A Generalized Sequential Quadratic Programming Framework for Large-Scale Optimization

Thesis Proposal

Denis Ridzal

Department of Computational and Applied Mathematics
Rice University, Houston, Texas
dridzal@caam.rice.edu

December 7, 2004

Thesis Committee: M. Heinkenschloss, W. Symes, T. Warburton, M. Pasquali
Outline

- Introduction:
  - SQP codes and their limitations,
  - algorithmic, software design, and application challenges,
  - proposed contributions.
- Composite-Step Trust-Region SQP Algorithms:
  - global convergence theory,
  - the augmented-system scheme.
- Inexactness in Linear System Solves.
- Overview of the Current Framework Implementation.
- Target Applications.
- Roadmap for Future Work.
Introduction

- SQP methods represent the **state of the art** for the solution of smooth nonlinear programming problems (NLPs), due to:
  - excellent convergence properties,
  - robustness proven in practice.
- Available SQP codes (NPSOL, SNOPT, KNITRO, etc.) are very effective for large-scale problems in $\mathbb{R}^n$ but otherwise limited in their applicability:
  - restricted to specific problem representations
  - closely intertwined with the underlying (direct) linear algebra
  - do not allow integration of application-specific subproblem solvers (multigrid, domain decomposition methods, FFT)
  ⇒ impossible to apply to many important optimization problems,
  ⇒ not suitable for very large-scale parallel computing environments

- **Goal:** Develop a general framework that makes SQP technology available to a larger class of problems.
Challenges

Algorithm Design

- Develop matrix-free code.
- Support iterative linear system solvers.
  - effects of iterative (i.e. inexact) solves on convergence
  - dynamically control stopping criteria for the solver
  - issue of preconditioning

Software Design

- Divorce optimization algorithm from the underlying linear algebra:
  - support abstract mathematical constructs (e.g. Hilbert spaces)
  - newer packages, for example, MOOCHO (Bartlet, SNL, 2002), address this issue through the use of an abstract vector library
- Divorce optimization algorithm from the problem representation:
  - extraction of a minimal problem-algorithm interface between a typical SQP algorithm and an arbitrary NLP (OO design)

Applications

- Parallel solution of an advection-dominated optimal control problem.
Proposed Contributions

Algorithm Design

- Analyze the issue of inexact linear system solves in a general composite-step trust-region SQP algorithm. *Full-space approach.*
- Suggest implementable ways of inexactness control.

Software Design

- Develop a research code that follows the general idea of problem-algorithm separation.
- Incorporate inexactness control mechanisms.

Applications

- Focus on the implementation of a KKT-system Neumann-Neumann domain-decomposition preconditioner for quadratic elliptic optimal control problems (Heinkenschloss & Nguyen, 2004).
- Consider its extensions to a class of advection-dominated problems.
Outline

► Introduction:
  ▶ SQP codes and their limitations,
  ▶ algorithmic, software design, and application challenges,
  ▶ overview of our contributions.

► Composite-Step Trust-Region SQP Algorithms:
  ▶ global convergence theory,
  ▶ the augmented-system scheme.

► Inexactness in Linear System Solves.

► Overview of the Current Framework Implementation.

► Target Applications.

► Roadmap for Future Work.
Problem Formulation

Solve NLP:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad c(x) = 0
\end{align*}
\]

where \( f : \mathcal{X} \to \mathbb{R} \) and \( c : \mathcal{X} \to \mathcal{Y} \), for some Hilbert spaces \( \mathcal{X} \) and \( \mathcal{Y} \), and \( f \) and \( c \) are twice continuously Fréchet differentiable.
Problem Formulation

Solve NLP:

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad c(x) = 0
\end{align*}
\]

where \( f : \mathcal{X} \to \mathbb{R} \) and \( c : \mathcal{X} \to \mathcal{Y} \), for some Hilbert spaces \( \mathcal{X} \) and \( \mathcal{Y} \), and \( f \) and \( c \) are twice continuously Fréchet differentiable.

▶ define Lagrangian functional \( \mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \):

\[
\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, c(x) \rangle_{\mathcal{Y}}
\]

▶ if regular point \( x^* \) is a local solution of the NLP, then there exists a \( \lambda^* \in \mathcal{Y} \) satisfying 1st order KKT necessary optimality conditions:

\[
\nabla_x f(x^*) + c_x(x^*)^* \lambda^* = 0\\
c(x^*) = 0
\]
Fundamentals of Trust-Region SQP

- Newton’s method applied to KKT system:

\[
\begin{pmatrix}
    \nabla_{xx} \mathcal{L}(x_k, \lambda_k) & c_x(x_k)^* \\
    c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
    s_k^x \\
    s_k^\lambda
\end{pmatrix}
= -\begin{pmatrix}
    \nabla_x f(x_k) + c_x(x_k)^* \lambda_k \\
    c(x_k)
\end{pmatrix}
\]

- Equivalent to KKT system for *quadratic programming problem* QP:

\[
\min \frac{1}{2} \langle \nabla_{xx} \mathcal{L}(x_k, \lambda_k) s_k^x, s_k^x \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s_k^x \rangle_X
\]

\[
\text{s.t. } c_x(x_k) s_k^x + c(x_k) = 0,
\]

under certain assumptions on \( \nabla_{xx} \mathcal{L}(x_k, \lambda_k) \).
Fundamentals of Trust-Region SQP

- Newton’s method applied to KKT system:

\[
\begin{pmatrix}
\nabla_{xx} \mathcal{L}(x_k, \lambda_k) & c_x(x_k) \\
- c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
s^x_k \\
- s^\lambda_k
\end{pmatrix} = - \begin{pmatrix}
\nabla_x f(x_k) + c_x(x_k)^* \lambda_k \\
c(x_k)
\end{pmatrix}
\]

- Equivalent to KKT system for *quadratic programming problem QP*:

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle \nabla_{xx} \mathcal{L}(x_k, \lambda_k) s^x_k, s^x_k \rangle_{\mathcal{X}} + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s^x_k \rangle_{\mathcal{X}} \\
\text{s.t.} & \quad c(x_k) s^x_k + c(x_k) = 0,
\end{align*}
\]

under certain assumptions on \( \nabla_{xx} \mathcal{L}(x_k, \lambda_k) \).

---

- **Trust-Region QP:**

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle H_k s^x_k, s^x_k \rangle_{\mathcal{X}} + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), s^x_k \rangle_{\mathcal{X}} \\
\text{s.t.} & \quad c(x_k) s^x_k + c(x_k) = 0 \\
& \quad \| s^x_k \|_{\mathcal{X}} \leq \Delta_k.
\end{align*}
\]
Fundamentals of Trust-Region SQP ... cont’d

Trust-region subproblem:

\[
\begin{align*}
\min & \quad \frac{1}{2} \langle H_k s^x_k, s^x_k \rangle \chi + \langle \nabla_x L(x_k, \lambda_k), s^x_k \rangle \chi \\
\text{s.t.} & \quad c_x(x_k)s^x_k + c(x_k) = 0 \\
& \quad \|s^x_k\| \chi \leq \Delta_k.
\end{align*}
\]

General iterative procedure:

- Compute Lagrange multipliers \(\lambda_k\).
- Compute step \(s^x_k\) by solving the trust-region subproblem.
- If \(s^x_k\) gives sufficient decrease for a chosen merit function, accept it. Otherwise, reject it, and reduce the trust region radius \(\Delta_k\).
Computation of Lagrange Multipliers

- A common practice is to approximately solve the stationarity equation:

\[ \nabla_x f(x_k) + c_x(x_k)^* \lambda = 0. \]

- General convergence requirement:
  Multipliers \( \lambda_k \) are uniformly bounded.

- Practical approach: Using the least-squares approach, we have

\[ \lambda_k = - (c_x(x_k)c_x(x_k)^*)^{-1} c_x(x_k) \nabla_x f(x_k), \]

which can be computed by solving the augmented system:

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
\lambda_k
\end{pmatrix}
=
\begin{pmatrix}
-\nabla_x f(x_k) \\
0
\end{pmatrix}.
\]
Solution of the Quadratic Subproblem

- **quasi-normal step** $n_k$: moves toward feasibility
- **tangential step** $t_k$: moves toward optimality while staying in the null space of the linearized constraints
Quasi-Normal Step

▶ Ask that $n_k$ approximately solve the problem:

$$\min \| c_x(x_k)n + c(x_k) \|_Y^2$$
\[\text{s.t.} \quad \|n\|_X \leq \zeta \Delta_k. \]

▶ General convergence requirements:
(1) For some $\kappa_1 > 0$ independent of $k$:

$$\|n_k\|_X \leq \kappa_1 \|c(x_k)\|_Y.$$

(2) For some $\kappa_2 > 0$, $\kappa_3 > 0$ independent of $k$:

$$\|c(x_k)\|_Y^2 - \|c_x(x_k)n_k + c(x_k)\|_Y^2 \geq \kappa_2 \|c(x_k)\|_Y \min \{\kappa_3 \|c(x_k)\|_Y, \Delta_k\}.$$
Quasi-Normal Step … cont’d

- **Practical approach:** Powell’s *dogleg method*. 

\[ c_x(x_k)s^x + c(x_k) = 0 \]
Quasi-Normal Step ... cont’d

- **Practical approach:** Powell’s *dogleg method*.
- **Newton (feasibility) step**

\[
n^N_k = -c(x_k)^* (c(x_k)c(x_k)^*)^{-1} c(x_k)
\]

can be computed by solving

\[
\begin{pmatrix}
  I & c(x_k)^* \\
  c(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
  z \\
  y
\end{pmatrix}
= \begin{pmatrix}
  0 \\
  c(x_k)
\end{pmatrix}
\]

and setting

\[
n^N_k = c(x_k)^* y.
\]
Tangential Step

▶ Ask that $t_k$ approximately solve a problem of this type:

$$
\begin{align*}
\min & \quad \frac{1}{2} \langle H_k(t + n_k), t + n_k \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), t + n_k \rangle_X \\
\text{s.t.} & \quad c_x(x_k)t = 0 \\
& \quad \|t + n_k\|_X \leq \Delta_k.
\end{align*}
$$

▶ Assume that there exists a bounded linear operator $W_k : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is a Hilbert space, such that $\text{Range}(W_k) = \text{Null}(c_x(x_k))$.

▶ Let $q(x_k, t) = \frac{1}{2} \langle H_k t, t \rangle_X + \langle \nabla_x \mathcal{L}(x_k, \lambda_k), t \rangle_X$.

▶ Let $t_k = W_k w_k$, where $w_k$ is the solution of

$$
\begin{align*}
\min & \quad q(x_k, n_k) + \langle W_k^* (H_k n_k + \nabla_x \mathcal{L}(x_k, \lambda_k)), w \rangle_Z \\
& \quad + \frac{1}{2} \langle W_k^* H_k W_k w, w \rangle_Z \\
\text{s.t.} & \quad \|n_k + W_k w\|_X \leq \Delta_k.
\end{align*}
$$
Tangential Step ... cont’d

- **General convergence requirement:**
  For $\kappa_4, \kappa_5, \kappa_6 > 0$ independent of $k$ we require:

  \[
  q(x_k, n_k) - q(x_k, n_k + t_k) \geq \\
  \kappa_4 \|W_k^* \nabla_x q(x_k, n_k)\| \min \{\kappa_5 \|W_k^* \nabla_x q(x_k, n_k)\| Z, \kappa_6 \Delta_k}\).
  \]

- **Practical approach:** *Projected Steihaug Conjugate Gradients.*
  Solve using a projected Steihaug CG in the full space. Use orthogonal projection operator $P_k : \mathcal{X} \rightarrow \mathcal{X}$,

  \[
P_k = I - c_x(x_k)^* (c_x(x_k) c_x(x_k)^*)^{-1} c_x(x_k).
  \]

- Compute $g = P_k r$, by solving the linear system

  \[
  \begin{pmatrix}
  I & c_x(x_k)^* \\
  c_x(x_k) & 0
  \end{pmatrix}
  \begin{pmatrix}
  g \\
  y
  \end{pmatrix}
  =
  \begin{pmatrix}
  r \\
  0
  \end{pmatrix}.
  \]
Acceptance of the Step (Rough Sketch)

- We use the augmented Lagrangian merit function
  \[
  \phi(x, \lambda; \rho) = \mathcal{L}(x, \lambda) + \rho \|c(x)\|_Y^2.
  \]

- Define actual reduction
  \[
  \text{ared}(s^x_k; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - \phi(x_k + s_k, \lambda_{k+1}; \rho_k),
  \]
  and predicted reduction
  \[
  \text{pred}(s^x_k; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - (q(x_k, s^x_k) + \langle \Delta \lambda_k, c_x(x_k) s^x_k + c(x_k) \rangle_Y + \rho_k \|c_x(x_k) s^x_k + c(x_k)\|_Y^2).
  \]

- Set \( \eta_1 > 0 \), independent of \( k \). If \( \frac{\text{ared}(s^x_k; \rho_k)}{\text{pred}(s^x_k; \rho_k)} > \eta_1 \), accept step.
General Convergence Result (Dennis, et al. 1994)

(1) $x_k \in \Omega$ and $x_k + s_k^x \in \Omega$, where $\Omega$ is an open subset of $\mathcal{X}$

(2) $f$ and $c$ are twice continuously Fréchet differentiable with Lipschitz continuous second derivatives

(3) $c_x(x)$ is surjective for all $x \in \Omega$

(4) The following function(al)s and operators are uniformly bounded over all $x \in \Omega$: $f(x)$, $\nabla_x f(x)$, $\nabla_{xx} f(x)$, $c(x)$, $c_x(x)$, $(c_x(x)c_x(x)^*)^{-1}$, and $c_{xx}(x)$.

(5) There exist uniformly bounded linear operators $W_k : \mathcal{Z} \rightarrow \mathcal{X}$, where $\mathcal{Z}$ is a Hilbert space, such that $\text{Range}(W_k) = \text{Null}(c_x(x_k))$.

(6) Operators $H_k$ are uniformly bounded.

(7) Lagrange multipliers $\lambda_k$ are uniformly bounded.

Theorem

Under the above problem assumptions, and convergence requirements on the quasi-normal and tangential steps, the sequence of iterates generated by the composite-step SQP algorithm satisfies

$$
\liminf_{k \to \infty} \left( \| W_k^* \nabla_x \mathcal{L}(x_k, \lambda_k) \|_{\mathcal{Z}} + \| c(x_k) \|_{\mathcal{Y}} \right) = 0.
$$
Outline

▶ Introduction:
  ▶ SQP codes and their limitations,
  ▶ algorithmic, software design, and application challenges,
  ▶ issues of inexactness and parallel preconditioning,
  ▶ overview of our contributions.

▶ Composite-Step Trust-Region SQP Algorithms:
  ▶ global convergence theory,
  ▶ the augmented-system scheme.

▶ Inexactness in Linear System Solves.
▶ Overview of the Current Framework Implementation.
▶ Target Applications.
▶ Roadmap for Future Work.
Inexactness: Existing Work

- Treatment of inexact Newton methods:
  - optimization: Dembo, Eisenstat, Steihaug, Dennis, Walker (1980s)
  - nonlinear equations: Eisenstat and Walker (1994)

- Connection with inexact SQP methods:
  - Dembo and Tulowitzki (1985) and Fontecilla (1985),
  - limited to local convergence analysis!

- Jäger and Sachs (1997) – line-search reduced-space SQP
  - first global convergence result
  - dependence on Lipschitz constants and derivative bounds

- Biros and Ghattas (2002) – quasi-Newton reduced-space SQP
  - dependence on derivative bounds

- Heinkenschloss and Vicente (2001) – reduced-space TRSQP
  - practical accuracy requirements that do not depend on Lipschitz constants or derivative bounds
  - limited to the reduced-space SQP approach
Inexactness: Goals

- Focus on full-space SQP methods, due to our choice of target applications and preconditioners.
- Devise accuracy requirements for iterative solvers.
  - must be easily implementable
  - must give optimal stopping criteria
- Short term:
  - revisit the classical composite-step trust-region SQP convergence results and focus on the algorithmic modules that are subject to inexactness
  - How can we enforce the existing convergence requirements?
- Long term:
  - suggest more effective approaches by relaxing the classical convergence results
Lagrange Multiplier Computation

- Example: Augmented-system approach.
- Uniform boundedness requirement easily imposed even in presence of inexactness.

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
\lambda_k
\end{pmatrix}
=
\begin{pmatrix}
-\nabla_x f(x_k) \\
0
\end{pmatrix},
\]
Lagrange Multiplier Computation

- Example: Augmented-system approach.
- Uniform boundedness requirement easily imposed even in presence of inexactness.

\[
\begin{pmatrix}
  I & c_x(x_k)^* \\
  c_x(x_k) & 0 \\
\end{pmatrix}
\begin{pmatrix}
  z \\
  \hat{\lambda}_k \\
\end{pmatrix}
= \begin{pmatrix}
  -\nabla_x f(x_k) + r^1_k \\
  r^2_k \\
\end{pmatrix},
\]

- \( r^1_k \in X, \|r^1_k\|_X \leq \epsilon \) and \( r^2_k \in Y, \|r^2_k\|_Y \leq \epsilon \) for some \( \epsilon > 0 \) independent of \( k \).
- We have:

\[
\lambda_k = - \left( c_x(x_k)c_x(x_k)^* \right)^{-1} c_x(x_k) \nabla_x f(x_k),
\]

\[
\hat{\lambda}_k = \lambda_k + \left( c_x(x_k)c_x(x_k)^* \right)^{-1} \left( c_x(x_k)r^1_k - r^2_k \right).
\]

- Uniform boundedness of \( \hat{\lambda}_k \) follows.
The classical convergence requirements can be satisfied in presence of inexactness:

- use explicitly inexact schemes, usually based on Krylov subspace methods (Golub and von Matt 1991, Sorensen 1994), which satisfy our convergence requirements by construction
- use modified relatives of “exact” algorithms (our dogleg approach is a good candidate)

Example: Consider the dogleg approach and inexactness arising in the Newton step computation.

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
-c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
z \\
y
\end{pmatrix}
= 
\begin{pmatrix}
r_k^1 \\
-c(x_k) + r_k^2
\end{pmatrix}
\]
Tangential Step Computation

► The inexact tangential step $\hat{t}_k$ usually solves:

$$\min \quad \frac{1}{2} \langle H_k(t + n_k), t + n_k \rangle_X + \langle \nabla_x L(x_k, \lambda_k) - r^1_k, t + n_k \rangle_X$$

s.t.

$$c_x(x_k)t = r^2_k$$

$$\|t + n_k\|_X \leq \Delta_k$$

► Related to the KKT optimality system:

$$\begin{pmatrix} H_k & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} -\nabla_x L(x_k, \lambda_k) + r^1_k \\ r^2_k \end{pmatrix}$$

► MAJOR COMPLICATION: $\hat{t}_k$ is no longer in the null space of the linearized constraints!!!
Tangential Step Computation \( \ldots \) cont’d

\[
\hat{t}_k = t_k + \delta_t^k + \delta_n^k \\
\hat{n}_k = n_k + \delta_n^k
\]

\[
c_x(x_k) s^x + c(x_k) = 0
\]

\[
c_x(x_k) t_k = 0
\]
Effective Tangential Step

- Effective tangential step $t_k + \delta^t_k$ needs to satisfy the previously stated convergence requirements.

- Let

$$\hat{q}(x_k, t) = \frac{1}{2} \langle H_k t, t \rangle \chi + \langle \nabla_x \mathcal{L}(x_k, \lambda_k) - r_k^1, t \rangle \chi.$$ 

- Then for $\hat{\kappa}_4, \hat{\kappa}_5, \hat{\kappa}_6 > 0$ independent of $k$ we require:

$$q(x_k, \hat{n}_k) - q(x_k, \hat{n}_k + t_k + \delta^t_k) \geq \hat{\kappa}_4 \| W_k^* \nabla_x q(x_k, \hat{n}_k) \| \zeta \min \left\{ \hat{\kappa}_5 \| W_k^* \nabla_x q(x_k, \hat{n}_k) \| \zeta, \hat{\kappa}_6 \Delta_k \right\}.$$
Effective Quasi-Normal Step

- Effective quasi-normal step $\hat{n}_k = n_k + \delta^n_k$ must satisfy:
  1. For some $\hat{\kappa}_1 > 0$ independent of $k$:
     \[ \|\hat{n}_k\|_X \leq \hat{\kappa}_1 \|c(x_k)\|_Y. \]
  2. For some $\hat{\kappa}_2 > 0, \hat{\kappa}_3 > 0$ independent of $k$:
     \[ \|c(x_k)\|_Y^2 - \|c_x(x_k)\hat{n}_k + c(x_k)\|_Y^2 \geq \hat{\kappa}_2 \|c(x_k)\|_Y \min \{\hat{\kappa}_3 \|c(x_k)\|_Y, \Delta_k\}. \]

- Must derive *easily implementable* and *flexible* conditions that guarantee (1) and (2).
Effective Quasi-Normal Step – Req. (1)

\[ \| \hat{n}_k \|_X \leq \hat{\kappa}_1 \| c(x_k) \|_Y \]

- Above condition implied by \( \| \delta^n_k \|_X \leq \hat{\kappa}_1 \| c(x_k) \|_Y \), with \( \hat{\kappa}_1 > 0 \).
- We further have \( c_x(x_k)\delta^n_k = c_x(x_k)\hat{t}_k = r^2_k \),
- and \( \delta^n_k = c_x(x_k)^* (c_x(x_k)c(x_k)^*)^{-1} r^2_k \).
- From boundedness properties and triangle inequality

\[ \| r^2_k \|_Y \leq \nu \| c(x_k) \|_Y \]

for some \( \nu > 0 \).
- Easily implementable. Not very flexible!
Effective Quasi-Normal Step – Req. (2)

\[ \| c \|^2_Y - \| c_x n_k + c \|^2_Y \geq \hat{\kappa}_2 \| c \|_Y \min \{ \hat{\kappa}_3 \| c \|_Y, \Delta_k \} \]

- Have

\[ \| c \|^2_Y - \| c_x n_k + c \|^2_Y = \| c \|^2_Y - \| c_x n_k + c \|^2_Y - [\| c_x \delta^n_k \|^2_Y + 2 \langle c_x n_k + c, c_x \delta^n_k \rangle] \]

- Note \[ \| c_x \delta^n_k \|^2_Y + 2 \langle c_x n_k + c, c_x \delta^n_k \rangle \leq \| r_k^2 \|^2_Y + 2 \| c_x n_k + c \|_Y \| r_k^2 \|_Y \]

- Impose for some \( 0 < \sigma < 1 \)

\[ \| r_k^2 \|^2_Y + 2 \| c_x n_k + c \|_Y \| r_k^2 \|_Y \leq \sigma (\| c \|^2_Y - \| c_x n_k + c \|^2_Y) \]

- Explicit requirement

\[ \| r_k^2 \|_Y \leq \sqrt{(1 - \sigma)\| c_x n_k + c \|^2_Y + \sigma \| c \|^2_Y - \| c_x n_k + c \|_Y} \]

- Easily implementable. Not very flexible!
Example

- Control of one-dimensional steady-state Burgers equation:

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} \int_0^1 (y(x) - y_d(x))^2 \, dx + \frac{\alpha}{2} \int_0^1 u(x)^2 \, dx \\
\text{s.t.} & \quad -\nu y_{xx}(x) + y_x(x)y(x) = f(x) + u(x), \quad x \in (0, 1) \\
& \quad y(0) = y(1) = 0.
\end{align*}
\]

where \( \alpha = 10^{-5}, \nu = 10^{-2}, y_d = \sin(2\pi x), \) and \( f = 0. \)

- Lagrange multipliers and quasi-normal are step computed exactly.

- Inexactness inside Steihaug CG iterations in the projections:

\[
\begin{pmatrix}
I & c_x(x_k)^* \\
c_x(x_k) & 0
\end{pmatrix}
\begin{pmatrix}
g_j \\
y
\end{pmatrix} - \begin{pmatrix}
r_j \\
0
\end{pmatrix} = \begin{pmatrix}
r_k^1 \\
r_k^2
\end{pmatrix},
\]

where we control \( \|r_k^1\|_X \leq tol_k \) and \( \|r_k^2\|_Y \leq tol_k. \)

- Set \( tol_k = \min \{\text{Req. (1)}, \text{Req. (2)}\} \rightarrow \text{QN direction only!} \)
Choose GMRES stopping tolerance

\[ tol_k = \min \left\{ \sqrt{(1 - \sigma)\|c x n_k + c\|_Y^2 + \sigma\|c\|_Y^2} - \|c x n_k + c\|_Y, \ \nu\|c\|_Y \right\} \]
Choose GMRES stopping tolerance

\[ tol_k = \min \left\{ \sqrt{(1 - \sigma) \| c x_n_k + c \|_Y^2 + \sigma \| c \|_Y^2} - \| c x_n_k + c \|_Y, \; \nu \| c \|_Y \right\} \]
Choose GMRES stopping tolerance

\[ tol_k = \min \left\{ \sqrt{(1 - \sigma)\|c x n_k + c\|^2_Y} + \sigma\|c\|^2_Y - \|c x n_k + c\|_Y, \ \nu\|c\|_Y \right\} \]

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>outer SQP iters</th>
<th>GMRES iters</th>
<th>tol in first iter</th>
<th>tol in last iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e-01</td>
<td>24</td>
<td>2570</td>
<td>9.0e-01</td>
<td>6.5e-08</td>
</tr>
<tr>
<td>1e-02</td>
<td>15</td>
<td>3782</td>
<td>2.9e-01</td>
<td>4.1e-07</td>
</tr>
<tr>
<td>1e-04</td>
<td>10</td>
<td>4220</td>
<td>2.9e-02</td>
<td>3.2e-07</td>
</tr>
<tr>
<td>1e-08</td>
<td>10</td>
<td>4940</td>
<td>2.9e-04</td>
<td>1.1e-09</td>
</tr>
<tr>
<td>tol = 1e-09</td>
<td>10</td>
<td>5918</td>
<td>1e-09</td>
<td>1e-09</td>
</tr>
</tbody>
</table>

Table: Computational benefits of inexactness.
Outline

▶ Introduction:
  ▶ SQP codes and their limitations,
  ▶ algorithmic, software design, and application challenges,
  ▶ issues of inexactness and parallel preconditioning,
  ▶ overview of our contributions.

▶ Composite-Step Trust-Region SQP Algorithms:
  ▶ global convergence theory,
  ▶ the augmented-system scheme.

▶ Inexactness in Linear System Solves.

▶ **Overview of the Current Framework Implementation.**

▶ Target Applications.

▶ Roadmap for Future Work.
Software Design Challenges Revisited

Separation of Optimization and Problem Information

- Identify main algorithmic components (algorithmic modules).
- Identify common traits (problem modules) in three classes of problems:
  - QR approach for the solution of small-scale problems in $\mathbb{R}^n$,
  - augmented-system approach,
  - basis-nonbasis approach (reduced-space SQP scheme).

$\Rightarrow$ Generic problem-algorithm interface

Separation of Optimization and Linear Algebra

- Use of the Standard Vector Library (Padula, Shannon, Symes).
Problem Modules of the Generic Interface

- objective information (evaluation and gradient)
- constraint information (evaluation and Jacobian)
- Hessian of Lagrangian information (direct representation – exact or approximate, or operator representation)
- Lagrange multiplier computation
- feasibility (Newton) step computation
- representation for the null space of the constraints
SQP Framework

SQP Algorithm

Generic Interface

Problem-Specific Interfaces and Implementations

- QR Approach (dense linear algebra)
- Augmented System Approach (sparse or iterative)
- Basis / Nonbasis Approach (sparse or iterative)
- ... etc.

User

User-Specified Implementation (LIGHT)

User-Specified Implementation (HEAVY)
Outline

▶ Introduction:
  ▶ SQP codes and their limitations,
  ▶ algorithmic, software design, and application challenges,
  ▶ issues of inexactness and parallel preconditioning,
  ▶ overview of our contributions.

▶ Composite-Step Trust-Region SQP Algorithms:
  ▶ global convergence theory,
  ▶ the augmented-system scheme.

▶ Inexactness in Linear System Solves.

▶ Overview of the Current Framework Implementation.

▶ Target Applications.

▶ Roadmap for Future Work.
Identification of Sources of Biochemical Attacks

Source inversion problem of determining an arbitrary source in a steady-state advection-diffusion equation, given a fixed velocity field and \( N_r \) pointwise measurements of concentrations:

\[
\min_{y,u} f(y,u) \equiv \frac{1}{2} \sum_{j=1}^{N_r} (y(x_j) - \hat{y}(x_j))^2 dx + \frac{\beta}{2} \mathcal{R}(u)
\]

subject to

\[-\epsilon \Delta y(x) + \mathbf{v}(x) \cdot \nabla y(x) = u(x), \quad x \in \Omega,
\]
\[y(x) = 0, \quad x \in \partial \Omega_d,
\]
\[\epsilon \frac{\partial}{\partial n} y(x) = g(x), \quad x \in \partial \Omega_n,
\]

where \( y \) and \( \hat{y} \) represent the predicted and observed concentrations, \( u \) is the source term driving the concentration in the system, \( \epsilon \) a diffusion coefficient, \( \mathbf{v} \) a fixed velocity field, \( g \) a given Neumann boundary function, and \( \mathcal{R}(u) \) a Tikhonov regularization term.
Problem Discretization

- Fully Discretized Problem (Discretize-Then-Optimize Approach):

\[
\begin{align*}
\min_{y,u} & \quad \frac{1}{2} y^T Q_h y + \tilde{q}^T y + \frac{\beta}{2} u^T R_h u \\
\text{s.t.} & \quad A_h y + B_h u - b = 0.
\end{align*}
\]

- KKT system:

\[
\begin{pmatrix}
Q_h & 0 & A_h^T \\
0 & \beta R_h & B_h^T \\
A_h & B_h & 0
\end{pmatrix}
\begin{pmatrix}
y \\
u \\
p
\end{pmatrix}
=
\begin{pmatrix}
-q \\
0 \\
b
\end{pmatrix}
\]

- Solved initially in Matlab (Umfpack). Discretization quantities computed using our implementation of the SUPG stabilized finite-element method for the advection-diffusion equation with linear elements and piecewise linear coefficients.
Problem Specs

- **Geometry:** A 2D model of an airport (meshed using Triangle).

![Geometry Diagram](image)

- **Sensors** are distributed uniformly at the nodes (simplification).

![Sensor Locations Diagram](image)

- **Velocity field:** Given by steady-state Stokes flow.

![Velocity Field Diagram](image)
Results

Sources

Simulated Attack – Exact Sources

Source Inversion – Computed Sources

Error in sources (log_{10})

Concentrations

Simulated Attack – Concentration Measurements

Source Inversion – Computed Concentrations

Error in concentration (log_{10})
Parallel Implementation

- C++ code, developed within the Trilinos solver framework (SNL).
- Utilizes its core parallel routine library *Epetra*, and the iterative solver library *AztecOO*. Subdomain solves done using Umfpack.
- Used *Metis* to partition the geometry:

![Partitioned geometry diagram]

- First attempt at simple domain-decomposition preconditioning in the context of advection-diffusion (no optimization) was successful.
Outline

- Introduction:
  - SQP codes and their limitations,
  - algorithmic, software design, and application challenges,
  - issues of inexactness and parallel preconditioning,
  - overview of our contributions.

- Composite-Step Trust-Region SQP Algorithms:
  - global convergence theory,
  - the augmented-system scheme.

- Inexactness in Linear System Solves.

- Overview of the Current Framework Implementation.

- Target Applications.

- Roadmap for Future Work.
Future Work

Algorithm Design

- The analysis of inexactness for the full-space augmented-system approach needs to be completed:
  - better criterion for the effects of tangential step inexactness on the effective quasi-normal step
  - have not considered the effects of the same source of inexactness on the effective tangential step
- Our analysis must address the issue of preconditioned solves.
- Approaches other than those based on ad hoc modifications of classical SQP convergence theories will be necessary.
  - the flexibility in the balance between feasibility, optimality, and sufficient decrease must be better utilized
- Consider globalization schemes other than merit function based trust-region approaches:
  - line search methods
  - filter methods
Future Work ... cont’d

Software Design

▶ Implement the augmented system and basis-nonbasis interfaces.
▶ Once solid theoretical results on inexactness are available, they must be included → possible redesign.
▶ Test our framework design on target applications.

Applications

▶ Finish work on the implementation of the Neumann-Neumann DD preconditioner for elliptic optimal control problems. Extend it to the advection-dominated case.
▶ Expand advection-diffusion application to a Navier-Stokes problem. Possible applications:
  ▶ active control of the HVAC flow for the airport model, with the goal to minimize the spread of the contaminant
  ▶ HVAC temperature control