An Inexact Sequential Quadratic Optimization Method for Nonlinear Optimization

Frank E. Curtis, Lehigh University

involving joint work with

Travis Johnson, Northwestern University
Daniel P. Robinson, Johns Hopkins University
Andreas Wächter, Northwestern University

SIAM Conference on Optimization — San Diego, CA

22 May 2014
Outline

Motivation

Algorithm Description

Numerical Experiments

Summary
Our goal is to solve a large-scale constrained nonlinear optimization problem:

\[
\min_{x \in \mathbb{R}^n} f(x) \\
\text{s.t. } c(x) = 0, \; \bar{c}(x) \leq 0.
\]

(NLP)

If (NLP) is infeasible, then at least we want to minimize constraint violation:

\[
\min_{x \in \mathbb{R}^n} v(x), \; \text{where } v(x) := \|c(x)\|_1 + \|[\bar{c}(x)]^+\|_1.
\]

(FP)

Any feasible point for (NLP) is an optimal solution of (FP).
Demands on algorithms for large-scale constrained nonlinear optimization:

- Scalable step computation
- Effectively handles negative curvature
- Superlinear convergence in primal-dual space
- Asymptotic monotonicity (consistent progress toward solution)
- Active-set detection and warm-starting

Traditional methods (SQO, IP, AL) only satisfy subsets of these demands.

- "I can solve my problem quickly enough with an IP method using matrix factorizations"... then you’re not interested in what I have to say.
- If you can only afford a few NLP iterations, are solving sequences of similar problems, and/or need a method that can converge fast locally, ...

---

1 Paraphrased from Zavala and Anitescu (2014)
Sequential quadratic optimization (SQO)

SQO advantages:
- “Parameter free” search direction computation (ideally)
- Strong global convergence properties and behavior
- Active-set identification $\rightarrow$ Newton-like local convergence

SQO disadvantages:
- No “best” way to handle negative curvature/inconsistent subproblems
- Quadratic subproblems (QPs) are expensive to solve exactly — not scalable!

Questions
- Employing scalable QP solvers, can we exploit inexact solves?
- From inexactness, do we lose global/local convergence? active set detection?
- Is there any benefit in applying a scalable method for the QP subproblems, as opposed to applying a similar scalable method to (NLP) directly?

Sit and solve QP? Or move on nonlinear problem as in filterSD; Fletcher (2012)
Inexact SQO vs. inexact SQO

Inexact SQO may mean inexact gradients with (in)exact QP solves:

▶ Dennis, El-Alem, Maciel (1997)
▶ Diehl, Walther, Bock, Kostina (2010)
▶ Jäger, Sachs (1997)
▶ Heinkenschloss, Vicente (2000)
▶ Walther, Vetukuri, Biegler (2012)

Inexact SQO may also mean exact gradients with inexact QP solves:

▶ Byrd, Curtis, Nocedal (2008)
▶ Izmailov, Solodov (2010)
▶ Leibfritz, Sachs (1999)
▶ Morales, Nocedal, Wu (2010)

In this talk, we are interested in the latter type of inexact SQO.
Algorithmic framework: Classic

NLP solver

approximation model

solution

QP solver

approximation model

solution

linear solver
Algorithmic framework: **Inexact**

NLP solver

- approximation model
- termination conditions

QP solver

- approximate solution
- step type

linear solver

- approximation model
- termination conditions

- approximate solution
- step type
Contributions:

- Broad, but consistent, framework; Gill, Robinson (2013), pdAL/sSQO
- Implementable termination conditions for inexact QP solves
- No specific QP solver required
- Global convergence guarantees (feasible and infeasible problems)
- Future work: Fast local convergence (feasible and infeasible problems)\(^2\)

Algorithmic features:

- Allows “generic” inexactness in QP solutions
- Convex combination of “optimality” and “feasibility” steps
- Negative curvature handled with dynamic Hessian modifications
- Separate multipliers for (NLP) and (FP)
- Dynamic updates for penalty parameter and Lagrange multipliers

\(^2\)Avoid using “Cauchy points” that only yield minimal progress for global convergence.
Fritz John and penalty functions

Define the Fritz John (FJ) function

\[ \mathcal{F}(x, y, \bar{y}, \mu) := \mu f(x) + c(x)^T y + \bar{c}(x)^T \bar{y} \]

and the \( \ell_1 \)-norm exact penalty function

\[ \phi(x, \mu) := \mu f(x) + v(x). \]

\( \mu \geq 0 \) acts as objective multiplier/penalty parameter.
Optimality conditions

\textbf{(NLP)}:
\[
\min_x f(x) \\
\text{s.t. } c(x) = 0, \; \bar{c}(x) \leq 0
\]

\textbf{(FP)}:
\[
\min_x v(x) := \left\| \left[ \begin{array}{c} c(x) \\ \bar{c}(x) \end{array} \right] \right\|_1
\]

\textbf{(PP)}:
\[
\min_x \phi(x, \mu) := \mu f(x) + v(x)
\]

\textbf{(FJ)}:
\[
\mathcal{F}(x, y, \bar{y}, \mu) := \\
\mu f(x) + c(x)^T y + \bar{c}(x)^T \bar{y}
\]

KKT conditions for (FP) and (PP) expressed with residual
\[
\rho(x, y, \bar{y}, \mu) := \\
\begin{bmatrix}
\mu g(x) + J(x)y + \bar{J}(x)\bar{y} \\
\min\{[c(x)]^+, e - y\} \\
\min\{[c(x)]^-, e + y\} \\
\min\{[\bar{c}(x)]^+, e - \bar{y}\} \\
\min\{[\bar{c}(x)]^-, e\}
\end{bmatrix}
\]

\begin{itemize}
  \item \textbf{FJ point}:
  \[
  \rho(x, y, \bar{y}, \mu) = 0, \; v(x) = 0, \; (y, \bar{y}, \mu) \neq 0
  \]
  \item \textbf{KKT point}:
  \[
  \rho(x, y, \bar{y}, \mu) = 0, \; v(x) = 0, \; \mu > 0
  \]
  \item \textbf{Infeasible stationary point}:
  \[
  \rho(x, y, \bar{y}, 0) = 0, \; v(x) > 0
  \]
\end{itemize}
Penalty function model and QP subproblem

(NLP):
\[
\begin{align*}
& \min_{x} \quad f(x) \\
& \text{s.t.} \quad c(x) = 0, \quad \bar{c}(x) \leq 0
\end{align*}
\]

(FP):
\[
\begin{align*}
& \min_{x} \quad v(x) := \left\| \begin{bmatrix} c(x) \\ \bar{c}(x) \end{bmatrix}^{+} \right\|_{1}
\end{align*}
\]

(PP):
\[
\begin{align*}
& \min_{x} \quad \phi(x, \mu) := \mu f(x) + v(x)
\end{align*}
\]

(FJ):
\[
\begin{align*}
& \mathcal{F}(x, y, \bar{y}, \mu) := \\
& \quad \mu f(x) + c(x)^{T} y + \bar{c}(x)^{T} \bar{y}
\end{align*}
\]

KKT residual:
\[
\rho(x, y, \bar{y}, \mu)
\]

Define a local model of \( \phi(\cdot, \mu) \) at \( x_{k} \):
\[
l_{k}(d, \mu) := \mu (f_{k} + g_{k}^{T} d) + \|c_{k} + J_{k}^{T} d\|_{1} + \|[\bar{c}_{k} + \bar{J}_{k}^{T} d]^{+}\|_{1}
\]

Reduction in this model yielded by a given \( d \):
\[
\Delta l_{k}(d, \mu) := \Delta l(0, \mu) - \Delta l(d, \mu)
\]

Subproblem of interest:
\[
\begin{align*}
& \min_{d} \quad -\Delta l_{k}(d, \mu) + \frac{1}{2} d^{T} H d \\
\end{align*}
\]

\( \Delta l_{k}(d, \mu) > 0 \) implies \( d \) is a direction of strict descent for \( \phi(\cdot, \mu) \) from \( x_{k} \)
Optimality conditions (for QP)

\[
\text{(NLP):} \quad \min_x f(x) \\
\text{s.t. } c(x) = 0, \quad \bar{c}(x) \leq 0
\]

\[
\text{(FP):} \quad \min_x v(x) := \left\| \begin{bmatrix} c(x) \\ \bar{c}(x) \end{bmatrix}^+ \right\|_1
\]

\[
\text{(PP):} \quad \min_x \phi(x, \mu) := \mu f(x) + v(x)
\]

\[
\text{(FJ):} \quad \mathcal{F}(x, y, \bar{y}, \mu) := \\
\mu g(x) + J(x)y + J(x)\bar{y}
\]

KKT residual:

\[
\rho(x, y, \bar{y}, \mu) := \left[ \begin{array}{c} \mu g(x) + J(x)y + J(x)\bar{y} \\
\min\{[c(x)]^+, e - y\} \\
\min\{[c(x)]^-, e + y\} \\
\min\{[\bar{c}(x)]^+, e - \bar{y}\} \\
\min\{[\bar{c}(x)]^-, e + \bar{y}\} \end{array} \right]
\]

KKT conditions for (QP) expressed with

\[
\rho_k(d, y, \bar{y}, \mu, H) := \left[ \begin{array}{c} \mu g_k + H d + J_k y + J_k \bar{y} \\
\min\{[c_k + J_k^T d]^+, e - y\} \\
\min\{[c_k + J_k^T d]^- , e + y\} \\
\min\{[\bar{c}_k + J_k^T \bar{d}]^+, e - \bar{y}\} \\
\min\{[\bar{c}_k + J_k^T \bar{d}]^- , e + \bar{y}\} \end{array} \right]
\]

Exact solution of (QP):

\[
\rho_k(d, y, \bar{y}, \mu, H) = 0
\]
Assumptions and well-posedness

Assumption

(1) The functions $f$, $c$, and $\bar{c}$ and their first derivatives are bounded and Lipschitz continuous in an open convex set containing $\{x_k\}$ and $\{x_k + d_k\}$.

(2) The QP solver can solve (QP) arbitrarily accurately for any $\mu \geq 0$.

Theorem (Well-posedness)

One of the following holds for our method, $iSQO$:

1. $iSQO$ terminates finitely with a KKT point or infeasible stationary point.

2. $iSQO$ generates an infinite sequence of iterates

   $\left( x_k, \begin{bmatrix} y'_k \\ \bar{y}'_k \end{bmatrix}, \begin{bmatrix} y''_k \\ \bar{y}''_k \end{bmatrix}, \mu_k \right)$ where $\begin{bmatrix} y'_k \\ y''_k \end{bmatrix} \in [-e, e]$, $\begin{bmatrix} \bar{y}'_k \\ \bar{y}''_k \end{bmatrix} \in [0, e]$, and $\mu_k > 0$. 
Global convergence

Theorem (Global convergence)

One of the following holds:

(a) $\mu_k = \mu$ for some $\mu > 0$ for all large $k$ and either every limit point of $\{x_k\}$ corresponds to a KKT point or is an infeasible stationary point;

(b) $\mu_k \to 0$ and every limit point of $\{x_k\}$ is an infeasible stationary point;

(c) $\mu_k \to 0$, all limit points of $\{x_k\}$ are feasible, and, with

$$K_\mu := \{k : \mu_{k+1} < \mu_k\},$$

every limit point of $\{x_k\}_{k \in K_\mu}$ corresponds to an FJ point where the MFCQ fails.

Corollary

If $\{x_k\}$ is bounded and every limit point of this sequence is a feasible point at which the MFCQ holds, then $\mu_k = \mu$ for some $\mu > 0$ for all large $k$ and every limit point of $\{x_k\}$ corresponds to a KKT point.
“Direct” scenario

(NLP):
\[
\begin{align*}
\min_x f(x) \\
s.t. \quad c(x) = 0, \quad \bar{c}(x) \leq 0
\end{align*}
\]

(FP):
\[
\min_x v(x) := \left\| \begin{bmatrix} c(x) \\ [\bar{c}(x)]^+ \end{bmatrix} \right\|_1
\]

(PP):
\[
\min_x \phi(x, \mu) := \mu f(x) + v(x)
\]

(FJ):
\[
\mathcal{F}(x, y, \bar{y}, \mu) := \\
\mu f(x) + c(x)^T y + \bar{c}(x)^T \bar{y}
\]

KKT residuals:
\[
\rho(x, y, \bar{y}, \mu) \\
\rho_k(d, y, \bar{y}, \mu, H)
\]

Local model of \(\phi\) at \(x_k\):
\[
l_k(d, \mu)
\]

Terminate the QP solver when the solution \(d_k, y_{k+1}, \bar{y}_{k+1}\) of (QP) with \(\mu = \mu_k\) satisfies
\[
\begin{align*}
\triangleright & y_{k+1} \in [-e, e], \quad \bar{y}_{k+1} \in [0, e] \\
\triangleright & \Delta l_k(d_k, \mu_k) \geq \theta \|d_k\|^2 > 0 \text{ for } \theta \in (0, 1) \\
\triangleright & \|\rho_k(d_k, y_{k+1}, \bar{y}_{k+1}, \mu_k, H_k)\| \leq \kappa \|\rho(x_k, y_k, \bar{y}_k, \mu_k)\|
\end{align*}
\]

If
\[
\begin{align*}
\triangleright & \Delta l_k(d_k, \mu_k) \geq \epsilon v_k \text{ for } \epsilon \in (0, 1) \\
\end{align*}
\]
then
\[
\begin{align*}
\triangleright & d_k \leftarrow d_k \text{ is the search direction} \\
\triangleright & \mu_{k+1} \leftarrow \mu_k
\end{align*}
\]
"Reference" scenario

(NLP):
\[
\begin{align*}
\min_x f(x) \\
\text{s.t. } c(x) = 0, \bar{c}(x) \leq 0
\end{align*}
\]

(FP):
\[
\begin{align*}
\min_x v(x) := \left\| \begin{bmatrix} c(x) \\ [\bar{c}(x)]^+ \end{bmatrix} \right\|_1
\end{align*}
\]

(PP):
\[
\begin{align*}
\min_x \phi(x, \mu) := \mu f(x) + v(x)
\end{align*}
\]

(FJ):
\[
\begin{align*}
\mathcal{F}(x, y, \bar{y}, \mu) := \\
\mu f(x) + c(x)^T y + \bar{c}(x)^T \bar{y}
\end{align*}
\]

KKT residuals:
\[
\begin{align*}
\rho(x, y, \bar{y}, \mu) \\
\rho_k(d, y, \bar{y}, \mu, H)
\end{align*}
\]

Local model of \( \phi \) at \( x_k \):
\[
\begin{align*}
l_k(d, \mu)
\end{align*}
\]

Terminate the QP solver when the solution \((d_k, y_{k+1}, \bar{y}_{k+1})\) of (QP) with \( \mu = 0 \) satisfies
\[
\begin{align*}
\text{▷ } y_{k+1} &\in [-e, e], \bar{y}_{k+1} \in [0, e] \\
\text{▷ } \Delta l_k(d_k, 0) &\geq \theta \|d_k\|^2 \text{ for } \theta \in (0, 1) \\
\text{▷ } \|\rho_k(d_k, y_{k+1}, \bar{y}_{k+1}, 0, H_k)\| &\leq \kappa \|\rho(x_k, y_k, \bar{y}_k, 0)\|
\end{align*}
\]

If
\[
\begin{align*}
\text{▷ } \Delta l_k(d_k, \mu_k) &\geq \epsilon \Delta l_k(d_k, 0) \text{ for } \epsilon \in (0, 1)
\end{align*}
\]

then
\[
\begin{align*}
\text{▷ } d_k &\leftarrow d_k \text{ is the search direction} \\
\text{▷ } \mu_{k+1} &\leftarrow \mu_k
\end{align*}
\]
"Combination" scenario

Choose the largest $\tau \in [0, 1]$ such that

$$d_k \leftarrow \tau d_k + (1 - \tau) d_k$$

yields

$$\Delta l_k(d_k, 0) \geq \epsilon \Delta l_k(d_k, 0)$$

then choose $\mu_{k+1} < \mu_k$ such that

$$\Delta l_k(d_k, \mu_{k+1}) \geq \beta \Delta l_k(d_k, 0)$$ for $\beta \in (0, 1)$
iSQO framework

repeat

(1) Check whether KKT point or infeasible stationary point has been obtained.
(2) Compute an inexact solution of (QP) with $\mu = \mu_k$.
   (a) If “Direct” scenario occurs, then go to step 4.

(3) Compute an inexact solution of (QP) with $\mu = 0$.
   (a) If “Reference” scenario occurs, then go to step 4.
   (b) If “Combination” scenario occurs, then go to step 4.

(4) Perform a backtracking line search to reduce $\phi(\cdot, \mu_k+1)$.
endrepeat
A few special cases make our actual algorithm slightly ;-) more complicated

- Landing on stationary points for $\phi(\cdot, \mu_k)$
  - We allow only a multiplier and/or penalty parameter update
- A tightened accuracy tolerance is needed in “combination” scenarios
  - We may require certain multipliers to be close to their bounds
  - (Think of identifying violated constraints)
- $H_k$ and/or $H_k$ may not be positive definite
  - We ask the QP solver to check the curvature along trial directions
  - (Dynamic inertia correction if trial curvature is too small/negative)

Actual algorithm involves six scenarios, but we have presented the “core” ideas
Implementation details: Experiments in paper

- Matlab implementation
- Test set involves 309 CUTEr problems with
  - at least one free variable
  - at least one general (non-bound) constraint
  - \#variables + \#constraints \leq 20,000 (because it’s Matlab!)
  - no failures due to BQPD
- BQPD for QP solves with indefinite Hessians; see (Fletcher, 2000)
- Simulated inexactness by perturbing QP solutions
- Termination conditions \((\epsilon_{tol} = 10^{-6} \text{ and } \epsilon_{\mu} = 10^{-8})\):
  \[
  \|\rho(x_k, y_k, \bar{y}_k, \mu_k)\|_{\infty} \leq \epsilon_{tol} \quad \text{and} \quad v_k \leq \epsilon_{tol};
  \]
  \[
  \|\rho(x_k, y_k, \bar{y}_k, 0)\|_{\infty} = 0 \quad \text{and} \quad v_k > 0; \quad \text{(Infeasible)}
  \]
  \[
  \|\rho(x_k, y_k, \bar{y}_k, 0)\|_{\infty} \leq \epsilon_{tol} \quad \text{and} \quad v_k > \epsilon_{tol} \quad \text{and} \quad \mu_k \leq \epsilon_{\mu} \quad \text{(Infeasible)}
  \]
- Investigate performance of inexact algorithm with \(\kappa = 0.01, 0.1, \text{ and } 0.5\)
Success statistics

Counts of termination messages for exact and three variants of inexact algorithm:

<table>
<thead>
<tr>
<th>Termination message</th>
<th>Exact</th>
<th>$\kappa = 0.01$</th>
<th>$\kappa = 0.1$</th>
<th>$\kappa = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal solution found</td>
<td>289</td>
<td>291</td>
<td>293</td>
<td>293</td>
</tr>
<tr>
<td>Infeasible stationary point found</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Iteration limit reached</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Termination statistics and reliability do not degrade with inexactness!
Inexactness levels

Observe “induced” relative residuals for QP solves:

\[ \kappa_I := \frac{\|\rho_k\|}{\|\rho\|} \]

For problem \( j \), we compute minimum (\( \kappa_I(j) \)) and mean (\( \bar{\kappa}_I(j) \)) values over run:

<table>
<thead>
<tr>
<th>min ( \kappa )</th>
<th>( \kappa_I \text{,min} )</th>
<th>( [0 - 10^{-8}] )</th>
<th>( [10^{-8} - 10^{-6}] )</th>
<th>( [10^{-6} - 10^{-4}] )</th>
<th>( [10^{-4} - 10^{-3}] )</th>
<th>( [10^{-3} - 0.01] )</th>
<th>( [0.01 - 0.1] )</th>
<th>( [0.1 - 0.5] )</th>
<th>( [0.5 - 1] )</th>
<th>( [1 - \infty] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>3.8e-03</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>277</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>3.0e-02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>254</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>9.0e-02</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>24</td>
<td>77</td>
<td>188</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mean</td>
<td>( \bar{\kappa}_I \text{,mean} )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>7.4e-03</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>270</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>7.0e-02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>280</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>3.5e-01</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>282</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Relative residuals generally need only be moderately smaller than parameter \( \kappa \)!
Iteration comparison

Considering the logarithmic outperforming factor

\[ r^j := - \log_2(\text{iter}^j_{\text{inexact}} / \text{iter}^j_{\text{exact}}), \]

we compare iteration counts of our inexact (\( \kappa = 0.01 \)) and exact algorithms:

Iteration counts do not degrade significantly with inexactness!
Implementation details: Experiments from last night

- Matlab implementation
- Test set involves 60 CUTEr problems that were all successfully solved with
  - augmented Lagrangian (AL) method
  - sequential quadratic optimization (SQO) method
  - iSQO method with AL method for QP subproblems
- AL method or CPLEX for QP solves: switch when
  \[ \| \rho(x_k, y_k, \bar{y}_k, \mu_k) \|_\infty \leq 10^{-2} \quad \text{and} \quad v_k \leq 10^{-2} \]
- Terminate solve for (NLP) when
  \[ \| \rho(x_k, y_k, \bar{y}_k, \mu_k) \|_\infty \leq 10^{-6} \quad \text{and} \quad v_k \leq 10^{-6} \]
Iteration comparison: AL vs. SQO

AL (left) with “cheap” iterations vs. SQO (right) with “expensive” iterations
Iteration comparison: SQO vs. iSQO

SQO (left) with “expensive” iterations vs. iSQO (right) with “cheaper” iterations
Iteration comparison: AL vs. iSQO(subproblems)

AL (left) with “cheap” iterations vs. iSQO (right) with few “expensive” iterations
Contributions:

- Developed, analyzed, and experimented with an inexact SQO method
- Allows generic inexactness in QP subproblem solves
- No specific QP solver required
- Global convergence guarantees established
- Numerical experiments suggest inexact algorithm is reliable
- Inexact solutions allowed without degradation of performance
“Exact” Algorithms:


“Inexact” Algorithms:


