An Interior-Point Algorithm with Inexact Step Computations for Large-scale Optimization

Frank E. Curtis, Lehigh University
Olaf Schenk, University of Basel
Andreas Wächter, IBM T. J. Watson Research

11th Copper Mountain Conference on Iterative Methods

April 5, 2010

Outline

Outline

Introduction

Interior-Point with Inexact Steps

Numerical Results

Summary and Future Work

Large-scale constrained optimization

Consider large-scale problems of the form

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

Large-scale constrained optimization

Consider large-scale problems of the form

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

- 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

Strengths

We propose 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 very encouraging so far

Weaknesses

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

Weaknesses

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

I'll close the talk with questions; you might have the answers!

Interior-point methods

- Add slacks to form the logarithmic-barrier subproblem

$$\begin{aligned} \min \ & f(x) - \mu \sum_{i \in \mathcal{I}} \ln s^i \\ \text{s.t. } & c^{\mathcal{E}}(x) = 0 \\ & c^{\mathcal{I}}(x) = s \end{aligned}$$

- The first-order optimality conditions are

$$\begin{aligned} \nabla f(x) + \nabla c^{\mathcal{E}}(x)\lambda^{\mathcal{E}} + \nabla c^{\mathcal{I}}(x)\lambda^{\mathcal{I}} &= 0 \\ -\mu S^{-1} e - \lambda^{\mathcal{I}} &= 0 \\ c^{\mathcal{E}}(x) &= 0 \\ c^{\mathcal{I}}(x) - s &= 0 \end{aligned}$$

along with $s > 0$

Newton's method

- Newton iteration involves the linear system

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

- Search direction computation followed by a line search

Usual questions

- How do we ensure global convergence?
- How do we solve ill-conditioned problems?
- How do we handle nonconvexity?

Usual answers

- 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

More questions

For large-scale problems:

- What if the derivative matrices cannot be stored?
- What if the derivative matrices cannot be factored?

We can use iterative in place of direct methods:

- Can we allow inexactness?
- How do we ensure global convergence, handle ill-conditioning, and handle nonconvexity if solutions are inexact?

Outline

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

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

If the constraint Jacobian is singular or ill-conditioned

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

Matrix modifications

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

Matrix modifications

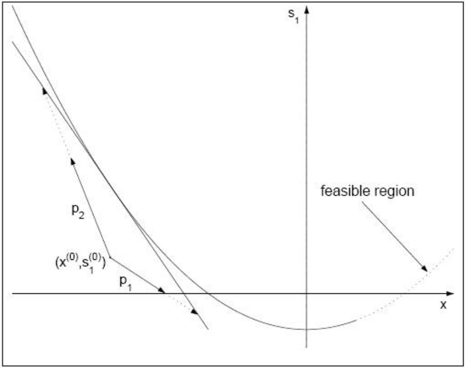
$$\begin{bmatrix} H_k & 0 & \nabla c_k^{\mathcal{E}} & \nabla c_k^{\mathcal{I}} \\ 0 & \Omega_k & 0 & -S_k \\ {\nabla c_k^{\mathcal{E}}}^T & 0 & -\xi I & 0 \\ {\nabla c_k^{\mathcal{I}}}^T & -S_k & 0 & -\xi I \end{bmatrix} \begin{bmatrix} d_k^x \\ \tilde{d}_k^s \\ \delta_k^{\mathcal{E}} \\ \delta_k^{\mathcal{I}}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

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

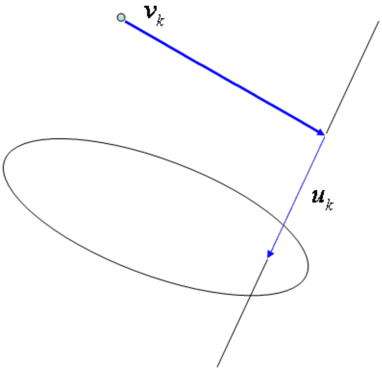
- When should this modification be performed?
- What value should ξ take? How large?
- How do we ensure that in the end we solve the right problem?

Failure of line search methods

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



Step decomposition

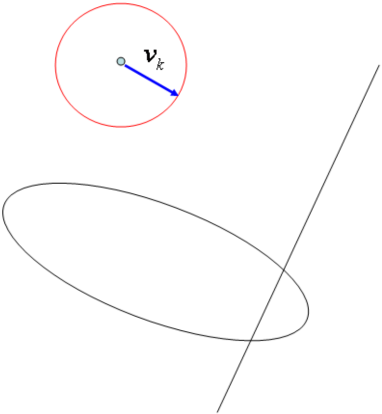


Normal step

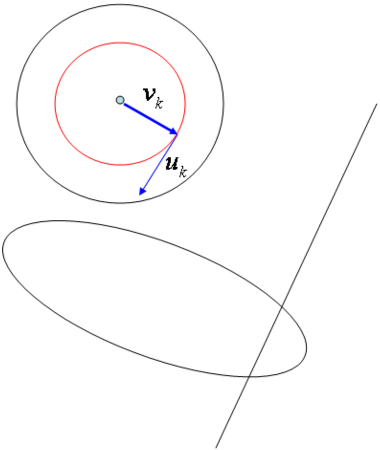
$$\min \frac{1}{2} \left\| \begin{bmatrix} c_k^{\mathcal{E}} \\ c_k^{\mathcal{I}} - s_k \end{bmatrix} + \begin{bmatrix} \nabla c_k^{\mathcal{E}\,T} & 0 \\ \nabla c_k^{\mathcal{I}\,T} & -S_k \end{bmatrix} \begin{bmatrix} v_k^x \\ \tilde{v}_k^s \end{bmatrix} \right\|^2$$

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

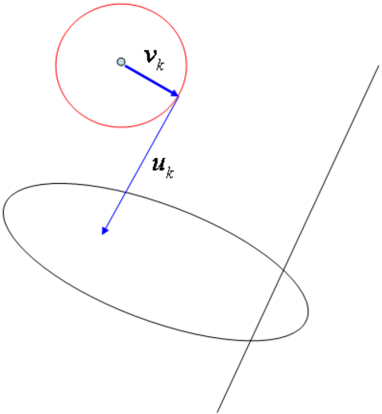
- QP w/ trust region constraint
- Trust region radius is zero at first-order minimizers of infeasibility
- Radius updates automatically
- Solve w/ CG or inexact dogleg



Tangential step



Tangential step



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

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

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

for some $\psi, \theta > 0$

- In our tests, modifications 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

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

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

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

How do we allow inexact solutions?

Consistency between the direction and the merit function

- In unconstrained optimization and nonlinear equations, there is always consistency (even w/ inexact steps) between the step computation and the function that measures progress
- In constrained optimization, however, our search direction is based on optimality conditions

$$\begin{bmatrix} \nabla f(x) + \nabla c^{\mathcal{E}}(x)\lambda^{\mathcal{E}} + \nabla c^{\mathcal{I}}(x)\lambda^{\mathcal{I}} \\ -\mu S^{-1}e - \lambda^{\mathcal{I}} \\ c^{\mathcal{E}}(x) \\ c^{\mathcal{I}}(x) - s \end{bmatrix} = 0$$

  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! A direction that may reduce the KKT error may not be a direction of descent for the merit function

Model reductions

- We ensure consistency by requiring model reductions
- 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 (2009))
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

Implementation details

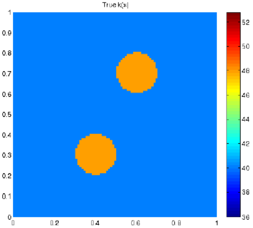
- Incorporated in `IPOPT` software package (Wächter)
  - `inexact_algorithm yes`
- Linear systems solved with `PARDISO` (Schenk)
  - SQMR (Freund (1994))
- Preconditioning in `PARDISO`
  - incomplete multilevel factorization with inverse-based pivoting
  - stabilized by symmetric-weighted matchings
- Optimality tolerance: 1e-8

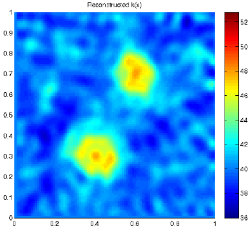
CUTEr and COPS collections

- 684 problems written in AMPL
- 580 solved successfully
- Robustness: $\sim$85%
- Original `IPOPT`: $\sim$94%

Parameter estimation for Helmholtz equation

Recover parameter k based on data collected from propagating waves





N	n	p	q	# iter	CPU sec (per iter)
32	14724	13824	1800	37	807.823 (21.833)
64	56860	53016	7688	25	3741.42 (149.66)
128	227940	212064	31752	20	54581.8 (2729.1)

Boundary control

$$\min \frac{1}{2}\int_\Omega (y(x) - y_t(x))^2 dx$$

$$\text{s.t.} \begin{cases} -\nabla \cdot (e^{y(x)} \cdot \nabla y(x)) & = & 20 & \text{in } \Omega \\ y(x) & = & u(x) & \text{on } \partial\Omega \\ 2.5 \leq u(x) & \leq & 3.5 & \text{on } \partial\Omega \end{cases}$$

where

$$y_t(x) = 3 + 10x_1(x_1 - 1)x_2(x_2 - 1)\sin(2\pi x_3)$$

N	n	p	q	# 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

$$\min \frac{1}{2}\int_{\Omega}(y(x)-y_t(x))^2 dx$$

$$\text{s.t.} \left\{ \begin{array}{rcl} -\Delta y(x) - 10(y(x)-37) & = & u^* M(x) u \quad \text{in } \Omega \\ 37.0 \leq y(x) & \leq & 37.5 \quad \text{on } \partial\Omega \\ 42.0 \leq y(x) & \leq & 44.0 \quad \text{in } \Omega_0 \end{array} \right.$$

where

$$u_j = a_j e^{i\phi_j}, \quad M_{jk}(x) = < E_j(x), E_k(x) >, \quad E_j = \sin(j x_1 x_2 x_3 \pi)$$

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)

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

Groundwater modeling

$$\min \frac{1}{2}\int_\Omega (y(x)-y_t(x))^2 dx + \frac{1}{2}\alpha\int_\Omega [\beta(u(x)-u_t(x))^2 + |\nabla(u(x)-u_t(x))|^2] dx$$

$$\text{s.t.} \left\{ \begin{array}{rcll} -\nabla\cdot(e^{u(x)}\cdot\nabla y_i(x)) & = & q_i(x) & \text{in } \Omega, \quad i=1,\ldots,6 \\ \nabla y_i(x)\cdot n & = & 0 & \text{on } \partial\Omega \\ \int_\Omega y_i(x)dx & = & 0, & i=1,\ldots,6 \\ -1 \leq u(x) & \leq & 2 & \text{in } \Omega \end{array} \right.$$

where

$$q_i = 100\sin(2\pi x_1)\sin(2\pi x_2)\sin(2\pi x_3)$$

N	n	p	q	# 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 approx. 20 hours for the first iteration

Outline

Introduction

Interior-Point with Inexact Steps

Numerical Results

Summary and Future Work

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 very 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, under review for SIAM Journal on Scientific Computing.