Matrix-free Primal-Dual Methods and Infeasibility Detection in Nonlinear Programming

Frank E. Curtis
New York University

involving joint work with
Richard H. Byrd, Jorge Nocedal, and Andreas Wächter

IBM, 2008

Outline

Matrix-free Primal-Dual Methods for Equality Constrained Optimization
- Motivation for Matrix-free Techniques
- Penalty Function Model Reductions and Handling Rank Deficiency
- Convergence Results and Numerical Experiments

Infeasibility Detection in Nonlinear Programming
- "Solving" Infeasible Problems
- Handling the Penalty Parameter in a Penalty-SQP Method
- Conclusion and Future Work

Outline

Matrix-free Primal-Dual Methods for Equality Constrained Optimization
- Motivation for Matrix-free Techniques
- Penalty Function Model Reductions and Handling Rank Deficiency
- Convergence Results and Numerical Experiments

Infeasibility Detection in Nonlinear Programming
- "Solving" Infeasible Problems
- Handling the Penalty Parameter in a Penalty-SQP Method
- Conclusion and Future Work

Equality constrained optimization

We consider *very large* problems of the form

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

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are smooth functions

- First, we describe a matrix-free primal-dual method for nice cases
- Then, we show how we handle (near) rank deficiency
- Assume strict convexity here, but we can handle non-convexity as well

First-order optimality

Defining the Lagrangian

$$\mathcal{L}(x, \lambda) \triangleq f(x) + \lambda^T c(x)$$

we are interested in finding a first-order optimal point; i.e., one satisfying

$$\nabla \mathcal{L} = \begin{bmatrix} g(x) + A(x)^T \lambda \\ c(x) \end{bmatrix} = 0$$

where $g(x)$ is the gradient of $f(x)$ and $A(x)$ is the Jacobian of $c(x)$

Method of choice: Newton/SQP

A Newton iteration from the point (x_k, λ_k) has the form

$$\begin{bmatrix} W(x_k, \lambda_k) & A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g(x_k) + A(x_k)^T \lambda_k \\ c(x_k) \end{bmatrix}$$

where $W(x_k, \lambda_k) \approx \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k)$, which is equivalent to solving the sequential quadratic programming (SQP) subproblem

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \ & f(x_k) + g(x_k)^T d + \tfrac{1}{2} d^T W(x_k, \lambda_k) d \\ \text{s.t. } & c(x_k) + A(x_k) d = 0 \end{aligned}$$

Algorithm

for $k = 0, 1, 2, \ldots$

- Evaluate f_k, g_k, c_k, A_k, and W_k
- Solve the *primal-dual* equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix}$$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \; & f(x_k) + g(x_k)^T d + \tfrac{1}{2} d^T W(x_k, \lambda_k) d \\ \text{s.t.} \; & c(x_k) + A(x_k) d = 0 \end{aligned}$$

- Update iterate $(x_k, \lambda_k) \leftarrow (x_k, \lambda_k) + (d_k, \delta_k)$

Algorithm, globalized with an exact penalty function

for $k = 0, 1, 2, \ldots$

- Evaluate f_k, g_k, c_k, A_k, and W_k
- Solve the *primal-dual* equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix}$$

$$\min_{d \in \mathbb{R}^n} \; f(x_k) + g(x_k)^T d + \tfrac{1}{2} d^T W(x_k, \lambda_k) d \quad \text{s.t. } c(x_k) + A(x_k) d = 0$$

- Set the penalty parameter π_k
- Perform a line search for the merit function

$$\phi(x; \pi_k) \triangleq f(x) + \pi_k \|c(x)\|$$

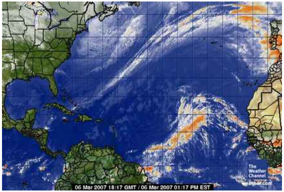
to find $\alpha_k \in (0, 1]$ satisfying the Armijo condition

$$\phi(x_k + \alpha_k d_k; \pi_k) \le \phi(x_k; \pi_k) + \eta \alpha_k D\phi(d_k; \pi_k)$$

- Update iterate $(x_k, \lambda_k) \leftarrow (x_k, \lambda_k) + \alpha_k (d_k, \delta_k)$

Example: Data assimilation in weather forecasting

- Goal: up-to-date global weather forecast for the next 7 to 10 days [1]



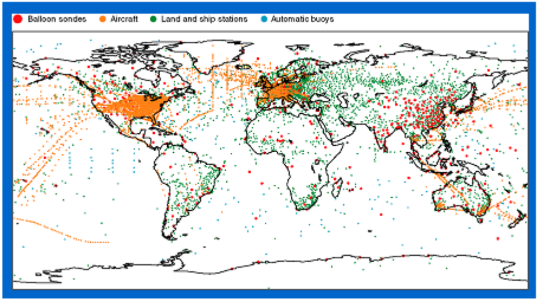
- If an entire *initial state* of the atmosphere (temperatures, pressures, wind patterns, humidities) were known at a certain point in time, then an accurate forecast could be obtained by integrating atmospheric model equations forward in time
- Flow described by Navier-Stokes and further sophistications of atmospheric physics and dynamics (none of which will be discussed here)

[1] (Fisher, Nocedal, Trémolet, and Wright, 2007)

In reality: Partial information known

Limited amount of data (satellites, buoys, planes, ground-based sensors)

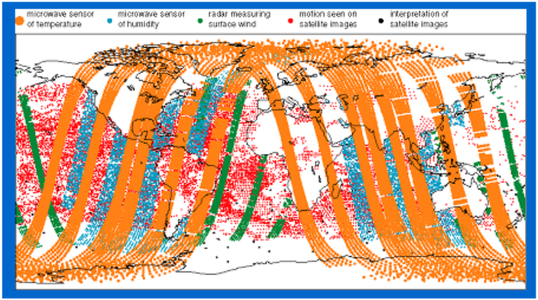
- Each observation is subject to error
- Nonuniformly distributed around the globe (satellite paths, densely-populated areas)



In reality: Partial information known

Limited amount of data (satellites, buoys, planes, ground-based sensors)

- Each observation is subject to error
- Nonuniformly distributed around the globe (satellite paths, densely-populated areas)



Data assimilation: Defining the unknowns

Currently in operational use at the European Centre for Medium-Range Weather Forecasts (ECMWF)

- We want values for an initial state, call it x^0
- For a given x^0, we could integrate our atmospheric models forward to forecast the state of the atmosphere at N time points

$$x^i = \mathcal{M}(x^{i-1}), \; i = 1, \dots, N$$

(x^i: state of the atmosphere at time i)

- Observe the atmosphere at these N time points

$$y^1, \dots, y^N$$

(y^i: observed state at time i)

- Let y^0 (background state) be values at initial time point obtained from previous forecast — carry over old information

Data assimilation as an optimization problem

Choose x^0 as the initial state "most likely" to have given the observed data:

$$\min_{x=(x^0,\ldots,x^N)} f(x) \triangleq \frac{1}{2}\|(x^0-y^0, x^1-y^1, \ldots, x^N-y^N)\|_R^2$$

$$\text{s.t. } c(x) = \begin{bmatrix} x^1 - \mathcal{M}(x^0) \\ x^2 - \mathcal{M}(x^1) \\ \vdots \\ x^N - \mathcal{M}(x^{N-1}) \end{bmatrix} = 0$$

- Objective: distance measure between observed and expected behavior
- In current forecasts, x^0 contains approximately 3×10^8 unknowns
- constraints are nonconvex (nonlinear operators $\mathcal{M}^i$)
- exact derivative information not available
- solutions needed in real-time
- ... bottom line: they cannot use contemporary SQP!

Working with matrices may be impractical

$$\boxed{\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix}}$$

What if...

- A_k, A_k^T, and W_k cannot be computed explicitly?
- A_k, A_k^T, and W_k cannot be stored?
- the *primal-dual matrix* cannot be factored?
- an iterative method may be more efficient?

If the products $A_k p$, $A_k^T q$, and $W_k y$ can be computed, we have answers...

Iterative step computations

From now on, let us assume that we have an iterative procedure for solving the primal-dual equations, which during each *inner iteration* yields (d_k, δ_k) solving

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}$$

for the residuals (ρ_k, r_k)

- How can we be sure that a given inexact step is *acceptable*?
- How small do the residuals need to be?

A naïve approach

Algorithm outline: given $0 < \kappa < 1$, for $k = 0, 1, 2, \ldots$

- Evaluate f_k, g_k, c_k, $A_k^T \lambda_k$
- Iteratively solve the *primal-dual* equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}$$

  until $\|(\rho_k, r_k)\| \leq \kappa \|(g_k + A_k^T \lambda_k, c_k)\|$
- Set the penalty parameter π_k
- Perform a line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \pi_k) \leq \phi(x_k; \pi_k) + \eta \alpha_k D\phi(d_k; \pi_k)$$

A naïve approach

Algorithm outline: given $0 < \kappa < 1$, for $k = 0, 1, 2, \ldots$

- Evaluate f_k, g_k, c_k, $A_k^T \lambda_k$
- Iteratively solve the *primal-dual* equations

$$\boxed{\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}}$$

  until $\|(\rho_k, r_k)\| \leq \kappa \|(g_k + A_k^T \lambda_k, c_k)\|$
- Set the penalty parameter π_k
- Perform a line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \pi_k) \leq \phi(x_k; \pi_k) + \eta \alpha_k \underbrace{D\phi(d_k; \pi_k)}_{>0 \ \forall \pi ?}$$

κ	2^{-1}	2^{-5}	2^{-10}
% Solved	45%	80%	86%

Optimization, not nonlinear equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}$$

$$\min_{d \in \mathbb{R}^n} \; f_k + g_k^T d + \tfrac{1}{2} d^T W_k d \quad \text{s.t. } c_k + A_k d = 0$$

Take (d_k, δ_k) and...

- ... "forget" about it being an inexact Newton step
- ... "forget" about it being an approximate SQP solution

We want a technique for determining if (d_k, δ_k) is acceptable that...

- ... allows for possibly very inexact solutions to Newton's equations
- ... integrates both step computation and step selection to solve the optimization problem

Central idea: Sufficient Model Reductions

Modern optimization algorithms work with models.

Take the penalty function

$$\phi(x;\pi) \triangleq f(x) + \pi\|c(x)\|$$

and consider the model

$$m_k(d;\pi) \triangleq f_k + g_k^T d + \pi\|c_k + A_k d\|$$

The reduction in m_k attained by d_k is computed easily as

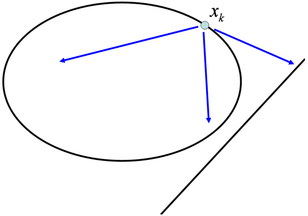
$$\begin{aligned}\Delta m_k(d_k;\pi) &\triangleq m_k(0;\pi) - m_k(d_k;\pi) \\ &= -g_k^T d_k + \pi(\|c_k\| - \|r_k\|)\end{aligned}$$

and yields

$$D\phi(d_k;\pi) \leq -\Delta m_k(d_k;\pi)$$

Main tool: “SMART” Tests

We develop two types of
Sufficient Merit function Approximation Reduction Termination Tests.

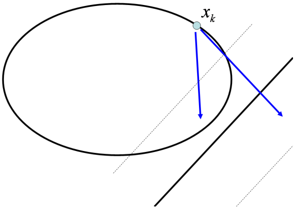


Termination Test I: A sufficient model reduction is attained for π_{k-1} (i.e., the most recent penalty parameter value):

$$\Delta m_k(d_k; \pi_{k-1}) = -g_k^T d_k + \pi_{k-1}(\|c_k\| - \|r_k\|) \gg 0$$

Main tool: “SMART” Tests

We develop two types of
Sufficient Merit function Approximation Reduction Termination Tests.



Termination Test II: A sufficient reduction in the constraint model is attained for some $\epsilon \in (0, 1)$

$$\|r_k\| \leq \epsilon \|c_k\|$$

Step acceptance criteria:

Model Reduction Condition. A step (d_k, δ_k) is acceptable if and only if

$$\Delta m_k(d_k; \pi_k) \geq \frac{1}{2} d_k^T W_k d_k + \sigma \pi_k \max\{\|c_k\|, \|c_k + A_k d_k\| - \|c_k\|\}$$

for some $\sigma \in (0,1)$ and an appropriate $\pi_k > 0$.

Termination Test I. For some $\sigma \in (0,1)$ and $\pi_k = \pi_{k-1}$ the Model Reduction Condition is satisfied and for some $\kappa \in (0,1)$ we have

$$\left\| \begin{bmatrix} \rho_k \\ r_k \end{bmatrix} \right\| \leq \kappa \left\| \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} \right\|$$

Termination Test II. For some $\epsilon \in (0,1)$ and $\beta > 0$ we have

$$\|r_k\| \leq \epsilon \|c_k\| \qquad \text{and} \qquad \|\rho_k\| \leq \beta \|c_k\|$$

and we set

$$\pi_k \geq \frac{g_k^T d_k + \frac{1}{2} d_k^T W_k d_k}{(1-\tau)(\|c_k\| - \|r_k\|)} \qquad \text{for } \tau \in (0,1)$$

Inexact SQP with SMART Tests[2]

Algorithm outline: for $k = 0, 1, 2 \ldots$

- Evaluate f_k, g_k, c_k, $A_k^T \lambda_k$
- Iteratively solve the *primal-dual* equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}$$

until Termination Test I or II holds

- Set the penalty parameter π_k
- Perform a line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \pi_k) \leq \phi(x_k; \pi_k) - \eta \alpha_k \Delta m_k(d_k; \pi_k)$$

[2] R. H. Byrd, F. E. Curtis, and J. Nocedal, "An Inexact SQP Method for Equality Constrained Optimization," to appear in SIAM Journal on Optimization.

(Near) Rank-deficient Jacobians

If at any point the Jacobian A of c is ill-conditioned or rank deficient, the Newton system

$$\begin{bmatrix} W(x_k,\lambda_k) & A(x_k)^T \\ A(x_k) & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g(x_k) + A(x_k)^T \lambda_k \\ c(x_k) \end{bmatrix}$$

and the SQP subproblem

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \ & f(x_k) + g(x_k)^T d + \tfrac{1}{2} d^T W(x_k,\lambda_k) d \\ \text{s.t.} \ & c(x_k) + A(x_k) d = 0 \end{aligned}$$

may not be well-defined or may lead to very long steps (i.e., $\|d_k\| \gg 0$, $\alpha_k \approx 0$, and algorithm may stall)

Even if we could solve the primal-dual equations exactly, the algorithm may fail

Regularizing the constraint model with trust regions

We decompose the step by first considering the trust region subproblem

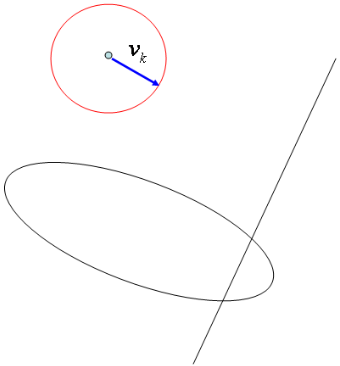
$$\min_{v \in \mathbb{R}^n} \ \tfrac{1}{2}\|c_k + A_k v\|^2$$
$$\text{s.t. } \|v\| \le \Omega_k$$

Notice that this subproblem fits well within our context of matrix-free optimization; e.g., apply CG/LSQR with Steihaug-Toint stop tests

Trust regions

The trust region keeps us in a local region of the search space:

$$\min_{v \in \mathbb{R}^n} \frac{1}{2}\|c_k + A_k v\|^2$$
$$\text{s.t. } \|v\| \leq \Omega_k$$



Trust regions

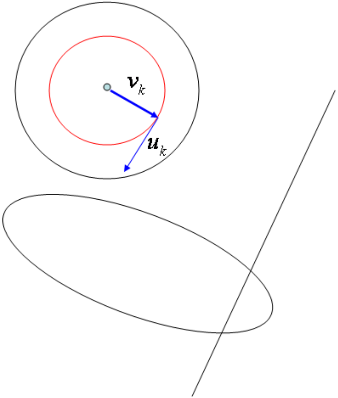
Once v is computed, we could consider computing a step toward optimality within a larger trust region:

$$\min_{u\in\mathbb{R}^n} \; (g_k + W_k v_k)^T u + \tfrac{1}{2} u^T W_k u$$
$$\text{s.t. } A_k u = 0, \quad \|u\| \le \Omega'_k,$$

but then we may need

$$Z_k \quad \text{s.t.} \quad A_k Z_k \approx 0$$

or to (approximately) project vectors onto the null space of A_k



Trust regions only for v!

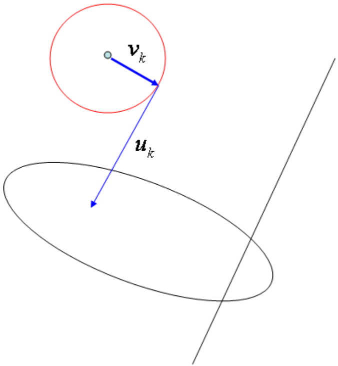
Instead, we set no trust region for u:

$$\min_{u \in \mathbb{R}^n} \ (g_k + W_k v_k)^T u + \tfrac{1}{2} u^T W_k u$$
$$\text{s.t. } A_k u = 0$$

which, with $d_k = v_k + u_k$, has the same solutions as

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = \begin{bmatrix} -(g_k + A_k^T \lambda_k) \\ A_k v_k \end{bmatrix}$$

Notice that this system is consistent (though perhaps (near) singular)



Setting the trust region radius

In fact, we propose a very specific form for the trust region radius:

$$\min_{v \in \mathbb{R}^n} \ \frac{1}{2}\|c_k + A_k v\|^2$$
$$\text{s.t. } \|v\| \leq \omega \|A_k^T c_k\|$$

for a given *constant* $\omega > 0$

- We incorporate problem information in the right-hand-side (note that a stationary point for the feasibility measure $\|c(x)\|$ has $\|A(x)^T c(x)\| = 0$)
- The radius is set dynamically without a heuristic update
- ω should be set to correspond to the reciprocal of the smallest allowable singular value of A_k

Inexact Newton with SMART Tests

Algorithm outline: for $k = 0, 1, 2 \ldots$

- Evaluate f_k, g_k, c_k, $A_k^T \lambda_k$
- Approximately solve (with an iterative method)

$$\min_{v \in \mathbb{R}^n} \tfrac{1}{2}\|c_k + A_k v\|^2$$

$$\text{s.t. } \|v\| \leq \omega \|A_k^T c_k\|$$

- Iteratively solve the *primal-dual* equations

$$\boxed{\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = - \begin{bmatrix} g_k + A_k^T \lambda_k \\ -A_k v_k \end{bmatrix} + \begin{bmatrix} \rho_k \\ r_k \end{bmatrix}}$$

<u>until a termination test is satisfied</u>

- Set the penalty parameter π_k
- Perform a line search to find $\alpha_k \in (0, 1]$ satisfying

$$\phi(x_k + \alpha_k d_k; \pi_k) \leq \phi(x_k; \pi_k) - \eta \alpha_k \Delta m_k(d_k; \pi_k)$$

Step acceptance criteria:[3]

Tangential Component Condition. The component u_k must satisfy

$$\|u_k\| \leq \psi \|v_k\| \quad \text{or} \quad (g_k + W_k v_k)^T u_k + \tfrac{1}{2} u_k^T W_k u_k \leq 0$$

Model Reduction Condition. A step (d_k, δ_k) is acceptable if and only if

$$\Delta m_k(d_k; \pi_k) \geq \tfrac{1}{2} u_k^T W_k u_k + \sigma \pi_k (\|c_k\| - \|c_k + A_k v_k\|)$$

for some $\sigma \in (0,1)$ and an appropriate $\pi_k > 0$.

Termination Test I. For some $\sigma \in (0,1)$ and $\pi_k = \pi_{k-1}$ the Tangential Component Condition holds, the Model Reduction Condition is satisfied, and for some $\kappa \in (0,1)$ we have

$$\left\| \begin{bmatrix} \rho_k \\ r_k \end{bmatrix} \right\| \leq \kappa \min \left\{ \left\| \begin{bmatrix} g_k + A_k^T \lambda_k \\ A_k v_k \end{bmatrix} \right\|, \left\| \begin{bmatrix} g_{k-1} + A_{k-1}^T \lambda_k \\ A_{k-1} v_{k-1} \end{bmatrix} \right\| \right\}$$

Termination Test II. For some $\epsilon \in (0,1)$ and $\beta > 0$, the Tangential Component Condition holds and we have

$$\begin{aligned} \|c_k\| - \|c_k + A_k d_k\| &\geq \epsilon(\|c_k\| - \|c_k + A_k v_k\|) \\ \text{and} \quad \|\rho_k\| &\leq \beta(\|c_k\| - \|c_k + A_k v_k\|), \\ \text{and we set} \quad \pi_k &\geq (g_k^T d_k + \tfrac{1}{2} u_k^T W_k u_k)/((1-\tau)(\|c_k\| - \|c_k + A_k d_k\|)) \end{aligned}$$

[3] F. E. Curtis, J. Nocedal, and A. Wächter, in preparation.

Main result

Assumptions: The generated sequence $\{x_k, \lambda_k\}$ is contained in a convex set over which f and c and their first derivatives are bounded, and the iterative linear system solver can solve the primal-dual equations to an arbitrary accuracy

Theorem: If all limit points satisfy the linear independence constraint qualification (LICQ), then $\{\pi_k\}$ is bounded and

$$\lim_{k\to\infty} \left\| \begin{bmatrix} g_k + A_k^T \lambda_{k+1} \\ c_k \end{bmatrix} \right\| = 0$$

Otherwise,

$$\lim_{k\to\infty} \left\| A_k^T c_k \right\| = 0$$

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

$$\lim_{k\to\infty} \left\| g_k + A_k^T \lambda_{k+1} \right\| = 0$$

Brief overview of analysis

- The step length (d_k, v_k, u_k) is explicitly or implicitly controlled...
- The reduction in the model of the penalty function satisfies

$$\Delta m_k(d_k; \pi_k) \geq \gamma(\|u_k\|^2 + \pi_k \|A_k^T c_k\|^2)$$

- In particular

$$\Delta m_k(d_k; \pi_k) \geq \gamma' \|A_k^T c_k\|^2 \Rightarrow \lim_{k \to \infty} \|A_k^T c_k\| = 0$$

- If $\{\pi_k\}$ remains bounded (guaranteed if LICQ holds), then

$$\lim_{k \to \infty} \left\| g_k + A_k^T \lambda_{k+1} \right\| = 0,$$

and otherwise $\pi \to \infty$

Implementation details

We use MINRES to solve the primal-dual equations

$$\begin{bmatrix} W_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \delta_k \end{bmatrix} = \begin{cases} -\begin{bmatrix} g_k + A_k^T \lambda_k \\ c_k \end{bmatrix} \\ -\begin{bmatrix} g_k + A_k^T \lambda_k \\ -A_k v_k \end{bmatrix} \end{cases}$$

and LSQR (algebraically equivalent to CG, but with better numerical properties) with Steihaug-Toint stop tests to solve the trust region subproblem

$$\min_{v \in \mathbb{R}^n} \tfrac{1}{2}\|c_k + A_k v\|^2$$
$$\text{s.t. } \|v\| \le \omega \|A_k^T c_k\|$$

All experiments performed in Matlab

Problems with rank-deficiency

Total of 73 problems from the CUTEr collection

- Original and perturbed models have

$$c_1(x) = 0 \quad \text{and} \quad \begin{cases} c_1(x) = 0 \\ c_1(x) - c_1^2(x) = 0 \end{cases}$$

  respectively
- Success rates:

	iSQP	TRINS
Original	95%	100%
Perturbed	46%	93%

- A few of the failures of TRINS was due to the Maratos effect, so second-order correction steps may be beneficial

Conclusion

We have...

- ... focused on a particular class of problems to which contemporary optimization techniques cannot be applied
- ... considered the fundamental question of how to ensure global convergence via a type of inexact SQP/Newton approach
- ... developed a methodology where inexact solutions are appraised based on the reductions obtained in linear models of an exact penalty function
- ... extended the algorithm and analysis for cases involving rank deficiency (and nonconvexity)

Outline

Matrix-free Primal-Dual Methods for Equality Constrained Optimization
Motivation for Matrix-free Techniques
Penalty Function Model Reductions and Handling Rank Deficiency
Convergence Results and Numerical Experiments

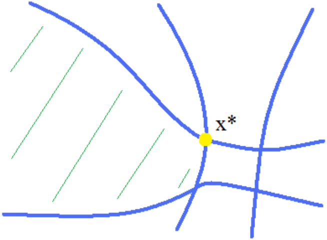
Infeasibility Detection in Nonlinear Programming
"Solving" Infeasible Problems
Handling the Penalty Parameter in a Penalty-SQP Method
Conclusion and Future Work

Infeasible Nonlinear Programming

We consider the optimization problems

$$(OPT) \triangleq \left\{ \begin{array}{l} \min\ f(x) \\ \text{ s.t. } c(x) \geq 0 \end{array} \right\} \text{ and } (FEAS) \triangleq \left\{ \min \sum_{i=1}^{t} \max\{-c^i(x), 0\} \right\}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are smooth functions

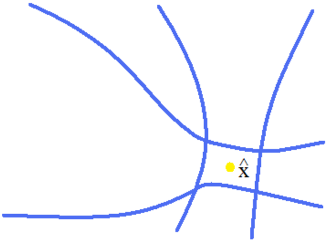


Infeasible Nonlinear Programming

We consider the optimization problems

$$(OPT) \triangleq \left\{ \begin{array}{l} \min \; f(x) \\ \text{ s.t. } \; c(x) \geq 0 \end{array} \right\} \quad \text{and} \quad (FEAS) \triangleq \left\{ \min \; \sum_{i=1}^{t} \max\{-c^i(x), 0\} \right\}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are smooth functions



Infeasible Nonlinear Programming

We consider the optimization problems

$$(OPT) \triangleq \left\{ \begin{array}{l} \min\ f(x) \\ \text{s.t. } c(x) \geq 0 \end{array} \right\} \quad \text{and} \quad (FEAS) \triangleq \left\{ \min \sum_{i=1}^{t} \max\{-c^i(x), 0\} \right\}$$

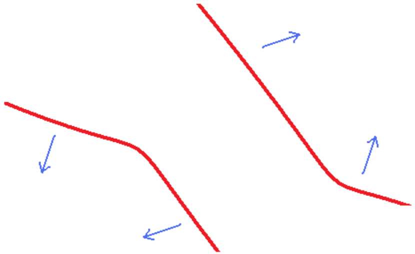
where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^t$ are smooth functions

- We want to solve (OPT) when a *feasible* point exists (i.e., $\exists x \in \mathbb{R}^n$ s.t. $c(x) \geq 0$)
- Otherwise, the algorithm should solve $(FEAS)$ when (OPT) is *infeasible*
- Many optimization methods focus on the efficient solution of (OPT), often with guarantees toward solutions of $(FEAS)$ if the problem is infeasible
- ... however, this latter feature is often treated as an afterthought and the rate at which the method converges can be exceedingly slow

Focus on active set methods

- Interior-point methods are known to behave poorly on infeasible problems:

$$\left\{ \begin{array}{l} \min\ f(x) - \mu \sum_{i=1}^{t} \ln s^i \\ \text{s.t. } c(x) - s = 0,\ s > 0 \end{array} \right\} \quad \Leftarrow \quad \text{true interior is empty}$$



Focus on active set methods

- Interior-point methods are known to behave poorly on infeasible problems:

$$\left\{ \begin{array}{l} \min\ f(x) - \mu \sum_{i=1}^{t} \ln s^i \\ \text{s.t. } c(x) - s = 0,\ s > 0 \end{array} \right\} \Leftarrow \quad \text{true interior is empty}$$

- Active-set methods present another option:
  Running `SNOPT` and `KNITRO` on NEOS:

Problem	SNOPT	KNITRO
optprloc1	11 itrs	10 itrs
optprloc2	14 itrs	44 itrs
optprloc3	30 itrs	29 itrs
c-reload-14c	37 itrs	1000+ itrs
batch	1000+ itrs	37 itrs

One option: Feasibility restoration

If the optimization problem (OPT) appears locally infeasible, then switch to an algorithm that exclusively attempts to solve the feasibility problem $(FEAS)$:[4]

$$(OPT) \triangleq \left\{ \begin{array}{l} \min \ f(x) \\ \text{ s.t. } \ c(x) \geq 0 \end{array} \right\} \quad \leftrightarrow \quad (FEAS) \triangleq \left\{ \min \ \sum_{i=1}^{t} \max\{-c^i(x), 0\} \right\}$$

If the algorithm iterates become (near) feasible, return to the optimization problem

[4] e.g., see Fletcher and Leyffer, 1997

A single algorithm for an entire problem family

Our goal is to design a *single* optimization algorithm designed for the fast solution of (OPT), or the fast solution of $(FEAS)$ when (OPT) is infeasible, that does not *switch* between two separate techniques

$$(OPT) \triangleq \begin{Bmatrix} \min & f(x) \\ \text{s.t.} & c(x) \geq 0 \end{Bmatrix} \quad \leftrightarrow \quad (FEAS) \triangleq \begin{Bmatrix} \min & e^T r \\ \text{s.t.} & c(x) + r \geq 0 \\ & r \geq 0 \end{Bmatrix}$$

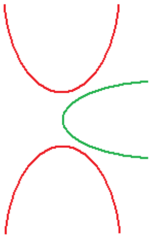
We combine (OPT) and $(FEAS)$ to define

$$(P) \triangleq \begin{Bmatrix} \min & \frac{1}{\pi} f(x) + e^T r \\ \text{s.t.} & c(x) + r \geq 0 \\ & r \geq 0 \end{Bmatrix}$$

where $\pi > 0$ is a penalty parameter to be updated dynamically

An ideal run of KNITRO

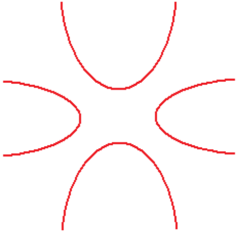
$$\begin{aligned} &\min\ x_1 \\ \text{s.t.}\ & -x_1^2 + x_2 - 1 \geq 0 \\ & -x_1^2 - x_2 - 1 \geq 0 \\ & x_1 - x_2^2 \geq 0 \\ & -x_1 + x_2^2 \geq 0 \end{aligned}$$



Iter	Objective	Feas err	Opt err	||Step||	pi
13	1.061997e-03	1.034e+00	1.000e+00	6.192e-02	1.000e+02
14	-6.689357e-05	1.000e+00	9.097e-01	3.379e-02	1.000e+02
15	-4.474151e-09	1.000e+00	9.999e-01	9.460e-05	1.000e+02
16	-2.001803e-17	1.000e+00	1.000e+00	6.327e-09	1.000e+02

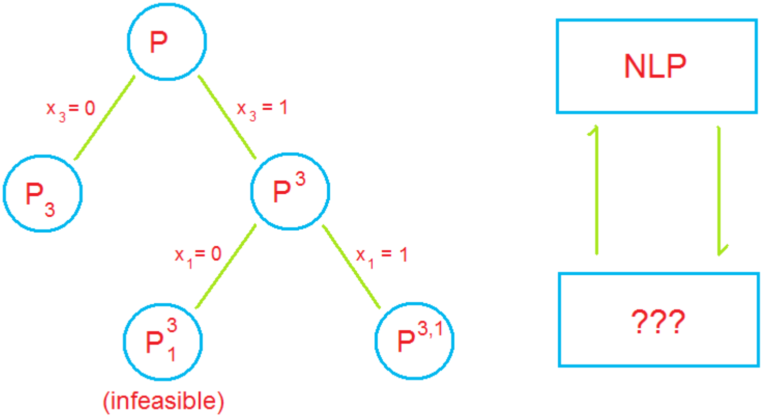
A *less than* ideal run of KNITRO

$$
\begin{aligned}
\min\ & x_1 + x_2 \\
\text{s.t.}\ & -x_1^2 + x_2 - 1 \geq 0 \\
& -x_1^2 - x_2 - 1 \geq 0 \\
& x_1 - x_2^2 - 1 \geq 0 \\
& -x_1 - x_2^2 - 1 \geq 0
\end{aligned}
$$

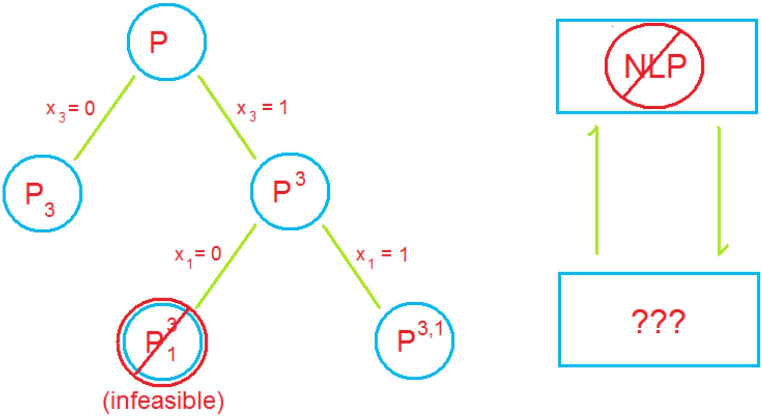


Iter	Objective	Feas err	Opt err	||Step||	pi
13	-5.000000e-07	1.000e+00	1.000e+00	0.000e+00	1.000e+06
14	-5.000000e-08	1.000e+00	1.000e+00	3.182e-07	1.000e+07
15	-5.000000e-08	1.000e+00	1.000e+00	0.000e+00	1.000e+07
16	-5.000000e-09	1.000e+00	1.000e+00	3.182e-08	1.000e+08

Effects compounded in MINLP methods



Effects compounded in MINLP methods



Summary

- There is a need for algorithms that converge quickly, regardless of whether the problem is feasible or infeasible
- Interior-point methods are known to perform poorly in infeasible cases, but active set methods seem promising
- Room for improvement in active set methods, too
- Feasibility restoration techniques are an option, but we prefer a smooth transition between solving (OPT) and solving $(FEAS)$
- When π remains finite, convergence can be fast since, after a point, we are solving a single problem
- However, we need to analyze the $\pi \to \infty$ case as well...

Our method for step computation and acceptance

We generate a step via the quadratic subproblem

$$(Q) \triangleq \begin{array}{ll} \min & q_k(d;\pi) \triangleq \frac{1}{\pi}\nabla f_k^T d + \frac{1}{2} d^T W_k d + e^T s \\ \text{s.t.} & c_k + \nabla c_k^T d + s \geq 0, \quad s \geq 0 \end{array}$$

where W_k is an approximation for the Hessian of the Lagrangian of (P), and we measure progress with the exact penalty function

$$\phi(x;\pi) \triangleq \frac{1}{\pi} f(x) + \sum_{i=1}^{t} \max\{-c^i(x), 0\}$$

We see later on that this SQP approach has the benefit that it can identify the correct *active set* near a "solution" point for π sufficiently large

A Penalty-SQP algorithm

Step 0. Initialize x_0 and set $\eta \in (0,1)$, $\tau \in (0,1)$ and $k \leftarrow 0$

Step 1. If x_k solves (OPT) or $(FEAS)$, then stop

Step 2. Compute a value for the penalty parameter, call it π_k

Step 3. Compute d_k by solving (Q) with $\pi \leftarrow \pi_k$

Step 4. Let α_k be the first member of the sequence $\{1, \tau, \tau^2, \dots\}$ s.t.

$$\phi(x_k; \pi_k) - \phi(x_k + \alpha_k d_k; \pi_k) \geq \eta\alpha_k[q_k(0; \pi_k) - q_k(d_k; \pi_k)]$$

Step 5. Update $x_{k+1} \leftarrow x_k + \alpha_k d_k$, go to Step 1

A Penalty-SQP algorithm

Step 0. Initialize x_0 and set $\eta \in (0, 1)$, $\tau \in (0, 1)$ and $k \leftarrow 0$

Step 1. If x_k solves (OPT) or $(FEAS)$, then stop

Step 2. Compute a value for the penalty parameter, call it π_k

Step 3. Compute d_k by solving (Q) with $\pi \leftarrow \pi_k$

Step 4. Let α_k be the first member of the sequence $\{1, \tau, \tau^2, \ldots\}$ s.t.

$$\phi(x_k; \pi_k) - \phi(x_k + \alpha_k d_k; \pi_k) \geq \eta \alpha_k [q_k(0; \pi_k) - q_k(d_k; \pi_k)]$$

Step 5. Update $x_{k+1} \leftarrow x_k + \alpha_k d_k$, go to Step 1

Strategy for fast convergence

Hitting a moving target:

$$x_k \longrightarrow x_\pi \longrightarrow \hat{x}$$

where

$$x_k \triangleq \text{ } k\text{th iterate of the algorithm}$$

$$x_\pi \triangleq \text{ solution of penalty problem } (P)$$

$$\hat{x} \triangleq \text{ infeasible stationary point of } (OPT)\text{, solution of } (FEAS)$$

We aim to show, for some $C, C' > 0$,

$$\begin{aligned} \|x_{k+1} - \hat{x}\| &\leq \|x_{k+1} - x_\pi\| + \|x_\pi - \hat{x}\| \\ &\leq C\|x_k - x_\pi\|^2 + O(1/\pi) \\ &\leq C'\|x_k - \hat{x}\|^2 + O(1/\pi), \end{aligned}$$

so convergence is quadratic if $(1/\pi) \propto \|x_k - \hat{x}\|^2$

Optimality conditions for problem (P)

First-order optimality conditions for

$$(P) \triangleq \left\{ \min \ \tfrac{1}{\pi} f(x) + e^T r, \quad \text{s.t. } c(x) + r \geq 0, \ r \geq 0 \right\} :$$

$$\left\{ \begin{aligned} \tfrac{1}{\pi}\nabla f(x) - \sum_{i\in\mathcal{I}} \lambda^i \nabla c^i(x) &= \ 0 \\ 1 - \lambda^i - \sigma^i &= \ 0, \quad i \in \mathcal{I} \\ \lambda^i (c^i(x) + r^i) &= \ 0, \quad i \in \mathcal{I} \\ \sigma^i r^i &= \ 0, \quad i \in \mathcal{I} \\ c^i(x) + r^i &\geq \ 0, \quad i \in \mathcal{I} \\ r, \lambda, \sigma &\geq \ 0 \end{aligned} \right\}$$

At an infeasible stationary point $\hat{x}$ we define

$$\hat{\mathcal{A}} = \{i : c^i(\hat{x}) = 0\}, \quad \hat{\mathcal{V}} = \{i : c^i(\hat{x}) < 0\}, \quad \hat{\mathcal{S}} = \{i : c^i(\hat{x}) > 0\}$$

as the sets of *active*, *violated*, and *strictly satisfied* constraints.

Assumptions

The point $(\hat{x}, \hat{r}, \hat{\lambda}, \hat{\sigma})$ is a first-order optimal solution of (P) at which the following conditions hold:

- (Regularity) $\nabla c(\hat{x})^T$ has full row rank;
- (Strict Complementarity) $\hat{\lambda}^i > 0$ for all $i \in \hat{\mathcal{A}}$;
- (Second Order Sufficiency) The Hessian of the Lagrangian for problem (P) with $\pi = \infty$, denoted by $\hat{W}$, satisfies $d^T \hat{W} d > 0$ for all $d \neq 0$ such that $\nabla c(\hat{x})^T d = 0$

The optimality conditions now reduce to: (define $\rho = 1/\pi$)

$$F(x, \lambda_{\hat{\mathcal{A}}}, \rho) = \begin{bmatrix} \rho \nabla f(x) - \sum_{i \in \hat{\mathcal{A}}} \lambda^i \nabla c^i(x) - \sum_{i \in \hat{\mathcal{V}}} \nabla c^i(x) \\ c_{\hat{\mathcal{A}}}(x) \end{bmatrix} = 0$$

$$\lambda_{\hat{\mathcal{A}}} \in (0, 1)$$

(all other values can be determined uniquely)

Lemma 1: $x_\pi \longrightarrow \hat{x}$

For all π sufficiently large the penalty problem (P) has a solution x_π with the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. Moreover,

$$\|x_\pi - \hat{x}\| = O(1/\pi)$$

Proof.

We have $F(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0) = 0$. Differentiating F yields:

$$\frac{\partial F(x, \lambda_{\hat{\mathcal{A}}}, \rho)}{\partial(x, \lambda_{\hat{\mathcal{A}}})} = \begin{bmatrix} W(x, \lambda_{\hat{\mathcal{A}}}, \rho) & -\nabla c_{\hat{\mathcal{A}}}(x) \\ \nabla c_{\hat{\mathcal{A}}}(x)^T & 0 \end{bmatrix},$$

which is nonsingular under our assumptions. The implicit function theorem then implies that there is an open neighborhood $\mathcal{N} \in \mathbb{R}$ containing $\rho = 0$ such that

$$F(x(\rho), \lambda_{\hat{\mathcal{A}}}(\rho), \rho) = 0 \quad \text{for all} \quad \rho \in \mathcal{N}.$$

Then, since $\hat{\lambda}_{\hat{\mathcal{A}}} \in (0, 1)$, $(x(\rho), \lambda_{\hat{\mathcal{A}}}(\rho), \rho)$ satisfies the first-order optimality conditions for ρ sufficiently small (π large)

Lemma 1: $x_\pi \longrightarrow \hat{x}$

For all π sufficiently large the penalty problem (P) has a solution x_π with the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. Moreover,

$$\|x_\pi - \hat{x}\| = O(1/\pi)$$

Example: (recall $\rho = 1/\pi$)

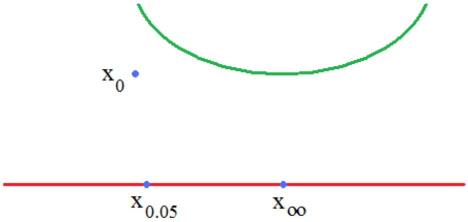
$$\begin{aligned} \min\ & \rho\left((x_1+1)^2 + (x_2-1)^2\right) + r_1 + r_2 \\ \text{s.t.}\ & -x_1^2 + x_2 - 1 + r_1 \geq 0 \\ & -100x_2 + r_2 \geq 0 \\ & (r_1, r_2) \geq 0 \end{aligned}$$

Lemma 1: $x_\pi \to \hat{x}$

For all π sufficiently large the penalty problem (P) has a solution x_π with the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. Moreover,

$$\|x_\pi - \hat{x}\| = O(1/\pi)$$

Example:

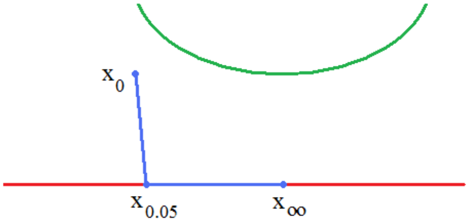


Lemma 1: $x_\pi \longrightarrow \hat{x}$

For all π sufficiently large the penalty problem (P) has a solution x_π with the same sets of active, violated, and strictly satisfied constraints as $\hat{x}$. Moreover,

$$\|x_\pi - \hat{x}\| = O(1/\pi)$$

Example:



Lemma 2: $x_k \longrightarrow x_\pi \longrightarrow \hat{x}$

For π sufficiently large and for x_k sufficiently close to x_π, the solution of the SQP subproblem identifies the same sets of active, violated, and strictly satisfied constraints as x_π (and $\hat{x}$). Then, standard Newton analysis for equality constrained optimization yields for some $C > 0$:

$$\|x_{k+1} - x_\pi\| \leq C\|x_k - x_\pi\|^2$$

Proof.

Similar to before, at $(x, \lambda_{\hat{\mathcal{A}}}, \rho) = (\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0)$ the SQP step is the solution $(d, \delta_{\hat{\mathcal{A}}}) = (0, \hat{\lambda}_{\hat{\mathcal{A}}})$ to:

$$\begin{bmatrix} W(x, \lambda_{\hat{\mathcal{A}}}, \rho) & -\nabla c_{\hat{\mathcal{A}}}(x) \\ \nabla c_{\hat{\mathcal{A}}}^T(x) & 0 \end{bmatrix} \begin{bmatrix} d \\ \delta_{\hat{\mathcal{A}}} \end{bmatrix} = - \begin{bmatrix} \rho \nabla f(x) - \sum_{i \in \hat{\mathcal{V}}} \nabla c^i(x) \\ c_{\hat{\mathcal{A}}}(x) \end{bmatrix}$$

This matrix is nonsingular and the solution varies continuously with $(x, \lambda_{\hat{\mathcal{A}}}, \rho)$ near $(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}}, 0)$, so since $\hat{\lambda}^i \in (0,1)$ for $i \in \hat{\mathcal{A}}$ the solution of the SQP subproblem can be obtained via this linear system (setting $\delta_{\hat{\mathcal{V}}} = 1$ and $\delta_{\hat{\mathcal{S}}} = 0$) for $(x, \lambda_{\hat{\mathcal{A}}})$ near $(\hat{x}, \hat{\lambda}_{\hat{\mathcal{A}}})$ and ρ small (π large) □

Main result

Thus, we find:

$$\begin{aligned} \|x_{k+1} - \hat{x}\| &\leq \|x_{k+1} - x_\pi\| + \|x_\pi - \hat{x}\| \text{ (triangle inequality)} \\ &\leq C\|x_k - x_\pi\|^2 + O(1/\pi) \text{ (Lemmas 1 and 2)} \\ &\vdots \\ &\leq C'\|x_k - \hat{x}\|^2 + O(1/\pi), \end{aligned}$$

so convergence is quadratic if $(1/\pi) \propto \|x_k - \hat{x}\|^2$; e.g., $1/\pi$ proportional to the squared optimality error of the problem $(FEAS)$

Summary

- We have discussed methods for the fast solution of infeasible optimization problems
- We have analyzed a penalty-SQP approach that transitions smoothly between solving an optimization problem and its feasibility problem counterpart
- We have shown that the approach can converge quadratically if the penalty parameter is handled correctly

Future work

- How can we construct a practical method for updating π that satisfies our condition? e.g., consider the auxiliary problem

$$\begin{aligned} \min \ & \sum s^i \\ \text{s.t. } & c_k + \nabla c_k^T d + s \geq 0, \quad s \geq 0 \end{aligned}$$

  and set π_k so that the reduction in linearized feasibility of the SQP problem is proportional to that achieved by the solution of this problem – can this do the trick?

- Can we relax our assumptions? For example, for many infeasible problems, the Hessian of the Lagrangian is not positive definite at $\hat{x}$