Large-Scale Nonlinear Optimization with Inexact Step Computations

Andreas Wächter

IBM T.J. Watson Research Center
Yorktown Heights, New York
andreasw@us.ibm.com

IPAM Workshop on Numerical Methods for Continuous Optimization

October 14, 2010

(joint with Frank Curtis, Johannes Huber, and Olaf Schenk)

Nonlinear Optimization

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & c(x) = 0 \\ & d(x) \leq 0 \end{aligned}$$

x	Variables
$f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$	Objective function
$c(x) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$	Equality constraints
$d(x) : \mathbb{R}^n \longrightarrow \mathbb{R}^p$	Inequality constraints

- Functions $f(x)$, $c(x)$, $d(x)$ are smooth (C^2); sparse derivatives
- Want to find local solution

Nonlinear Optimization

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ s.t. \quad & c(x) = 0 \\ & d(x) \leq 0 \end{aligned}$$

x	Variables
$f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$	Objective function
$c(x) : \mathbb{R}^n \longrightarrow \mathbb{R}^m$	Equality constraints
$d(x) : \mathbb{R}^n \longrightarrow \mathbb{R}^p$	Inequality constraints

- Functions $f(x)$, $c(x)$, $d(x)$ are smooth (C^2); sparse derivatives
- Want to find local solution

Barrier method:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^p} \quad & f(x) - \mu \sum_{i=1}^{p} \ln(s^{(i)}) \\ s.t. \quad & c(x) = 0 \\ & d(x) + s = 0 \end{aligned}$$

- Solve sequence of barrier problems
- In following, concentrate on NLPs only with equality constraints (fixed μ)

Motivation

Example PDE constrained optimization problem:

- 3D inverse problem [Haber et al.]
- "All-at-once" approach
  - Finite difference discretization on regular grid
  - $N = 32$ in each dimension
- Size of NLP:
  - 229,376 variables
  - 196,608 equality and 65,536 inequality constraints
- Using IPOPT with Pardiso linear solver (direct factorization)

Motivation

Example PDE constrained optimization problem:

- 3D inverse problem [Haber et al.]
- "All-at-once" approach
  - Finite difference discretization on regular grid
  - $N = 32$ in each dimension
- Size of NLP:
  - 229,376 variables
  - 196,608 equality and 65,536 inequality constraints
- Using IPOPT with Pardiso linear solver (direct factorization)
- About 15 hours for first iteration!

Motivation

Example PDE constrained optimization problem:

- 3D inverse problem [Haber et al.]
- "All-at-once" approach
  - Finite difference discretization on regular grid
  - $N = 32$ in each dimension
- Size of NLP:
  - 229,376 variables
  - 196,608 equality and 65,536 inequality constraints
- Using IPOPT with Pardiso linear solver (direct factorization)
- About 15 hours for first iteration!
  - # nonzeros in sparse step computation matrix: 6,329,867
  - # nonzeros in factor: 2,503,219,440
  - Fill-in factor: 395 (!)
- Want to be able to use iterative linear solver

Motivation

Example PDE constrained optimization problem:

- 3D inverse problem [Haber et al.]
- "All-at-once" approach
  - Finite difference discretization on regular grid
  - $N = 32$ in each dimension
- Size of NLP:
  - 229,376 variables
  - 196,608 equality and 65,536 inequality constraints
- Using IPOPT with Pardiso linear solver (direct factorization)
- About 15 hours for first iteration!
  - # nonzeros in sparse step computation matrix: 6,329,867
  - # nonzeros in factor: 2,503,219,440
  - Fill-in factor: 395 (!)
- Want to be able to use iterative linear solver

Looking for other applications with large fill-in (ideas?)

Outline

- Newton's method for $F(x) = 0$ with inexact step computation
- NLP algorithm with inexact step computation
- Global convergence result
- Numerical experiments (3D PDE problems)

Newton's Method

NLP

$$\min_{x \in \mathbb{R}^n} \; f(x)$$
$$s.t. \; c(x) = 0$$

Optimality Conditions

$$\begin{aligned} \nabla f(x) + \nabla c(x) y &= 0 \\ c(x) &= 0 \end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix}$$

- $W_k = \nabla_{xx} \mathcal{L}(x_k, y_k)$ Hessian of Lagrangian (or approximation)
  - $\mathcal{L}(x, y) = f(x) + y^T c(x)$

Newton's Method

NLP

$$\min_{x \in \mathbb{R}^n} \; f(x)$$
$$s.t. \; c(x) = 0$$

Optimality Conditions

$$\begin{aligned} \nabla f(x) + \nabla c(x) y &= 0 \\ c(x) &= 0 \end{aligned}$$

Apply Newton's Method

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix}$$

- $W_k = \nabla_{xx} \mathcal{L}(x_k, y_k)$ Hessian of Lagrangian (or approximation)
  - $\mathcal{L}(x, y) = f(x) + y^T c(x)$
- Simple approach:
  - Consider as root finding problem $F(z) = 0$ (with $z = (x, y)$)

Basic Line Search Algorithm for $F(z) = 0$

Given: Starting point z_0; set $k \leftarrow 0$

1. Solve linear system to compute search direction

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

2. Perform backtracking line search to determine step size $\alpha_k \in (0, 1]$ satisfying Armijo condition ($\eta \in (0, 1)$):

$$\phi(z_k + \alpha_k \Delta z_k) \leq \phi(z_k) + \eta \alpha_k D\phi(z_k; \Delta z_k)$$

   Here: Merit function $\phi(z) = \|F(z)\|_2^2$

3. Update iterates $z_{k+1} = z_k + \alpha_k \Delta z_k$

4. Increase $k \leftarrow k + 1$ and go back to Step 1.

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$\phi(z_{k+1}) - \phi(z_0) \quad = \quad \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$\phi(z_{k+1}) - \phi(z_0) = \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

$$\leq \sum_{i=0}^{k} \eta \alpha_i D\phi(z_i; \Delta z_i)$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$\begin{aligned} \phi(z_{k+1}) - \phi(z_0) &= \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i)) \\ \leq \sum_{i=0}^{k} \eta\alpha_i D\phi(z_i; \Delta z_i) &\leq -2\eta\bar{\alpha} \sum_{i=0}^{k} \|F(z_i)\|_2^2 \end{aligned}$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$L \leq \phi(z_{k+1}) - \phi(z_0) = \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

$$\leq \sum_{i=0}^{k} \eta\alpha_i D\phi(z_i; \Delta z_i) \leq -2\eta\bar{\alpha} \sum_{i=0}^{k} \|F(z_i)\|_2^2$$

Simple Global Convergence Proof

- Step computation

$$\nabla F(z_k)^T \Delta z_k = -F(z_k)$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k = -2\|F(z_k)\|_2^2 < 0$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$L \leq \phi(z_{k+1}) - \phi(z_0) = \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

$$\leq \sum_{i=0}^{k} \eta\alpha_i D\phi(z_i; \Delta z_i) \leq -2\eta\bar{\alpha} \sum_{i=0}^{k} \|F(z_i)\|_2^2$$

  4. Therefore: $\lim_k F(z_k) = 0$

Simple Global Convergence Proof

- Inexact step computation [Dembo, Eisenstat, Steihaug (1982)]

$$\nabla F(z_k)^T \Delta z_k = -F(z_k) + r_k \qquad (\|r_k\| \le \kappa \|F(z_k)\|_2, \quad \kappa \in (0,1))$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k \le -2\|F(z_k)\|_2^2 + 2F(z_k)^T r_k$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \ge \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$L \le \phi(z_{k+1}) - \phi(z_0) = \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

$$\le \sum_{i=0}^{k} \eta \alpha_i D\phi(z_i; \Delta z_i) \le -2\eta\bar{\alpha} \sum_{i=0}^{k} \|F(z_i)\|_2^2$$

  4. Therefore: $\lim_k F(z_k) = 0$

Simple Global Convergence Proof

- Inexact step computation [Dembo, Eisenstat, Steihaug (1982)]

$$\nabla F(z_k)^T \Delta z_k = -F(z_k) + r_k \qquad (\|r_k\| \leq \kappa \|F(z_k)\|_2, \quad \kappa \in (0,1))$$

- Merit function $\phi(z) = \|F(z)\|_2^2$

$$D\phi(z_k; \Delta z_k) = 2F(z_k)^T \nabla F(z_k)^T \Delta z_k \leq -2(1-\kappa)\|F(z_k)\|_2^2$$

- Proof outline:
  1. Show Δz_k are bounded (e.g., if $F(z_k)$ and $\nabla F(z_k)^{-1}$ bounded)
  2. Therefore step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
  3. Then

$$L \leq \phi(z_{k+1}) - \phi(z_0) = \sum_{i=0}^{k} (\phi(z_{i+1}) - \phi(z_i))$$

$$\leq \sum_{i=0}^{k} \eta \alpha_i D\phi(z_i; \Delta z_i) \leq -2\eta\bar{\alpha} \sum_{i=0}^{k} (1-\kappa)\|F(z_i)\|_2^2$$

  4. Therefore: $\lim_k F(z_k) = 0$

Discussion

- Root finding algorithm $F(z) = 0$ can easily incorporate inexactness of linear solver
  - Natural consistency between step computation and merit function
  - Can be used for solving PDEs

Discussion

- Root finding algorithm $F(z) = 0$ can easily incorporate inexactness of linear solver
  - Natural consistency between step computation and merit function
  - Can be used for solving PDEs
- Requires $\nabla F(z_k)^{-1}$ is uniformly bounded

$$\nabla F(z_k) = \begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix}$$

Violated, e.g, if $\nabla c(x_k)$ loses rank

- In barrier approach, this also corresponds to violation of LICQ

Discussion

- Root finding algorithm $F(z) = 0$ can easily incorporate inexactness of linear solver
  - Natural consistency between step computation and merit function
  - Can be used for solving PDEs
- Requires $\nabla F(z_k)^{-1}$ is uniformly bounded

$$\nabla F(z_k) = \begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix}$$

  Violated, e.g, if $\nabla c(x_k)$ loses rank
  - In barrier approach, this also corresponds to violation of LICQ
- Maximizers and stationary points also satisfy $F(z) = 0$
  - Would like to encourage convergence to minimizers

Merit Function For Optimization

Classical penalty function: $\phi_\nu(x) = f(x) + \nu\|c(x)\|$

- For ν sufficiently large:

  x_* local solution of NLP $\iff$ x_* local minimizer of $\phi_\nu(x)$

Merit Function For Optimization

Classical penalty function: $\phi_\nu(x) = f(x) + \nu\|c(x)\|$

- For ν sufficiently large:

  x_* local solution of NLP $\iff$ x_* local minimizer of $\phi_\nu(x)$

Algorithm:

1. Solve linear system to compute search direction

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix}$$

2. Perform backtracking line search to determine step size α_k with

   $\phi(x_k + \alpha_k \Delta x_k) \leq \phi(x_k) + \eta \alpha_k D\phi(x_k; \Delta x_k)$ [Need $D\phi(x_k; \Delta x_k) < 0$!]

3. Update iterates $(x_{k+1}, y_{k+1}) = (x_k, y_k) + \alpha_k(\Delta x_k, \Delta y_k)$

4. Increase $k \leftarrow k+1$ and go back to Step 1.

Ensuring Descent

If $\nabla c(x_k)^T \Delta x_k + c(x_k) = 0$ then

$$D\phi_{\nu_k}(x_k; \Delta x_k) = \nabla f(x_k)^T \Delta x_k - \nu_k \| c(x_k) \|$$

Ensuring Descent

If $\nabla c(x_k)^T \Delta x_k + c(x_k) = 0$ then

$$D\phi_{\nu_k}(x_k; \Delta x_k) = \nabla f(x_k)^T \Delta x_k - \nu_k \| c(x_k) \|$$

To guarantee $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$:

- Before line search, possibly increase ν

  Careful: Need to avoid $\nu \to \infty$!

Ensuring Descent

If $\nabla c(x_k)^T \Delta x_k + c(x_k) = 0$ then

$$D\phi_{\nu_k}(x_k; \Delta x_k) = \nabla f(x_k)^T \Delta x_k - \nu_k \| c(x_k) \|$$

To guarantee $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$:

- Before line search, possibly increase ν
  Careful: Need to avoid $\nu \to \infty$!
- One option: "Convexify" local model

$$\begin{bmatrix} W_k + \delta I & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix}$$

  - Make sure that $v^T (W_k + \delta I) v > 0$ for all v with $\nabla c(x_k)^T v = 0$
    $(\delta \geq 0)$
  - Can be verified by looking at inertia (requires factorization)
  - E.g., in IPOPT : Choose $\delta \geq 0$ so that inertia is as desired

Simple Convergence Proof

Show first (based on "strongish" assumptions):

- Steps Δx_k are bounded
- Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
- $\nu_k = \bar{\nu}$ for large k
- $D\phi_{\bar{\nu}}(x_k; \Delta x_k) \leq -\theta \|\text{OptCond}_k\|$ $(\theta > 0)$

Simple Convergence Proof

Show first (based on "strongish" assumptions):

- Steps Δx_k are bounded
- Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0
- $\nu_k = \bar{\nu}$ for large k
- $D\phi_{\bar{\nu}}(x_k; \Delta x_k) \leq -\theta \|\text{OptCond}_k\|$ $\qquad (\theta > 0)$

Then, as before, using Armijo condition:

$$
\begin{aligned}
L \quad &\leq \quad \phi_{\bar{\nu}}(x_{k+1}) - \phi_{\bar{\nu}}(x_0) \quad = \quad \sum_{i=0}^{k} (\phi_{\bar{\nu}}(x_{i+1}) - \phi_{\bar{\nu}}(x_i)) \\
&\leq \quad \sum_{i=0}^{k} \eta \alpha_i D\phi_{\bar{\nu}}(x_i; \Delta x_i) \quad \leq \quad -\eta \bar{\alpha} \theta \sum_{i=0}^{k} \|\text{OptCont}_i\|
\end{aligned}
$$

Therefore: $\lim_k \|\text{OptCont}_k\| = 0$

Discussion of Convergence

- Required for this convergence proof:
  - $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$
  - $\nu_k = \bar{\nu}$ for large k
  - Steps Δx_k are bounded
  - Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

Discussion of Convergence

- Required for this convergence proof:
  - $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$
  - $\nu_k = \bar{\nu}$ for large k
  - Steps Δx_k are bounded
  - Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

- Standard algorithm/proof fails if
  - Inertia cannot be computed
    - ★ iterative linear solver

Discussion of Convergence

- Required for this convergence proof:
  - $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$
  - $\nu_k = \bar{\nu}$ for large k
  - Steps Δx_k are bounded
  - Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

- Standard algorithm/proof fails if
  - Inertia cannot be computed
    - ★ iterative linear solver
  - Step does not satisfy linear system exactly
    - ★ iterative linear solver

Discussion of Convergence

- Required for this convergence proof:
  - $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$
  - $\nu_k = \bar{\nu}$ for large k
  - Steps Δx_k are bounded
  - Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

- Standard algorithm/proof fails if
  - Inertia cannot be computed
    - ★ iterative linear solver
  - Step does not satisfy linear system exactly
    - ★ iterative linear solver
  - Constraint gradients become linearly dependent
    - ★ strong assumption

Discussion of Convergence

- Required for this convergence proof:
  - $D\phi_{\nu_k}(x_k; \Delta x_k) < 0$
  - $\nu_k = \bar{\nu}$ for large k
  - Steps Δx_k are bounded
  - Step size $\alpha_k \geq \bar{\alpha} > 0$ bounded away from 0

- Standard algorithm/proof fails if
  - Inertia cannot be computed
    - ★ iterative linear solver
  - Step does not satisfy linear system exactly
    - ★ iterative linear solver
  - Constraint gradients become linearly dependent
    - ★ strong assumption
  - For barrier methods:
    The fraction-to-the-boundary rule leads to $\alpha_k \to 0$
    - ★ fundamental problem for line-search barrier methods
      [W., Biegler, 2000]

Goal

Devise an algorithm that

- can work with an iterative linear solver
- can deal with rank-deficient constraint Jacobian
- overcomes convergence problems of line-search barrier methods

Goal

Devise an algorithm that

- can work with an iterative linear solver
- can deal with rank-deficient constraint Jacobian
- overcomes convergence problems of line-search barrier methods

Main ingredients:

- Step decomposition
  - helps to bound step (no need to assume full rank of $\nabla c(x_k)$)
  - handle nonconvexity

Goal

Devise an algorithm that

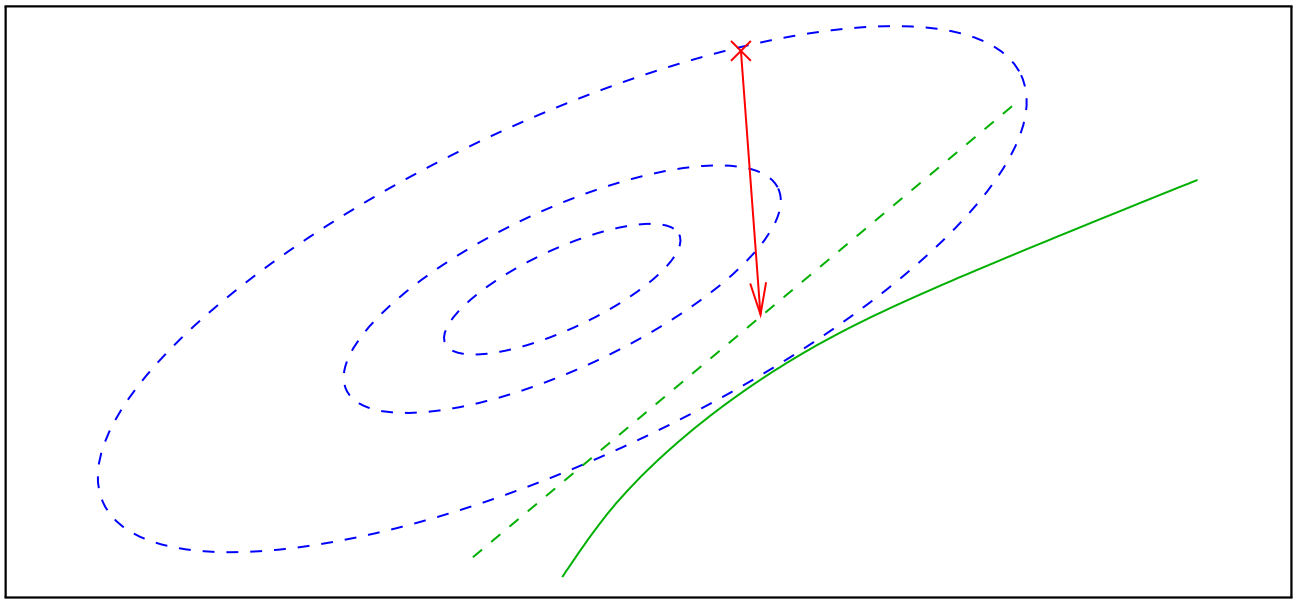
- can work with an iterative linear solver
- can deal with rank-deficient constraint Jacobian
- overcomes convergence problems of line-search barrier methods

Main ingredients:

- Step decomposition
  - helps to bound step (no need to assume full rank of $\nabla c(x_k)$)
  - handle nonconvexity
- Termination test of linear solver based on model of merit function
  - consistency between inexact solution and merit function
  - does not ignore optimization nature of problem

Useful Equivalence

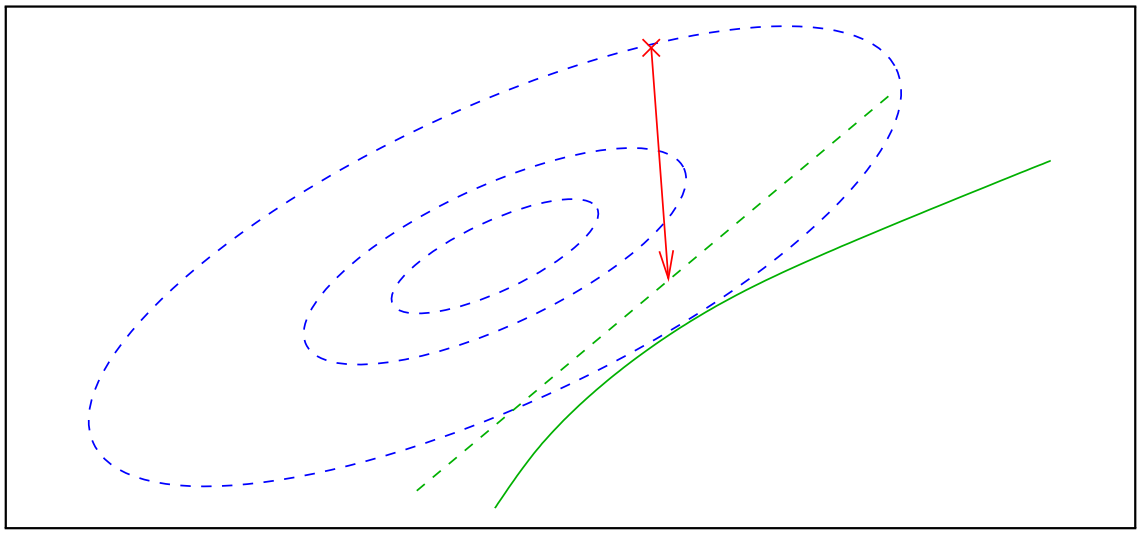
$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix}$$



$$\begin{aligned} \min \quad & \frac{1}{2} \Delta x_k^T W_k \Delta x_k + \nabla f(x_k)^T \Delta x_k \\ s.t. \quad & \nabla c(x_k)^T \Delta x_k + c(x_k) = 0 \end{aligned}$$

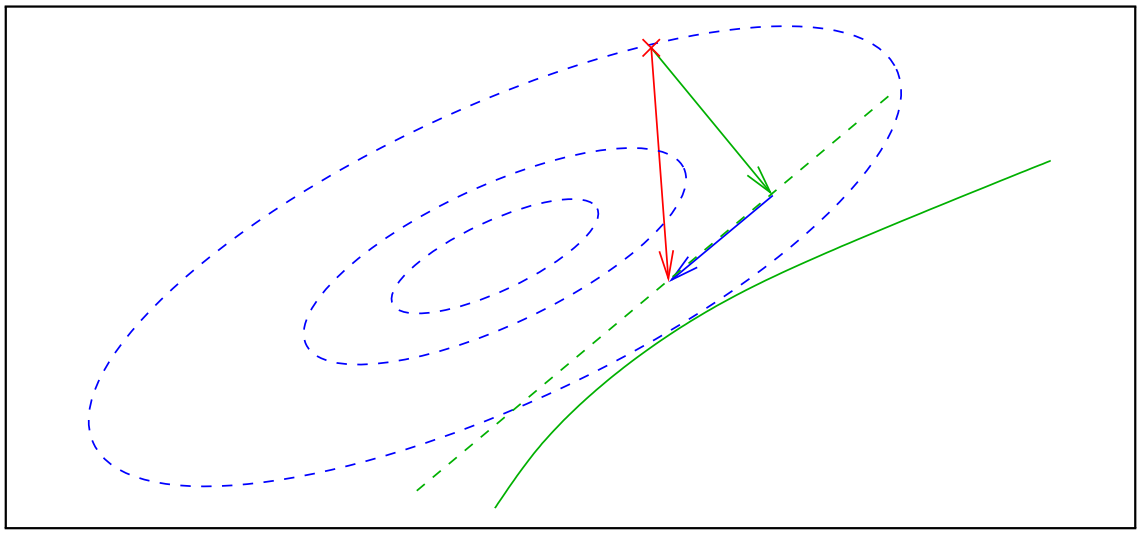
Step Decomposition

$$\begin{aligned} \min \quad & \frac{1}{2}\Delta x_k^T W_k \Delta x_k + \nabla f(x_k)^T \Delta x_k \\ s.t. \quad & \nabla c(x_k)^T \Delta x_k + c(x_k) = 0 \end{aligned}$$



Step Decomposition

$$\begin{aligned} \min \quad & \frac{1}{2}\Delta x_k^T W_k \Delta x_k + \nabla f(x_k)^T \Delta x_k \\ s.t. \quad & \nabla c(x_k)^T \Delta x_k + c(x_k) = 0 \end{aligned}$$

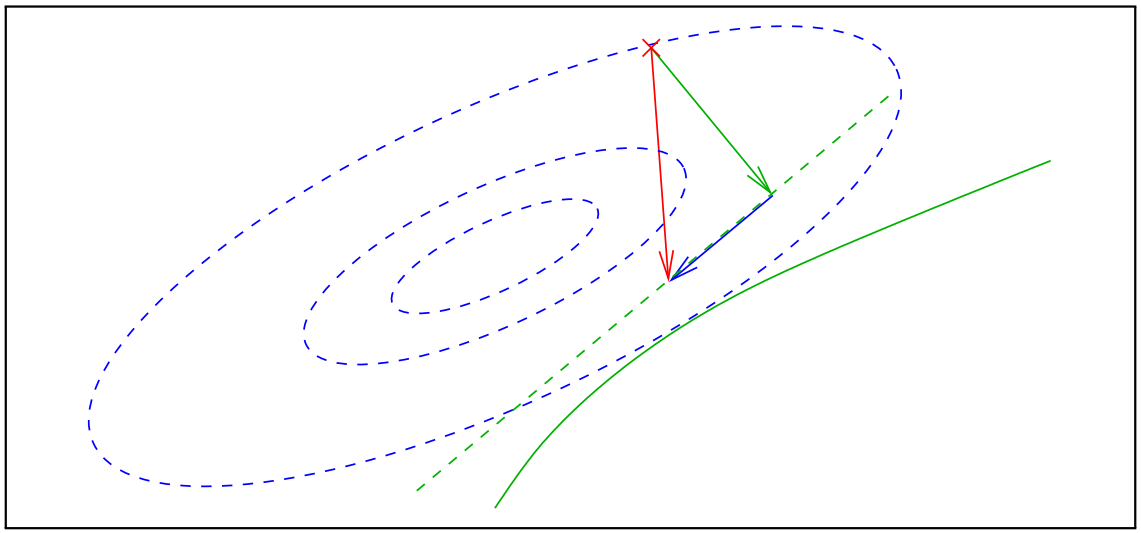


- $\Delta x_k = n_k + t_k$
  - Normal step n_k

$$\nabla c(x_k)^T n_k + c(x_k) = 0$$

Step Decomposition

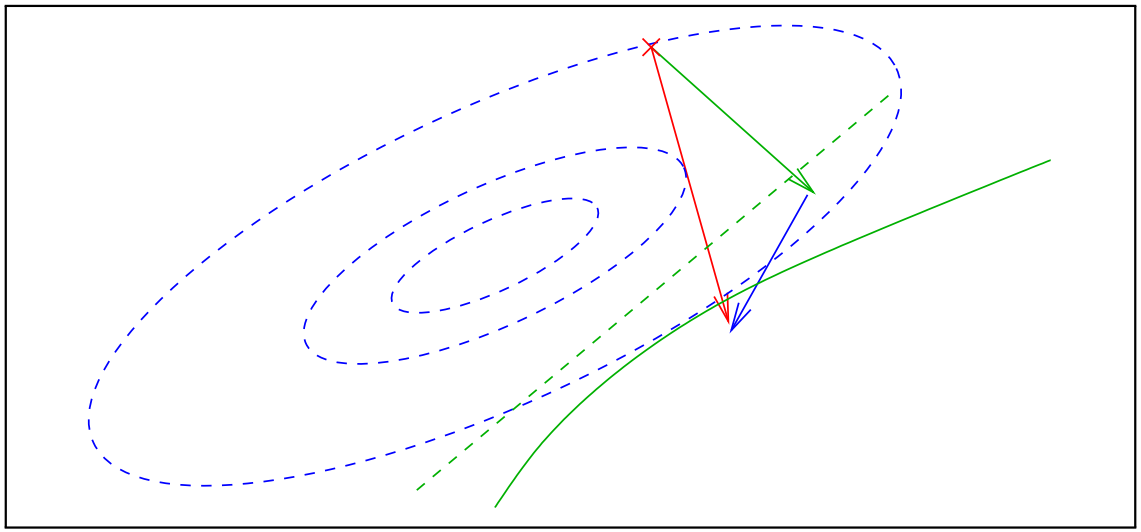
$$\begin{aligned} \min \quad & \frac{1}{2}\Delta x_k^T W_k \Delta x_k + \nabla f(x_k)^T \Delta x_k \\ s.t. \quad & \nabla c(x_k)^T \Delta x_k + c(x_k) = 0 \end{aligned}$$



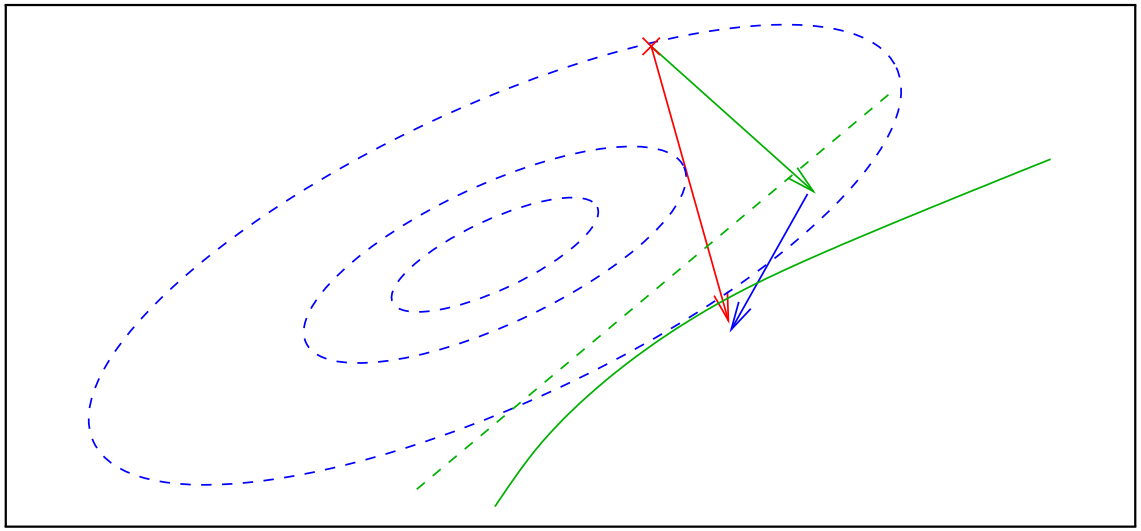
- $\Delta x_k = n_k + t_k$
  - Normal step n_k
  - Tangential step t_k

$$\begin{aligned} \min_t \quad & \frac{1}{2} t^T W_k t + \left(W_k^T n_k + \nabla f(x_k) \right)^T t \\ s.t. \quad & \nabla c(x_k)^T t = 0 \end{aligned}$$

Inexact Steps (Normal Component)



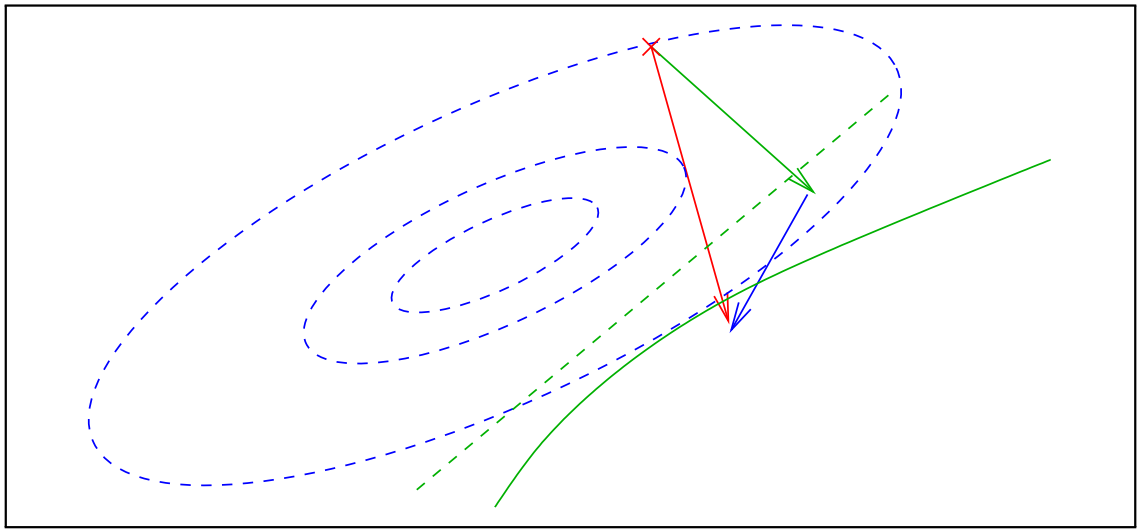
Inexact Steps (Normal Component)



- Normal component

$$\nabla c(x_k)^T n_k + c(x_k) = 0$$

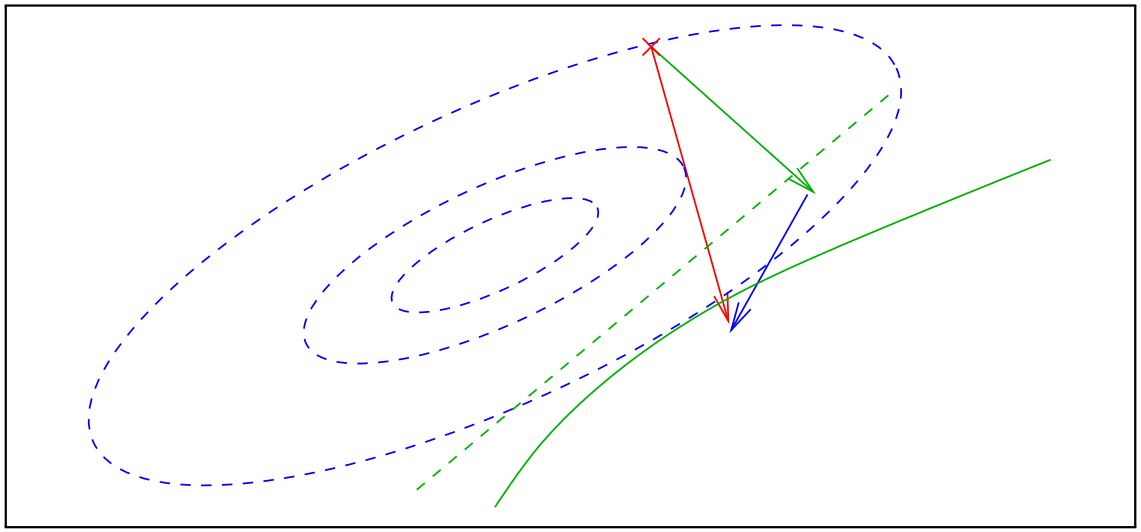
Inexact Steps (Normal Component)



- Normal component

$$\min_{n} \quad \|\nabla c(x_k)^T n + c(x_k)\|_2^2$$

Inexact Steps (Normal Component)



- Normal component

$$\begin{aligned} \min_{n} \quad & \|\nabla c(x_k)^T n + c(x_k)\|_2^2 \\ s.t. \quad & \|n\|_2 \leq \omega \|\nabla c(x_k) c(x_k)\|_2 \end{aligned}$$

($\omega > 0$ fixed, usually large)

Inexact Steps (Normal Component)

$$\begin{aligned} \min_{n} \quad & \|\nabla c(x_k)^T n + c(x_k)\|_2^2 \\ s.t. \quad & \|n\|_2 \leq \omega \|\nabla c(x_k) c(x_k)\|_2 \end{aligned}$$

- Trust-region subproblem
  - Controls size of normal component
  - Trust region inactive if $\omega \geq [\sigma_{\min}(\nabla c(x_k))]^{-1}$
  - $\nabla c(x_k) c(x_k)$ goes to zero if x_k converges to minimizer of $\|c(x)\|_2^2$

Inexact Steps (Normal Component)

$$\begin{aligned} \min_{n} \quad & \|\nabla c(x_k)^T n + c(x_k)\|_2^2 \\ s.t. \quad & \|n\|_2 \leq \omega \|\nabla c(x_k) c(x_k)\|_2 \end{aligned}$$

- Trust-region subproblem
  - Controls size of normal component
  - Trust region inactive if $\omega \geq [\sigma_{\min}(\nabla c(x_k))]^{-1}$
  - $\nabla c(x_k) c(x_k)$ goes to zero if x_k converges to minimizer of $\|c(x)\|_2^2$
- Inexact: Make at least as much progress as steepest descent step
  - "Cauchy step" n^C

Inexact Steps (Normal Component)

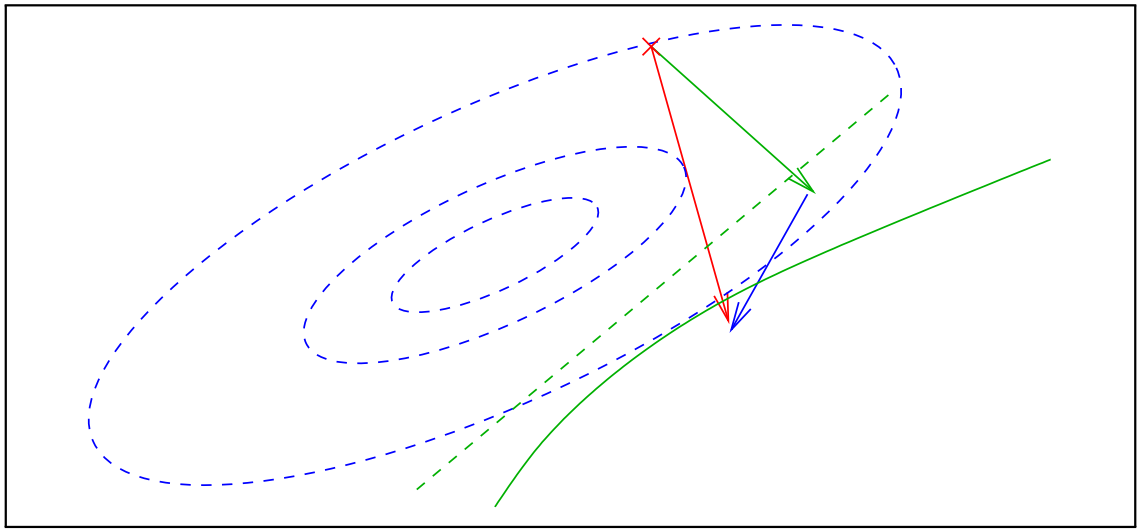
$$\begin{aligned} \min_{n} \quad & \|\nabla c(x_k)^T n + c(x_k)\|_2^2 \\ s.t. \quad & \|n\|_2 \leq \omega \|\nabla c(x_k) c(x_k)\|_2 \end{aligned}$$

- Trust-region subproblem
  - Controls size of normal component
  - Trust region inactive if $\omega \geq [\sigma_{\min}(\nabla c(x_k))]^{-1}$
  - $\nabla c(x_k) c(x_k)$ goes to zero if x_k converges to minimizer of $\|c(x)\|_2^2$
- Inexact: Make at least as much progress as steepest descent step
  - "Cauchy step" n^C
- In practice:
  - Shortest norm solution:

$$\begin{bmatrix} I & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \delta \\ n^N \end{pmatrix} = - \begin{pmatrix} 0 \\ c(x_k) \end{pmatrix}$$

    (solve only approximately)
  - Choose combination of n^C and n^N satisfying trust region giving best objective

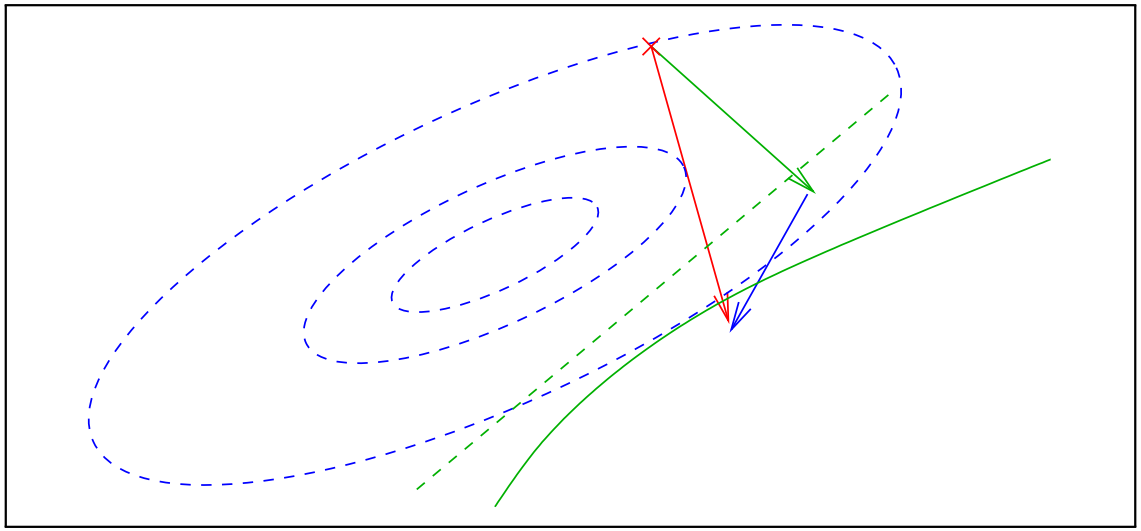
Inexact Steps (Tangential Component)



- Tangential component

$$\begin{aligned} \min_{t} \quad & \frac{1}{2} t^T W_k t + \left(W_k^T n_k + \nabla f(x_k) \right)^T t \\ s.t. \quad & \nabla c(x_k)^T t = 0 \end{aligned}$$

Inexact Steps (Tangential Component)



- Tangential component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} t_k \\ \Delta y_k \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k + W_k n_k \\ 0 \end{pmatrix}$$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} t_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k + W_k n_k \\ 0 \end{pmatrix}$$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} t_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k + W_k n_k \\ 0 \end{pmatrix}$$

- Rewrite using $\Delta x_k = n_k + t_k$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix}$$

- Rewrite using $\Delta x_k = n_k + t_k$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \widetilde{\Delta x_k} \\ \widetilde{\Delta y_k} \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} + \begin{pmatrix} \widetilde{\rho_k} \\ \widetilde{r_k} \end{pmatrix}$$

- Rewrite using $\widetilde{\Delta x_k} = n_k + \widetilde{t_k}$ ($\widetilde{\Delta x_k}$ current inexact solution)

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \widetilde{\Delta x_k} \\ \widetilde{\Delta y_k} \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} + \begin{pmatrix} \widetilde{\rho_k} \\ \widetilde{r_k} \end{pmatrix}$$

- Rewrite using $\widetilde{\Delta x_k} = n_k + \widetilde{t_k}$ ($\widetilde{\Delta x_k}$ current inexact solution)

Require: (simplified)

1. Residual

$$\|\widetilde{\rho_k}\| \le \kappa \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} \right\| \qquad \kappa \in (0,1)$$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \widetilde{\Delta x_k} \\ \widetilde{\Delta y_k} \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} + \begin{pmatrix} \widetilde{\rho_k} \\ \widetilde{r_k} \end{pmatrix}$$

- Rewrite using $\widetilde{\Delta x_k} = n_k + \widetilde{t_k}$ ($\widetilde{\Delta x_k}$ current inexact solution)

Require: (simplified)

1. Residual

$$\|\widetilde{\rho_k}\| \le \kappa \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} \right\| \qquad \kappa \in (0,1)$$

2. Tangential component condition: Satisfy at least one of

$$\begin{aligned} \widetilde{t_k}^T W_k \widetilde{t_k} &\ge \theta \|\widetilde{t_k}\|^2 & \theta > 0 \\ \|\widetilde{t_k}\| &\le \psi \|n_k\| & \psi > 0 \end{aligned}$$

Inexact Tangential Component

$$\begin{bmatrix} W_k & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{pmatrix} \widetilde{\Delta x_k} \\ \widetilde{\Delta y_k} \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} + \begin{pmatrix} \widetilde{\rho_k} \\ \widetilde{r_k} \end{pmatrix}$$

- Rewrite using $\widetilde{\Delta x_k} = n_k + \widetilde{t_k}$ ($\widetilde{\Delta x_k}$ current inexact solution)

Require: (simplified)

1. Residual

$$\|\widetilde{\rho_k}\| \le \kappa \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ -\nabla c(x_k)^T n_k \end{pmatrix} \right\| \qquad \kappa \in (0,1)$$

2. Tangential component condition: Satisfy at least one of

$$\begin{aligned} \widetilde{t_k}^T W_k \widetilde{t_k} &\ge \theta \|\widetilde{t_k}\|^2 & \theta > 0 \\ \|\widetilde{t_k}\| &\le \psi \|n_k\| & \psi > 0 \end{aligned}$$

Update $W_k \leftarrow W_k + \delta I$ if both violated

Model Reduction Condition

- Merit function: $\phi_{\nu_k}(x) = f(x) + \nu_k \|c(x)\|$
- Linear model:

$$m_{\nu_k}(x_k, \Delta x) = f(x_k) + \nabla f(x_k)^T \Delta x + \nu_k \| c(x_k) + \nabla c(x_k)^T \Delta x \|$$

Model Reduction Condition

- Merit function: $\phi_{\nu_k}(x) = f(x) + \nu_k \|c(x)\|$
- Linear model:

$$m_{\nu_k}(x_k, \Delta x) = f(x_k) + \nabla f(x_k)^T \Delta x + \nu_k \|c(x_k) + \nabla c(x_k)^T \Delta x\|$$

- Predicted reduction:

$$\Delta m_{\nu_k}(x_k; \Delta x) = m_{\nu_k}(x_k, 0) - m_{\nu_k}(x_k, \Delta x)$$

- Can show

$$D\phi_{\nu_k}(x_k; \Delta x) \leq -\Delta m_{\nu_k}(x_k; \Delta x)$$

Model Reduction Condition

- Merit function: $\phi_{\nu_k}(x) = f(x) + \nu_k \|c(x)\|$
- Linear model:
$$m_{\nu_k}(x_k, \Delta x) = f(x_k) + \nabla f(x_k)^T \Delta x + \nu_k \|c(x_k) + \nabla c(x_k)^T \Delta x\|$$
- Predicted reduction:
$$\Delta m_{\nu_k}(x_k; \Delta x) = m_{\nu_k}(x_k, 0) - m_{\nu_k}(x_k, \Delta x)$$
- Can show
$$D\phi_{\nu_k}(x_k; \Delta x) \leq -\Delta m_{\nu_k}(x_k; \Delta x)$$

Require:

- $\Delta m_{\nu_k}(x_k; \widetilde{\Delta x_k})$ is sufficiently positive,
($\Longrightarrow$ guarantee direction of descent for current ν_k) **or**

Model Reduction Condition

- Merit function: $\phi_{\nu_k}(x) = f(x) + \nu_k \|c(x)\|$
- Linear model:
$$m_{\nu_k}(x_k, \Delta x) = f(x_k) + \nabla f(x_k)^T \Delta x + \nu_k \|c(x_k) + \nabla c(x_k)^T \Delta x\|$$
- Predicted reduction:
$$\Delta m_{\nu_k}(x_k; \Delta x) = m_{\nu_k}(x_k, 0) - m_{\nu_k}(x_k, \Delta x)$$
- Can show
$$D\phi_{\nu_k}(x_k; \Delta x) \leq -\Delta m_{\nu_k}(x_k; \Delta x)$$

Require:

- $\Delta m_{\nu_k}(x_k; \widetilde{\Delta x_k})$ is sufficiently positive,
($\Longrightarrow$ guarantee direction of descent for current ν_k) **or**
- $\|c(x_k)\| - \|c(x_k) - \nabla c(x_k)^T \widetilde{\Delta x_k}\|$ is sufficiently positive
($\Longrightarrow$ "controlled" update of ν_k to get $D\phi_{\nu_k}(x_k; \widetilde{\Delta x_k}) < 0$)

Theoretical Convergence Result

Assumption

$f(x)$, $c(x)$, $\nabla f(x)$, $\nabla c(x)$ are bounded and Lipschitz continuous, (unmodified) W_k are bounded.

Theoretical Convergence Result

Assumption

$f(x)$, $c(x)$, $\nabla f(x)$, $\nabla c(x)$ are bounded and Lipschitz continuous, (unmodified) W_k are bounded.

Theorem (Curtis, Nocedal, W, 2009)

- *If all limit points of $\nabla c(x_k)$ have full rank then ν_k is bounded and*

$$\lim_k \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix} \right\| = 0.$$

Theoretical Convergence Result

Assumption

$f(x)$, $c(x)$, $\nabla f(x)$, $\nabla c(x)$ are bounded and Lipschitz continuous, (unmodified) W_k are bounded.

Theorem (Curtis, Nocedal, W, 2009)

- *If all limit points of $\nabla c(x_k)$ have full rank then ν_k is bounded and*

$$\lim_k \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix} \right\| = 0.$$

- *Otherwise*

$$\lim_k \|\nabla c(x_k) c(x_k)\| = 0 \qquad (\textit{stationary for:} \quad \min \frac{1}{2}\|c(x)\|_2^2)$$

Theoretical Convergence Result

Assumption

$f(x)$, $c(x)$, $\nabla f(x)$, $\nabla c(x)$ are bounded and Lipschitz continuous, (unmodified) W_k are bounded.

Theorem (Curtis, Nocedal, W, 2009)

- *If all limit points of $\nabla c(x_k)$ have full rank then ν_k is bounded and*

$$\lim_k \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix} \right\| = 0.$$

- *Otherwise*

$$\lim_k \|\nabla c(x_k) c(x_k)\| = 0 \qquad (\textit{stationary for:} \quad \min \frac{1}{2}\|c(x)\|_2^2)$$

and if ν_k is bounded

$$\lim_k \|\nabla f(x_k) + \nabla c(x_k) y_k\| = 0.$$

Theoretical Convergence Result

Assumption

$f(x)$, $c(x)$, $\nabla f(x)$, $\nabla c(x)$ are bounded and Lipschitz continuous, (unmodified) W_k are bounded.

Theorem (Curtis, Nocedal, W, 2009)

- *If all limit points of $\nabla c(x_k)$ have full rank then ν_k is bounded and*

$$\lim_k \left\| \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k \\ c(x_k) \end{pmatrix} \right\| = 0.$$

- *Otherwise*

$$\lim_k \|\nabla c(x_k) c(x_k)\| = 0 \qquad (\textit{stationary for:} \quad \min \frac{1}{2}\|c(x)\|_2^2)$$

and if ν_k is bounded

$$\lim_k \|\nabla f(x_k) + \nabla c(x_k) y_k\| = 0.$$

Similar result for interior point method with inequalities (Curtis, Schenk, W, 2010)

Inequality Constraints / Barrier Problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^p} \quad & f(x) - \mu \sum_{i=1}^{p} \ln(s^{(i)}) \\ s.t. \quad & c(x) = 0 \\ & d(x) + s = 0 \qquad (s > 0) \end{aligned}$$

$$\begin{bmatrix} W_k & 0 & \nabla c(x_k) & \nabla d(x_k) \\ 0 & \mu S_k^{-2} & 0 & I \\ \nabla c(x_k)^T & 0 & 0 & 0 \\ \nabla d(x_k)^T & I & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ \Delta s_k \\ \Delta y_k^c \\ \Delta y_k^d \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k^c + \nabla d(x_k) y_k^d \\ -\mu S_k^{-1} e \\ c(x_k) \\ d(x_k) + s_k \end{pmatrix}$$

Inequality Constraints / Barrier Problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^p} \quad & f(x) - \mu \sum_{i=1}^{p} \ln(s^{(i)}) \\ s.t. \quad & c(x) = 0 \\ & d(x) + s = 0 \qquad (s > 0) \end{aligned}$$

$$\begin{bmatrix} W_k & 0 & \nabla c(x_k) & \nabla d(x_k) \\ 0 & \mu I & 0 & S_k \\ \nabla c(x_k)^T & 0 & 0 & 0 \\ \nabla d(x_k)^T & S_k & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ S_k^{-1}\Delta s_k \\ \Delta y_k^c \\ \Delta y_k^d \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k^c + \nabla d(x_k) y_k^d \\ -\mu e \\ c(x_k) \\ d(x_k) + s_k \end{pmatrix}$$

- Trick: Scale slack variables: $\Delta\tilde{s} = S_k^{-1}\Delta s_k$

Inequality Constraints / Barrier Problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^p} \quad & f(x) - \mu \sum_{i=1}^{p} \ln(s^{(i)}) \\ s.t. \quad & c(x) = 0 \\ & d(x) + s = 0 \qquad (s > 0) \end{aligned}$$

$$\begin{bmatrix} W_k & 0 & \nabla c(x_k) & \nabla d(x_k) \\ 0 & \mu I & 0 & S_k \\ \nabla c(x_k)^T & 0 & 0 & 0 \\ \nabla d(x_k)^T & S_k & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ S_k^{-1}\Delta s_k \\ \Delta y_k^c \\ \Delta y_k^d \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k^c + \nabla d(x_k) y_k^d \\ -\mu e \\ c(x_k) \\ d(x_k) + s_k \end{pmatrix}$$

- Trick: Scale slack variables: $\Delta\tilde{s} = S_k^{-1}\Delta s_k$
- Now all quantities in linear system are uniformly bounded
  - Much of equality-only analysis applies, use same termination tests

Inequality Constraints / Barrier Problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n, s\in\mathbb{R}^p} \quad & f(x) - \mu \sum_{i=1}^{p} \ln(s^{(i)}) \\ s.t. \quad & c(x) = 0 \\ & d(x) + s = 0 \qquad (s > 0) \end{aligned}$$

$$\begin{bmatrix} W_k & 0 & \nabla c(x_k) & \nabla d(x_k) \\ 0 & \mu I & 0 & S_k \\ \nabla c(x_k)^T & 0 & 0 & 0 \\ \nabla d(x_k)^T & S_k & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x_k \\ S_k^{-1}\Delta s_k \\ \Delta y_k^c \\ \Delta y_k^d \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla c(x_k) y_k^c + \nabla d(x_k) y_k^d \\ -\mu e \\ c(x_k) \\ d(x_k) + s_k \end{pmatrix}$$

- Trick: Scale slack variables: $\Delta\tilde{s} = S_k^{-1}\Delta s_k$
- Now all quantities in linear system are uniformly bounded
  - Much of equality-only analysis applies, use same termination tests
- Special consideration
  - Fraction to boundary: $s_k + \alpha_k \Delta s_k \geq (1-\tau) s_k > 0$
  - Slack reset: $s_{k+1} \leftarrow \max\{s_{k+1}, -d(x_{k+1})\}$

Some Computational Results

- IPOPT [W, Laird, Biegler]
  - primal-dual interior point method
  - originally: filter line-search method
  - added inexact algorithm with penalty function
- PARDISO [Schenk, Gärtner]
  - Parallel Direct Linear Solver
  - includes iterative linear solvers (e.g., SQMR)
  - preconditioner: multi-level incomplete factorization with weighted graph matchings and inverse-based pivoting

Some Computational Results

- IPOPT [W, Laird, Biegler]
  - primal-dual interior point method
  - originally: filter line-search method
  - added inexact algorithm with penalty function
- PARDISO [Schenk, Gärtner]
  - Parallel Direct Linear Solver
  - includes iterative linear solvers (e.g., SQMR)
  - preconditioner: multi-level incomplete factorization with weighted graph matchings and inverse-based pivoting

Experiments:

- Robustness test with CUTE/COPS collection
  - 684 AMPL models
  - Inexact method solved 85% (original IPOPT 94%)

Some Computational Results

- IPOPT [W, Laird, Biegler]
  - primal-dual interior point method
  - originally: filter line-search method
  - added inexact algorithm with penalty function
- PARDISO [Schenk, Gärtner]
  - Parallel Direct Linear Solver
  - includes iterative linear solvers (e.g., SQMR)
  - preconditioner: multi-level incomplete factorization with weighted graph matchings and inverse-based pivoting

Experiments:

- Robustness test with CUTE/COPS collection
  - 684 AMPL models
  - Inexact method solved 85% (original IPOPT 94%)
- PDE-constrained optimization problems
  - Implemented in Matlab; finite differences; uniform grids

Boundary Control

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\int_{\Omega} (y(x) - y_t(x))^2 \, dx \\
s.t. \quad & -\nabla(e^{y(x)} \cdot \nabla y(x)) = 20 && \text{in } \Omega = [0,1]^3 \\
& y(x) = u(x) && \text{on } \partial\Omega \\
& 2.5 \le u(x) \le 3.5 && \text{on } \partial\Omega
\end{aligned}
$$

Target profile: $y_t(x) = 3 + 10x^{(1)}(x^{(1)} - 1)x^{(2)}(x^{(2)} - 1)\sin(2\pi x^{(3)})$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	4096	2744	2704	13	2.8144	(0.2165)
32	32768	27000	11536	13	103.65	(7.9731)
64	262144	238328	47632	14	5332.3	(380.88)

Boundary Control

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\int_{\Omega} (y(x) - y_t(x))^2 \, dx \\
s.t. \quad & -\nabla(e^{y(x)} \cdot \nabla y(x)) = 20 && \text{in } \Omega = [0,1]^3 \\
& y(x) = u(x) && \text{on } \partial\Omega \\
& 2.5 \le u(x) \le 3.5 && \text{on } \partial\Omega
\end{aligned}
$$

Target profile: $y_t(x) = 3 + 10x^{(1)}(x^{(1)} - 1)x^{(2)}(x^{(2)} - 1)\sin(2\pi x^{(3)})$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	4096	2744	2704	13	2.8144	(0.2165)
32	32768	27000	11536	13	103.65	(7.9731)
64	262144	238328	47632	14	5332.3	(380.88)

Original IPOPT with $N = 32$ requires 238 seconds per iteration

Hyperthermia Treatment Planning

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\int_{\Omega} (y(x) - y_t(x))^2 \, dx \\
s.t. \quad & -\Delta y(x) - 10(y(x) - 37) = u^* M(x) u \qquad \text{in } \Omega = [0,1]^3 \\
& 37.0 \le y(x) \le 37.5 \qquad \text{on } \partial\Omega \\
& 42.0 \le y(x) \le 44.0 \qquad \text{in } \Omega_{\text{Tumor}} \subset \Omega
\end{aligned}
$$

Controls: $u_j = a_j e^{i\phi_j} \quad j = 1, \ldots, 10$

Data: $M_{jk}(x) =< E_j(x), E_k(x) >, \quad E_j = \sin(j\pi x_1 x_2 x_3)$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	4116	2744	2994	68	22.893	(0.3367)
32	32788	27000	13034	51	3055.9	(59.920)

Hyperthermia Treatment Planning

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\int_{\Omega} (y(x) - y_t(x))^2 \, dx \\
s.t. \quad & -\Delta y(x) - 10(y(x) - 37) = u^* M(x) u \qquad \text{in } \Omega = [0,1]^3 \\
& 37.0 \le y(x) \le 37.5 \qquad \text{on } \partial\Omega \\
& 42.0 \le y(x) \le 44.0 \qquad \text{in } \Omega_{\text{Tumor}} \subset \Omega
\end{aligned}
$$

Controls: $u_j = a_j e^{i\phi_j} \quad j = 1, \ldots, 10$

Data: $M_{jk}(x) =< E_j(x), E_k(x) >, \quad E_j = \sin(j\pi x_1 x_2 x_3)$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	4116	2744	2994	68	22.893	(0.3367)
32	32788	27000	13034	51	3055.9	(59.920)

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

Inverse Problem

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\sum_{i=1}^{6}\int_{\Omega}(y_i(x)-y_{i,t}(x))^2\,dx+\frac{1}{2}\int_{\Omega}\left(\beta(u(x)-u_t(x))^2+|\nabla(u(x)-u_t(x))|^2\right) \\
s.t. \quad & -\nabla\cdot(e^{u(x)}\cdot\nabla y_i(x))=q_i(x) && \text{in } \Omega=[0,1]^3, && i=1,\ldots,6 \\
& \nabla y_i(x)\cdot n=0 && \text{on } \partial\Omega, && i=1,\ldots,6 \\
& \int_{\Omega} y_i(x)\,dx=0 && && i=1,\ldots,6 \\
& -1\le u(x)\le 1 && \text{in } \Omega &&
\end{aligned}
$$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	28672	24576	8192	18	206.416	(11.4676)
32	229376	196608	65536	20	1963.64	(98.1820)
64	1835008	1572864	524288	21	134418.	(6400.85)

Inverse Problem

$$
\begin{aligned}
\min_{y,u} \quad & \frac{1}{2}\sum_{i=1}^{6}\int_\Omega (y_i(x)-y_{i,t}(x))^2\,dx + \frac{1}{2}\int_\Omega \left(\beta(u(x)-u_t(x))^2 + |\nabla(u(x)-u_t(x))|^2\right) \\
s.t. \quad & -\nabla\cdot(e^{u(x)}\cdot\nabla y_i(x)) = q_i(x) && \text{in } \Omega=[0,1]^3, && i=1,\dots,6 \\
& \nabla y_i(x)\cdot n = 0 && \text{on } \partial\Omega, && i=1,\dots,6 \\
& \int_\Omega y_i(x)\,dx = 0 && && i=1,\dots,6 \\
& -1 \le u(x) \le 1 && \text{in } \Omega &&
\end{aligned}
$$

N	# var	# eq	# ineq	# iter	CPU sec	(per iter)
16	28672	24576	8192	18	206.416	(11.4676)
32	229376	196608	65536	20	1963.64	(98.1820)
64	1835008	1572864	524288	21	134418.	(6400.85)

Original IPOPT with $N = 32$ requires 15 hours for the first iteration (reduction by factor 550)

PDE-Constrained Optimization

- Very active research area
- Often try to use of preconditioners for original PDE
  - Extremely efficient in solving linear systems for state variables
  - Optimization methods often based on step decomposition
    - Projections into the null space of constraint Jacobian
    - Work with reduced Hessian
  - Those often lead to triple iterative loops
  - Can handle constraints on control variables

PDE-Constrained Optimization

- Very active research area
- Often try to use of preconditioners for original PDE
  - Extremely efficient in solving linear systems for state variables
  - Optimization methods often based on step decomposition
    - ★ Projections into the null space of constraint Jacobian
    - ★ Work with reduced Hessian
  - Those often lead to triple iterative loops
  - Can handle constraints on control variables
- Potential difficulty: Inequalities involving state variables
  - If interior-point method is used:
    - ★ Reduced Hessian becomes very ill-conditioned
    - ★ Not clear how to find efficient preconditioner

Our Ongoing Work

- We hope that our approach can make a difference
  - for problems with many state constraints
  - when no efficient PDE preconditioner is known
    (e.g., some version of Helmholtz equation)
- Find an interesting problem with many state constraints
- Utilize standard PDE techniques
  - Discretization using meshes (e.g., finite elements)
  - Adaptive mesh refinement (e.g., [Ziems, Ulbrich])
- Scalable distributed-memory parallel implementation
- If successful, distribute code

Example: Server Room Cooling

Problem:

- Heat generating equipment in room must be cooled
- Want to place/control air conditioners to minimize cooling costs
- Suggested by Henderson, Lopez, Mello (IBM Research)

Formulation Of Airflow Problem

- We assume
  - Air flow has no internal friction (irrotational)
  - Air speed is far below speed of sound (incompressible)

- Then air flow can be described by velocity potential Φ
  - Air velocity: $v(x) = \nabla\Phi(x)$
  - Potential satisfies Laplace's equation

$$\Delta\Phi(x) = \sum_{i=1}^{n} \frac{\partial^2\Phi}{\partial x_i^2}(x) = 0 \quad \text{for} \quad x \in \Omega$$

  - $\Omega \subseteq \mathbb{R}^n$: interior of server room without equipment ($n = 2, 3$)

Boundary conditions

$$\Delta\Phi(x) = 0 \quad \text{for} \quad x \in \Omega$$

- Walls and equipment: No airflow through the surface

$$\frac{\partial\Phi(x)}{\partial n} = 0 \quad \text{for} \quad x \in \Gamma_{\text{wall}}$$

($\frac{\partial}{\partial n}$: outward normal derivative)

- At AC locations in walls: Controllable fan speed is incoming flow

$$\frac{\partial\Phi(x)}{\partial n} = -v_{\text{AC}_i} \quad \text{for} \quad x \in \Gamma_{\text{AC}_i}$$

- At exhausts: Assume "hole in the wall" ("grounded potential")

$$\Phi(x) = 0 \quad \text{for} \quad x \in \Gamma_{\text{Exh}_j}$$

- Unique solution $\Phi \in H^1(\Omega)$, continuously dependent on v_{AC_i}

Optimal Control Problem

$$
\begin{aligned}
\min \quad & \sum c_i v_{\mathrm{AC}_i} & \\
s.t. \quad & \Delta\Phi(x) = 0 & \text{for } x \in \Omega \\
& \frac{\partial\Phi(x)}{\partial n} = 0 & \text{for } x \in \Gamma_{\mathrm{wall}} \\
& \frac{\partial\Phi(x)}{\partial n} = -v_{\mathrm{AC}_i} & \text{for } x \in \Gamma_{\mathrm{AC}_i} \\
& \Phi(x) = 0 & \text{for } x \in \Gamma_{\mathrm{Exh}_j} \\
& \|\nabla\Phi(x)\|_2^2 \geq v_{\mathrm{min}}^2 & \text{for } x \in \Gamma_{\mathrm{hot}} \\
& v_{\mathrm{AC}_i} \geq 0 &
\end{aligned}
$$

- Objective: Minimize costs of running ACs
- Constraints: Require minimum air speed at heat sources of equipment (nonconvex)

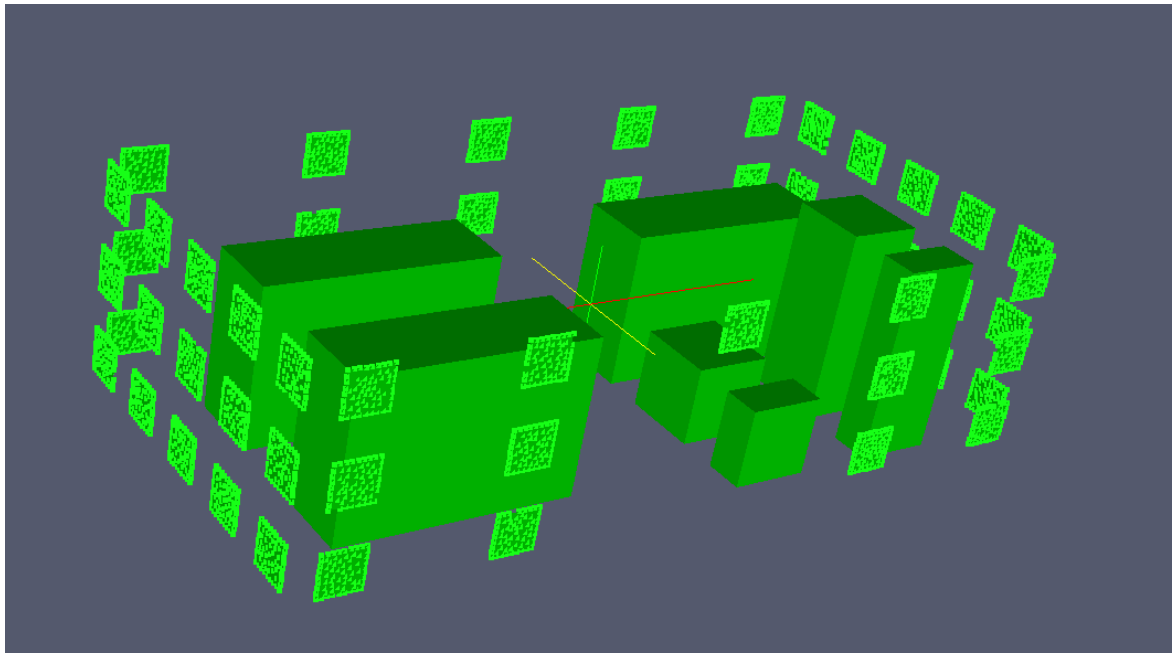
Software

- Finite Element Library: `libmesh` [Kirk, Peterson, Stogner, Carey]
  - Manages the finite-element mesh
  - Interfaces to mesh generators (we use `tetgen` [Si, Gärtner] for 3D)
  - Handles refinement of mesh (not yet used)
  - Distributed memory implementation (uses `PETSc`)

- Optimization code: IPOPT (inexact algorithm)
  - Variation: Step decomposition only when necessary
  - Using the "flexible penalty function" [Curtis, Nocedal, 2008]
  - Using distributed memory version of IPOPT [Dash, W]
    - ★ handles linear algebra objects in parallel
    - ★ supports parallel function evaluations
    - ★ not (yet...) officially released

- Linear Solver: PARDISO (iterative linear solver)
  - No distributed memory implementation
  - In the future: use PSPIKE [Samel, Sathe, Schenk]

3D Server Room Cooling Example



- 7 servers of different sizes
- 57 ACs located all around the walls
- One exhaust in top front
- Constraint: Minimum air speed at front and back sides of servers

Computational Results

Problem size ($\Omega = [0, 10] \times [0, 5] \times [0, 3]$)

h	#var	#eq	#ineq	#nnz KKT
0.1	43,816	43,759	4,793	777,139
0.05	323,191	323,134	19,128	5,759,809

Exact algorithm

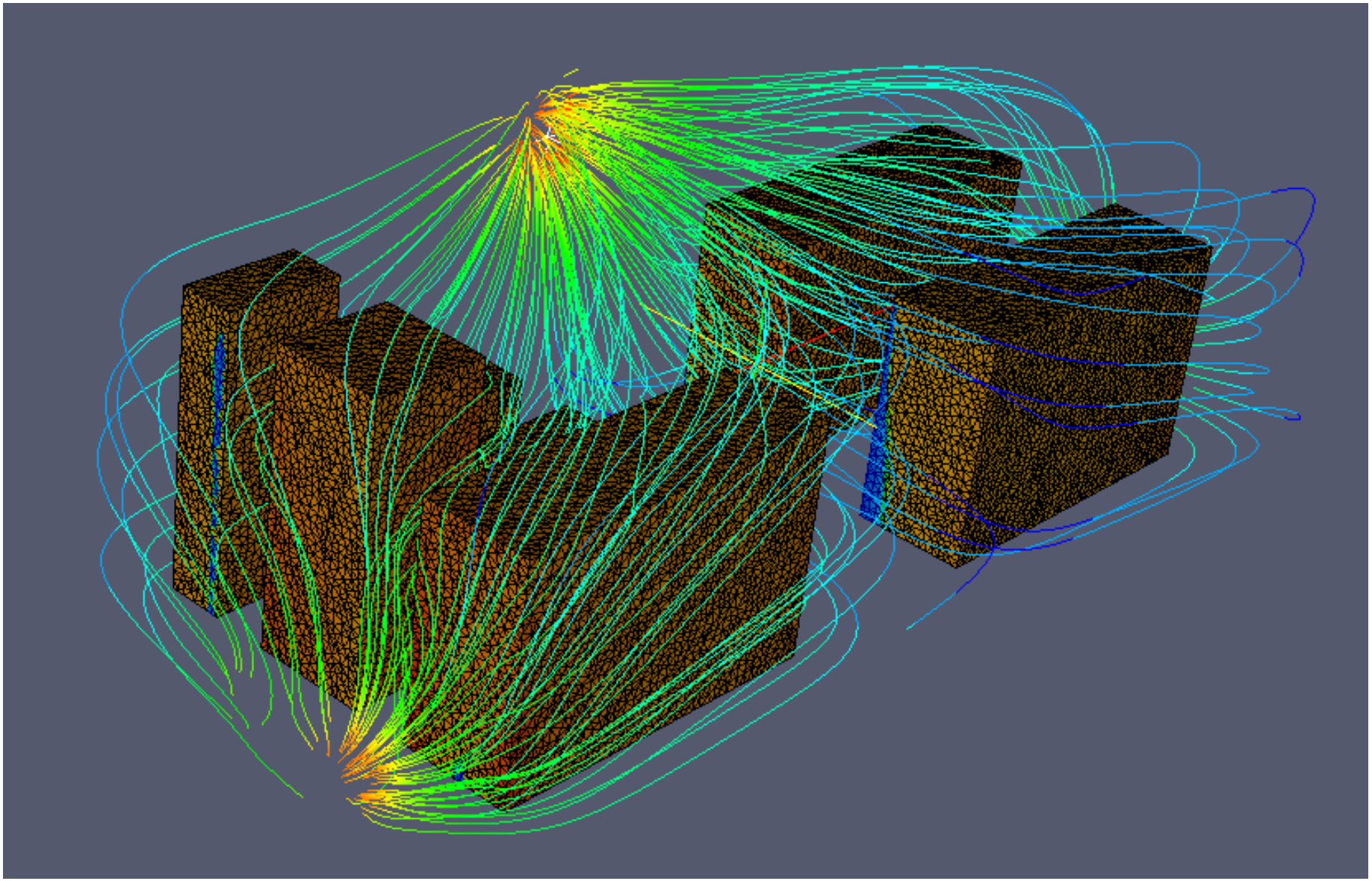
h	#iter	$f(x^*)$	CPU(algo)	CPU(eval)	fill-in
0.1	54	278.10	1,766.02	341.12	54.1
0.05	75	586.36	179,256.62	3,752.15	117.3

Inexact algorithm

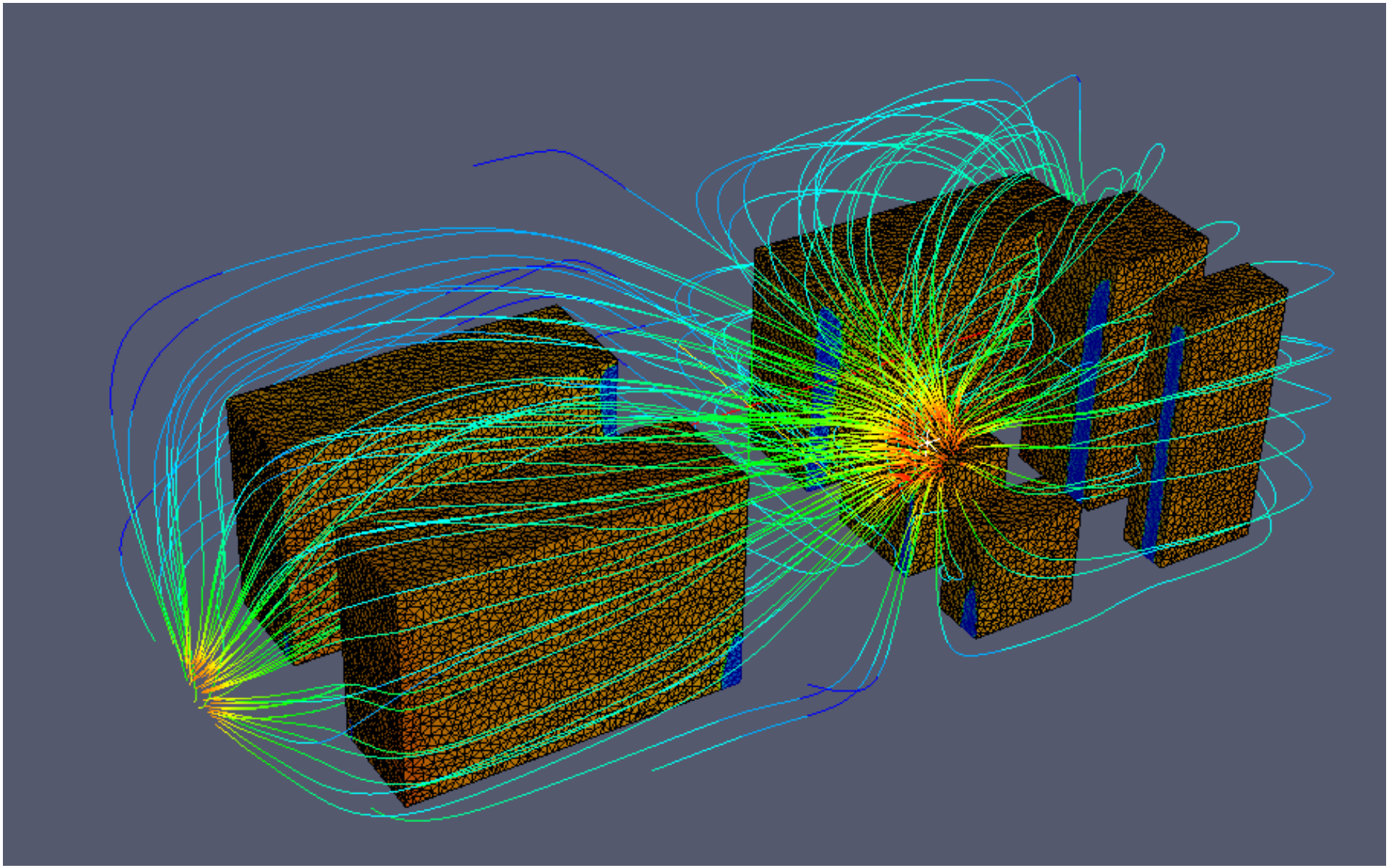
h	#iter	$f(x^*)$	CPU(algo)	CPU(eval)
0.1	47	278.10	1,697.47	307.47
0.05	54	586.36	28,518.41	2,658.18

(different ACs active for $h = 0.1$ and $h = 0.05$)

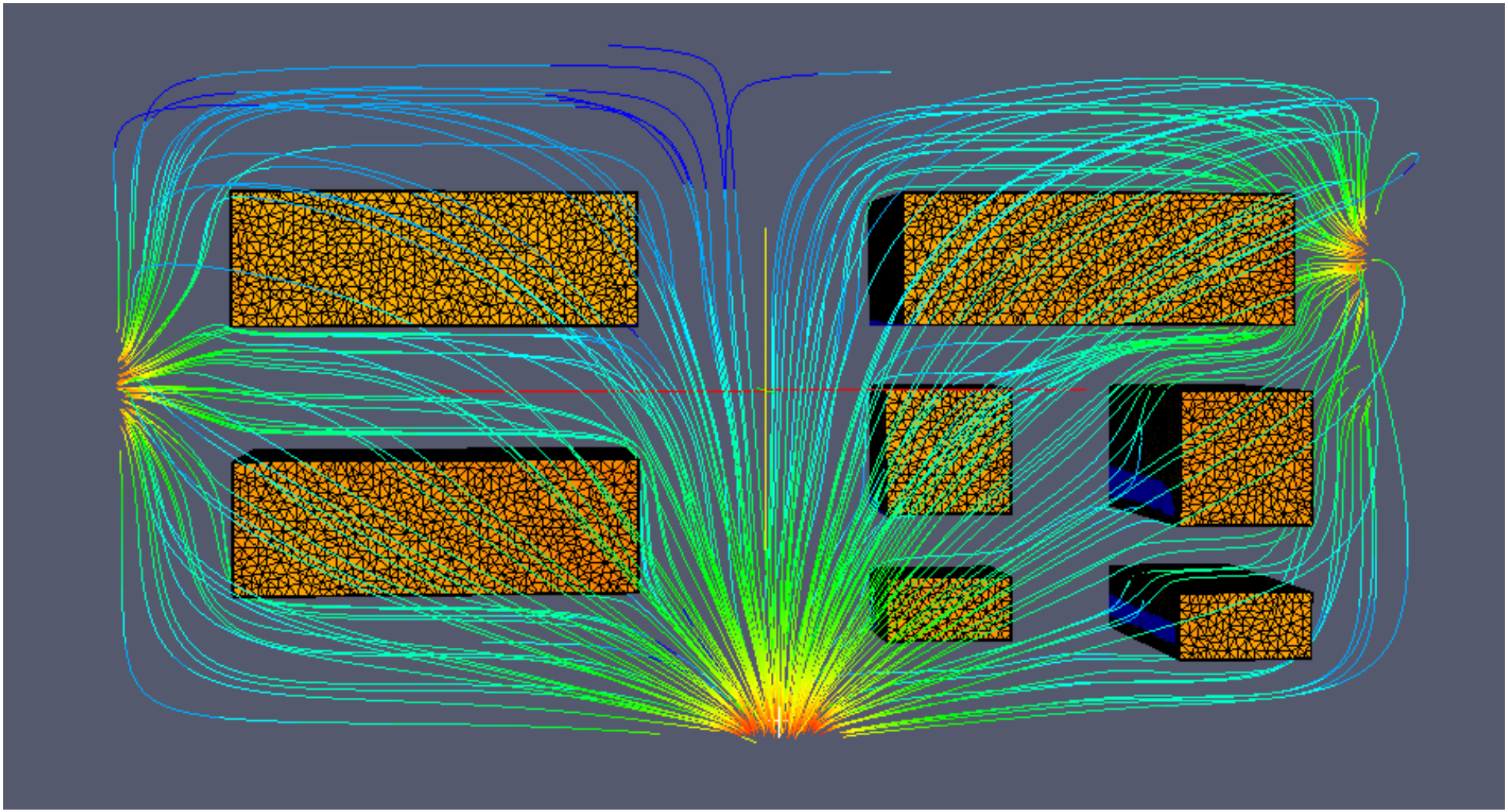
Optimal Solution ($h = 0.05$)



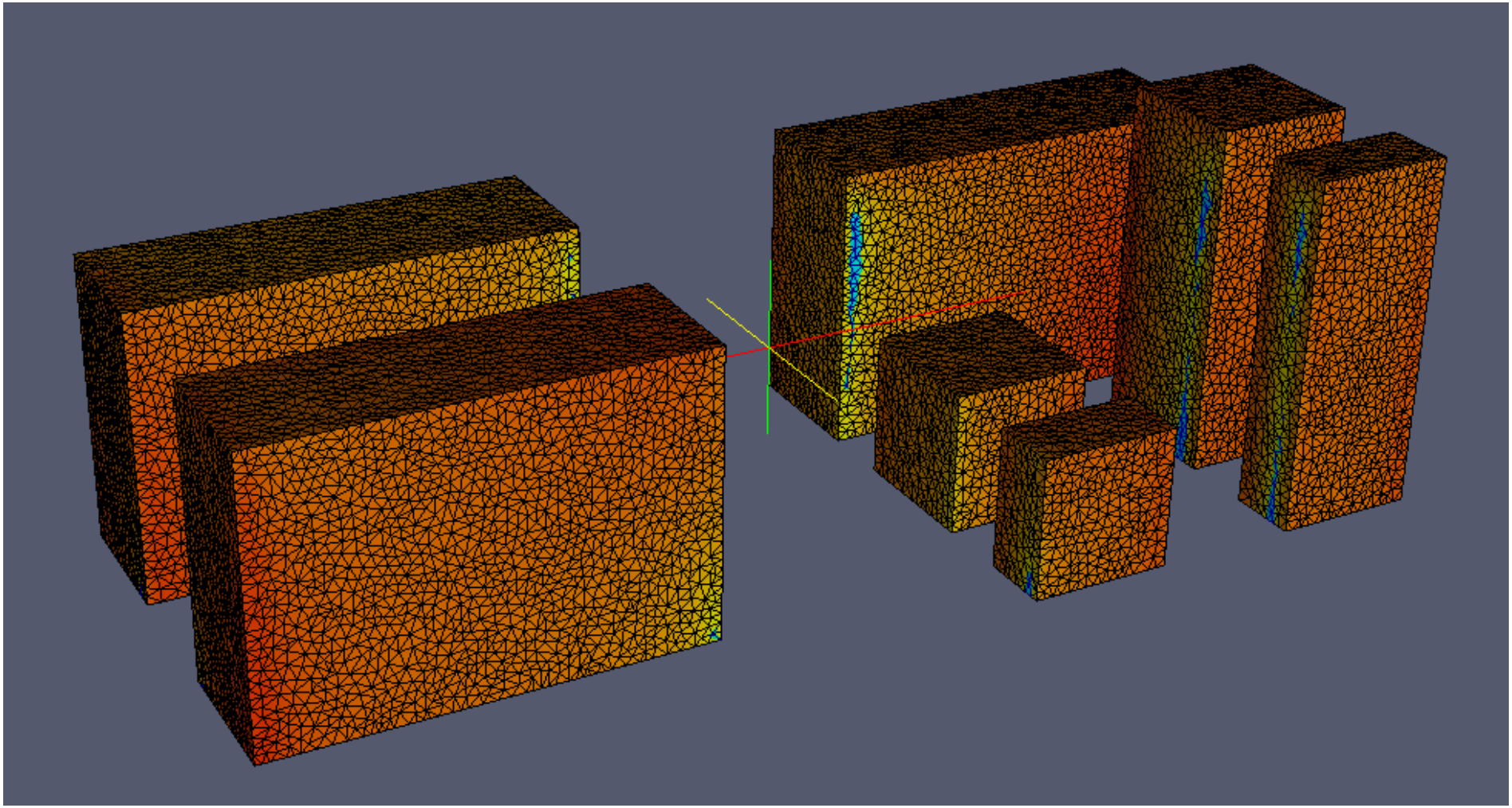
Optimal Solution ($h = 0.05$)



Optimal Solution ($h = 0.05$)



Optimal Solution (Active Constraints)



Future Plans

- These computations are still preliminary
- Further improvements:
  - Include temperature into the model
    - ★ Nonlinear PDEs
  - Include adaptive mesh refinement within the optimization
  - Improve preconditioner
  - Perform computations in parallel
    - ★ Function evaluations (libmesh) and Ipopt are already parallel
    - ★ Explore distributed memory iterative linear solver PSPIKE
  - Potentially release code as part of Ipopt distribution

Conclusions

- Nonlinear optimization with inexact step computations
  - Termination tests for iterative linear solver
  - Consider optimization nature
  - Based on model of merit function
  - Theoretical convergence properties under mild assumptions
- Encouraging results for PDE constrained problems
  - Very good speed up on some examples
  - Promising preliminary results for problem with state constraints
  - A lot left to do
- (Looking for other NLP applications with large fill-in — ideas?)