\widehat{L} & 1 \\ \widehat{\Delta} & 1 & 0 \end{pmatrix}$$

Compute the determinant of the symbol ($\widehat{\Delta}_h = h^{-2}2(\cos(\theta_1) + cos(\theta_2) - 2)$)

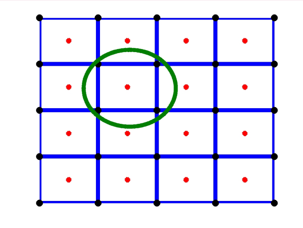
$$|\det(H)(\theta)| = 1 + \alpha\widehat{L}\widehat{\Delta}_h^2$$

Look at high frequencies

Solving the KKT system - multigrid

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & O \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Box smoothing - solve the equation locally



Solving the KKT system - multigrid

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & O \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Need:

- smoother - box smoothing, others?(in progress)
- coarse grid approximation
- solution on the coarsest grid (may not be so coarse)

Solving the KKT system - multigrid

- Case by case development
- Hard to generalize, even when BC change
- May worth the effort if the same type of problem is repeatedly solved

Solving the KKT system - The reduced Hessian

Nocedal & Wright 1999

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathbf{O} & c_y^\top \\ \mathbf{O} & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

- Eliminate s_y
  $c_y s_y + c_u s_u = \ldots$
- Eliminate s_λ
  $\mathcal{L}_{yy} s_u + c_u^\top s_\lambda = \ldots$
- Obtain an equation for s_u

$$H_r s_u = \underbrace{\left(c_u^\top c_y^{-\top} \mathcal{L}_{yy} c_y^{-1} c_u + \mathcal{L}_{uu} \right)}_{\text{the reduced Hessian}} s_u = \text{rhs}$$

The reduced Hessian in Fourier space

Use LFA to study the properties of the reduced Hessian.
Load problem

$$\min\ \frac{1}{2}\|y-d\|^2+\frac{\alpha}{2}\|Lu\|^2 \quad \text{s.t } \Delta y - u = 0$$

$$\widehat{H}(\theta) = \begin{pmatrix} 1 & & \widehat{\Delta}_h \\ & \alpha\widehat{L} & 1 \\ \widehat{\Delta} & 1 & 0 \end{pmatrix}$$

The symbol of the reduced Hessian ($\widehat{\Delta}_h = h^{-2}2(\cos(\theta_1)+cos(\theta_2)-2)$)

$$\widehat{\Delta}_h^{-2}+\alpha\widehat{L}$$

Very unstable for small α

More on the reduced Hessian method

$$H_r s_u = \left(c_u^\top c_y^{-\top} \mathcal{L}_{yy} c_y^{-1} c_u + \mathcal{L}_{uu}\right) s_u = \text{rhs}$$

- For QP with linear constraints the reduced Hessian is equivalent to the Hessian of the unconstrained approach
- The reduced Hessian represents an integro-differential equation
- Efficient solvers for the reduced Hessian is an open question, recent work [Biros & Dugan]

Even more on the reduced Hessian method

The reduced Hessian can be viewed as a block factorization of the (permuted) KKT system H. & Ascher 2001, Biros & Gahttas 2005, Dollar & Wathen 2006

$$\begin{pmatrix} c_y & \mathbf{O} & c_u \\ \mathcal{L}_{yy} & c_y^\top & \mathbf{O} \\ \mathbf{O} & c_u & \mathcal{L}_{uu} \end{pmatrix}^{-1} =$$

$$\begin{pmatrix} c_y^{-1} & \mathbf{O} & -JH_r^{-1} \\ \mathbf{O} & c_y^{-\top} & -c_y^{-\top}JH_r^{-1} \\ \mathbf{O} & \mathbf{O} & H_r^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{O} & \mathbf{O} \\ c_y^{-1} & \mathbf{I} & \mathbf{O} \\ -J^\top c_y^{-1} & -J^\top & \mathbf{I} \end{pmatrix}$$

$$J = c_y^{-1} c_u$$

$$H_r = J^\top J + \mathcal{L}_{uu}$$

Solving the KKT system - iterative methods and preconditioners

Solve

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Using some Krylov method (MINRES, SYMQMR, GMRES, ...)

- Indefinite
- Highly ill-conditioned
- A must: Preconditioner

Many of the preconditioners developed for general optimization problems are not useful

Solving the KKT system - iterative methods and preconditioners

Solve

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Using some Krylov method (MINRES, SYMQMR, GMRES, ...)

- Indefinite
- Highly ill-conditioned
- A must: Preconditioner

Many of the preconditioners developed for general optimization problems are not useful

Solving the KKT system - iterative methods and preconditioners

Solve

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Using some Krylov method (MINRES, SYMQMR, GMRES, ...)

- Indefinite
- Highly ill-conditioned
- A must: Preconditioner

Many of the preconditioners developed for general optimization problems are not useful

Solving the KKT system - iterative methods and preconditioners

$$\begin{pmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yu} & c_y^\top \\ \mathcal{L}_{yu}^\top & \mathcal{L}_{uu} & c_u^\top \\ c_y & c_u & \mathbf{O} \end{pmatrix} \begin{pmatrix} s_y \\ s_u \\ s_\lambda \end{pmatrix} = \text{rhs}$$

Preconditioners based on the approximate reduced Hessian method H. & Ascher 2001, Biros & Ghattas 2005

Preconditioners based on the reduced Hessian method

$$\begin{pmatrix} c_y & \mathbf{O} & c_u \\ \mathcal{L}_{yy} & c_y^\top & \mathbf{O} \\ \mathbf{O} & c_u & \mathcal{L}_{uu} \end{pmatrix}^{-1} \approx$$

$$\begin{pmatrix} \widehat{c}_y^{-1} & \mathbf{O} & -\widehat{J}H_r^{-1} \\ \mathbf{O} & \widehat{c}_y^{-\top} & -\widehat{c}_y^{-\top}\widehat{J}\widehat{H}_r^{-1} \\ \mathbf{O} & \mathbf{O} & \widehat{H}_r^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{O} & \mathbf{O} \\ \widehat{c}_y^{-1} & \mathbf{I} & \mathbf{O} \\ -\widehat{J}^\top \widehat{c}_y^{-1} & -\widehat{J}^\top & \mathbf{I} \end{pmatrix}$$

$$\widehat{J} = \widehat{c}_y^{-1} c_u$$

$$\widehat{H}_r = ??$$

Preconditioners based on the reduced Hessian method

$$\begin{pmatrix} c_y & \mathbf{O} & c_u \\ \mathcal{L}_{yy} & c_y^\top & \mathbf{O} \\ \mathbf{O} & c_u & \mathcal{L}_{uu} \end{pmatrix}^{-1} \approx \begin{pmatrix} \widehat{c}_y^{-1} & \mathbf{O} & -\widehat{J}H_r^{-1} \\ \mathbf{O} & \widehat{c}_y^{-\top} & -\widehat{c}_y^{-\top}\widehat{J}\widehat{H}_r^{-1} \\ \mathbf{O} & \mathbf{O} & \widehat{H}_r^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{O} & \mathbf{O} \\ \widehat{c}_y^{-1} & \mathbf{I} & \mathbf{O} \\ -\widehat{J}^\top \widehat{c}_y^{-1} & -\widehat{J}^\top & \mathbf{I} \end{pmatrix}$$

Approximating c_y and H_r

- $\widehat{c}_y$ - standard PDE approximation
- $\widehat{H}_r$ - BFGS, other QN, approximate inverse, ...
- Can prove mesh independence under some assumptions

Other Preconditioners

Other approaches

- Domain Decomposition, [Heinkenschloss 02]
- Augmented Lagrangian, [Greif & Golub 03]
- Schur complement based
- See excellent review paper by Benzi
  Everything you wanted to know about KKT systems but was afraid to ask

No magic bullet, application dependent (as they should be!)

Taking a step

$$\min \mathcal{J}(y,u) \quad \text{s.t}\; c(y,u) = 0$$

Guess u_0, y_0
while not converge

- Evaluate $\mathcal{J}_k, c_k$, $\nabla \mathcal{L}_k, c_y, c_u$ and an approximation to the Hessian (the KKT system)
- Approximately solve the KKT system for a step
- Take a (guarded) step
- Check if need to project to the constraint

Questions

while not converge

- Evaluate $\mathcal{J}_k, c_k$, $\nabla\mathcal{L}_k, c_y, c_u$ and an approximation to the Hessian (the KKT system)
  How accurate should the Hessian/Jacobian be?
- Approximately solve the KKT system for a step
  To what tolerance?
- Take a (guarded) step
  How should we judicially pick a step?
- Check if need to project to the constraint
  why and when should we project?

Questions

while not converge

- Evaluate $\mathcal{J}_k, c_k$, $\nabla\mathcal{L}_k, c_y, c_u$ and an approximation to the Hessian (the KKT system)
  How accurate should the Hessian/Jacobian be?
- Approximately solve the KKT system for a step
  To what tolerance?
- Take a (guarded) step
  How should we judicially pick a step?
- Check if need to project to the constraint
  why and when should we project?

Questions

while not converge

- Evaluate $\mathcal{J}_k, c_k$, $\nabla \mathcal{L}_k, c_y, c_u$ and an approximation to the Hessian (the KKT system)
  How accurate should the Hessian/Jacobian be?
- Approximately solve the KKT system for a step
  To what tolerance?
- Take a (guarded) step
  How should we judicially pick a step?
- Check if need to project to the constraint
  why and when should we project?

Questions

while not converge

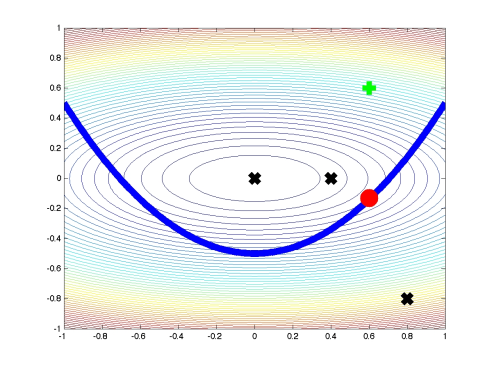
- Evaluate $\mathcal{J}_k, c_k$, $\nabla\mathcal{L}_k, c_y, c_u$ and an approximation to the Hessian (the KKT system)
  How accurate should the Hessian/Jacobian be?
- Approximately solve the KKT system for a step
  To what tolerance?
- Take a (guarded) step
  How should we judicially pick a step?
- Check if need to project to the constraint
  why and when should we project?

How well should we solve the KKT system?

- treat the problem as a system of nonlinear equations we can use inexact Newton's theory - ignore optimization aspects
- for traditional SQP algorithms require accurate solutions
- Can we use SQP with inaccurate solution of the sub-problem?

  Leibfritz & Sachs 1999, Heinkenschloss & Vicente 2001
- Recent work by Curtis Nocedal and Bird on inexact SQP methods, based on a penalty function

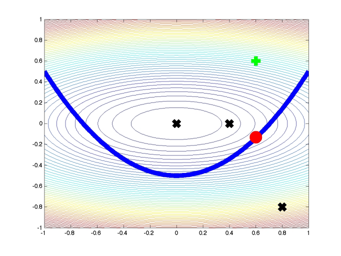
Choosing a step



The dilemma

- Should I decrease the Objective?
- Should I become more feasible?

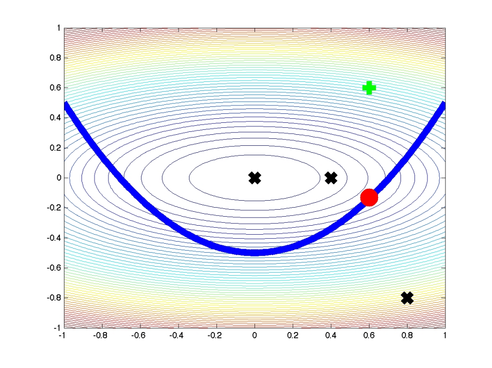
Choosing a step



merit function approach: $\mathcal{L}_\mu = f(y,u) + \mu h(c(y,u))$

- Use the L_1 or L_2 merit functions
- Disadvantage - need an estimate of the Lagrange multipliers

Choosing a step



Filter Fletcher & Leyffer 2002

- either reduce the objective or
- improve feasibility
- No need for Lagrange multipliers

Projecting back to the constraint

- In most cases feasibility is much more important than optimality
- Project the solution when getting close or before termination
- Can help with convergence (secondary correction)

Projecting back to the constraint

- In most cases feasibility is much more important than optimality
- Project the solution when getting close or before termination
- Can help with convergence (secondary correction)

Projecting back to the constraint

- In most cases feasibility is much more important than optimality
- Project the solution when getting close or before termination
- Can help with convergence (secondary correction)

Projecting back to the constraint - beyond optimization

- Accuracy of the optimization can be low
- Accuracy of the PDE should be high
- When should we project?

Multilevel

- Multilevel approach is computational effective
- In many cases, avoid local minima
- Help choosing parameters (e.g regularization, interior point)
- Hard to prove

Grid Sequencing

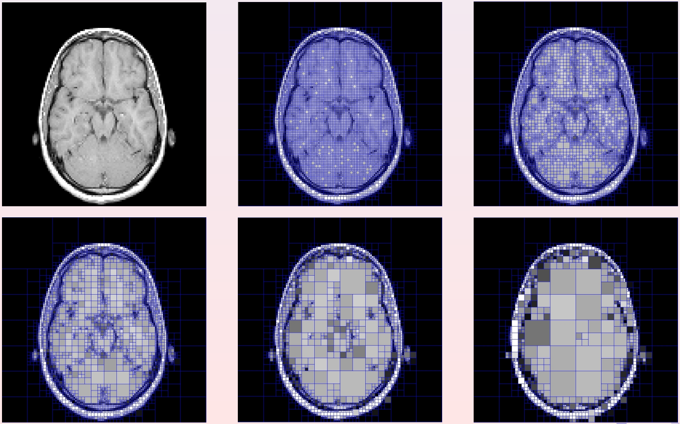
The problems we solve have an underline continuous structure.
Use this structure for continuation

Main idea: *Solution of the problem on a coarse grid can approximate the problem on a fine grid.*

Use coarse grids to evaluate parameters within the optimization. Moŕe , Burger, Ascher & H., H. & Modersitzki, H., H. & Benzi

Adaptive Multilevel Grid Sequencing

- Rather than refine everywhere, refine only where needed H., Heldman & Ascher [07], Bungrath [08]
- Requires data structures, discretization techniques, refinement techniques
- Can save an order of magnitude in calculation

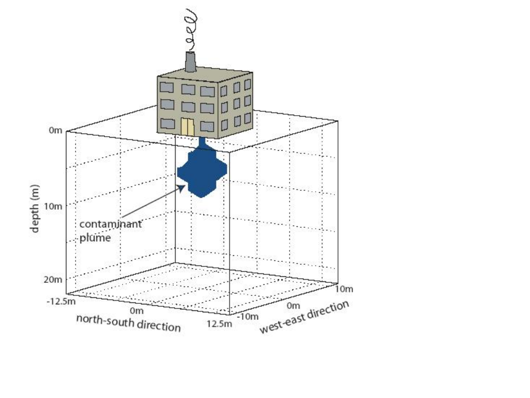


Examples

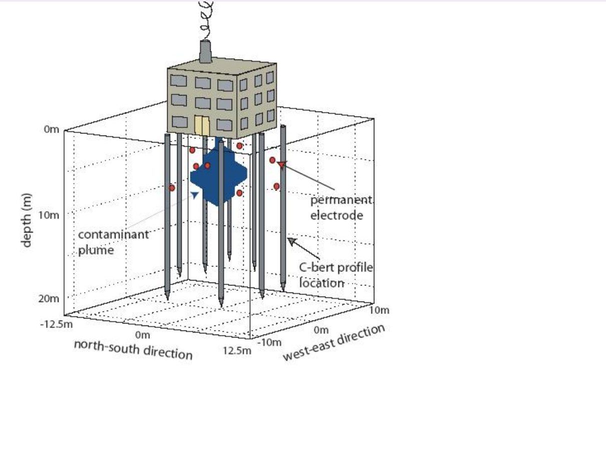
And this is how its really done Dora the explorer

Application: Impedance Tomography

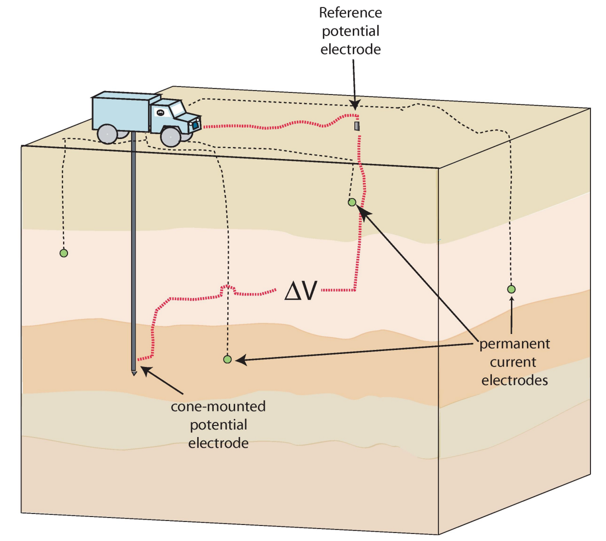
Joint project with R. Knight and A. Pidlovski, Stanford Environmental Geophysics Group



Application: Impedance Tomography



Application: Impedance Tomography



Application: Impedance Tomography

The mathematical problem

The constraint (PDE)

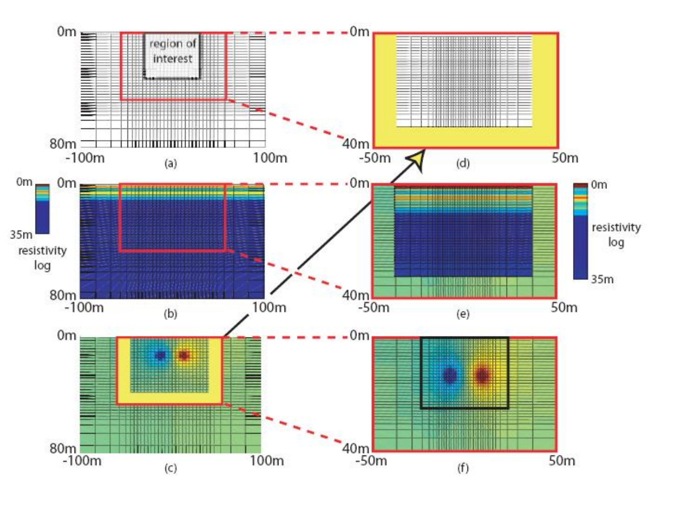
$$c(y,u) = \nabla \times \mu^{-1} \nabla \times y - i\omega\sigma y = i\omega s_j \quad j = 1 \ldots k$$

(with some BC)

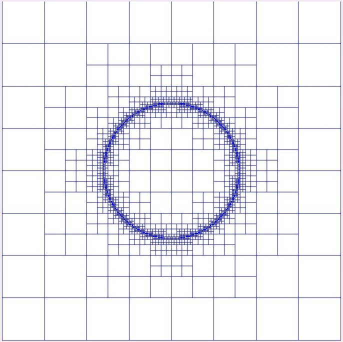
The Objective function

$$\min \quad \frac{1}{2} \underbrace{\|Q(y - y^{\mathrm{obs}})\|^2}_{\text{misfit}} + \underbrace{\alpha}_{\text{regpar}} \quad \overbrace{R(u)}^{\text{regularization}}$$

Discretization - I



Discretization - II



Discretization

use $128 \times 128 \times 64$ cells
$\#$ of states $= k \times \#$ of controls

In practical experiments $k \approx 10 - 1000$

The discrete mathematical problem

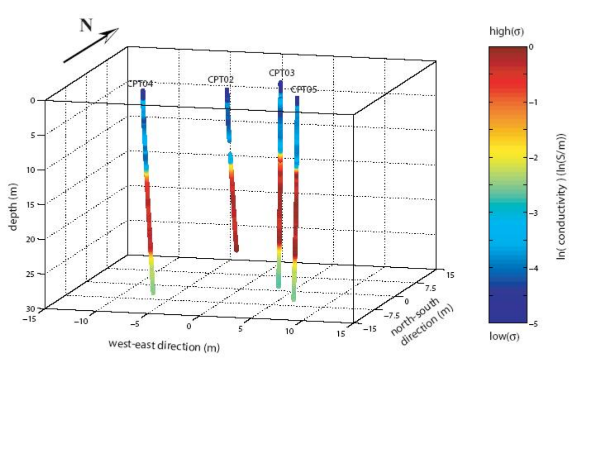
The constraint (PDE)

$$c_h(y_h, u_h) = A(u_h)y_h - q_h = D^T S(u_h) D y_h - q_h = 0$$

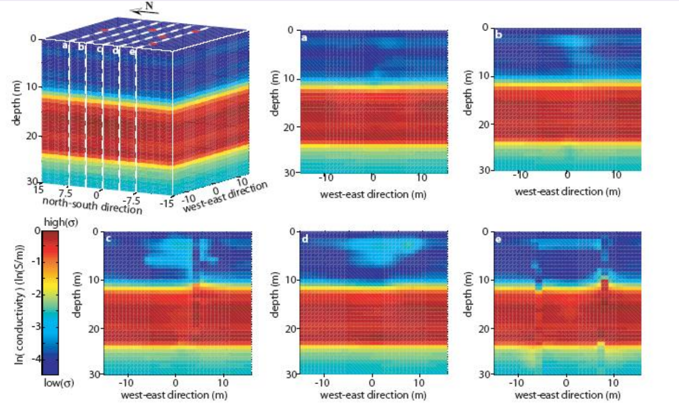
The Objective function

$$\min \quad \frac{1}{2} \underbrace{\|Q(y_h - y^{\text{obs}})\|^2}_{\text{misfit}} + \underbrace{\alpha}_{\text{regpar}} \overbrace{R(u_h)}^{\text{regularization}}$$

The Data - 63 sources



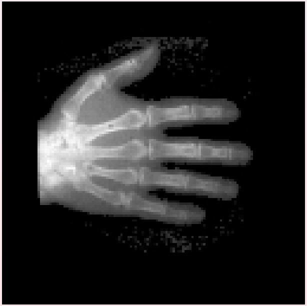
The Inversion



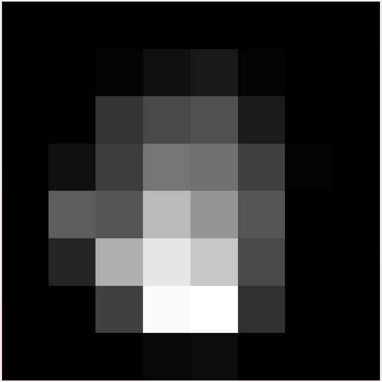
Application - Image Registration

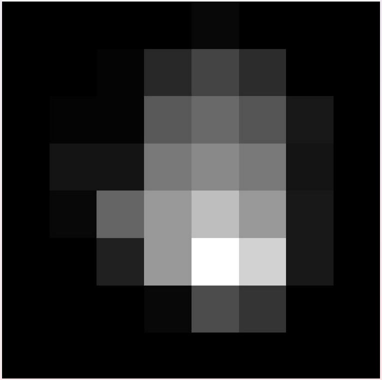
Joint work with S. Heldmann and J. Modesitzki, Lübeck, Germany

$$\begin{array}{ll} \min & \frac{1}{2}\|y(T) - R\|^2 + \frac{1}{2}\alpha S(u) \\ \text{s.t} & y_t + u^\top \nabla y = 0 \quad y(0) = y_0 \end{array}$$

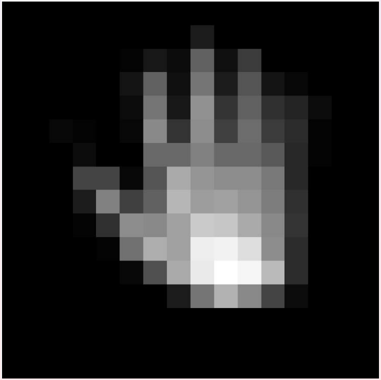


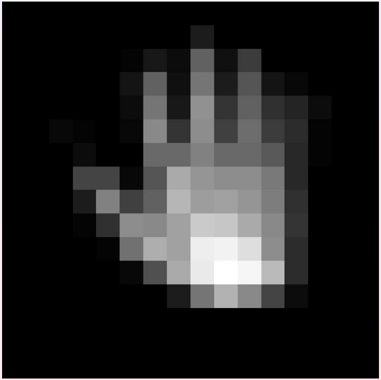
Example - ML



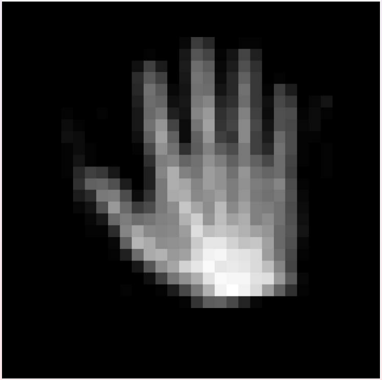


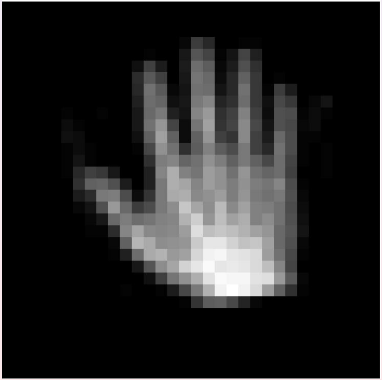
Example - ML



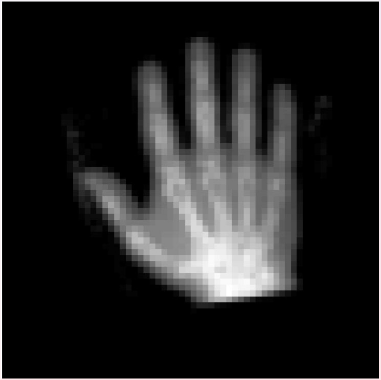


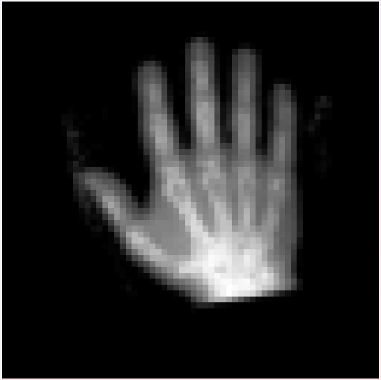
Example - ML





Example - ML



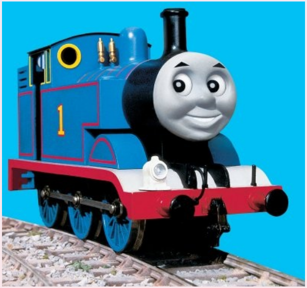


Example - ML

Model Problems

Sometimes, you can learn a lot from small things Thomas the engine



Goal

- PDE optimization problems are difficult to implement
- Suggest some *simple* model problems we can experiment with
- Develop optimization algorithms, preconditioners, grounded to reality
- Will not cover all PDE-optimization problems but not all PDE's are Poisson equation either
- Much of the development in PDE's was motivated by the 5 point stencil!

The problems/implementation

Parameter identification problems

- Assume smooth enough problems (discretize optimize not a problem)
- Consider elliptic, parabolic and hyperbolic problems
- Use regular grids and finite difference/volume for simplicity
- Code in matlab
- Modular, BYOPC (bring your own preconditioner)

The problems

The PDE's

- Elliptic

$$\nabla \cdot \exp^u \nabla y - q = 0; \quad \mathbf{n} \cdot y = 0$$

- Parabolic

$$y_t - \nabla \cdot \exp^u \nabla y = 0; \quad \mathbf{n} \cdot y = 0; \quad y(x,0) = y_0$$

- Hyperbolic

$$y_t - \vec{u}^\top \nabla y = 0; \quad \mathbf{n} \cdot y = 0; \quad y(x,0) = y_0$$

The code

Download:

`http://www.mathcs.emory.edu/ haber/code.html`

Very simple to get started (matlab demo ...)

Takes some time to run, elliptic problem on n^3 grid has $6n^3 + n^3 + 6n^3$ variables

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work

Outline/Summary

- Introduction
- Difficulties and PDE aspects
- The optimization framework
- Solving the KKT system
- Optimization algorithms
- Examples
- Summary and future work