Chapter 14: Sequential Quadratic Programming

Outline

Outline

Local SQP Method

We begin by considering the equality-constrained problem

$$\min \quad f(x) \tag{1.1a}$$
$$s.t. \quad c(x) = 0. \tag{1.1b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c = (c_1, \ldots, c_m)^T$ and $c_i : \mathbb{R}^n \to \mathbb{R}$ are smooth functions.

- The essential idea of SQP is to model (1.1) 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 so that the overall SQP algorithm has good convergence properties and good practical performance.
- Perhaps the simplest derivation of SQP methods, which we now present, views them as an application of Newton's method to the KKT optimality conditions for (1.1).

Local SQP Method

We know that the Lagrangian function for this problem is

$$\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x).$$

We use $A(x)$ to denote the Jacobian matrix of the constraints, that is,

$$A(x)^T = [\nabla c_1(x), \nabla c_2(x), \cdots, \nabla c_m(x)], \tag{1.2}$$

where $c_i(x)$ is the ith component of the vector $c(x)$. 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 λ:

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0. \tag{1.3}$$

Any solution (x^*, λ^*) of the equality-constrained problem (1.1) for which $A(x^*)$ has full rank satisfies (1.3). One approach that suggests itself is to solve the nonlinear equations (1.3) by using Newton's method.

Local SQP Method

The Jacobian of (1.3) with respect to x and λ is given by

$$F'(x,\lambda) = \begin{bmatrix} \nabla^2_{xx}\mathcal{L}(x,\lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix}. \tag{1.4}$$

The Newton step form the iterate (x_k, λ_k) is given by

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix}, \tag{1.5}$$

where p_k and p_λ solve the Newton-KKT system

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix} \tag{1.6}$$

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

Local SQP Method

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 $A(x)d = 0$.

Nonsingularity is a consequence of the above condition.

- The first condition is the LICQ condition, which we assume throughout this chapter.
- The second condition holds whenever (x,λ) is close to the optimum (x^*,λ^*) and the second-order sufficient condition is satisfied at the solution.

The Newton iteration (1.5)-(1.6) 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^*.

SQP Framework

There is an alternative way to view the iteration (1.5), (1.6). Suppose that at the iterate (x_k, λ_k) we define the quadratic program

$$\min_p \quad f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \tag{1.7a}$$

$$\text{s.t.} \qquad A_k p + c_k = 0. \tag{1.7b}$$

If above assumptions hold, this problem has a unique solution (p_k, l_k) that satisfies

$$\nabla_{xx}^2 \mathcal{L}_k p_k + \nabla f_k - A_k^T l_k = 0, \tag{1.8a}$$

$$A_k p_k + c_k = 0. \tag{1.8b}$$

The vectors p_k and l_k can be identified with the solution of the Newton equations (1.6). If we subtract $A_k^T \lambda_k$ from both sides of the first equation in (1.6), we obtain

$$\begin{bmatrix} \nabla_{xx}^2 \mathcal{L}_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{1.9}$$

Hence, by nonsingularity of the coefficient matrix, we have that $\lambda_{k+1} = l_k$ and p_k solves (1.7) and (1.6).

SQP Framework

The new iterate (x_{k+1}, λ_{k+1}) can therefore be defined either as the solution of the quadratic program (1.7) or as the iterate generated by Newton's method (1.5) and (1.6) 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,
- whereas the SQP framework enables us to derive practical algorithms and to extend the technique to the inequality-constrained case.

SQP Framework

Algorithm 1: Local SQP Algorithm for solving (1.1)

Choose an initial pair (x_0, λ_0); set $k \leftarrow 0$;

repeat until a convergence test is satisfied

Evaluate f_k, ∇f_k, $\nabla^2_{xx}\mathcal{L}_k$, c_k and A_k;

Solve (1.7) 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)

SQP Framework

- We should note in passing that, in the objective (1.7a) 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 (1.7b) makes the two choices equivalent.
- In this case, (1.7a) is a quadratic approximation of the Lagrangian function.
- This fact provides a motivation for our choices of the quadratic model (1.7):
  - We first replace the nonlinear program (1.1) by the problem of minimizing the Lagrangian subject to the equality constraints $c(x) = 0$;
  - then make a quadratic approximation of the Lagrangian and a linear approximation of the constraints to obtain (1.7).

Inequality Constraints - IQP

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

$$
\begin{aligned}
\min \quad & f(x) && (1.10a)\\
s.t. \quad & c_i(x) = 0, \quad i \in \mathcal{E}, && (1.10b)\\
& c_i(x) \geq 0, \quad i \in \mathcal{I}. && (1.10c)
\end{aligned}
$$

The first approach solves at every iteration the quadratic subprogram

$$
\begin{aligned}
\min \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p && (1.11a)\\
s.t. \quad & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, && (1.11b)\\
& \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}, && (1.11c)
\end{aligned}
$$

The new iterate is given by $x_k + p_k, \lambda_{k+1}$ where p_k and λ_{k+1} are the solution and the corresponding Lagrangian multiplier of (1.11). A local SQP method for (1.10) follows from Algorithm 1 with the modification that the step is computed from (1.11). 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 approach the set of active constraints $\mathcal{A}_k$ at the solution of (1.11) 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 1

Suppose that x^* is a solution point of (1.11) at which the KKT conditions are satisfied for some λ^*. Suppose, too, that LICQ condition, the strict complementarity condition, and the second-order sufficient conditions hold at (x^*, λ^*). Then if (x_k, λ_k) is sufficiently close to (x^*, λ^*), there is a local solution of the subproblem (1.11) whose active set $\mathcal{A}_k$ is the same as the active set $\mathcal{A}(x^*)$ of the nonlinear program (1.10) at x^*.

Inequality Constraints - IQP

- The main drawback of IQP method is the expense of solving the general quadratic program (1.11), 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 (1.7), 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 (1.11) 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 (1.7a) 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 $\|p\|_\infty \leq \Delta_k$ 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 (1.7) 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.

Enforcing Convergence

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

- To ensure that the SQP method converges from remote starting points it is common to use a merit function ϕ to control the size of the steps (in line search methods) or to determine whether a step is acceptable and whether the trust- region radius needs to be modified (in trust-region methods). 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.
- 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. So we discuss the conditions on the problem and on the merit functions that ensure that the functions show a decrease on a step generated by the SQP method.

Merit Functions

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 (1.1).

Merit Functions

The ℓ_1 merit function for (1.1) takes the form

$$\phi_1(x;\mu) = f(x) + \mu\|c(x)\|_1. \tag{2.1}$$

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

$$\phi_1(x_k + \alpha_k p_k; \mu_k) \leq \phi_1(x_k; \mu_k) + \eta\alpha_k D(\phi_1(x_k; \mu_k); p_k), \qquad \eta \in (0,1), \tag{2.2}$$

where $D(\phi_1(x_k;\mu);p_k)$ denotes the directional derivatives of ϕ_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 μ is chosen sufficiently large, as we show in the following result.

Theorem 2

Let p_k and λ_{k+1} be generated by the SQP iteration (1.6). Then the directional derivative of ϕ_1 in the direction p_k satisfies

$$D(\phi_1(x_k;\mu);p_k) = \nabla f_k^T p_k - \mu\|c_k\|_1. \tag{2.3}$$

Moreover, we have that

$$D(\phi_1(x_k;\mu);p_k) \le -p_k^T \nabla_{xx}^2 \mathcal{L}_k p_k - (\mu - \|\lambda_{k+1}\|_\infty)\|c_k\|_1. \tag{2.4}$$

It follows from (2.4) that p_k will be a descent direction for ϕ_1 if $p_k \neq 0$, $\nabla_{xx}^2 \mathcal{L}_k$ is positive defined and

$$\mu > \|\lambda_{k+1}\|_\infty. \tag{2.5}$$

One strategy for choosing the new value of the penalty parameter μ in $\phi_1(x;\mu)$ at every iteration is to increase the previous value, if necessary, so as to satisfy (2.5), with some margin.

Outline

Handling Inconsistent Linearizations

A common difficulty in SQP methods is that the linearizations (1.11b)-(1.11c) 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.

Handling Inconsistent Linearizations

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

$$\min \quad f(x) + \mu \sum_{i\in\mathcal{E}} (v_i + w_i) + \mu \sum_{i\in\mathcal{I}} t_i \tag{3.1a}$$

$$\text{s.t.} \quad c_i(x) = v_i + w_i, \qquad i \in \mathcal{E} \tag{3.1b}$$

$$c_i(x) \geq -t_i, \qquad i \in \mathcal{I} \tag{3.1c}$$

$$v, w, t \geq 0, \tag{3.1d}$$

where μ is a nonnegative penalty parameter. If the nonlinear problem (1.10) has a feasible solution and μ is sufficiently large, the solutions to (3.1) and (1.10) 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 μ is large enough, then the auxiliary problem (3.1) usually determines a stationary point of the infeasibility measure.

Full Quasi-Newton Approximations

- The Hessian of the Lagrangian $\nabla^2_{xx} L(x_k, \lambda_k)$ is made up of second derivatives of the objective function and constraints. In some applications, this information is not easy to compute, so it is useful to consider replacing the Hessian $\nabla^2_{xx} L(x_k, \lambda_k)$ by a quasi-Newton approximation. Normally we define

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla \mathcal{L}(x_k, \lambda_{k+1}).$$

- Since the BFGS and SR1 formulae have proved to be successful in the context of unconstrained optimization, we can employ them here as well. SR1 formula is normally used in trust region framework.
- In line search framework, to guarantee the positive definiteness of B_{k+1}, BFGS updating requires that s_k and y_k satisfy the curvature condition $s_k^T y_k > 0$, which may not hold when applying line search with merit functions.

Damped BFGS Updating

The damped BFGS updating computes B_{k+1} as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{r_k r_k^T}{s_k^T r_k},$$

where $r_k = \theta_k y_k + (1 - \theta_k) B_k s_k$, and

$$\theta_k = \begin{cases} 1, & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{if } s_k^T y_k < 0.2 s_k^T B_k s_k. \end{cases}$$

It is easy to see that

$$s_k^T r_k \geq 0.2 s_k^T B_k s_k,$$

so the positive definiteness of B_k can be preserved.

The Maratos Effect

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. It is illustrated by the following example, in which the SQP steps p_k give rise to a quadratic convergence rate but cause an increase both in the objective function value and the constraint norm. As a result, these steps will be rejected by many merit functions.

Consider the problem

$$\min f(x_1, x_2) = 2(x_1^2 + x_2^2 - 1) - x_1, \text{ s.t. } x_1^2 + x_2^2 - 1 = 0. \tag{3.2}$$

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 $\nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) = I$.

The Maratos Effect

Let us consider an iterate x_k of the form $x_k = (\cos\theta, \sin\theta)^T$, which is feasible for any value of θ. We now generate a search direction p_k by solving the subproblem (1.7) with $\nabla^2_{xx}\mathcal{L}_k = I$. Since

$$f(x_k) = -\cos\theta,\ \nabla f(x) = \begin{pmatrix} 4\cos\theta - 1 \\ 4\sin\theta \end{pmatrix},\ A(x_k)^T = \begin{pmatrix} 2\cos\theta \\ 2\sin\theta \end{pmatrix}, \tag{3.3}$$

the quadratic subproblem (1.7) takes the form

$$\min_{p_1,p_2} \quad -\cos\theta + (4\cos\theta - 1)p_1 + 4\sin\theta p_2 + \tfrac{1}{2}p_1^2 + \tfrac{1}{2}p_2^2 \tag{3.4}$$

$$s.t. \quad p_2 = -\cot\theta p_1. \tag{3.5}$$

By solving this subproblem, we obtain

$$p_k = \begin{pmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{pmatrix}, \tag{3.6}$$

which yields a new trial point

$$x_k + p_k = \begin{pmatrix} \cos\theta + \sin^2\theta \\ \sin\theta(1-\cos\theta) \end{pmatrix}. \tag{3.7}$$

The Maratos Effect

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

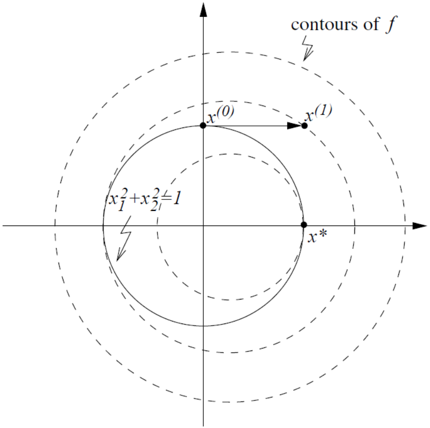
$$\frac{\|x_k + p_k - x^*\|_2}{\|x_k - x^*\|_2^2} = \frac{1}{2}.$$

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

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

- This example shows that 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 (3.6), so that any algorithm based on a merit function of this type will suffer from 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.
    - We can use a merit function that does not suffer from the Maratos effect. An example is Fletcher's augmented Lagrangian function.
    - We can 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.
    - We may allow the merit function ϕ to increase on certain iterations; that is, we can use a nonmonotone strategy.

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 (1.11). If this step yields an increase in the merit function ϕ_1, a possible cause is that our linear approximation to the constraints are not sufficiently accurate. To overcome this deficiency, we could re-solve (1.11) with the linear terms $c_i(x_k) + \nabla c_i(x_k)^T p$ replaced by the quadratic approximations

$$c_i(x_k) + \nabla c_i(x_k)^T p + \frac{1}{2} p^T \nabla^2 c_i(x_k) p. \tag{3.8}$$

However, even if the Hessian of the constraints are individually available, the resulting quadratically constrained 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

$$c_i(x_k + p_k) \approx c_i(x_k) + \nabla c_i(x_k)^T p_k + \frac{1}{2} p_k^T \nabla^2 c_i(x_k) p_k. \tag{3.9}$$

Even though we don't know what the step p would be if we used the quadratic approximations in the subproblem, we will assume that second-order step p will not be too different from p_k, then

$$p^T \nabla^2 c_i(x_k)^T p = p_k^T \nabla^2 c_i(x_k)^T p_k. \tag{3.10}$$

So we can approximate the last term in (3.8) as follows:

$$\frac{1}{2} p^T \nabla^2 c_i(x_k) p = \frac{1}{2} p_k^T \nabla^2 c_i(x_k) p_k. \tag{3.11}$$

Second-Order Correction

By making this substitution in (3.8) and using (3.9), we obtain the second-order correction subproblem

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

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 (1.11) 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.

Outline

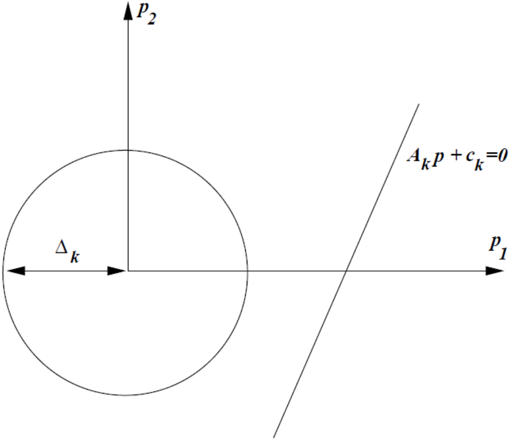
The Trust-Region Subproblem

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

$$
\begin{aligned}
\min \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx} \mathcal{L}_k p && \text{(4.1a)} \\
s.t. \quad & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, && \text{(4.1b)} \\
& \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}, && \text{(4.1c)} \\
& \|p\| \leq \Delta_k. && \text{(4.1d)}
\end{aligned}
$$

Even if the constraints (4.1b), (4.1c) are compatible, this problem may not always have a solution because of the trust-region constraint (4.1d).

Demo

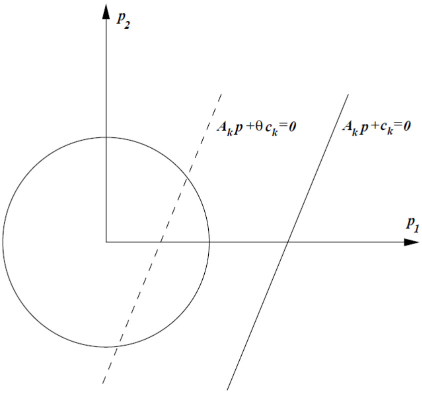


Inconsistent constraints in trust-region model.

The Trust-Region Subproblem

- To resolve the possible conflict between satisfying the linear constraints and the trust-region constraint, it is not appropriate simply to increase Δ_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



$$
\begin{aligned}
\min_{p} \quad & f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p && \text{(4.2a)} \\
s.t. \quad & A_k p + c_k = r_k, && \text{(4.2b)} \\
& \|p\|_2 \le \Delta_k. && \text{(4.2c)}
\end{aligned}
$$

Approach I: Shifting the Constraints

Phase 1: Solve the subproblem

$$\begin{aligned} \min_{p} \quad & \|A_k v + c_k\|_2^2 \\ s.t. \quad & \|v\|_2 \leq 0.8\Delta_k \end{aligned}$$

with solution denoted as v_k. Define

$$r_k = A_k v_k + c_k.$$

Phase 2: Solve the subproblem (4.2a) yielding p_k.

A merit function that fits well with this approach is the nonsmooth l_2 function

$$\phi_2(x;\mu) = f(x) + \mu\|c(x)\|_2.$$

Approach II: Two Elliptical Constraints

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

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

There are several ways to choose the bound π_k. Regardless of the value of π_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: Sℓ_1QP(Sequential ℓ_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 (1.11).
- The Sℓ_1QP approach moves the linearized constraints into the objective of the quadratic program, in the form of an ℓ_1 penalty term, leaving only the trust region as a constraint. This strategy yields the following subproblem:

$$\begin{array}{ll} \min\limits_{p} & f_k + \nabla f_k^T p + \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p \\ & +\mu_k \sum_{i\in\mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| + \mu_k \sum_{i\in\mathcal{I}} [c_i(x_k) + \nabla c_i(x_k)^T p]^- \\ \text{s.t.} & \|p\|_\infty \leq \Delta_k. \end{array}$$

- A natural choice about merit function is l_1 function $\phi_1(x;\mu) = f(x) + \mu\|c_{\mathcal{E}}(x)\|_1 + \mu\|[c_{\mathcal{I}}(x)]^-\|_1$.

Approach IV: Augmented Lagrangian Trust Region Method

To avoid the nonsmoothness of l_1 function in Sl_1QP, the augmented Lagrangian function can be used to define the objective of subproblems as well as the merit function. For equality-constrained case, define the subproblem as

$$\begin{aligned} \min_p \quad & f_k + (\nabla f_k - A_k^T\lambda_k)^T p + \tfrac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}_k p + \tfrac{\mu_k}{2}\|A_k p + c_k\|^2 \\ \text{s.t.} \quad & \|p\| \le \Delta_k. \end{aligned}$$

For general case, slack variables are introduced to handle inequality constraints. So bound constraints will be presented in subproblems. And the merit function can be the augmented Lagrangian function.

Outline

Global Convergence and Superlinear Convergence Rate

Under some assumption, all the limit points of the sequences $\{x_k\}$ generated by the SQP algorithms are KKT points of the nonlinear program (1.10). and a superlinear rate of convergence 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.
- 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!