# Practical Optimization Algorithms and Applications

## Chapter XIII: Sequential Quadratic Programming

Lingfeng NIU

Research Center on Fictitious Economy & Data Science,
University of Chinese Academy of Sciences

niulf@ucas.ac.cn

# Outline

One of the most effective methods for nonlinearly constrained optimization generates steps by solving quadratic subproblems. This sequential quadratic programming (SQP) approach can be used both in line search and trust-region frameworks, and it is appropriate for small or large problems.

All the methods consider in this chapter are active-set methods; a more descriptive title for this chapter would perhaps be "Active-Set Methods for Nonlinear Programming".

# Introduction

There are two types of active-set SQP methods.

- In the IQP approach, a general inequality-constrained quadratic program is solved at each iteration, with the twin goals of computing a step and generating an estimate of the optimal active set.
- EQP methods decouple these computations. They first compute an estimate of the optimal active set, then solve an equality-constrained quadratic program to find the step.

In this chapter we study both IQP and EQP methods.

# Introduction

Our development of SQP methods will be done in two stages.

- First, we consider local methods that motivates the SQP approach and that allows us to introduce the step computation and Hessian approximation techniques in a simple setting.
- Second, we consider practical line search and trust-region methods that achieve convergence from remote starting points.

Throughout the chapter we give consideration to the algorithmic demands of solving large problems.

# Outline

# Local SQP Method

We begin by considering the equality-constrained problem

$$\begin{aligned} \min \quad & f(x) && (1a) \\ s.t. \quad & c(x) = 0. && (1b) \end{aligned}$$

where $f : \Re^n \to \Re$ and $c : \Re^n \to \Re^m$ are smooth functions.

The essential idea of SQP is to model (1) at the current iterate $x_k$ by a quadratic programming subproblem and to use the minimizer of this subproblem to define a new iterate $x_{k+1}$.

The challenge is to design the quadratic subproblem so that it yields a good step for the underlying constrained optimization problem and so that the overall SQP algorithm has good convergence properties and good practical performance.

# Local SQP Method

Perhaps the simplest derivation of SQP methods views them as an application of Newton's method to the KKT optimality conditions for (1).

The Lagrangian function for problem (1) is $\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x)$. By specializing the KKT conditions to the equality-constrained case, we obtain a system of $n + m$ equations in the $n + m$ unknowns $x$ and $\lambda$:

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0. \tag{2}$$

Here $A(x)$ is the Jacobian matrix of the constraints, that is,

$$A(x)^T = [\nabla c_1(x), \nabla c_2(x), \cdots, \nabla c_m(x)],$$

and $c_i(x)$ is the $i$th component of the vector $c(x)$.

Any solution $(x^*, \lambda^*)$ of the equality-constrained problem (1) for which $A(x^*)$ has full rank satisfies (2). One approach that suggests itself is to solve the nonlinear equations (2) by using Newton's method.

The Jacobian of (2) with respect to $x$ and $\lambda$ is given by

$$F'(x,\lambda) = \begin{bmatrix} \nabla^2_{xx}\mathcal{L}(x,\lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix}. \tag{3}$$

The Newton step from the iterate $(x_k, \lambda_k)$ is given by

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix}, \tag{4}$$

where $p_k$ and $p_\lambda$ solve the Newton-KKT system

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix} \tag{5}$$

This iteration, which is sometimes called the *Newton-Lagrange method*, is well-defined when the KKT matrix is nonsingular.

Nonsingularity is a consequence of the following conditions:
**Assumption 1:**

- The constraint Jacobian $A(x)$ has full row rank;
- The matrix $\nabla^2_{xx}\mathcal{L}(x,\lambda)$ is positive definite on the tangent space of the constraints, i.e., $d^T\nabla^2_{xx}\mathcal{L}(x,\lambda)d > 0$ for all $d \neq 0$ such that $A(x)d = 0$.

The first assumption is the linear independence constraint qualification. The second condition holds whenever $(x,\lambda)$ is close to the optimum $(x^*,\lambda^*)$ and the second-order sufficient condition is satisfied at the solution.

The Newton iteration (4), (5) can be shown to be quadratically convergent under these assumptions and constitutes an excellent algorithm for solving equality-constrained problems, provided that the starting point is close enough to $x^*$.

# Local SQP Method

There is an alternative way to view the iteration (4), (5). Suppose that at the iterate $(x_k, \lambda_k)$ we define the quadratic program

$$\begin{aligned} \min_p \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p && \text{(6a)} \\ \text{s.t.} \quad & A_k p + c_k = 0. && \text{(6b)} \end{aligned}$$

If Assumptions 1 hold, this problem has a unique solution $(p_k, l_k)$ that satisfies

$$\begin{aligned} \nabla_{xx}^2 \mathcal{L}_k p_k + \nabla f_k - A_k^T l_k &= 0, && \text{(7a)} \\ A_k p_k + c_k &= 0. && \text{(7b)} \end{aligned}$$

The vectors $p_k$ and $l_k$ can be identified with the solution of the Newton equations (5). If we subtract $A_k^T \lambda_k$ from both sides of the first equation in (5), we obtain

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{8}$$

Hence, by nonsingularity of the coefficient matrix, we have that $\lambda_{k+1} = l_k$ and $p_k$ solves (6) and (5).

# Local SQP Method

The new iterate $(x_{k+1}, \lambda_{k+1})$ can therefore be defined

- either as the solution of the quadratic program (6);
- or as the iterate generated by Newton's method (4) and (5) applied to the optimality conditions of the problem.

We refer to this interesting relationship as the *equivalence between SQP and Newton's method*. Both view points are useful.

- The Newton point of view facilitates the analysis.
- The SQP framework enables us to derive practical algorithms and to extend the technique to the inequality-constrained case.

# Local SQP Method

**Algorithm 1**: Local SQP Algorithm for solving (1)
Choose an initial pair $(x_0, \lambda_0)$; set $k \leftarrow 0$;
**repeat** until a convergence test is satisfied
    Evaluate $f_k$, $\nabla f_k$, $\nabla^2_{xx}\mathcal{L}_k$, $c_k$ and $A_k$;
    Solve (6) to obtain $p_k$ and $l_k$;
    Set $x_{k+1} \leftarrow x_k + p_k$ and $\lambda_{k+1} \leftarrow l_k$;
**end(repeat)**

# Local SQP Method

We should note in passing that, in the objective (6a) of the quadratic program, we could replace the linear term $\nabla f_k^T p$ by $\nabla_x \mathcal{L}(x_k, \lambda_k)^T p$, since the constraint (6b) makes the two choices equivalent.

In this case, (6a) is a quadratic approximation of the Lagrangian function. This fact provides a motivation for our choices of the quadratic model (6):

- We first replace the nonlinear program (1) by the problem of minimizing the Lagrangian subject to the equality constraints (6b),
- then make a quadratic approximation of the Lagrangian and a linear approximation of the constraints to obtain (6).

There are two ways of designing SQP methods for solving the general nonlinear programming problem

$$
\begin{aligned}
\min \quad & f(x) && (9a)\\
s.t. \quad & c_i(x) = 0, \quad i \in \mathcal{E}, && (9b)\\
& c_i(x) \geq 0, \quad i \in \mathcal{I}. && (9c)
\end{aligned}
$$

The first approach solves at every iteration the quadratic subprogram

$$
\begin{aligned}
\min \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p && (10a)\\
s.t. \quad & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, && (10b)\\
& \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}, && (10c)
\end{aligned}
$$

The new iterate is given by $x_k + p_k, \lambda_{k+1}$ where $p_k$ and $\lambda_{k+1}$ are the solution and the corresponding Lagrangian multiplier of (10). A local SQP method for (9) follows from Algorithm 1 with the modification that the step is computed from (10). This approach is referred to as the *IQP (inequality-constrained QP)* approach, and has proved to be quite successful in practice.

# Inequality Constraints - IQP

In this IQP approach the set of active constraints $\mathcal{A}_k$ at the solution of (10) continues our guess of the active set at the solution of the nonlinear program.

If the SQP method is able to correctly identify this optimal active set (and not change its guess at a subsequent iteration) then it will act like a Newton method for equality-constrained optimization and will converges rapidly. The following result gives conditions under which this desirable behavior takes place.

**Theorem**

*Suppose that $x^*$ is a solution point of (9) at which the KKT conditions are satisfied for some $\lambda^*$. Suppose, too, that the linear independence constraint qualification, the strict complementarity condition, and the second-order sufficient conditions hold at $(x^*, \lambda^*)$. Then if $(x_k, \lambda_k)$ is sufficiently close to $(x^*, \lambda^*)$, there is a local solution of the subproblem (10) whose active set $\mathcal{A}_k$ is the same as the active set $\mathcal{A}(x^*)$ of the nonlinear program (9) at $x^*$.*

# Inequality Constraints - IQP

The main drawback of IQP method is the expense of solving the general QP (10), which can be high when the problem is large.

As the iterates of the SQP method converge to the solution, however, solving the quadratic subproblem becomes very economical if we carry information from the previous iteration to make a good guess of the optimal solution of the current subproblem, which is called *warm-start strategy*.

We can achieve significant savings in the solution of the quadratic subproblem by warm-start procedures. For example, we can initialize the working set for each QP subproblem to be the final active set from the previous SQP iteration.

# Inequality Constraints - EQP

The second approach selects a subset of constraints at each iteration to be the so called *working set*, and solves only equality-constrained subproblems of the form (6), where the constraints in the working sets are imposed as equalities and all other constraints are ignored.

The working set is updated at every iteration by rules based on the Lagrange multiplier estimates, or by solving an auxiliary subproblem.

This EQP approach has the advantage that the equality-constrained quadratic subproblems are less expensive to solve than (10) and require less sophisticated software.

# Inequality Constraints - EQP

- An example of an EQP method is the gradient projection method. In this method, the working set is determined by minimizing the quadratic model along the path obtained by projecting the steepest descent direction onto the feasible region.
- Another variant of the EQP method makes use of the method of *successive linear programming*. This approach constructs a linear program by omitting the quadratic term $p^T \nabla_{xx}^T \mathcal{L}_k p$ from (6a) and adding a trust-region constraint $\|p\|_\infty \leq \delta_k$ to the subproblem. The active set of the resulting linear programming subproblem is taken to be the working set for the current iteration. The method then fixes the constraints in the working set and solves an equality-constrained quadratic program (with the term $p^T \nabla_{xx}^T \mathcal{L}_k p$ reinserted) to obtain the SQP step.

# Outline

# Enforcing Convergence

To be practical, an SQP method must be able to converge from remote starting points and on nonconvex problems. We now outline how the local SQP strategy can be adapted to meet these goals.

# Enforcing Convergence - Line Search Method

Line search method can be used to globalize *SQP* methods.

- If $\nabla^2_{xx}\mathcal{L}_k$ is positive definite on the tangent space of the active constraints, the quadratic subproblem (6) has a unique solution.
- When $\nabla^2_{xx}\mathcal{L}_k$ does not have this property, line search methods either replace it by a positive definite approximation $B_k$ or modify $\nabla^2_{xx}\mathcal{L}_k$ directly during the process of matrix factorization. In all these cases, the subproblem will be well-defined, but the modification may introduce unwanted distortions in the model.

# Enforcing Convergence - Trust Region Method

Trust-region SQP methods add a constraint to the subproblem, limiting the step to a region where the model is considered to be reliable. Because they impose a trust region bound on the step, they are able to use Hessians $\nabla^2_{xx}\mathcal{L}_k$ that fail to satisfy the convexity properties.

Complications may arise, however, because the inclusion of the trust region may cause the subproblem to become infeasible. At some iterations, it is necessary to relax the constraints, which complicates the algorithm and increases its computational cost.

Due to this tradeoffs, neither of the two SQP approaches - line search or trust region - is currently regarded as clearly superior to the other.

# Enforcing Convergence - Step-Acceptance Mechanisms

The techniques used to accept or reject steps also impacts the efficiency of SQP methods.

- In unconstrained optimization, the merit function is simply the objective $f$, and it remains fixed throughout the minimization procedure.
- For constrained problems, we use devices such as a merit function or a filter. The parameters or entries used in these devices must be updated in a way that is compatible with the step produced by the SQP method.

# Merit Functions

SQP methods often use a merit function to decide whether a trial step should be accepted.

- In the line search methods, the merit function controls the size of the steps;
- In the trust region methods, it determines whether a step is accepted or rejected and whether the trust-region radius needs to be adjusted.

It plays the role of the objective function in unconstrained optimization, since we insist that each step provide a sufficient reduction in it. A variety of merit functions have been used in conjunction with SQP methods, including nonsmooth penalty functions and augmented Lagrangians.

For the purpose of step computation and evaluation of a merit function, inequality constraints $c(x) \geq 0$ are often converted to the form

$$\bar{c}(x, s) = c(x) - s = 0,$$

where $s \geq 0$ is a vector of slacks. (The condition $s \geq 0$ is typically not monitored by the merit function.) Therefore, in the discussion follows that all constraints are in the form of equalities, and we focus our attention on the equality-constrained problem (1).

The $\ell_1$ merit function for (1) takes the form

$$\phi_1(x; \mu) = f(x) + \mu \|c(x)\|_1. \tag{11}$$

# Merit Functions

In a line search method, a step $\alpha_k p_k$ will be accepted if the following sufficient decrease condition holds:

$$\phi_1(x_k + \alpha_k; \mu_k) \leq \phi_1(x_k; \mu_k) + \eta \alpha_k D(\phi_1(x_k; \mu); p_k), \qquad \eta \in (0, 1), \tag{12}$$

where $D(\phi_1(x_k; \mu); p_k)$ denotes the directional derivatives of $\phi_1$ in the direction $p_k$. This requirement is analogous to the Armijo condition for unconstrained optimization provided that $p_k$ is a descent direction, that is, $D(\phi_1(x_k; \mu); p_k) < 0$.

# Merit Functions

This descent condition holds if the penalty parameter $\mu$ is chosen sufficiently large, as we show in the following result.

**Theorem**

*Let $p_k$ and $\lambda_{k+1}$ be generated by the SQP iteration (5). Then the directional derivative of $\phi_1$ in the direction $p_k$ satisfies*

$$D(\phi_1(x_k;\mu);p_k) = \nabla f_k^T p_k - \mu\|c_k\|_1. \tag{13}$$

*Moreover, we have that*

$$D(\phi_1(x_k;\mu);p_k) \le -p_k^T \nabla_{xx}^2 \mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1. \tag{14}$$

It follows from (14) that $p_k$ will be a descent direction for $\phi_1$ if $p_k \neq 0$, $\nabla_{xx}^2 \mathcal{L}_k$ is positive defined and

$$\mu > \|\lambda_{k+1}\|_\infty. \tag{15}$$

# Filters

Filter techniques are step acceptance mechanisms based on ideas from multiobjective optimization.

Nonlinear programming has two goals: minimization of the objective function and the satisfaction of the constraints. If we define a measure of infeasibility as

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} |c_i(x)|^-,$$

we can write these two goals as

$$\min_x f(x) \text{ and } \min_x h(x).$$

Unlike merit functions, which combine both problems into a single minimization problem, filter methods keep the two goals separate.

# Filters

Filter methods accept a trial step $x^+$ as a new iterate if the pair $(f(x^+), h(x^+))$ is not *dominated* by a previous pair $(f_l, h_l) = (f(x_l), h(x_l))$ generated by the algorithm. These concepts are defined as follows.

## Definition

(a) A pair $(f_k, h_k)$ is said to dominate another pair $(f_l, h_l)$ if both $f_k \leq f_l$ and $h_k \leq h_l$.
(b) A filter is a list of pairs $(f_l, h_l)$ such that no pair dominates any other.
(c) An iterate $x_k$ is said to be acceptable to the filter if $(f_k, h_k)$ is not dominated by any other pair in the filter.

When an iterate $x_k$ is acceptable to the filter, we (normally) add $(f_k, h_k)$ to the filter and remove any pairs that are dominated by $(f_k, h_k)$.

# Outline

# The Inconsistence of Linearized Constraints

A common difficulty in line search SQP methods is that the linearizations (10b), (10c) of the nonlinear constraints may give rise to an infeasible subproblem.

Consider, for example, the case where $n = 1$ and where the constraints are $x \leq 1$ and $x^2 \geq 4$. When we linearize these constraints at $x_k = 1$, we obtain the inequalities

$$-p \geq 0 \text{ and } 2p - 3 \geq 0,$$

which are inconsistent.

To overcome this difficulty, we can define a relaxation of the SQP subproblem that is guaranteed to be feasible.

# Handling Inconsistent Linearizations

For example, the SNOPT program for large-scale optimization solves the following auxiliary problem to deal with inconsistencies of the linearized constraints, which is called the *elastic mode*

$$
\begin{aligned}
\min \quad & f(x) + \mu \textstyle\sum_{i\in\mathcal{E}} (v_i + w_i) + \mu \textstyle\sum_{i\in\mathcal{I}} t_i && \text{(16a)}\\
\text{s.t.} \quad & c_i(x) = v_i - w_i, \qquad i \in \mathcal{E} && \text{(16b)}\\
& c_i(x) \geq -t_i, \qquad i \in \mathcal{I} && \text{(16c)}\\
& v, w, t \geq 0, && \text{(16d)}
\end{aligned}
$$

where $\mu$ is a nonnegative penalty parameter.

- If the nonlinear problem (9) has a feasible solution and $\mu$ is sufficiently large, the solutions to (16) and (9) are identical($x^*, v_i^* = w_i^* = 0, i \in \mathcal{E}$ and $t_i^* = 0, i \in \mathcal{I}$).
- If, on the other hand, there is no feasible solution to the nonlinear problem and $\mu$ is large enough, then the auxiliary problem (16) usually determines a stationary point of the infeasibility measure.

# The Maratos Effect

Although the merit function is needed to induce global convergence, we do not want it to interfere with "good" steps-those that make progress toward a solution.

Some algorithms based on merit functions or filters may fail to converge rapidly because they reject steps that make good progress towards a solution. This undesirable phenomenon is often called the *Maratos effect*, because it was first observed by Maratos.

# The Maratos Effect

Consider the problem

$$\min f(x_1, x_2) = 2(x_1^2 + x_2^2 - 1) - x_1, \text{ s.t. } x_1^2 + x_2^2 - 1 = 0. \tag{17}$$

It is easy to verify that the optimal solution is $x^* = (1, 0)^T$, that the corresponding Lagrange multiplier is $\lambda^* = \frac{3}{2}$, and that $\nabla^2_{xx}\mathcal{L}(x^*, \lambda^*) = I$.

Let us consider an iterate $x_k$ of the form $x_k = (\cos\theta, \sin\theta)^T$, which is feasible for any value of $\theta$. We now generate a search direction $p_k$ by solving the subproblem (6) with $\nabla^2_{xx}\mathcal{L}_k = I$. Since

$$f(x_k) = -\cos\theta,\ \nabla f(x) = \begin{pmatrix} 4\cos\theta - 1 \\ 4\sin\theta \end{pmatrix},\ A(x_k)^T = \begin{pmatrix} 2\cos\theta \\ 2\sin\theta \end{pmatrix}, \tag{18}$$

the quadratic subproblem (6) takes the form

$$\begin{aligned} \min \quad & -\cos\theta + (4\cos\theta - 1)p_1 + 4\sin\theta p_2 + \tfrac{1}{2}p_1^2 + \tfrac{1}{2}p_2^2 && (19) \\ s.t. \quad & p_2 = -\cot\theta p_1. && (20) \end{aligned}$$

# The Maratos Effect

By solving this subproblem, we obtain

$$p_k = \begin{pmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{pmatrix}, \tag{21}$$

which yields a new trial point

$$x_k + p_k = \begin{pmatrix} \cos\theta + \sin^2\theta \\ \sin\theta(1-\cos\theta) \end{pmatrix}. \tag{22}$$

If $\sin\theta \neq 0$, we have that

$$\|x_k + p_k - x^*\|_2 = 2\sin^2(\theta/2), \qquad \|x_k - x^*\|_2 = 2|\sin(\theta/2)|,$$

and therefore

$$\frac{\|x_k + p_k - x^*\|_2}{\|x_k - x^*\|_2^2} = \frac{1}{2}.$$

Hence, this step approaches the solution at a rate consistent with Q-quadratic convergence.

# The Maratos Effect

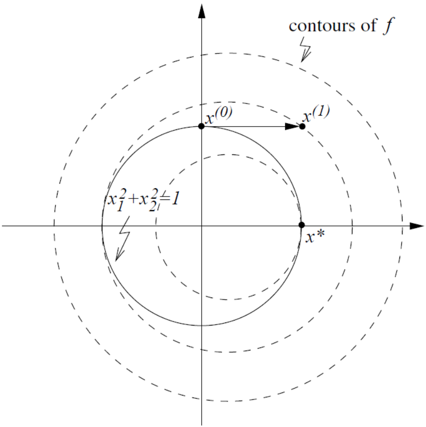
Note, however, that

$$\begin{aligned} f(x_k + p_k) &= \sin^2\theta - \cos\theta > -\cos\theta = f(x_k), \\ c(x_k + p_k) &= sin^2\theta > c(x_k) = 0, \end{aligned}$$

so that both the objective function value and constraint violation increase over this step. Then any merit function of the form

$$\phi_1(x;\mu) = f(x) + \mu h(c(x)),$$

(where $h(\cdot)$ is a nonnegative function satisfying $h(0) = 0$) will reject the step (21), so that any algorithm based on a merit function of this type will suffer from the Maratos effect.




$\theta = \frac{\pi}{2}$ and the SQP method moves from $x(0) = (1, 0)$ to $x(1) = (1, 1)$.

# The Maratos Effect

If no measures are taken, the Maratos effect can dramatically slow down SQP methods. Not only does it interfere with good steps away from the solution, but it can also prevent superlinear convergence from taking place. Techniques for avoiding the Maratos effect include the following.

- Use a merit function that does not suffer from the Maratos effect.
- Use a second-order correction in which we add to $p_k$ a step $\hat{p}_k$, which is computed at $c(x_k + p_k)$ and which decrease the constraint violation.
- Use a nonmonotone strategy: allow the merit function $\phi$ to increase on certain iterations.

## Definition

A merit function $\phi(x;\mu)$ is exact if there is a positive scalar $\mu^*$ such that for any $\mu > \mu^*$, any local solution of the nonlinear programming problem (9) is a local minimizer of $\phi(x;\mu)$.

We can show that the merit function of the following type

$$\phi(x;\mu) = f(x) + \mu\|c_{\mathcal{E}}(x)\| + \mu\|[c_{\mathcal{I}}(x)]^-\|, \tag{23}$$

*must* be nonsmooth to be exact.

# Exact Nonsmooth Penalty Functions

For simplicity, we restrict our attention to the case where only equality constraints are present, and assemble the constraint functions $c_i(x), i \in \mathcal{E}$, into a vector $c(x)$. Consider a merit function of the form

$$\phi(x; \mu) = f(x) + \mu h(c(x)),$$

where $h : \Re^m \to \Re$ is a function satisfying the properties $h(y) \geq 0$ for all $y \in \Re^m$ and $h(0) = 0$. Suppose for contradiction that $h$ is differentiable. Since $h$ has a minimizer at zero, we have $\nabla h(0) = 0$. Now, if $x^*$ is a local solution of the problem (1), we have $c(x^*) = 0$ and therefore $\nabla(c(x^*)) = 0$. Hence, since $x^*$ is a local minimizer of $\phi(x; \mu)$, we have that

$$0 = \nabla\phi(x^*; \mu) = \nabla f(x^*) + \mu\nabla c(x^*)\nabla h(c_1(x^*)) = \nabla f(x^*).$$

However, it is not generally true that the gradient of $f$ vanishes at the solution of a constrained optimization problem, so our original assumption that $h$ is differentiable must be incorrect, and our claim is proved.

# Fletcher's Augmented Lagrangian

Some merit functions are both smooth and exact. To ensure that both properties hold, we must include additional terms in the merit function. For equality-constrained problem (1), Fletcher's augmented Lagrangian is given by

$$\phi_F(x;\mu) = f(x) - \lambda(x)^T c(x) + \frac{1}{2}\mu \sum_{i\in\mathcal{E}} c_i(x)^2, \tag{24}$$

where $\mu > 0$ is the penalty parameter and

$$\lambda(x) = [A(x)A(x)^T]^{-1} A(x)\nabla f(x). \tag{25}$$

(Here $A(x)$ denotes the Jacobian of $c(x)$.) Although this merit function has some interesting theoretical properties, it has practical limitations, including the expense of solving for $\lambda(x)$ in (25).

# Second-Order Correction

By adding a correction term that provides further decrease in the constraints, the SQP iteration overcomes the difficulties associated with the Maratos effect.

Suppose that the SQP method has computed a step $p_k$ from (10). If this step yields an increase in the merit function $\phi_1$, a possible cause is that our linear approximation to the constraints are not sufficiently accurate.

To overcome this deficiency, we could re-solve (10) with the linear terms $c_i(x_k) + \nabla c_i(x_k)^T p$ replaced by the quadratic approximations

$$c_i(x_k) + \nabla c_i(x_k)^T p + \frac{1}{2} p^T \nabla^2 c_i(x_k) p. \tag{26}$$

However, even if the Hessian of the constraints are individually available, the resulting quadratically constraints subproblem may be too difficult to solve.

# Second-Order Correction

Instead we evaluate the constraint values at the new point $x_k + p_k$ and makes use of the following approximations. By Taylor's theorem, we have

$$c_i(x_k + p_k) \approx c_i(x_k) + \nabla c_i(x_k)^T p_k + \frac{1}{2} p_k^T \nabla^2 c_i(x_k) p_k. \tag{27}$$

Assuming that the (still unknown) second-order step $p$ will not be too different from $p_k$. we can approximate the last term in (26) as follows:

$$\frac{1}{2} p^T \nabla^2 c_i(x_k) p = \frac{1}{2} p_k^T \nabla^2 c_i(x_k) p_k. \tag{28}$$

By making this substitution in (26) and using (27), we obtain the second-order correction subproblem

$$\begin{array}{ll} \min & \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ \text{s.t.} & \nabla c_i(x_k)^T p + d_i = 0 \qquad i \in \mathcal{E}, \\ & \nabla c_i(x_k)^T p + d_i \geq 0 \qquad i \in \mathcal{I}. \end{array}$$

where

$$d_i = c_i(x_k + p_k) - \nabla c_i(x_k)^T p_k, \qquad i \in \mathcal{E} \cup \mathcal{I}.$$

# Second-Order Correction

The second-order correction steps requires evaluation of the constraints $c_i(x_k + p_k)$ for $i \in \mathcal{E} \cup \mathcal{I}$, and therefore it is preferable not to apply it every time the merit function increase. One strategy is to use it only if the increase in the merit functions is accompanied by a increase in the constraint norm.

It can be shown that when the step $p_k$ is generated by the SQP method (10) then, near a solution satisfying second-order sufficient conditions, the algorithm above takes either the full step $p_k$ or the corrected step $p_k + \hat{p}_k$. The merit function does not interfere with the iteration, so superlinear convergence is attained, as in the local algorithm.

# Nonmonotone Strategy

The inefficiencies caused by the Maratos effect can also be avoided by occasionally accepting steps that increase the merit function; such steps are called *relaxed steps*. There is a limit to our liberality, however. If a sufficient reduction of the merit function has not been obtained within a certain number of iterates of the relaxed step ($\hat{t}$ iterates, say), then we return to the iterate before the relaxed step and perform a normal step, using a line search or some other technique to force a reduction in the merit function.

Our hope is, of course, that the relaxed step is a good step in the sense of making progress toward the solution, even though it increases the merit function. The step taken immediately after the relaxed step serves a similar purpose to the second-order correction step above; that is, it corrects the SQP step for its not taking sufficient account of curvature information for the constraint functions in its formulation of the SQP subproblem.

# Watchdog Strategy

In contrast with the second-order correction, which aims only to improve satisfaction of the constraints, this nonmonotone strategy always takes regular steps $p_k$ of the algorithm that aim both for improved feasibility and optimality. The hope is that any increase in the merit function over a single step will be temporary, and that subsequent steps will more than compensate for it.

We now describe a particular instance of this technique, which is often called the *watchdog strategy*. We set $\hat{t} = 1$, so that we allow the merit function to increase on just a single step before insisting on a sufficient decrease of some type. We focus our discussion on a line search SQP algorithm that uses the $\ell_1$ merit function. We assume that the penalty parameter $\mu$ is not changed until a successful cycle has been completed. For simplicity we omit the dependence of $\phi$ on $\mu$.

# Algorithm: Watchdog

Choose constant $\eta \in (0, 0.5)$;
Choose initial point $x_0$ and approximate Hessian $B_0$;
Set $k \leftarrow 0$, $\mathcal{S} \leftarrow \{0\}$;
**repeat**
    Evaluate $f_k, \nabla f_k, c_k, A_k$;
    **if** termination test satisfied
        **STOP** with approximate solution $x_k$;
    Compute the SQP step $p_k$;
    Set $x_{k+1} \leftarrow x_k + p_k$;
    Update $B_k$ using a quasi-Newton formula to obtain $B_{k+1}$;
    **if** $\phi_1(x_{k+1}) \le \phi_1(x_k) + \eta D\phi_1(x_k; p_k)$
        $k \leftarrow k + 1$;
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$;
    **else**
        Compute the SQP step $p_{k+1}$;
        Find $\alpha_{k+1}$ such that
            $\phi_1(x_{k+2}) \le \phi_1(x_{k+1}) + \eta \alpha_{k+1} D\phi_1(x_{k+1}; p_{k+1})$;
        Set $x_{k+2} \leftarrow x_{k+1} + \alpha_{k+1} p_{k+1}$;
        Update $B_{k+1}$ using a quasi-Newton formula to obtain $B_{k+2}$;

**if** $\phi_1(x_{k+1}) \leq \phi(x_k)$ **or** $\phi_1(x_{k+2}) \leq \phi_1(x_k) + \eta D\phi_1(x_k; p_k)$

$k \leftarrow k + 2$;

$\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$;

**else if** $\phi_1(x_{k+2}) > \phi_1(x_k)$

Find $\alpha_k$ such that $\phi_1(x_{k+3}) \leq \phi_1(x_k) + \eta\alpha_k D\phi_1(x_k; p_k)$;

Compute $x_{k+3} = x_k + \alpha_k p_k$;

**else**

Compute the SQP step $p_{k+2}$;

Find $\alpha_{k+2}$ such that

$$\phi_1(x_{k+3}) \leq \phi_1(x_{k+2}) + \eta\alpha_{k+2} D\phi_1(x_{k+2}; p_{k+2});$$

Set $x_{k+3} \leftarrow x_{k+2} + \alpha_{k+2} p_{k+2}$;

Update $B_{k+2}$ using a quasi-Newton formula to obtain $B_{k+3}$;

$k \leftarrow k + 3$;

$\mathcal{S} \leftarrow \mathcal{S} \cup \{k\}$;

**end (if)**

**end (if)**

**end (repeat)**

# Watchdog Strategy

The set $S$ is not required by the algorithm and is introduced only to identify the iterates for which a sufficient merit function reduction was obtained. Note that at least one third of the iterates have their indices in $S$. By using this fact, one can show that the SQP method using the watchdog technique is locally convergent, that for all sufficiently large $k$ the step length is $\alpha_k = 1$, and that the rate of convergence is superlinear.

In practice, it may be advantageous to allow increases in the merit function for more than 1 iteration. Values of $\hat{t}$ such as 5 or 10 are typical. A potential advantage of the watchdog technique over the second-order correction strategy is that it may require fewer evaluations of the constraints. Indeed, in the best case, most of the steps will be full SQP steps, and there will rarely be a need to return to an earlier point.

# Outline

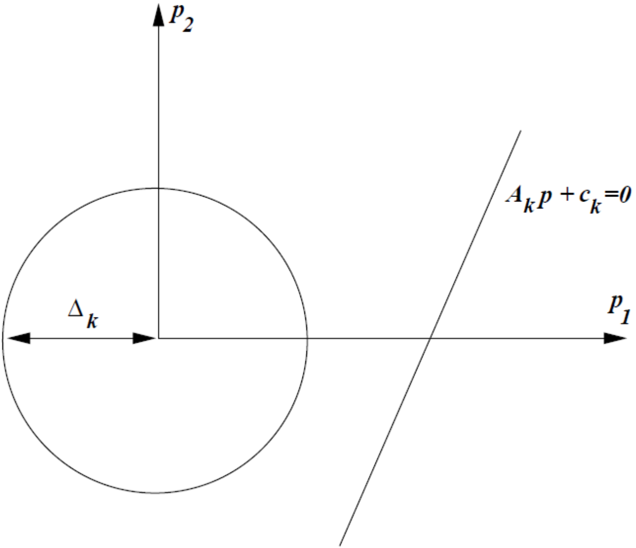
# The Trust-Region Subproblem

By adding a trust-region constraint, we obtain the new model

$$
\begin{aligned}
\min \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p && \text{(29a)}\\
s.t. \quad & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, && \text{(29b)}\\
& \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}, && \text{(29c)}\\
& \|p\| \leq \Delta_k. && \text{(29d)}
\end{aligned}
$$

Even if the constraints (29b), (29c) are compatible, this problem may not always have a solution because of the trust-region constraint (29d).

# Demo



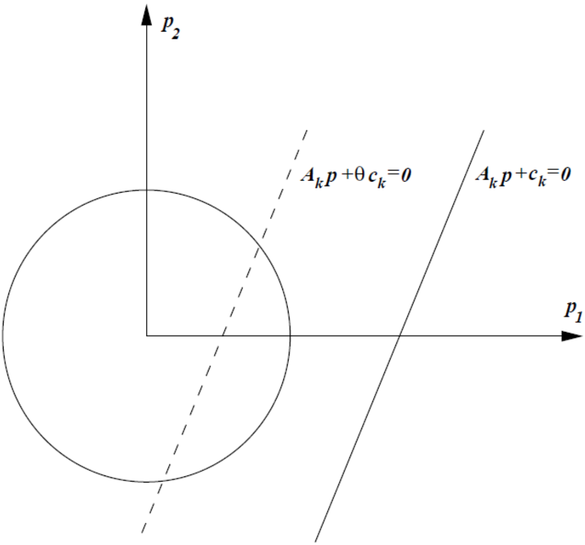
# The Trust-Region Subproblem

How to resolve the possible conflict between satisfying the linear constraints and the trust-region constraint?

It is not appropriate simply to increase $\Delta_k$ until the set of steps $p$ satisfying the linear constraints intersects the trust region. This approach would defeat the purpose of using the trust region in the first place as a way to define a region within which we trust the model to accurately reflect the behavior of the true objective and constraint functions, and it would harm the convergence properties of the algorithm.

A more appropriate viewpoint is that there is no reason to try to satisfy the equality constraints exactly at every step; rather, we should aim to improve the feasibility of these constraints at each step and to satisfy them exactly only in the limit. This point of view leads to different techniques for reformulating the trust-region subproblem.

# Approach I: Shifting the Constraints



# Approach I: Shifting the Constraints

In this approach, we need choose parameter $\theta$ and reformulate the SQP subproblem as following

$$\begin{array}{ll} \min\limits_{p} & f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ \text{s.t.} & A_k p + \theta c_k = 0, \\ & \|p\|_2 \leq \triangle_k. \end{array}$$

# Approach II: Two Elliptical Constraints

Another modification of the approach is to reformulate the SQP subproblem as

$$
\begin{aligned}
\min_{p} \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\
\text{s.t.} \quad & \|A_k p + c_k\|_2 = \pi_k, \\
& \|p\|_2 \le \triangle_k.
\end{aligned}
$$

There are several ways to choose the bound $\pi_k$. Regardless of the value of $\pi_k$, the above subproblem is more difficult to solve than a standard trust-region problem. Various techniques for finding exact or approximate solutions have been proposed, and all are satisfactory for the case in which $n$ is small or $A_k$ and $\nabla_{xx}^2 \mathcal{L}_k$ are dense. However, efficient algorithms for this subproblem are still not established for the large-scale case.

# Approach III: Sequential $\ell_1$ Quadratic Programming

The two approaches above were developed specifically with equality-constrained optimization in mind, and it is not trivial to extend them to inequality-constrained problems. The approach to be described here, however, handles inequality constraints in a straightforward way, so we describe it in terms of the general problem (10).

The S$\ell_1$QP approach moves the linearized constraints into the objective of the quadratic program, in the form of an $\ell_1$ penalty term, leaving only the trust region as a constraint. This strategy yields the following subproblem:

$$
\begin{aligned}
\min_p \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\
& + \mu_k \textstyle\sum_{i \in \mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| + \mu_k \textstyle\sum_{i \in \mathcal{I}} [c_i(x_k) + \nabla c_i(x_k)^T p]^- \\
\text{s.t.} \quad & \|p\|_\infty \le \triangle_k.
\end{aligned}
$$

# Outline

# Global Convergence and Superlinear Convergence Rate

Numerical experience has shown that the SQP methods often converge to a solution from remote starting points. In fact, under some assumption, all the limit points of the sequences $\{x_k\}$ generated by the SQP algorithms are KKT points of the nonlinear program (9).

Furthermore, under certain assumption, a superlinear rate of convergence of the SQP methods also can be obtained.

# Outline

# Perspective and Software

SQP method are most efficient if the number of active constraints is nearly as large as the number of variables, that is, if the number of free variables is relatively small. They require few evaluations of the functions, in comparison with augmented Lagrangian methods and can be more robust on badly scaled problems than the nonlinear interior-point methods.

# Perspective and Software

SQP method are most efficient if the number of active constraints is nearly as large as the number of variables, that is, if the number of free variables is relatively small. They require few evaluations of the functions, in comparison with augmented Lagrangian methods and can be more robust on badly scaled problems than the nonlinear interior-point methods.

Two established SQP software packages are SNOPT and FILTERSQP. The former code follows a line search approach, while the latter implements a trust-region strategy using a filter for step acceptance.

Thanks for your attention!