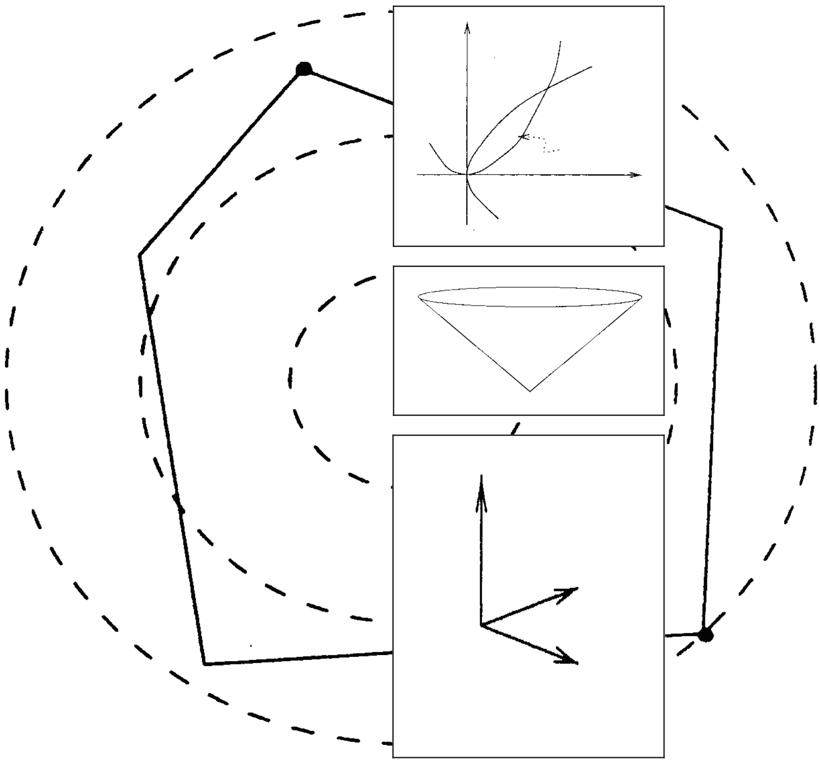
CHAPTER 18



Sequential Quadratic Programming

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. Unlike sequential linearly constrained methods (Chapter 17), which are effective when most of the constraints are linear, SQP methods show their strength when solving problems with significant nonlinearities.

Our development of SQP methods will be done in two stages. First we will present a local algorithm that motivates the SQP approach and that allows us to introduce the step computation and Hessian approximation techniques in a simple setting. We then consider practical line search and trust-region methods that achieve convergence from remote starting points.

18.1 LOCAL SQP METHOD

Let us begin by considering the equality-constrained problem

$$\min f(x) \tag{18.1a}$$
$$\text{subject to} \quad c(x) = 0, \tag{18.1b}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions. Problems containing only equality constraints are not very common in practice, but an understanding of (18.1) is crucial in the design of SQP methods for problems with general constraints.

The essential idea of SQP is to model (18.1) 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. 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 (18.1).

From (12.28) 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,

$$A(x)^T = [\nabla c_1(x), \nabla c_2(x), \ldots, \nabla c_m(x)], \tag{18.2}$$

where $c_i(x)$ is the ith component of the vector $c(x)$. By specializing the first-order (KKT) conditions (12.30) to the equality-constrained case, we obtain a system of $n + m$ equations in the $n + m$ unknowns x and λ:

$$F(x, \lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0. \tag{18.3}$$

If A_* has full rank, any solution (x^*, λ^*) of the equality-constrained problem (18.1) satisfies (18.3). One approach that suggests itself is to solve the nonlinear equations (18.3) by using Newton's method, as described in Chapter 11.

The Jacobian of (18.3) is given by

$$\begin{bmatrix} W(x, \lambda) & -A(x)^T \\ A(x) & 0 \end{bmatrix}, \tag{18.4}$$

where W denotes the Hessian of the Lagrangian,

$$W(x, \lambda) = \nabla^2_{xx} \mathcal{L}(x, \lambda). \tag{18.5}$$

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

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix}, \tag{18.6}$$

where p_k and p_λ solve the *KKT system*

$$\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix}. \tag{18.7}$$

This iteration, which is sometimes called the *Newton–Lagrange method*, is well-defined when the KKT matrix is nonsingular. We saw in Chapter 16 that nonsingularity is a consequence of the following conditions.

Assumption 18.1.

(a) *The constraint Jacobian A_k has full row rank.*

(b) *The matrix W_k is positive definite on the tangent space of the constraints, i.e., $d^T W_k d > 0$ for all $d \neq 0$ such that $A_k d = 0$.*

The first assumption is the linear independence constraint qualification discussed in Chapter 12 (see Definition 12.1), 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 (see Theorem 12.6). The Newton iteration (18.6), (18.7) 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 (18.6), (18.7). Suppose that at the iterate (x_k, λ_k) we define the quadratic program

$$\min_p \ \tfrac{1}{2} p^T W_k p + \nabla f_k^T p \tag{18.8a}$$

$$\text{subject to} \quad A_k p + c_k = 0. \tag{18.8b}$$

If Assumptions 18.1 hold, this problem has a unique solution (p_k, μ_k) that satisfies

$$W_k p_k + \nabla f_k - A_k^T \mu_k = 0, \tag{18.9a}$$

$$A_k p_k + c_k = 0. \tag{18.9b}$$

A key observation is that p_k and μ_k can be identified with the solution of the Newton equations (18.7). If we subtract $A_k^T \lambda_k$ from both sides of the first equation in (18.7), we obtain

$$\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix}. \tag{18.10}$$

Hence, by nonsingularity of the coefficient matrix, we have that $p = p_k$ and $\lambda_{k+1} = \mu_k$.

We refer to this interesting relationship as *the equivalence between SQP and Newton's method*: If Assumptions 18.1 hold at x_k, then the new iterate (x_{k+1}, λ_{k+1}) can be defined either as the solution of the quadratic program (18.8) or as the iterate generated by Newton's method (18.6), (18.7) applied to the optimality conditions of the problem. These alternative interpretations are quite 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.

We now state the SQP method in its simplest form.

Algorithm 18.1 (Local SQP Algorithm).

Choose an initial pair (x_0, λ_0);
for $k = 0, 1, 2, \ldots$
  Evaluate f_k, ∇f_k, $W_k = W(x_k, \lambda_k)$, c_k, and A_k;
  Solve (18.8) to obtain p_k and μ_k;
  $x_{k+1} \leftarrow x_k + p_k$; $\lambda_{k+1} \leftarrow \mu_k$;
  if convergence test satisfied
    STOP with approximate solution (x_{k+1}, λ_{k+1});
end (for).

It is straightforward to establish a local convergence result for this algorithm, since we know that it is equivalent to Newton's method applied to the nonlinear system $F(x, \lambda) = 0$. More specifically, if Assumptions 18.1 hold at a solution (x^*, λ^*) of (18.1), if f and c are twice differentiable with Lipschitz continuous second derivatives, and if the initial point (x_0, λ_0) is sufficiently close to (x^*, λ^*), then the iterates generated by Algorithm 18.1 converge quadratically to (x^*, λ^*) (see Section 18.10).

We should note in passing that in the objective (18.8a) 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 (18.8b) makes the two choices equivalent. In this case, (18.8a) is a quadratic approximation of the Lagrangian function, and this leads to an alternative motivation of the SQP method. We replace the nonlinear program (18.1) by the problem of minimizing the Lagrangian subject to the equality constraints (18.1b). By making a quadratic approximation of the Lagrangian and a linear approximation of the constraints we obtain (18.8).

INEQUALITY CONSTRAINTS

The SQP framework can be extended easily to the general nonlinear programming problem

$$\min \quad f(x) \tag{18.11a}$$
$$\text{subject to} \quad c_i(x) = 0, \quad i \in \mathcal{E}, \tag{18.11b}$$
$$c_i(x) \geq 0, \quad i \in \mathcal{I}. \tag{18.11c}$$

To define the subproblem we now linearize both the inequality and equality constraints to obtain

$$\min \tfrac{1}{2} p^T W_k p + \nabla f_k^T p \tag{18.12a}$$
$$\text{subject to} \quad \nabla c_i(x_k)^T p + c_i(x_k) = 0, \quad i \in \mathcal{E}, \tag{18.12b}$$
$$\nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \quad i \in \mathcal{I}. \tag{18.12c}$$

We can use one of the algorithms for quadratic programming described in Chapter 16 to solve this problem.

A local SQP method for (18.11) follows from Algorithm 18.1 above, with one modification: The step p_k and the new multiplier estimate λ_{k+1} are defined as the solution and the corresponding Lagrange multiplier of (18.12). The following result shows that this approach eventually identifies the optimal active set for the inequality-constrained problem (18.11). Recall that strict complementarity is said to hold at a solution pair (x^*, λ^*) if there is no index $i \in \mathcal{I}$ such that $\lambda_i^* = c_i(x^*) = 0$.

Theorem 18.1.

Suppose that x^* is a solution point of (18.11). Assume that the Jacobian A_* of the active constraints at x^* has full row rank, that $d^T W_* d > 0$ for all $d \neq 0$ such that $A_* d = 0$, and that strict complementarity holds. Then if (x_k, λ_k) is sufficiently close to (x^*, λ^*), there is a local solution of the subproblem (18.12) whose active set $\mathcal{A}_k$ is the same as the active set $\mathcal{A}(x^*)$ of the nonlinear program (18.11) at x^*.

As the iterates of the SQP method approach a minimizer satisfying the conditions given in the theorem, the active set will remain fixed. The subproblem (18.12) behaves like an equality-constrained quadratic program, since we can eventually ignore the inequality constraints that do not fall into the active set $\mathcal{A}(x^*)$, while treating the active constraints as equality constraints.

IQP VS. EQP

There are two ways of implementing the SQP method for solving the general nonlinear programming problem (18.11). The first approach solves at every iteration the quadratic subprogram (18.12), taking the active set at the solution of this subproblem as a guess of the optimal active set. This approach is referred to as the IQP (inequality-constrained QP) approach, and has proved to be quite successful in practice. Its main drawback is the expense of solving the general quadratic program (18.12), 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. This *hot-start* strategy is described later.

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 (18.8), 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 (18.12) and require less sophisticated software.

An example of an EQP method is the gradient projection method described in Section 16.6. 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 obtains a linear program by omitting the quadratic term $p^T W_k p$ from (18.12a), applying a trust-region constraint $\|p\| \leq \Delta_k$ on the step p (see (18.45c)), and taking the active set of this subproblem to be the working set for the current iteration. It then fixes the constraints in the working set and solves an equality-constrained quadratic program (with the term $p^T W_k p$ reinserted) to obtain the step.

18.2 PREVIEW OF PRACTICAL SQP METHODS

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.

We begin by drawing an analogy with unconstrained optimization. In its simplest form, the Newton iteration for minimizing a function f takes a step to the minimum of the quadratic model

$$m_k(p) = f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla^2 f_k p.$$

This framework is useful near the solution where the Hessian $\nabla^2 f(x_k)$ is normally positive definite and the quadratic model has a well-defined minimizer. When x_k is not close to the solution, however, the model function m_k may not be convex. Trust-region methods ensure that the new iterate is always well-defined and useful by restricting the candidate step p_k to some neighborhood of the origin. Line search methods modify the Hessian in $m_k(p)$ to make it positive definite (possibly replacing it by a quasi-Newton approximation B_k), to ensure that p_k is a descent direction for the objective function f.

Similar strategies are used to globalize SQP methods. If W_k is positive definite on the tangent space of the constraints, the quadratic subproblem (18.8) has a unique solution. When W_k does not have this property, line search methods either replace it by a positive definite approximation B_k or modify W_k directly during the process of matrix factorization. A third possibility is to define W_k as the Hessian of an *augmented Lagrangian* function having certain convexity properties. In all these cases, the subproblem (18.8) will be well-defined.

Trust-region SQP methods add a constraint to the subproblem, limiting the step to a region where the model (18.8) is considered to be reliable. Because they impose a trust-region bound on the step, they are able to use Hessians W_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 these tradeoffs, neither one of the two SQP approaches—line search or trust region—can be regarded as clearly superior to the other.

The technique used to solve the line search and trust-region subproblems has a great impact in the efficiency and robustness of SQP methods, particularly for large problems. For line search methods, the quadratic programming algorithms described in Chapter 16 can be used, whereas trust-region methods require special techniques.

Another important question in the development of SQP methods is the choice of a merit function that guides the algorithms toward the solution. In unconstrained optimization the merit function is simply the objective f, and it is fixed throughout the course of the minimization. SQP methods can use any of the merit functions discussed in Chapter 15, but the parameters of these merit functions may need to be adjusted on some iterations to ensure that the direction obtained from the subproblem is indeed a descent direction with respect to this function. The rules for updating these parameters require careful consideration, since they have a strong influence on the practical behavior of SQP methods.

In the remainder of this chapter we expand on these ideas to produce practical SQP algorithms. We first discuss a variety of techniques for solving the quadratic subproblems (18.8) and (18.12), and observe the effect that they have in the form of the SQP iteration. We then consider various formulations of the quadratic model that ensure their adequacy in a line search context, studying the case in which W_k is the exact Lagrangian Hessian and also the case in which it is a quasi-Newton approximation. We present two important merit functions and show that the SQP directions are descent directions for these functions. This discussion will set the stage for our presentation of practical line search and trust-region SQP methods.