Inexact Newton Methods for Nonlinear Optimization

Frank E. Curtis, Lehigh University

involving joint work with
Richard H. Byrd, University of Colorado at Boulder
Jorge Nocedal, Northwestern University
Olaf Schenk, University of Basel
Andreas Wächter, IBM T. J. Watson Research

SIAM CSE Conference 2011

March 4, 2011

Outline

Outline

Motivation

Interior-Point with Inexact Steps

Numerical Results

Summary and Future Work

Large-scale constrained optimization

Consider large-scale problems of the form

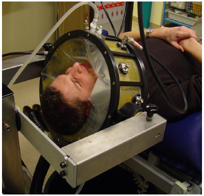
$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } c^{\mathcal{E}}(x) = 0 \\ &\qquad c^{\mathcal{I}}(x) \geq 0. \end{aligned}$$

For example, PDE-constrained optimization:

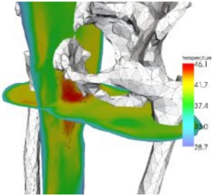
- True problem of interest is infinite-dimensional;
- Equality constraints include a discretized PDE;
- $x = (y, u)$ is composed of states y and controls u;
- Inequality constraints include control (and state) bounds.

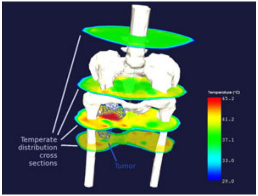
Motivating example 1: Hyperthermia treatment

- Regional hyperthermia is a cancer therapy that aims at heating large and deeply seated tumors by means of radio wave adsorption.



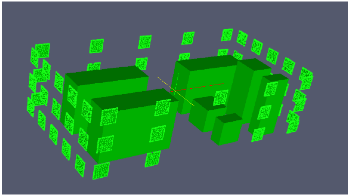
- Computer modeling and numerical optimization can be used to plan the therapy to heat the tumor while minimizing damage to nearby cells.





Motivating example 2: Server room cooling

- Heat generating equipment in a server room must be cooled.



- Numerical optimization can be used to help place and control air conditioners to satisfying cooling demands while minimizing costs.
- Problem suggested by Henderson, Lopez, Mello (IBM Research).

Strengths and weaknesses

We propose an algorithm for general-purpose large-scale nonlinear optimization:

- It can handle ill-conditioned/rank-deficient and nonconvex problems.
- Inexactness is allowed and controlled with implementable conditions.
- Algorithm is globally convergent, can handle control and state constraints.
- Numerical results are encouraging (but much more to do).

Aim to have an algorithm for PDE-constrained optimization, but so far:

- We solve for a single discretization.
- We use finite-dimensional norms.
- Our implementation does not exploit structure.
- We need further experimentation on interesting problems.

Questions and answers

- How do we compute search directions?
  - Sequential quadratic programming
  - Augmented Lagrangian methods
  - Interior-point methods
- How do we ensure global convergence?
  - KKT conditions (convex case)
  - Merit/penalty function
  - Filter
- How do we solve ill-conditioned problems?
  - Matrix modifications
  - Trust regions
- How do we handle nonconvexity?
  - Matrix modifications
  - Trust regions
- What if derivative matrices cannot be stored/factored?
  - Reduced space methods
  - Iterative methods on primal-dual system
  - Inexact calculations

Outline

Interior-point methods

- Add slacks ($s > 0$) to form the logarithmic-barrier subproblem

$$\boxed{\begin{aligned} \min\ & f(x) - \mu \sum_{i \in \mathcal{I}} \ln s^i \\ \text{s.t. } & c^{\mathcal{E}}(x) = 0 \\ & c^{\mathcal{I}}(x) = s \end{aligned}} \quad \Longrightarrow \quad \boxed{\begin{aligned} \nabla f(x) + \nabla c^{\mathcal{E}}(x)\lambda^{\mathcal{E}} + \nabla c^{\mathcal{I}}(x)\lambda^{\mathcal{I}} &= 0 \\ -\mu S^{-1} e - \lambda^{\mathcal{I}} &= 0 \\ c^{\mathcal{E}}(x) &= 0 \\ c^{\mathcal{I}}(x) - s &= 0 \end{aligned}}$$

- Newton iteration involves the linear system

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \mu S_k^{-2} & 0 & -I \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu S_k^{-1} e - \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}$$

Scaling and slack reset

- We begin by scaling the Newton system

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}$$

- Primal-dual matrix now has nicer properties
- The use of a slack reset

$$s_k \geq \max\{0, c^{\mathcal{I}}(x_k)\}$$

  allows easier infeasibility detection

Rank deficiency and nonconvexity

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}$$

If the constraint Jacobian is singular or ill-conditioned

- The system may be inconsistent
- The search direction $(d_k^x, \tilde{d}_k^s, \delta_k^{\mathcal{E}}, \delta_k^{\mathcal{I}})$ may blow up
- The line search may break down

If the Hessian is not positive definite on the null space of the Jacobian

- The system may be inconsistent
- The search direction $(d_k^x, \tilde{d}_k^s)$ may not be a descent direction
- The line search may fail

Matrix modifications

A common remedy is to modify the primal-dual matrix:

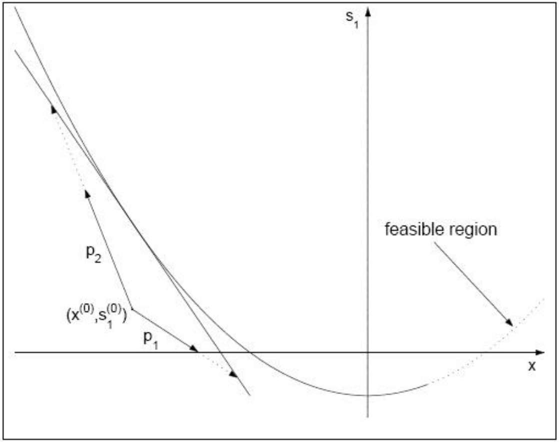
$$\begin{bmatrix} H_k+\xi_1 I & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k+\xi_1 I & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & -\xi_2 I & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & -\xi_2 I \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}$$

However, without matrix factorizations (i.e., no idea of the inertia)

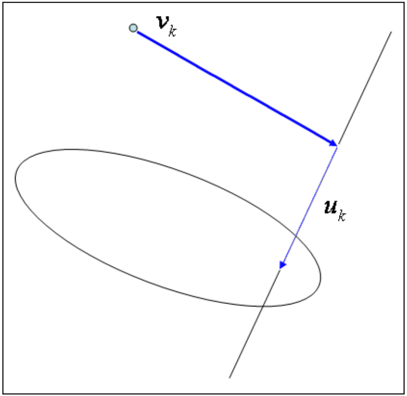
- When should these modifications be performed?
- What values should ξ_1 and ξ_2 take? How large?
- How do we ensure that in the end we solve the right problem?

Failure of line search methods

- Recall the counter-example of Wächter and Biegler (2000)



Our approach: Step decomposition

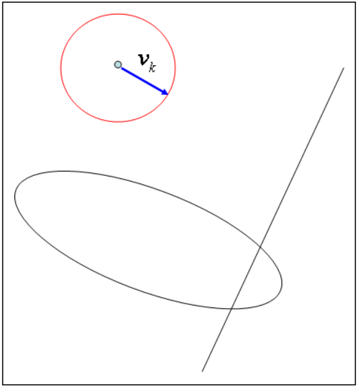


Normal step within a trust region

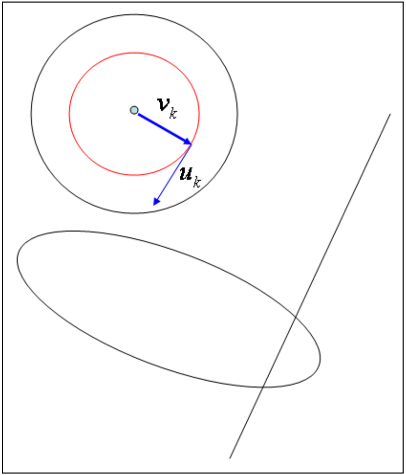
$$\min \frac{1}{2}\left\|\begin{bmatrix} c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} + \begin{bmatrix} \nabla c_k^{\mathcal{E}\,T} & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k \end{bmatrix}\begin{bmatrix} v_k^x \\ \tilde{v}_k^s \end{bmatrix}\right\|^2$$

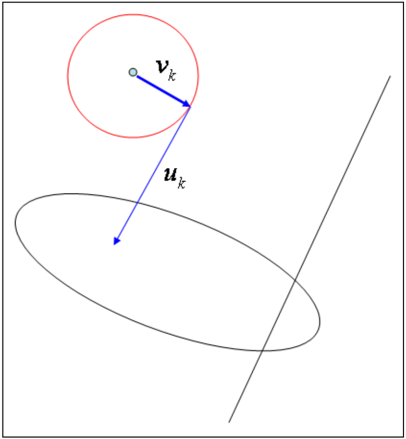
$$\text{s.t.} \quad \left\|\begin{bmatrix} v_k^x \\ \tilde{v}_k^s \end{bmatrix}\right\| \leq \omega \left\|\begin{bmatrix} \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & -S_k \end{bmatrix}\begin{bmatrix} c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}\right\|$$

- QP w/ trust region constraint
- Trust region radius is zero at first-order minimizers of infeasibility
- Radius updates automatically
- Solve w/ CG or inexact dogleg



Tangential step (within a trust region?)





Nonconvexity

- During primal-dual step computation, convexify the Hessian

$$\begin{bmatrix} H_k + \xi I & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k + \xi I & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix}$$

(i.e. increase ξ) whenever

$$\begin{bmatrix} u_k^x \\ \tilde{u}_k^s \end{bmatrix} > \psi \begin{bmatrix} v_k^x \\ \tilde{v}_k^s \end{bmatrix}$$

$$\begin{bmatrix} u_k^x \\ \tilde{u}_k^s \end{bmatrix}^T \begin{bmatrix} H_k + \xi I & 0 \\ 0 & \Omega + \xi I \end{bmatrix} \begin{bmatrix} u_k^x \\ \tilde{u}_k^s \end{bmatrix} < \theta \begin{bmatrix} u_k^x \\ \tilde{u}_k^s \end{bmatrix}^2$$

for some $\psi, \theta > 0$

- In our tests, modifications (often) are few and early
- We avoid having to develop conditions for inexact projections

Primal-dual step computation

We can be brave and approach the full system (avoid normal step)

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix}$$

... or compute a normal step, then approach the perturbed system

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ -\nabla c_k^{\mathcal{E}\,T} v_k^x \\ -\nabla c_k^{\mathcal{I}\,T} v_k^x + d_k^s \end{bmatrix}$$

Either way, how do we allow inexact solutions?

Consistency between the direction and the merit function

- In unconstrained optimization and nonlinear equations, there is consistency (even w/ inexact steps) between the step computation and merit function.
- In constrained optimization, however, our search direction is based on optimality conditions, but we judge progress by a merit function

$$\phi(x, s; \pi) \triangleq f(x) - \mu \sum_{i \in \mathcal{I}} \ln s^i + \pi \left\| \begin{bmatrix} c^{\mathcal{E}}(x) \\ c^{\mathcal{I}}(x) - s \end{bmatrix} \right\|$$

- Consistency is not automatic!
- Define the model of $\phi(x, s; \pi)$ at (x_k, s_k):

$$m_k(d^x, \tilde{d}^s; \pi) \triangleq f_k + \nabla f_k^T d^x - \mu \sum_{i \in \mathcal{I}} \ln s_k^i - \mu \tilde{d}^s + \pi \left(\left\| \begin{bmatrix} c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} + \begin{bmatrix} \nabla c_k^{\mathcal{E}\,T} & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k \end{bmatrix} \begin{bmatrix} d^x \\ \tilde{d}^s \end{bmatrix} \right\| \right)$$

- d_k is acceptable if

$$\Delta m_k(d_k^x, \tilde{d}_k^s; \pi) \triangleq m_k(0, 0; \pi_k) - m_k(d_k^x, \tilde{d}_k^s; \pi) \gg 0$$

- This ensures descent (and more)

Termination tests

$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ \nabla c_k^{\mathcal{E}\,T} & 0 & 0 & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k & 0 & 0 \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}} \end{bmatrix} = - \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} + \begin{bmatrix} \rho_k^x \\ \rho_k^s \\ \rho_k^{\mathcal{E}} \\ \rho_k^{\mathcal{I}} \end{bmatrix}$$

Search direction is acceptable if

- (TT1) dual residual is sufficiently small, tangential component is bounded by normal component or by sufficient convexity, and model reduction is sufficiently large for current penalty parameter
- (TT2) dual residual is sufficiently small, tangential component is bounded by normal component or by sufficient convexity, and sufficient progress in linearized feasibility (model reduction obtained with increase in penalty parameter)
- (TT3) sufficient progress in reducing dual infeasibility

Interior-point algorithm with inexact step computations

(C., Schenk, and Wächter (2010))
for $k = 0, 1, 2, \ldots$

- Approximately solve for a normal step (optional?)
- Iteratively solve the primal-dual equations until TT1, TT2, or TT3 is satisfied, modifying the Hessian matrix when appropriate
- If only termination test 2 is satisfied, then increase π
- Backtrack from $\alpha_k \leftarrow 1$ to satisfy fraction-to-the-boundary and sufficient decrease conditions for the merit function ϕ
- Update the iterate
- Reset the slacks

Convergence (inner iteration)

Assumption

The sequence $\{(x_k, s_k, \lambda_k^{\mathcal{E}}, \lambda_k^{\mathcal{I}})\}$ *is contained in a convex set* Ω *over which* f, $c^{\mathcal{E}}$, $c^{\mathcal{I}}$, *and their first derivatives are bounded and Lipschitz continuous*

Theorem

If all limit points of the constraint Jacobians have full row rank, then

$$\lim_{k\to\infty} \left\| \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \\ c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} \right\| = 0.$$

Otherwise,

$$\lim_{k\to\infty} \left\| \begin{bmatrix} \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & -S_k \end{bmatrix} \begin{bmatrix} c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} \right\| = 0$$

and if $\{\pi_k\}$ *is bounded, then*

$$\lim_{k\to\infty} \left\| \begin{bmatrix} \nabla f_k + \nabla c_k^{\mathcal{E}} \lambda_k^{\mathcal{E}} + \nabla c_k^{\mathcal{I}} \lambda_k^{\mathcal{I}} \\ -\mu e - S_k \lambda_k^{\mathcal{I}} \end{bmatrix} \right\| = 0$$

Convergence (outer iteration)

Theorem

If the algorithm yields a sufficiently accurate solution to the barrier subproblem for each $\{\mu_j\} \to 0$ *and if the linear independence constraint qualification (LICQ) holds at a limit point* $\bar{x}$ *of* $\{x_j\}$*, then there exist Lagrange multipliers* $\bar{\lambda}$ *such that the first-order optimality conditions of the nonlinear program are satisfied*

Outline

Motivation

Interior-Point with Inexact Steps

Numerical Results

Summary and Future Work

Implementation details

- Incorporated in IPOPT software package (Wächter, Laird, Biegler):
  - interior-point algorithm with inexact step computations;
  - flexible penalty function for promoting faster convergence (Curtis, Nocedal);
  - tests on $\sim$ 700 CUTEr problems yields robustness (almost) on par with original IPOPT.
- Linear systems solved with PARDISO (Schenk, Gärtner):
  - includes iterative linear system solvers, e.g., SQMR (Freund);
  - incomplete multilevel factorization with inverse-based pivoting;
  - stabilized by symmetric-weighted matchings.
- Server cooling room example coded w/ libmesh (Kirk, Peterson, Stogner, Carey)

Hyperthermia treatment planning

Let $u_j = a_j e^{i\phi_j}$ and $M_{jk}(x) = \langle E_j(x), E_k(x)\rangle$ where $E_j = \sin(jx_1x_2x_3\pi)$:

$$\begin{array}{l} \min \ \frac{1}{2}\int_\Omega (y(x) - y_t(x))^2 dx \\ \text{s.t.} \left\{ \begin{array}{rclll} -\Delta y(x) - 10(y(x) - 37) & = & u^* M(x) u & & \text{in } \Omega \\ 37.0 \leq y(x) & \leq & 37.5 & & \text{on } \partial\Omega \\ 42.0 \leq y(x) & \leq & 44.0 & & \text{in } \Omega_0 \end{array} \right. \end{array}$$

Original `IPOPT` with $N = 32$ requires 408 seconds per iteration.

N	n	p	q	# iter	CPU sec (per iter)
16	4116	2744	2994	68	22.893 (0.3367)
32	32788	27000	13034	51	3055.9 (59.920)

Server room cooling

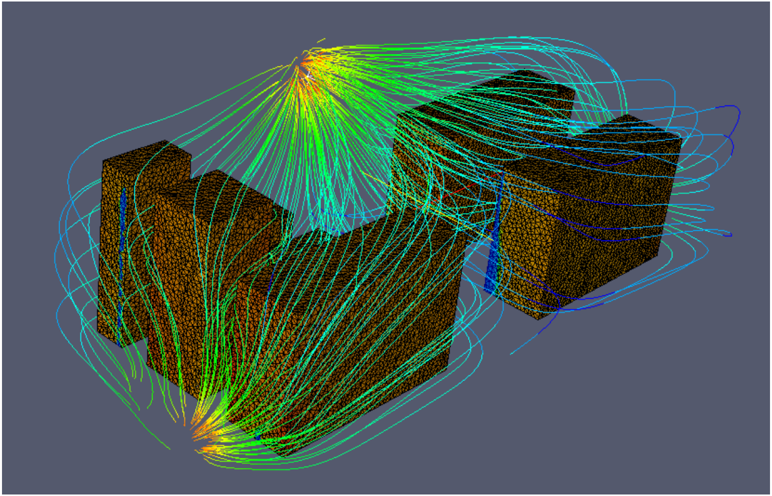
Let $\phi(x)$ be the air flow velocity potential:

$$
\begin{array}{l}
\min \sum c_i v_{AC_i} \\
\text{s.t.} \left\{ \begin{array}{rcll}
\nabla \phi(x) & = & 0 & \text{in } \Omega \\
\partial_n \phi(x) & = & 0 & \text{on } \partial\Omega_{wall} \\
\partial_n \phi(x) & = & -v_{AC_i} & \text{on } \partial\Omega_{AC_i} \\
\phi(x) & = & 0 & \text{in } \Omega_{Exh_j} \\
\|\nabla \phi(x)\|_2^2 & \geq & v_{min}^2 & \text{on } \partial\Omega_{hot} \\
v_{AC_i} & \geq & 0 &
\end{array} \right.
\end{array}
$$

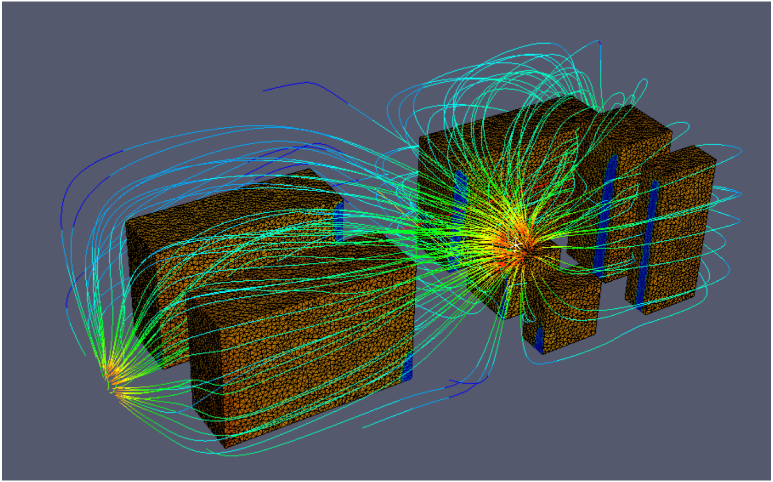
Original `IPOPT` with $h = 0.05$ requires 2390.09 seconds per iteration.

h	n	p	q	# iter	CPU sec (per iter)
0.10	43816	43759	4793	47	1697.47 (36.1164)
0.05	323191	323134	19128	54	28518.4 (528.119)

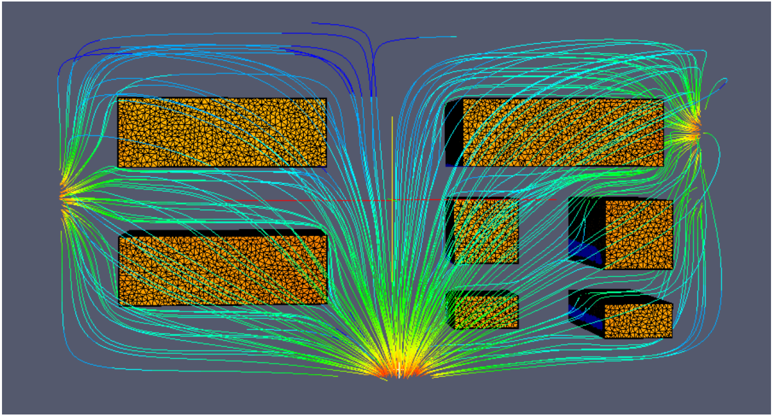
Server room cooling solution



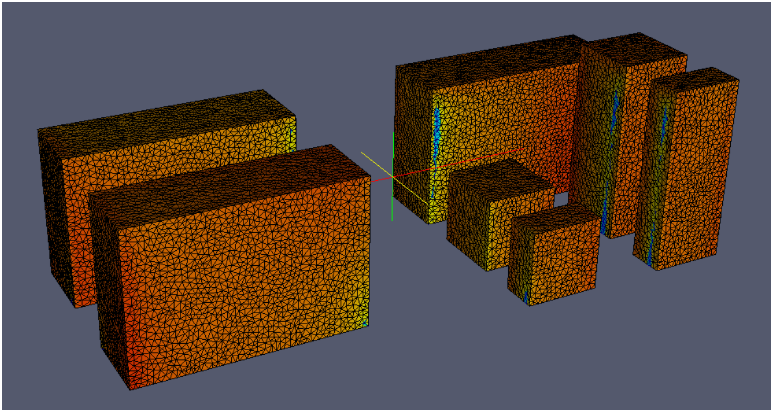
Server room cooling solution



Server room cooling solution



Server room cooling solution (active constraints)



Outline

Summary

We proposed an algorithm for large-scale nonlinear optimization:

- It can handle ill-conditioned/rank-deficient problems
- It can handle nonconvex problems
- Inexactness is allowed and controlled with loose conditions
- The conditions are implementable (in fact, implemented)
- The algorithm is globally convergent
- It can handle problems with control and state constraints
- Numerical results are encouraging so far

Future work and questions

What are we missing (to *really* solve PDE-constrained problems)?

- PDE-specific preconditioners
- Use of appropriate norms
- Mesh refinement, error estimators

What does it take to transform an algorithm for finite-dimensional optimization into one for solving infinite-dimensional problems?

- Can the finite-dimensional solver be a black-box?
- If not, to what extent do the outer and inner algorithms need to be coupled? (Do *all* components of the finite-dimensional solver need to be checked for their effect on the infinite-dimensional problem?)

What interesting problems may be solved with our approach?

References

- "An Inexact SQP Method for Equality Constrained Optimization," R. H. Byrd, F. E. Curtis, and J. Nocedal, SIAM Journal on Optimization, Volume 19, Issue 1, pg. 351–369, 2008.
- "An Inexact Newton Method for Nonconvex Equality Constrained Optimization," R. H. Byrd, F. E. Curtis, and J. Nocedal, Mathematical Programming, Volume 122, Issue 2, pg. 273–299, 2010.
- "A Matrix-free Algorithm for Equality Constrained Optimization Problems with Rank-Deficient Jacobians," F. E. Curtis, J. Nocedal, and A. Wächter, SIAM Journal on Optimization, Volume 20, Issue 3, pg. 1224–1249, 2009.
- "An Interior-Point Algorithm for Large-Scale Nonlinear Optimization with Inexact Step Computations," F. E. Curtis, O. Schenk, and A. Wächter, SIAM Journal on Scientific Computing, Volume 32, Issue 6, pg. 3447–3475, 2010.