Chapter 12
Primal Methods

In this chapter we initiate the presentation, analysis, and comparison of algorithms designed to solve constrained minimization problems. The four chapters that consider such problems roughly correspond to the following classification scheme. Consider a constrained minimization problem having n variables and m constraints. Methods can be devised for solving this problem that work in spaces of dimension $n - m$, n, m, or $n + m$. Each of the following chapters corresponds to methods in one of these spaces. Thus, the methods in the different chapters represent quite different approaches and are founded on different aspects of the theory. However, there are also strong interconnections between the methods of the various chapters, both in the final form of implementation and in their performance. Indeed, there soon emerges the theme that the rates of convergence of most practical algorithms are determined by the structure of the Hessian of the Lagrangian much like the structure of the Hessian of the objective function determines the rates of convergence for a wide assortment of methods for unconstrained problems. Thus, although the various algorithms of these chapters differ substantially in their motivation, they are ultimately found to be governed by a common set of principles.

12.1 Advantage of Primal Methods

We consider the question of solving the general nonlinear programming problem

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}) \leqslant \mathbf{0} \end{aligned} \tag{12.1}$$

where $\mathbf{x}$ is of dimension n, while f, $\mathbf{g}$, and $\mathbf{h}$ have dimensions 1, p, and m, respectively. It is assumed throughout the chapter that all of the functions have continuous partial derivatives of order three. Geometrically, we regard the problem as that of minimizing f over the region in E^n defined by the constraints.

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3_12

By a *primal method* of solution we mean a search method that works on the original problem directly by searching through the feasible region for the optimal solution. Each point in the process is feasible and the value of the objective function constantly decreases. For a problem with n variables and having m equality constraints only, primal methods work in the feasible space, which has dimension $n - m$.

Primal methods possess three significant advantages that recommend their use as general procedures applicable to almost all nonlinear programming problems. First, since each point generated in the search procedure is feasible, if the process is terminated before reaching the solution (as practicality almost always dictates for nonlinear problems), the terminating point is feasible. Thus this final point is a feasible and probably nearly optimal solution to the original problem and therefore may represent an acceptable solution to the practical problem that motivated the nonlinear program. A second attractive feature of primal methods is that, often, it can be guaranteed that if they generate a convergent sequence, the limit point of that sequence must be at least a local constrained minimum. Finally, a major advantage is that most primal methods do not rely on special problem structure, such as convexity, and hence these methods are applicable to general nonlinear programming problems.

Primal methods are not, however, without major disadvantages. They require a phase I procedure (see Sect. 3.5) to obtain an initial feasible point, and they are all plagued, particularly for problems with nonlinear constraints, with computational difficulties arising from the necessity to remain within the feasible region as the method progresses. Some methods can fail to converge for problems with inequality constraints unless elaborate precautions are taken.

The convergence rates of primal methods are competitive with those of other methods, and particularly for linear constraints, they are often among the most efficient. On balance their general applicability and simplicity place these methods in a role of central importance among nonlinear programming algorithms.

12.2 Feasible Direction Methods

The idea of feasible direction methods is to take steps through the feasible region of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{12.2}$$

where $\mathbf{d}_k$ is a direction vector and α_k is a nonnegative scalar. The scalar is chosen to minimize the objective function f with the restriction that the point $\mathbf{x}_{k+1}$ and the line segment joining $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ be feasible. Thus, in order that the process of minimizing with respect to α be nontrivial, an initial segment of the ray $\mathbf{x}_k + \alpha\mathbf{d}_k$, $\alpha > 0$ must be contained in the feasible region. This motivates the use of *feasible directions* for the directions of search. We recall from Sect. 7.1 that a vector $\mathbf{d}_k$ is a *feasible direction* (at $\mathbf{x}_k$) if there is an $\bar{\alpha} > 0$ such that $\mathbf{x}_k + \alpha\mathbf{d}_k$ is feasible for all α, $0 \leqslant \alpha \leqslant \bar{\alpha}$. A feasible direction method can be considered as a natural extension of our unconstrained descent methods. Each step is the composition of selecting a feasible direction and a constrained line search.

Let us consider the problem with linear inequality constraints

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} \leqslant b_i,\ i = 1, \ldots, m. \end{array} \tag{12.3}$$

Example 1 (Frank-Wolfe Method). One of the earliest proposals for a feasible direction method uses a sequential linear programming subproblem approach. Given a feasible point $\mathbf{x}_k$, the direction vector

$$\mathbf{d}_k = \mathbf{x}_k^* - \mathbf{x}_k$$

where $\mathbf{x}_k^*$ is a solution to the linear program

$$\begin{array}{ll} \text{minimize} & \nabla f(\mathbf{x}_k)\mathbf{x} \\ \text{subject to} & \mathbf{a}_i^T\mathbf{x} \leqslant b_i,\ i = 1, \ldots, m. \end{array}$$

Example 2 (Simplified Zoutendijk Method). Another proposal solves a sequence of linear subprograms as follows. Given a feasible point, $\mathbf{x}_k$, let I be the set of indices representing active constraints, that is, $\mathbf{a}_i^T\mathbf{x}_k = b_i$ for $i \in I$. The direction vector $\mathbf{d}_k$ is then chosen as a solution to the linear program

$$\begin{array}{lll} \text{minimize} & \nabla f(\mathbf{x}_k)\mathbf{d} & \\ \text{subject to} & \mathbf{a}_i^T\mathbf{d} \leqslant 0, & i \in I \\ & \sum\limits_{i=1}^{n} |d_i| = 1, & \end{array} \tag{12.4}$$

where $\mathbf{d} = (d_1,\ d_2,\ \ldots,\ d_n)$. The last equation is a normalizing equation that ensures a bounded solution. (Even though it is written in terms of absolute values, the problem can be converted to a linear program; see Exercise 1.) The other constraints assure that vectors of the form $\mathbf{x}_k + \alpha\mathbf{d}_k$ will be feasible for sufficiently small $\alpha > 0$, and subject to these conditions, $\mathbf{d}$ is chosen to line up as closely as possible with the negative gradient of f. In some sense this will result in the locally best direction in which to proceed. The overall procedure progresses by generating feasible directions in this manner, and moving along them to decrease the objective.

There are two major shortcomings of feasible direction methods that require that they be modified in most cases. The first shortcoming is that for general problems there may not exist any feasible directions. If, for example, a problem had nonlinear equality constraints, we might find ourselves in the situation depicted by Fig. 12.1 where no straight line from $\mathbf{x}_k$ has a feasible segment. For such problems it is necessary either to relax our requirement of feasibility by allowing points to deviate slightly from the constraint surface or to introduce the concept of moving along curves rather than straight lines.

Fig. 12.1 No feasible direction

A second shortcoming is that in simplest form most feasible direction methods, such as the simplified Zoutendijk method, are not globally convergent. They are subject to *jamming* (sometimes referred to as *zigzagging*) where the sequence of points generated by the process converges to a point that is not even a constrained local minimum point. This phenomenon can be explained by the fact that the algorithmic map is not closed.

It is possible to develop feasible direction algorithms that are closed and hence not subject to jamming. Some procedures for doing so are discussed in Exercises 4–7. However, such methods can become somewhat complicated. A simpler approach for treating inequality constraints is to use an active set method, as discussed in the next section.

12.3 Active Set Methods

The idea underlying active set methods is to partition inequality constraints into two groups: those that are to be treated as active and those that are to be treated as inactive. The constraints treated as inactive are essentially ignored.

Consider the constrained problem

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{g}(\mathbf{x}) \leqslant \mathbf{0}, \end{aligned} \tag{12.5}$$

which for simplicity of the current discussion is taken to have inequality constraints only. The inclusion of equality constraints is straightforward, as will become clear.

The necessary conditions for this problem are

$$\begin{aligned} \nabla f(\mathbf{x}) + \boldsymbol{\lambda}^T \nabla \mathbf{g}(\mathbf{x}) &= \mathbf{0} \\ \mathbf{g}(\mathbf{x}) &\leqslant \mathbf{0} \\ \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) &= 0 \\ \boldsymbol{\lambda} &\geqslant \mathbf{0}. \end{aligned} \tag{12.6}$$

(See Sect. 11.8.) These conditions can be expressed in a somewhat simpler form in terms of the set of active constraints. Let A denote the index set of active constraints; that is, A is the set of i such that $g_i(\mathbf{x}^*) = 0$. Then the necessary conditions (12.6) become

$$
\begin{aligned}
\nabla f(\mathbf{x}) + \sum_{i \in A} \lambda_i \nabla g_i(\mathbf{x}) &= \mathbf{0} \\
g_i(\mathbf{x}) &= 0, \quad i \in A \\
g_i(\mathbf{x}) &< 0, \quad i \notin A \\
\lambda_i &\geqslant 0, \quad i \in A \\
\lambda_i &= 0, \quad i \notin A
\end{aligned}
\tag{12.7}
$$

The first two lines of these conditions correspond identically to the necessary conditions of the equality constrained problem obtained by requiring the active constraints to be zero. The next line guarantees that the inactive constraints are satisfied, and the sign requirement of the Lagrange multipliers guarantees that every constraint that is active *should* be active.

It is clear that if the active set were known, the original problem could be replaced by the corresponding problem having equality constraints only. Alternatively, suppose an active set was guessed and the corresponding equality constrained problem solved. Then if the other constraints were satisfied and the Lagrange multipliers turned out to be nonnegative, that solution would be correct.

The idea of active set methods is to define at each step, or at each phase, of an algorithm a set of constraints, termed the *working set*, that is to be treated as the active set. The working set is chosen to be a subset of the constraints that are actually active at the current point, and hence the current point is feasible for the working set. The algorithm then proceeds to move on the surface defined by the working set of constraints to an improved point. At this new point the working set may be changed. Overall, then, an active set method consists of the following components: (1) determination of a current working set that is a subset of the current active constraints, and (2) movement on the surface defined by the working set to an improved point.

There are several methods for determining the movement on the surface defined by the working set. (This surface will be called the *working surface*.) The most important of these methods are discussed in the following sections. The direction of movement is generally determined by first-order or second-order approximations of the functions at the current point in a manner similar to that for unconstrained problems. The asymptotic convergence properties of active set methods depend entirely on the procedure for moving on the working surface, since near the solution the working set is generally equal to the correct active set, and the process simply moves successively on the surface determined by those constraints.

Changes in Working Set

Suppose that for a given working set W the problem with equality constraints

$$
\begin{aligned}
&\text{minimize} \quad f(\mathbf{x}) \\
&\text{subject to} \quad g_i(\mathbf{x}) = 0, \quad i \in W
\end{aligned}
$$

is solved yielding the point $\mathbf{x}_W$ that satisfies $g_i(\mathbf{x}_W) < 0,\ i \notin W$. This point satisfies the necessary conditions

$$\nabla f(\mathbf{x}_W) + \sum_{i \in W} \lambda_j \nabla g_i(\mathbf{x}_W) = \mathbf{0}. \tag{12.8}$$

If $\lambda_i \geqslant 0$ for all $i \in W$, then the point $\mathbf{x}_W$ is a local solution to the original problem. If, on the other hand, there is an $i \in W$ such that $\lambda_i < 0$, then the objective can be decreased by relaxing constraint i. This follows directly from the sensitivity interpretation of Lagrange multipliers, since a small decrease in the constraint value from 0 to $-c$ would lead to a change in the objective function of $\lambda_i c$, which is negative. Thus, by dropping the constraint i from the working set, an improved solution can be obtained. The Lagrange multiplier of a problem thereby serves as an indication of which constraints should be dropped from the working set. This is illustrated in Fig. 12.2. In the figure, $\mathbf{x}$ is the minimum point of f on the surface (a curve in this case) defined by $g_1(x) = 0$. However, it is clear that the corresponding Lagrange multiplier λ_1 is negative, implying that g_1 should be dropped. Since ∇f points outside, it is clear that a movement toward the interior of the feasible region will indeed decrease f.

During the course of minimizing $f(\mathbf{x})$ over the working surface, it is necessary to monitor the values of the other constraints to be sure that they are not violated, since all points defined by the algorithm must be feasible. It often happens that while moving on the working surface a new constraint boundary is encountered. It is then convenient to add this constraint to the working set, proceeding on a surface of one lower dimension than before. This is illustrated in Fig. 12.3. In the figure the working constraint is just $g_1 = 0$ for $\mathbf{x}_1,\ \mathbf{x}_2,\ \mathbf{x}_3$. A boundary is encountered at the next step, and therefore $g_2 = 0$ is adjoined to the set of working constraints.

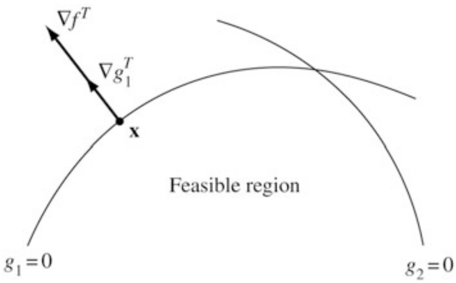

Fig. 12.2 Constraint to be dropped

A complete active set strategy for systematically dropping and adding constraints can be developed by combining the above two ideas. One starts with a given working set and begins minimizing over the corresponding working surface. If new constraint boundaries are encountered, they may be added to the working set, but no constraints are dropped from the working set. Finally, a point is obtained that minimizes f

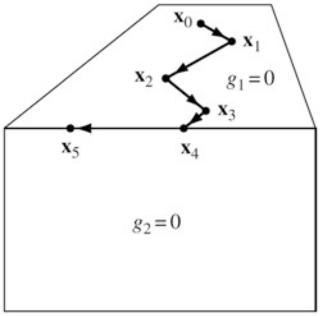

Fig. 12.3 Constraint added to working set

with respect to the current working set of constraints. The corresponding Lagrange multipliers are determined, and if they are all nonnegative the solution is optimal. Otherwise, one or more constraints with negative Lagrange multipliers are dropped from the working set. The procedure is reinitiated with this new working set, and f will strictly decrease on the next step.

An active set method built upon this basic active set strategy requires that a procedure be defined for minimization on a working surface that allows constraints to be added to the working set when they are encountered, and that, after dropping a constraint, insures that the objective is strictly decreased. Such a method is guaranteed to converge to the optimal solution, as shown below.

Active Set Theorem. *Suppose that for every subset W of the constraint indices, the constrained problem*

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & g_i(\mathbf{x}) = 0, \quad i \in W \end{array} \tag{12.9}$$

is well-defined with a unique nondegenerate solution (that is, for all $i \in W$, $\lambda_i \neq 0$). Then the sequence of points generated by the basic active set strategy converges to the solution of the inequality constrained problem (12.6).

Proof. After the solution corresponding to one working set is found, a decrease in the objective is made, and hence it is not possible to return to that working set. Since there are only a finite number of working sets, the process must terminate. ∎

The difficulty with the above procedure is that several problems with incorrect active sets must be solved. Furthermore, the solutions to these intermediate problems must, in general, be exact global minimum points in order to determine the correct sign of the Lagrange multipliers and to assure that during the subsequent descent process the current working surface is not encountered again.

In practice one deviates from the ideal basic method outlined above by dropping constraints using various criteria before an exact minimum on the working surface is found. Convergence cannot be guaranteed for many of these methods, and indeed they are subject to zigzagging (or jamming) where the working set changes an infinite number of times. However, experience has shown that zigzagging is very rare for many algorithms, and in practice the active set strategy with various refinement is often very effective.

It is clear that a fundamental component of an active set method is the algorithm for solving a problem with equality constraints only, that is, for minimizing on the working surface. Such methods and their analyses are presented in the following sections.

12.4 The Gradient Projection Method

The gradient projection method is motivated by the ordinary method of steepest descent for unconstrained problems. The negative gradient is projected onto the working surface in order to define the direction of movement. We present it here in a simplified form that is based on a pure active set strategy.

Linear Constraints

Consider first problems of the form

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{a}_i^T\mathbf{x} \leqslant b_i, \quad i \in I_1 \\ &&& \mathbf{a}_i^T\mathbf{x} = b_i, \quad i \in I_2 \end{aligned} \tag{12.10}$$

having linear equalities and inequalities.

A feasible solution to the constraints, if one exists, can be found by application of the phase I procedure of linear programming; so we shall always assume that our descent process is initiated at such a feasible point. At a given feasible point $\mathbf{x}$ there will be a certain number q of active constraints satisfying $\mathbf{a}_i^T\mathbf{x} = b_i$ and some inactive constraints $\mathbf{a}_i^T\mathbf{x} < b_i$. We initially take the working set $W(\mathbf{x})$ to be the set of active constraints.

At the feasible point $\mathbf{x}$ we seek a feasible direction vector $\mathbf{d}$ satisfying $\nabla f(\mathbf{x})\mathbf{d} < 0$, so that movement in the direction $\mathbf{d}$ will cause a decrease in the function f. Initially, we consider directions satisfying $\mathbf{a}_i^T\mathbf{d} = 0$, $i \in W(\mathbf{x})$ so that all working constraints remain active. This requirement amounts to requiring that the direction vector $\mathbf{d}$ lie in the tangent subspace $\mathbf{M}$ defined by the working set of constraints. The particular direction vector that we shall use is the projection of the negative gradient onto this subspace.

To compute this projection let $\mathbf{A}_q$ be defined as composed of the rows of working constraints. Assuming regularity of the constraints, as we shall always assume, $\mathbf{A}_q$ will be a $q \times n$ matrix of rank $q < n$. The tangent subspace M in which $\mathbf{d}$ must lie is the subspace of vectors satisfying $\mathbf{A}_q\mathbf{d} = \mathbf{0}$. This means that the subspace N

consisting of the vectors making up the rows of $\mathbf{A}_q$ (that is, all vectors of the form $\mathbf{A}_q^T\boldsymbol{\lambda}$ for $\boldsymbol{\lambda} \in E^q$) is orthogonal to M. Indeed, any vector can be written as the sum of vectors from each of these two complementary subspaces. In particular, the negative gradient vector $-\mathbf{g}_k$ can be written

$$-\mathbf{g}_k = \mathbf{d}_k + \mathbf{A}_q^T\boldsymbol{\lambda}_k \tag{12.11}$$

where $\mathbf{d}_k \in M$ and $\boldsymbol{\lambda}_k \in E^q$. We may solve for $\boldsymbol{\lambda}_k$ through the requirement that $\mathbf{A}_q\mathbf{d}_k = \mathbf{0}$. Thus

$$\mathbf{A}_q\mathbf{d}_k = -\mathbf{A}_q\mathbf{g}_k - (\mathbf{A}_q\mathbf{A}_q^T)\boldsymbol{\lambda}_k = \mathbf{0}, \tag{12.12}$$

which leads to

$$\boldsymbol{\lambda}_k = -(\mathbf{A}_q\mathbf{A}_q^T)^{-1}\mathbf{A}_q\mathbf{g}_k \tag{12.13}$$

and

$$\mathbf{d}_k = -[\mathbf{I} - \mathbf{A}_q^T(\mathbf{A}_q\mathbf{A}_q^T)^{-1}\mathbf{A}_q]\mathbf{g}_k = -\mathbf{P}_k\mathbf{g}_k. \tag{12.14}$$

The matrix

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{A}_q^T(\mathbf{A}_q\mathbf{A}_q^T)^{-1}\mathbf{A}_q] \tag{12.15}$$

is called the projection matrix corresponding to the subspace M. Action by it on any vector yields the projection of that vector onto M. See Exercises 8 and 9 for other derivations of this result.

We easily check that if $\mathbf{d}_k \neq \mathbf{0}$, then it is a direction of descent. Since $\mathbf{g}_k + \mathbf{d}_k$ is orthogonal to $\mathbf{d}_k$, we have

$$\mathbf{g}_k^T\mathbf{d}_k = (\mathbf{g}_k^T + \mathbf{d}_k^T - \mathbf{d}_k^T)\mathbf{d}_k = -|\mathbf{d}_k|^2.$$

Thus if $\mathbf{d}_k$ as computed from (12.14) turns out to be nonzero, it is a feasible direction of descent on the working surface.

We next consider selection of the step size. As α is increased from zero, the point $\mathbf{x} + \alpha\mathbf{d}$ will initially remain feasible and the corresponding value of f will decrease. We find the length of the feasible segment of the line emanating from $\mathbf{x}$ and then minimize f over this segment. If the minimum occurs at the endpoint, a new constraint will become active and will be added to the working set.

Next, consider the possibility that the projected negative gradient is zero. We have in that case

$$\nabla f(\mathbf{x}_k) + \boldsymbol{\lambda}_k^T\mathbf{A}_q = \mathbf{0}, \tag{12.16}$$

and the point $\mathbf{x}_k$ satisfies the necessary conditions for a minimum on the working surface. If the components of $\boldsymbol{\lambda}_k$ corresponding to the active inequalities are all non-negative, then this fact together with (12.16) implies that the Karush-Kuhn-Tucker conditions for the original problem are satisfied at $\mathbf{x}_k$ and the process terminates. In this case the $\boldsymbol{\lambda}_k$ found by projecting the negative gradient is essentially the Lagrange multiplier vector for the original problem (except that zero-valued multipliers must be appended for the inactive constraints).

If, however, at least one of those components of $\boldsymbol{\lambda}_k$ is negative, it is possible, by relaxing the corresponding inequality, to move in a new direction to an improved point. Suppose that λ_{jk}, the jth component of $\boldsymbol{\lambda}_k$, is negative and the indexing is arranged so that the corresponding constraint is the inequality $\mathbf{a}_j^T\mathbf{x} \leqslant b_j$. We determine the new direction vector by relaxing the jth constraint and projecting the negative gradient onto the subspace determined by the remaining $q-1$ active constraints. Let $\mathbf{A}_{\bar{q}}$ denote the matrix $\mathbf{A}_q$ with row $\mathbf{a}_j$ deleted. We have for some $\bar{\boldsymbol{\lambda}}_k$

$$-\mathbf{g}_k = \mathbf{A}_q^T\boldsymbol{\lambda}_k \tag{12.17}$$

$$-\mathbf{g}_k = \bar{\mathbf{d}}_k + \mathbf{A}_{\bar{q}}^T\bar{\boldsymbol{\lambda}}_k, \tag{12.18}$$

where $\bar{\mathbf{d}}_k$ is the projection of $-\mathbf{g}_k$ using $\mathbf{A}_{\bar{q}}$. It is immediately clear that $\bar{\mathbf{d}}_k \neq \mathbf{0}$, since otherwise (12.18) would be a special case of (12.17) with $\lambda_{jk} = 0$ which is impossible, since the rows of $\mathbf{A}_q$ are linearly independent. From our previous work we know that $\mathbf{g}_k^T\bar{\mathbf{d}}_k < 0$. Multiplying the transpose of (12.17) by $\bar{\mathbf{d}}_k$ and using $\mathbf{A}_{\bar{q}}\bar{\mathbf{d}}_k = \mathbf{0}$ we obtain

$$0 > \mathbf{g}_k^T\bar{\mathbf{d}}_k = -\lambda_{jk}\mathbf{a}_j^T\bar{\mathbf{d}}_k. \tag{12.19}$$

Since $\lambda_{jk} < 0$ we conclude that $\mathbf{a}_j^T\bar{\mathbf{d}}_k < 0$. Thus the vector $\bar{\mathbf{d}}_k$ is not only a direction of descent, but it is a feasible direction, since $\mathbf{a}_j^T\bar{\mathbf{d}}_k = 0$, $i \in W(\mathbf{x}_k)$, $i \neq j$, and $\mathbf{a}_j^T\bar{\mathbf{d}}_k < 0$. Hence j can be dropped from $W(\mathbf{x}_k)$.

In summary, one step of the algorithm is as follows: Given a feasible point $\mathbf{x}$

1. Find the subspace of active constraints M, and form $\mathbf{A}_q$, $W(\mathbf{x})$.
2. Calculate $\mathbf{P} = \mathbf{I} - \mathbf{A}_q^T(\mathbf{A}_q\mathbf{A}_q^T)^{-1}\mathbf{A}_q$ and $\mathbf{d} = -\mathbf{P}\nabla f(\mathrm{x})^T$.
3. If $\mathbf{d} \neq \mathbf{0}$, find α_1 and α_2 achieving, respectively,

$$\max\{\alpha : \mathbf{x} + \alpha\mathbf{d} \text{ is feasible}\}$$
$$\min\{f(\mathbf{x} + \alpha\mathbf{d}) : 0 \leqslant \alpha \leqslant \alpha_1\}.$$

   Set $\mathbf{x}$ to $\mathbf{x} + \alpha_2\mathbf{d}$ and return to (12.1).
4. If $\mathbf{d} = \mathbf{0}$, find $\boldsymbol{\lambda} = -(\mathbf{A}_q\mathbf{A}_q^T)^{-1}\mathbf{A}_q\nabla f(\mathbf{x})^T$.
   (a) If $\lambda_j \geqslant 0$, for all j corresponding to active inequalities, stop; $\mathbf{x}$ satisfies the Karush-Kuhn-Tucker conditions.
   (b) Otherwise, delete the row from $\mathbf{A}_q$ corresponding to the inequality with the most negative component of $\boldsymbol{\lambda}$ (and drop the corresponding constraint from $W(\mathbf{x})$) and return to (12.2).

The projection matrix need not be recomputed in its entirety at each new point. Since the set of active constraints in the working set changes by at most one constraint at a time, it is possible to calculate one required projection matrix from the previous one by an updating procedure. (See Exercise 11.) This is an important feature of the gradient projection method and greatly reduces the computation required at each step.

Example. Consider the problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \text{subject to} & 2x_1 + x_2 + x_3 + 4x_4 = 7 \\ & x_1 + x_2 + 2x_3 + x_4 = 6 \\ & x_i \geqslant 0, \ i = 1, 2, 3, 4. \end{array} \tag{12.20}$$

Suppose that given the feasible point $\mathbf{x} = (2, 2, 1, 0)$ we wish to find the direction of the projected negative gradient. The active constraints are the two equalities and the inequality $x_4 \geqslant 0$. Thus

$$\mathbf{A}_q = \begin{bmatrix} 2 & 1 & 1 & 4 \\ 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{12.21}$$

and hence

$$\mathbf{A}_q\mathbf{A}_q^T = \begin{bmatrix} 22 & 9 & 4 \\ 9 & 7 & 1 \\ 4 & 1 & 1 \end{bmatrix}.$$

After considerable calculation we then find

$$(\mathbf{A}_q\mathbf{A}_q^T)^{-1} = \frac{1}{11}\begin{bmatrix} 6 & -5 & -19 \\ -5 & 6 & 14 \\ -19 & 14 & 73 \end{bmatrix}$$

and finally

$$\mathbf{P} = \frac{1}{11}\begin{bmatrix} 1 & -3 & 1 & 0 \\ -3 & 9 & -3 & 0 \\ 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{12.22}$$

The gradient at the point (2, 2, 1, 0) is $\mathbf{g} = (2, 4, 2, -3)$ and hence we find

$$\mathbf{d} = -\mathbf{Pg} = \frac{1}{11}(-8, 24, -8, 0),$$

or normalizing by 8/11

$$\mathbf{d} = (-1, 3, -1, 0). \tag{12.23}$$

It can be easily verified that movement in this direction does not violate the constraints.

Nonlinear Constraints

In extending the gradient projection method to problems of the form

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \ \mathbf{g}(\mathbf{x}) \leqslant \mathbf{0}, \end{array} \tag{12.24}$$

the basic idea is that at a feasible point $\mathbf{x}_k$ one determines the active constraints and projects the negative gradient onto the subspace tangent to the surface determined by these constraints. This vector, if it is nonzero, determines the direction for the next step. The vector itself, however, is not in general a feasible direction, since the surface may be curved as illustrated in Fig. 12.4. It is therefore not always possible to move along this projected negative gradient to obtain the next point.

What is typically done in the face of this difficulty is essentially to search along a curve on the constraint surface, the direction of the curve being defined by the projected negative gradient. A new point is found in the following way: First, a move is made along the projected negative gradient to a point $\mathbf{y}$. Then a move is made in the direction perpendicular to the tangent plane at the original point to a nearby feasible point on the working surface, as illustrated in Fig. 12.4. Once this point is found the value of the objective is determined. This is repeated with various $\mathbf{y}$'s until a feasible point is found that satisfies one of the standard descent criteria for improvement relative to the original point.

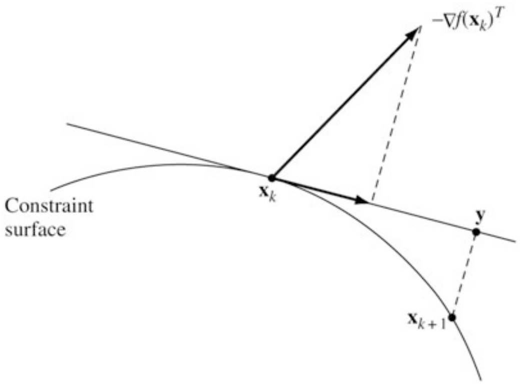

Fig. 12.4 Gradient projection method

This procedure of tentatively moving away from the feasible region and then coming back introduces a number of additional difficulties that require a series of interpolations and nonlinear equation solutions for their resolution. A satisfactory general routine implementing the gradient projection philosophy is therefore of necessity quite complex. It is not our purpose here to elaborate on these details but simply to point out the general nature of the difficulties and the basic devices for surmounting them.

One difficulty is illustrated in Fig. 12.5. If, after moving along the projected negative gradient to a point $\mathbf{y}$, one attempts to return to a point that satisfies the old active constraints, some inequalities that were originally satisfied may then be violated. One must in this circumstance use an interpolation scheme to find a new point $\bar{\mathbf{y}}$ along the negative gradient so that when returning to the active constraints no originally nonactive constraint is violated. Finding an appropriate $\bar{\mathbf{y}}$ is to some extent a trial and error process. Finally, the job of returning to the active constraints is itself a nonlinear problem which must be solved with an iterative technique. Such a technique is described below, but within a finite number of iterations, it cannot exactly reach the surface. Thus typically an error tolerance δ is introduced, and throughout the procedure the constraints are satisfied only to within δ.

Computation of the projections is also more difficult in the nonlinear case. Lumping, for notational convenience, the active inequalities together with the equalities into $\mathbf{h}(\mathbf{x}_k)$, the projection matrix at $\mathbf{x}_k$ is

$$\mathbf{P}_k = \mathbf{I} - \nabla\mathbf{h}(\mathbf{x}_k)^T[\nabla\mathbf{h}(\mathbf{x}_k)\nabla\mathbf{h}(\mathbf{x}_k)^T]^{-1}\nabla\mathbf{h}(\mathbf{x}_k). \tag{12.25}$$

At the point $\mathbf{x}_k$ this matrix can be updated to account for one more or one less constraint, just as in the linear case. When moving from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$, however, $\nabla\mathbf{h}$ will change and the new projection matrix cannot be found from the old, and hence this matrix must be recomputed at each step.

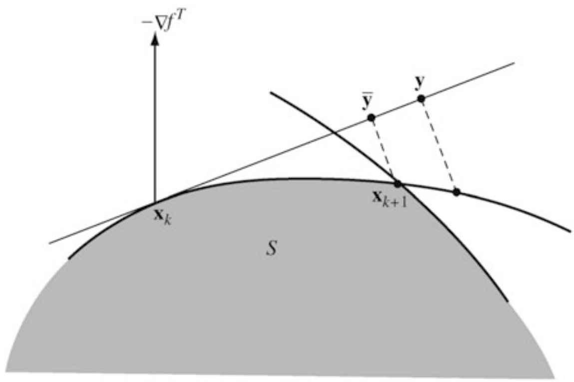

Fig. 12.5 Interpolation to obtain feasible point

The most important new feature of the method is the problem of returning to the feasible region from points outside this region. The type of iterative technique employed is a common one in nonlinear programming, including interior-point methods of linear programming, and we describe it here. The idea is, from any point near $\mathbf{x}_k$, to move back to the constraint surface in a direction orthogonal to the tangent plane at $\mathbf{x}_k$. Thus from a point $\mathbf{y}$ we seek a point of the form $\mathbf{y} + \nabla\mathbf{h}(\mathbf{x}_k)^T\boldsymbol{\alpha} = \mathbf{y}^*$ such that $\mathbf{h}(\mathbf{y}^*) = \mathbf{0}$. As shown in Fig. 12.6 such a solution may not always exist, but it does for $\mathbf{y}$ sufficiently close to $\mathbf{x}_k$.

To find a suitable first approximation to $\boldsymbol{\alpha}$, and hence to $\mathbf{y}^*$, we linearize the equation at $\mathbf{x}_k$ obtaining

$$\mathbf{h}(\mathbf{y} + \nabla\mathbf{h}(\mathbf{x}_k)^T\boldsymbol{\alpha}) \simeq \mathbf{h}(\mathbf{y}) + \nabla\mathbf{h}(\mathbf{x}_k)\nabla\mathbf{h}(\mathbf{x}_k)^T\boldsymbol{\alpha}, \tag{12.26}$$

the approximation being accurate for $|\boldsymbol{\alpha}|$ and $|\mathbf{y} - \mathbf{x}|$ small. This motivates the first approximation

$$\alpha_1 = -[\nabla \mathbf{h}(\mathbf{x}_k)\nabla \mathbf{h}(\mathbf{x}_k)^T]^{-1}\mathbf{h}(\mathbf{y}) \tag{12.27}$$

$$\mathbf{y}_1 = \mathbf{y} - \nabla \mathbf{h}(\mathbf{x}_k)^T[\nabla \mathbf{h}(\mathbf{x}_k)\nabla \mathbf{h}(\mathbf{x}_k)^T]^{-1}\mathbf{h}(\mathbf{y}). \tag{12.28}$$

Substituting $\mathbf{y}_1$ for $\mathbf{y}$ and successively repeating the process yields the sequence $\{\mathbf{y}_j\}$ generated by

$$\mathbf{y}_{j+1} = \mathbf{y}_j - \nabla \mathbf{h}(\mathbf{x}_k)^T[\nabla \mathbf{h}(\mathbf{x}_k)\nabla \mathbf{h}(\mathbf{x}_k)^T]^{-1}\mathbf{h}(\mathbf{y}_j), \tag{12.29}$$

which, started close enough to $\mathbf{x}_k$ and the constraint surface, will converge to a solution $\mathbf{y}^*$. We note that this process requires the same matrices as the projection operation.

The gradient projection method has been successfully implemented and has been found to be effective in solving general nonlinear programming problems. Successful implementation resolving the several difficulties introduced by the requirement of staying in the feasible region requires, as one would expect, some degree of skill. The true value of the method, however, can be determined only through an analysis of its rate of convergence.

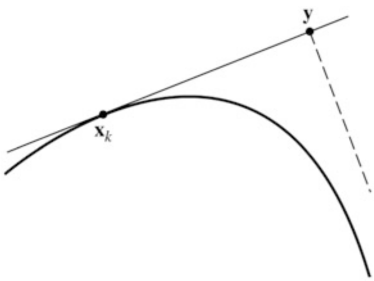

Fig. 12.6 Case in which it is impossible to return to surface

12.5 Convergence Rate of the Gradient Projection Method

An analysis that directly attacked the nonlinear version of the gradient projection method, with all of its iterative and interpolative devices, would quickly become monstrous. To obtain the asymptotic rate of convergence, however, it is not necessary to analyze this complex algorithm directly—instead it is sufficient to analyze an alternate simplified algorithm that asymptotically duplicates the gradient projection method near the solution. Through the introduction of this idealized algorithm we show that the rate of convergence of the gradient projection method is governed by the eigenvalue structure of the Hessian of the Lagrangian restricted to the constraint tangent subspace.

Geodesic Descent

For simplicity we consider first the problem having only equality constraints

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{h}(\mathbf{x}) = 0. \end{aligned} \tag{12.30}$$

The constraints define a continuous surface Ω in E^n.

In considering our own difficulties with this problem, owing to the fact that the surface is nonlinear thereby making directions of descent difficult to define, it is well to also consider the problem as it would be viewed by a small bug confined to the constraint surface who imagines it to be his total universe. To him the problem seems to be a simple one. It is unconstrained, with respect to his universe, and is only $(n - m)$-dimensional. He would characterize a solution point as a point where the gradient of f (as measured on the surface) vanishes and where the appropriate $(n - m)$-dimensional Hessian of f is positive semidefinite. If asked to develop a computational procedure for this problem, he would undoubtedly suggest, since he views the problem as unconstrained, the method of steepest descent. He would compute the gradient, as measured on his surface, and would move along what would appear to him to be straight lines.

Exactly what the bug would compute as the gradient and exactly what he would consider as straight lines would depend basically on how distance between two points on his surface were measured. If, as is most natural, we assume that he inherits his notion of distance from the one which we are using in E^n, then the path $\mathbf{x}(t)$ between two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on his surface that minimizes $\int_{x_1}^{x_2} |\dot{\mathbf{x}}(t)|dt$ would be considered a straight line by him. Such a curve, having minimum arc length between two given points, is called a *geodesic*.

Returning to our own view of the problem, we note, as we have previously, that if we project the negative gradient onto the tangent plane of the constraint surface at a point $\mathbf{x}_k$, we cannot move along this projection itself and remain feasible. We might, however, consider moving along a curve which had the same initial heading as the projected negative gradient but which remained on the surface. Exactly which such curve to move along is somewhat arbitrary, but a natural choice, inspired perhaps by the considerations of the bug, is a geodesic. Specifically, at a given point on the surface, we would determine the geodesic curve passing through that point that had an initial heading identical to that of the projected negative gradient. We would then move along this geodesic to a new point on the surface having a lesser value of f.

The idealized procedure then, which the bug would use without a second thought, and which we would use if it were computationally feasible (which it definitely is not), would at a given feasible point $\mathbf{x}_k$ (see Fig. 12.7):

1. Calculate the projection $\mathbf{p}$ of $-\nabla f(\mathbf{x}_k)^T$ onto the tangent plane at $\mathbf{x}_k$.
2. Find the geodesic, $\mathbf{x}(t)$, $t \geqslant 0$, of the constraint surface having $\mathbf{x}(0) = \mathbf{x}_k$, $\dot{\mathbf{x}}(0) = \mathbf{p}$.
3. Minimize $f(\mathbf{x}(t))$ with respect to $t \geqslant 0$, obtaining t_k and $\mathbf{x}_{k+1} = \mathbf{x}(t_k)$.

At this point we emphasize that this technique (which we refer to as geodesic descent) is proposed essentially for theoretical purposes only. It does, however, capture the main philosophy of the gradient projection method. Furthermore, as the step size of the methods go to zero, as it does near the solution point, the distance between the point that would be determined by the gradient projection method and the point found by the idealized method goes to zero even faster. Thus the asymptotic rates of convergence for the two methods will be equal, and it is, therefore, appropriate to concentrate on the idealized method only.

Our bug confined to the surface would have no hesitation in estimating the rate of convergence of this method. He would simply express it in terms of the smallest and largest eigenvalues of the Hessian of f as measured on his surface. It should not be surprising, then, that we show that the asymptotic convergence ratio is

$$\left(\frac{A-a}{A+a}\right)^2, \tag{12.31}$$

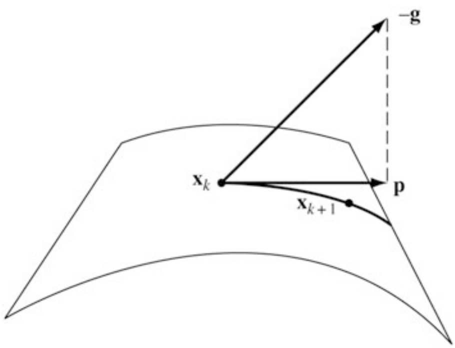

Fig. 12.7 Geodesic descent

where a and A are, respectively, the smallest and largest eigenvalues of $\mathbf{L}$, the Hessian of the Lagrangian, restricted to the tangent subspace M. This result parallels the convergence rate of the method of steepest descent, but with the eigenvalues determined from the same restricted Hessian matrix that is important in the general theory of necessary and sufficient conditions for constrained problems. This rate, which almost invariably arises when studying algorithms designed for constrained problems, will be referred to as the *canonical rate*.

We emphasize again that, since this convergence ratio governs the convergence of a large family of algorithms, it is the formula itself rather than its numerical value that is important. For any given problem we do not suggest that this ratio be evaluated, since this would be extremely difficult. Instead, the potency of the result derives from the fact that fairly comprehensive comparisons among algorithms can be made, on the basis of this formula, that apply to general classes of problems rather than simply to particular problems.

The remainder of this section is devoted to the analysis that is required to establish the convergence rate. Since this analysis is somewhat involved and not crucial for an understanding of remaining material, some readers may wish to simply read the theorem statement and proceed to the next section.

Geodesics

Given the surface $\mathbf{\Omega} = \{\mathbf{x} : \mathbf{h}(\mathbf{x}) = \mathbf{0}\} \subset E^n$, a smooth curve, $\mathbf{x}(t) \in \mathbf{\Omega}$, $0 \leqslant t \leqslant T$ starting at $\mathbf{x}(0)$ and terminating at $\mathbf{x}(T)$ that minimizes the total arc length

$$\int_0^T |\dot{\mathbf{x}}(t)|dt$$

with respect to all other such curves on $\mathbf{\Omega}$ is said to be a *geodesic* connecting $\mathbf{x}(0)$ and $\mathbf{x}(T)$.

It is common to parameterize a geodesic $\mathbf{x}(t)$, $0 \leqslant t \leqslant T$ so that $|\dot{\mathbf{x}}(t)| = 1$. The parameter t is then itself the arc length. If the parameter t is also regarded as time, then this parameterization corresponds to moving along the geodesic curve with unit velocity. Parameterized in this way, the geodesic is said to be *normalized*. On any linear subspace of E^n geodesics are straight lines. On a three-dimensional sphere, the geodesics are arcs of great circles.

It can be shown, using the calculus of variations, that any normalized geodesic on $\mathbf{\Omega}$ satisfies the condition

$$\ddot{\mathbf{x}}(t) = \nabla\mathbf{h}^T(\mathbf{x}(t))\boldsymbol{\omega}(t) \tag{12.32}$$

for some function $\boldsymbol{\omega}$ taking values in E^m. Geometrically, this condition says that if one moves along the geodesic curve with unit velocity, the acceleration at every point will be orthogonal to the surface. Indeed, this property can be regarded as the fundamental defining characteristic of a geodesic. To stay on the surface $\mathbf{\Omega}$, the geodesic must also satisfy the equation

$$\nabla\mathbf{h}(\mathbf{x}(t))\dot{\mathbf{x}}(t) = \mathbf{0}, \tag{12.33}$$

since the velocity vector at every point is tangent to $\mathbf{\Omega}$. At a regular point $\mathbf{x}_0$ these two differential equations, together with the initial conditions $\mathbf{x}(0) = \mathbf{x}_0, \dot{\mathbf{x}}(0)$ specified, and $|\dot{\mathbf{x}}(0)| = 1$, uniquely specify a curve $\mathbf{x}(t)$, $t \geqslant 0$ that can be continued as long as points on the curve are regular. Furthermore, $|\dot{\mathbf{x}}(t)| = 1$ for $t \geqslant 0$. Hence geodesic curves emanate in every direction from a regular point. Thus, for example, at any point on a sphere there is a unique great circle passing through the point in a given direction.

Lagrangian and Geodesics

Corresponding to any regular point $\mathbf{x} \in \boldsymbol{\Omega}$ we may define a corresponding Lagrange multiplier $\boldsymbol{\lambda}(\mathbf{x})$ by calculating the projection of the gradient of f onto the tangent subspace at $\mathbf{x}$, denoted $M(\mathbf{x})$. The matrix that, when operating on a vector, projects it onto $M(\mathbf{x})$ is

$$\mathbf{P}(\mathbf{x}) = \mathbf{I} - \nabla\mathbf{h}(\mathbf{x})^T[\nabla\mathbf{h}(\mathbf{x})\nabla\mathbf{h}(\mathbf{x})^T]^{-1}\nabla\mathbf{h}(\mathbf{x}),$$

and it follows immediately that the projection of $\nabla f(\mathbf{x})^T$ onto $M(\mathbf{x})$ has the form

$$\mathbf{y}(\mathbf{x}) = [\nabla f(\mathbf{x}) + \boldsymbol{\lambda}(\mathbf{x})^T\nabla\mathbf{h}(\mathbf{x})]^T, \tag{12.34}$$

where $\boldsymbol{\lambda}(\mathbf{x})$ is given explicitly as

$$\boldsymbol{\lambda}(\mathbf{x})^T = -\nabla f(\mathbf{x})\nabla\mathbf{h}(\mathbf{x})^T[\nabla\mathbf{h}(\mathbf{x})\nabla\mathbf{h}(\mathbf{x})^T]^{-1}. \tag{12.35}$$

Thus, in terms of the Lagrangian function $l(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{x})$, the projected gradient is

$$\mathbf{y}(\mathbf{x}) = l_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))^T. \tag{12.36}$$

If a local solution to the original problem occurs at a regular point $\mathbf{x}^* \in \boldsymbol{\Omega}$, then as we know

$$l_{\mathbf{x}}(\mathbf{x}^*, \boldsymbol{\lambda}(\mathbf{x}^*)) = \mathbf{0}, \tag{12.37}$$

which states that the projected gradient must vanish at $\mathbf{x}^*$. Defining $\mathbf{L}(\mathbf{x}) = l_{\mathbf{xx}}(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x})) = \mathbf{F}(\mathbf{x}) + \boldsymbol{\lambda}(\mathbf{x})^T\mathbf{H}(\mathbf{x})$ we also know that at $\mathbf{x}^*$ we have the second-order necessary condition that $\mathbf{L}(\mathbf{x}^*)$ is positive semidefinite on $M(\mathbf{x}^*)$; that is, $\mathbf{z}^T\mathbf{L}(\mathbf{x}^*)\mathbf{z} \geqslant 0$ for all $\mathbf{z} \in M(\mathbf{x}^*)$. Equivalently, letting

$$\overline{\mathbf{L}}(\mathbf{x}) = \mathbf{P}(\mathbf{x})\mathbf{L}(\mathbf{x})\mathbf{P}(\mathbf{x}), \tag{12.38}$$

it follows that $\overline{\mathbf{L}}(\mathbf{x}^*)$ is positive semidefinite.

We then have the following fundamental and simple result, valid along a geodesic.

Proposition 1. *Let* $\mathbf{x}(t)$, $0 \leqslant t \leqslant T$, *be a geodesic on* $\boldsymbol{\Omega}$. *Then*

$$\frac{d}{dt}f(\mathbf{x}(t)) = l_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))\dot{\mathbf{x}}(t) \tag{12.39}$$

$$\frac{d^2}{dt^2}f(\mathbf{x}(t)) = \dot{\mathbf{x}}(t)^T\mathbf{L}(\mathbf{x}(t))\dot{\mathbf{x}}(t). \tag{12.40}$$

Proof. We have

$$\frac{d}{dt}f(\mathbf{x}(t)) = \nabla f(\mathbf{x}(t))\dot{\mathbf{x}}(t) = l_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))\dot{\mathbf{x}}(t),$$

the second equality following from the fact that $\dot{\mathbf{x}}(t) \in M(\mathbf{x})$. Next,

$$\frac{d^2}{dt^2}f(\mathbf{x}(t)) = \dot{\mathbf{x}}(t)^T\mathbf{F}(\mathbf{x}(t))\dot{\mathbf{x}}(t) + \nabla f(\mathbf{x}(t))\ddot{\mathbf{x}}(t). \tag{12.41}$$

But differentiating the relation $\boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}(t)) = 0$ twice, for fixed $\boldsymbol{\lambda}$, yields

$$\dot{\mathbf{x}}(t)^T \boldsymbol{\lambda}^T \mathbf{H}(\mathbf{x}(t))\dot{\mathbf{x}}(t) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}(t))\ddot{\mathbf{x}}(t) = \mathbf{0}.$$

Adding this to (12.41), we have

$$\frac{d^2}{dt^2} f(\mathbf{x}(t)) = \dot{\mathbf{x}}(t)^T (\mathbf{F} + \boldsymbol{\lambda}^T \mathbf{H})\dot{\mathbf{x}}(t) + (\nabla f(\mathbf{x}) + \boldsymbol{\lambda}^T \nabla \mathbf{h}(\mathbf{x}))\ddot{\mathbf{x}}(t),$$

which is true for any fixed $\boldsymbol{\lambda}$. Setting $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{x})$ determined as above, $(\nabla \mathbf{f} + \boldsymbol{\lambda}^T \nabla \mathbf{h})^T$ is in $M(\mathbf{x})$ and hence orthogonal to $\ddot{\mathbf{x}}(t)$, since $\mathbf{x}(t)$ is a normalized geodesic. This gives (12.40). ∎

It should be noted that we proved a simplified version of this result in Chap. 11. There the result was given only for the optimal point $\mathbf{x}^*$, although it was valid for any curve. Here we have shown that essentially the same result is valid at any point provided that we move along a geodesic.

Rate of Convergence

We now prove the main theorem regarding the rate of convergence. We assume that all functions are three times continuously differentiable and that every point in a region near the solution $\mathbf{x}^*$ is regular. This theorem only establishes the rate of convergence and not convergence itself so for that reason the stated hypotheses assume that the method of geodesic descent generates a sequence $\{\mathbf{x}_k\}$ converging to $\mathbf{x}^*$.

> ***Theorem.*** *Theorem. Let $\mathbf{x}^*$ be a local solution to the problem (12.30) and suppose that A and $a > 0$ are, respectively, the largest and smallest eigenvalues of $\mathbf{L}(\boldsymbol{x}^*)$ restricted to the tangent subspace $M(\boldsymbol{x}^*)$. If $\{\boldsymbol{x}_k\}$ is a sequence generated by the method of geodesic descent that converges to $\mathbf{x}^*$, then the sequence of objective values $\{f(\boldsymbol{x}_k)\}$ converges to $f(\boldsymbol{x}^*)$ linearly with a ratio no greater than $[(A - a)/(A + a)]^2$.*

Proof. Without loss of generality we may assume $f(\mathbf{x}^*) = 0$. Given a point $\mathbf{x}_k$ it will be convenient to define its distance from the solution point $\mathbf{x}^*$ as the arc length of the geodesic connecting $\mathbf{x}^*$ and $\mathbf{x}_k$. Thus if $\mathbf{x}(t)$ is a parameterized version of the geodesic with $\mathbf{x}(0) = \mathbf{x}^*$, $|\dot{\mathbf{x}}(t)| = 1$, $\mathbf{x}(T) = \mathbf{x}_k$, then T is the distance of $\mathbf{x}_k$ from $\mathbf{x}^*$. Associated with such a geodesic we also have the family $\mathbf{y}(t)$, $0 \leqslant t \leqslant T$, of corresponding projected gradients $\mathbf{y}(t) = l_{\mathbf{x}}(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))^T$, and Hessians $\mathbf{L}(t) = \mathbf{L}(\mathbf{x}(t))$. We write $\mathbf{y}_k = \mathbf{y}(\mathbf{x}_k)$, $\mathbf{L}_k = \mathbf{L}(\mathbf{x}_k)$.

We now derive an estimate for $f(\mathbf{x}_k)$. Using the geodesic discussed above we can write (setting $\dot{\mathbf{x}}_k = \dot{\mathbf{x}}(T)$)

$$f(\mathbf{x}^*) - f(\mathbf{x}_k) = -f(\mathbf{x}_k) = -\mathbf{y}_k^T \dot{\mathbf{x}}_k T + \frac{1}{2} T^2 \dot{\mathbf{x}}_k^T \mathbf{L}_k \dot{\mathbf{x}}_k + o(T^2), \tag{12.42}$$

which follows from Proposition 1. We also have

$$\mathbf{y}_k = -\mathbf{y}(\mathbf{x}^*) + \mathbf{y}(\mathbf{x}_k) = \dot{\mathbf{y}}_k T + o(T). \tag{12.43}$$

But differentiating (12.34) we obtain

$$\dot{\mathbf{y}}_k = \mathbf{L}_k \dot{\mathbf{x}}_k + \nabla \mathbf{h}(\mathbf{x}_k)^T \dot{\boldsymbol{\lambda}}_k^T, \tag{12.44}$$

and hence if $\mathbf{P}_k$ is the projection matrix onto $M(\mathbf{x}_k) = M_k$, we have

$$\mathbf{P}_k \dot{\mathbf{y}}_k = \mathbf{P}_k \mathbf{L}_k \dot{\mathbf{x}}_k. \tag{12.45}$$

Multiplying (12.43) by $\mathbf{P}_k$ and accounting for $\mathbf{P}_k \mathbf{y}_k = \mathbf{y}_k$ we have

$$\mathbf{P}_k \dot{\mathbf{y}}_k T = \mathbf{y}_k + o(T). \tag{12.46}$$

Substituting (12.45) into this we obtain

$$\mathbf{P}_k \mathbf{L}_k \dot{\mathbf{x}}_k T = \mathbf{y}_k + o(T).$$

Since $\mathbf{P}_k \dot{\mathbf{x}}_k = \dot{\mathbf{x}}_k$ we have, defining $\overline{\mathbf{L}}_k = \mathbf{P}_k \mathbf{L}_k \mathbf{P}_k$,

$$\overline{\mathbf{L}}_k \dot{\mathbf{x}}_k T = \mathbf{y}_k + o(T). \tag{12.47}$$

The matrix $\overline{\mathbf{L}}_k$ is related to $\mathbf{L}_{M_k}$, the restriction of $\mathbf{L}_k$ to M_k, the only difference being that while $\mathbf{L}_{M_k}$ is defined only on M_k, the matrix $\overline{\mathbf{L}}_k$ is defined on all of E^n but in such a way that it agrees with $\mathbf{L}_{M_k}$ on M_k and is zero on $M_k^\perp$. The matrix $\overline{\mathbf{L}}_k$ is not invertible, but for $\mathbf{y}_k \in M_k$ there is a unique solution $\mathbf{z} \in M_k$ to the equation $\overline{\mathbf{L}}_k \mathbf{z} = \mathbf{y}_k$ which we denote† $\overline{\mathbf{L}}_k^{-1} \mathbf{y}_k$. With this notation we obtain from (12.47)

$$\dot{\mathbf{x}}_k T = \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k + o(T). \tag{12.48}$$

Substituting this last result into (12.42) and accounting for $|\mathbf{y}_k| = O(T)$ (see (12.43)) we have

$$f(\mathbf{x}_k) = \frac{1}{2} \mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k + o(T^2), \tag{12.49}$$

which expresses the objective value at $\mathbf{x}_k$ in terms of the projected gradient.

Since $|\dot{\mathbf{x}}_k| = 1$ and since $\overline{\mathbf{L}}_k \to \overline{\mathbf{L}}^*$ as $\mathbf{x}_k \to \mathbf{x}^*$, we see from (12.47) that

$$o(T) + aT \leqslant |\mathbf{y}_k| \leqslant AT + o(T), \tag{12.50}$$

which means that not only do we have $|\mathbf{y}_k| = O(T)$, which was known before, but also $|\mathbf{y}_k| \neq o(T)$. We may therefore write our estimate (12.49) in the alternate form

† Actually a more standard procedure is to define the pseudoinverse $\overline{\mathbf{L}}_k^\dagger$, and then $\mathbf{z} = \overline{\mathbf{L}}_k^\dagger \mathbf{y}_k$.

$$f(\mathbf{x}_k) = \frac{1}{2}\mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k \left(1 + \frac{o(T^2)}{\mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k} \right), \tag{12.51}$$

and since $o(T^2) \neq \mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k = O(T^2)$, we have

$$f(\mathbf{x}_k) = \frac{1}{2}\mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k (1 + O(T)), \tag{12.52}$$

which is the desired estimate.

Next, we estimate $f(\mathbf{x}_{k+1})$ in terms of $f(\mathbf{x}_k)$. Given $\mathbf{x}_k$ now let $\mathbf{x}(t)$, $t \geqslant 0$, be the normalized geodesic emanating from $\mathbf{x}_k \equiv \mathbf{x}(0)$ in the direction of the negative projected gradient, that is,

$$\dot{\mathbf{x}}(0) \equiv \dot{\mathbf{x}}_k = -\mathbf{y}_k / |\mathbf{y}_k|.$$

Then

$$f(\mathbf{x}(t)) = f(\mathbf{x}_k) + t\mathbf{y}_k^T \dot{\mathbf{x}}_k + \frac{t^2}{2}\dot{\mathbf{x}}_k^T \mathbf{L}_k \dot{\mathbf{x}}_k + o(t^2). \tag{12.53}$$

This is minimized at

$$t_k = -\frac{\mathbf{y}_k^T \dot{\mathbf{x}}_k}{\dot{\mathbf{x}}_k^T \mathbf{L}_k \dot{\mathbf{x}}_k} + o(t_k). \tag{12.54}$$

In view of (12.50) this implies that $t_k = O(T)$, $t_k \neq o(T)$. Thus t_k goes to zero at essentially the same rate as T. Thus we have

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) - \frac{1}{2}\frac{(\mathbf{y}_k^T \dot{\mathbf{x}}_k)^2}{\dot{\mathbf{x}}_k^T \mathbf{L}_k \dot{\mathbf{x}}_k} + o(T^2). \tag{12.55}$$

Using the same argument as before we can express this as

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) = \frac{1}{2}\frac{(\mathbf{y}_k^T \mathbf{y}_k)^2}{\mathbf{y}_k^T \mathbf{L}_k \mathbf{y}_k}(1 + O(T)), \tag{12.56}$$

which is the other required estimate.

Finally, dividing (12.56) by (12.52) we find

$$\frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})}{f(\mathbf{x}_k)} = \frac{(\mathbf{y}_k^T \mathbf{y}_k)^2 (1 + O(T))}{(\mathbf{y}_k^T \mathbf{L}_k \mathbf{y}_k)(\mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k)}, \tag{12.57}$$

and thus

$$f(\mathbf{x}_{k+1}) = \left[1 - \frac{(\mathbf{y}_k^T \mathbf{y}_k)^2 (1 + O(T))}{(\mathbf{y}_k^T \mathbf{L}_k \mathbf{y}_k)(\mathbf{y}_k^T \overline{\mathbf{L}}_k^{-1} \mathbf{y}_k)} \right] f(\mathbf{x}_k). \tag{12.58}$$

Using the fact that $\mathbf{L}_k \to \mathbf{L}^*$ and applying the Kantorovich inequality leads to

$$f(\mathbf{x}_{k+1}) \leqslant \left[\left(\frac{A - a}{A + a} \right)^2 + O(T) \right] f(\mathbf{x}_k). \blacksquare \tag{12.59}$$

Problems with Inequalities

The idealized version of gradient projection could easily be extended to problems having nonlinear inequalities as well as equalities by following the pattern of Sect. 12.4. Such an extension, however, has no real value, since the idealized scheme cannot be implemented. The idealized procedure was devised only as a technique for analyzing the asymptotic rate of convergence of the analytically more complex, but more practical, gradient projection method.

The analysis of the idealized version of gradient projection given above, nevertheless, does apply to problems having inequality as well as equality constraints. If a computationally feasible procedure is employed that avoids jamming and does not bounce on and off constraint boundaries an infinite number of times, then near the solution the active constraints will remain fixed. This means that near the solution the method acts just as if it were solving a problem having the active constraints as equality constraints. Thus the asymptotic rate of convergence of the gradient projection method applied to a problem with inequalities is also given by (12.59) but with $\mathbf{L}(\mathbf{x}^*)$ and $M(\mathbf{x}^*)$ (and hence a and A) determined by the active constraints at the solution point $\mathbf{x}^*$. In every case, therefore, the rate of convergence is determined by the eigenvalues of the same restricted Hessian that arises in the necessary conditions.

12.6 The Reduced Gradient Method

From a computational viewpoint, the reduced gradient method, discussed in this section and the next, is closely related to the simplex method of linear programming in that the problem variables are partitioned into basic and nonbasic groups. From a theoretical viewpoint, the method can be shown to behave very much like the gradient projection method.

Linear Constraints

Consider the problem

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geqslant \mathbf{0}, \end{aligned} \tag{12.60}$$

where $\mathbf{x} \in E^n$, $\mathbf{b} \in E^m$, $\mathbf{A}$ is $m \times n$, and f is a function in C^2. The constraints are expressed in the format of the standard form of linear programming. For simplicity of notation it is assumed that each variable is required to be non-negative—if some variables were free, the procedure (but not the notation) would be somewhat simplified.

We invoke the *nondegeneracy assumptions* that every collection of m columns from $\mathbf{A}$ is linearly independent and every basic solution to the constraints has m

strictly positive variables. With these assumptions any feasible solution will have at most $n-m$ variables taking the value zero. Given a vector $\mathbf{x}$ satisfying the constraints, we partition the variables into two groups: $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ where $\mathbf{y}$ has dimension m and $\mathbf{z}$ has dimension $n - m$. This partition is formed in such a way that all variables in $\mathbf{y}$ are strictly positive (for simplicity of notation we indicate the basic variables as being the first m components of $\mathbf{x}$ but, of course, in general this will not be so). With respect to the partition, the original problem can be expressed as

$$\text{minimize} \quad f(\mathbf{y}, \mathbf{z}) \tag{12.61a}$$

$$\text{subject to} \quad \mathbf{By} + \mathbf{Cz} = \mathbf{b} \tag{12.61b}$$

$$\mathbf{y} \geqslant \mathbf{0}, \ \mathbf{z} \geqslant \mathbf{0}, \tag{12.61c}$$

where, of course, $\mathbf{A} = [\mathbf{B}, \ \mathbf{C}]$. We can regard $\mathbf{z}$ as consisting of the independent variables and $\mathbf{y}$ the dependent variables, since if $\mathbf{z}$ is specified, (12.61b) can be uniquely solved for $\mathbf{y}$. Furthermore, a small change $\mathbf{\Delta z}$ from the original value that leaves $\mathbf{z} + \mathbf{\Delta z}$ nonnegative will, upon solution of (12.61b), yield another feasible solution, since $\mathbf{y}$ was originally taken to be strictly positive and thus $\mathbf{y} + \mathbf{\Delta y}$ will also be positive for small $\mathbf{\Delta y}$. We may therefore move from one feasible solution to another by selecting a $\mathbf{\Delta z}$ and moving $\mathbf{z}$ on the line $\mathbf{z} + \alpha\mathbf{\Delta z}$, $\alpha \geqslant 0$. Accordingly, $\mathbf{y}$ will move along a corresponding line $\mathbf{y} + \alpha\mathbf{\Delta y}$. If in moving this way some variable becomes zero, a new inequality constraint becomes active. If some independent variable becomes zero, a new direction $\mathbf{\Delta z}$ must be chosen. If a dependent (basic) variable becomes zero, the partition must be modified. The zero-valued basic variable is declared independent and one of the strictly positive independent variables is made dependent. Operationally, this interchange will be associated with a pivot operation.

The idea of the reduced gradient method is to consider, at each stage, the problem only in terms of the independent variables. Since the vector of dependent variables $\mathbf{y}$ is determined through the constraints (12.61b) from the vector of independent variables $\mathbf{z}$, the objective function can be considered to be a function of $\mathbf{z}$ only. Hence a simple modification of steepest descent, accounting for the constraints, can be executed. The gradient with respect to the independent variables $\mathbf{z}$ (the *reduced gradient*) is found by evaluating the gradient of $f(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\ \mathbf{Cz}\ \mathbf{z})$. It is equal to

$$\mathbf{r}^T = \boldsymbol{\nabla}_{\mathbf{z}} f(\mathbf{y}, \ \mathbf{z}) - \boldsymbol{\nabla}_{\mathbf{y}} f(\mathbf{y}, \ \mathbf{z})\mathbf{B}^{-1}\mathbf{C}. \tag{12.62}$$

It is easy to see that a point $(\mathbf{y}, \mathbf{z})$ satisfies the first-order necessary conditions for optimality if and only if

$$\begin{aligned} r_i &= 0 \quad \text{for all} \quad z_i > 0 \\ r_i &\geqslant 0 \quad \text{for all} \quad z_i = 0. \end{aligned}$$

In the active set form of the reduced gradient method the vector $\mathbf{z}$ is moved in the direction of the reduced gradient on the working surface. Thus at each step, a direction of the form

$$\Delta z_i = \begin{cases} -r_i, i \notin W(\mathbf{z}) \\ 0, \quad i \in W(\mathbf{z}) \end{cases}$$

is determined and a descent is made in this direction. The working set is augmented whenever a new variable reaches zero; if it is a basic variable, a new partition is also formed. If a point is found where $r_i = 0$ for all $i \notin W(\mathbf{z})$ (representing a vanishing reduced gradient on the working surface) but $r_j < 0$ for some $j \in W(\mathbf{z})$, then j is deleted from $W(\mathbf{z})$ as in the standard active set strategy.

It is possible to avoid the pure active set strategy by moving away from our active constraint whenever that would lead to an improvement, rather than waiting until an exact minimum on the working surface is found. Indeed, this type of procedure is often used in practice. One version progresses by moving the vector $\mathbf{z}$ in the direction of the overall negative reduced gradient, except that zero-valued components of $\mathbf{z}$ that would thereby become negative are held at zero. One step of the procedure is as follows:

1. Let $\mathbf{\Delta} z_i = \begin{cases} -r_i & \text{if } r_i < 0 \text{ or } z_i > 0 \\ 0 & \text{otherwise.} \end{cases}$
2. If $\mathbf{\Delta z}$ is zero, stop; the current point is a solution. Otherwise, find $\mathbf{\Delta y} = -\mathbf{B}^{-1}\mathbf{C}\Delta\mathbf{z}$.
3. Find $\alpha_1,\ \alpha_2,\ \alpha_3$ achieving, respectively,

$$\max\{\alpha : \mathbf{y} + \alpha\mathbf{\Delta y} \geqslant 0\}$$
$$\max\{\alpha : \mathbf{z} + \alpha\mathbf{\Delta z} \geqslant 0\}$$
$$\min\{f(\mathbf{x} + \alpha\mathbf{\Delta x}) : 0 \leqslant \alpha \leqslant \alpha_1,\ 0 \leqslant \alpha \leqslant \alpha_2\}$$

   Let $\overline{\mathbf{x}} = \mathbf{x} + \alpha_3\mathbf{\Delta x}$.
4. If $\alpha_3 < \alpha_1$, return to (12.1). Otherwise, declare the vanishing variable in the dependent set independent and declare a strictly positive variable in the independent set dependent. Update $\mathbf{B}$ and $\mathbf{C}$.

Example. We consider the example presented in Sect. 12.4 where the projected negative gradient was computed:

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \text{subject to} & 2x_1 + x_2 + x_3 + 4x_4 = 7 \\ & x_1 + x_2 + 2x_3 + x_4 = 6 \\ & x_i \geqslant 0,\ i = 1, 2, 3, 4. \end{array}$$

We are given the feasible point $\mathbf{x} = (2, 2, 1, 0)$. We may select any two of the strictly positive variables to be the basic variables. Suppose $\mathbf{y} = (x_1,\ x_2)$ is selected. In standard form the constraints are then

$$x_1 + 0 - x_3 + 3x_4 = 1$$
$$0 + x_2 + 3x_3 - 2x_4 = 5$$
$$x_i \geqslant 0, \quad i = 1, 2, 3, 4.$$

The gradient at the current point is $\mathbf{g} = (2, 4, 2, -3)$. The corresponding reduced gradient (with respect to $\mathbf{z} = (x_3, x_4)$) is then found by *pricing-out* in the usual manner. The situation at the current point can then be summarized by the tableau

Variable	x_1	x_2	x_3	x_4	
Constraints	1	0	−1	3	1
	0	1	3	−2	5
$\mathbf{r}^T$	0	0	−8	−1	
Current value	2	2	1	0	

Tableau for Example

In this solution x_3 and x_4 would be increased together in a ratio of eight to one. As they increase, x_1 and x_2 would follow in such a way as to keep the constraints satisfied. Overall, in E^4, the implied direction of movement is thus

$$\mathbf{d} = (5, -22, 8, 1).$$

If the reader carefully supplies the computational details not shown in the presentation of the example as worked here and in Sect. 12.4, he will undoubtedly develop a considerable appreciation for the relative simplicity of the reduced gradient method.

It should be clear that the reduced gradient method can, as illustrated in the example above, be executed with the aid of a tableau. At each step the tableau of constraints is arranged so that an identity matrix appears over the m dependent variables, and thus the dependent variables can be easily calculated from the values of the independent variables. The reduced gradient at any step is calculated by evaluating the n-dimensional gradient and "pricing out" the dependent variables just as the reduced cost vector is calculated in linear programming. And when the partition of basic and non-basic variables must be changed, a simple pivot operation is all that is required.

Global Convergence

The perceptive reader will note the direction finding algorithm that results from the second form of the reduced gradient method is not closed, since slight movement away from the boundary of an inequality constraint can cause a sudden change in the direction of search. Thus one might suspect, and correctly so, that this method is subject to jamming. However, a trivial modification will yield a closed mapping; and hence global convergence. This is discussed in Exercise 19.

Nonlinear Constraints

The *generalized reduced gradient method* solves nonlinear programming problems in the *standard form*

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \ \mathbf{a} \leqslant \mathbf{x} \leqslant \mathbf{b}, \end{aligned}$$

where $\mathbf{h}(\mathbf{x})$ is of dimension m. A general nonlinear programming problem can always be expressed in this form by the introduction of slack variables, if required, and by allowing some components of $\mathbf{a}$ and $\mathbf{b}$ to take on the values $+\infty$ or $-\infty$, if necessary.

In a manner quite analogous to that of the case of linear constraints, we introduce a *nondegeneracy* assumption that, at each point $\mathbf{x}$, hypothesizes the existence of a partition of $\mathbf{x}$ into $\mathbf{x} = (\mathbf{y},\ \mathbf{z})$ having the following properties:

(i) $\mathbf{y}$ is of dimension m, and $\mathbf{z}$ is of dimension $n - m$.
(ii) If $\mathbf{a} = (\mathbf{a}_{\mathrm{y}},\ \mathbf{a}_{\mathrm{z}})$ and $\mathbf{b} = (\mathbf{b}_{\mathrm{y}},\ \mathbf{b}_{\mathrm{z}})$ are the corresponding partitions of $\mathbf{a}$, $\mathbf{b}$, then $\mathbf{a}_{\mathrm{y}} < \mathbf{y} < \mathbf{b}_{\mathrm{y}}$.
(iii) The $m \times m$ matrix $\nabla_{\mathbf{y}}\mathbf{h}(\mathbf{y},\ \mathbf{z})$ is nonsingular at $\mathbf{x} = (\mathbf{y},\ \mathbf{z})$.

Again $\mathbf{y}$ and $\mathbf{z}$ are referred to as the vectors of *dependent* and *independent variables*, respectively.

The reduced gradient (with respect to $\mathbf{z}$) is in this case:

$$\mathbf{r}^T = \nabla_{\mathbf{z}} f(\mathbf{y},\ \mathbf{z}) + \boldsymbol{\lambda}^T \nabla_{\mathrm{z}} \mathbf{h}(\mathbf{y},\ \mathbf{z}),$$

where $\boldsymbol{\lambda}$ satisfies

$$\nabla_{\mathbf{y}} f(\mathbf{y},\ \mathbf{z}) + \boldsymbol{\lambda}^T \nabla_{\mathrm{y}} \mathbf{h}(\mathbf{y},\ \mathbf{z}) = \mathbf{0}.$$

Equivalently, we have

$$\mathbf{r}^T = \nabla_{\mathbf{z}} f(\mathbf{y},\ \mathbf{z}) - \nabla_{\mathbf{y}} f(\mathbf{y},\ \mathbf{z})[\nabla_{\mathbf{y}} \mathbf{h}(\mathbf{y},\ \mathbf{z})]^{-1} \nabla_{\mathbf{z}} \mathbf{h}(\mathbf{y},\ \mathbf{z}). \tag{12.63}$$

The actual procedure is roughly the same as for linear constraints in that moves are taken by changing $\mathbf{z}$ in the direction of the negative reduced gradient (with components of $\mathbf{z}$ on their boundary held fixed if the movement would violate the bound). The difference here is that although $\mathbf{z}$ moves along a straight line as before, the vector of dependent variables $\mathbf{y}$ must move nonlinearly to continuously satisfy the equality constraints. Computationally, this is accomplished by first moving linearly along the tangent to the surface defined by $\mathbf{z} \to \mathbf{z} + \boldsymbol{\Delta}\mathbf{z},\ \mathbf{y} \to \mathbf{y} + \boldsymbol{\Delta}\mathbf{y}$ with $\boldsymbol{\Delta}\mathbf{y} = -[\nabla_{\mathbf{y}}\mathbf{h}]^{-1}\nabla_{\mathbf{z}}\mathbf{h}\boldsymbol{\Delta}\mathbf{z}$. Then a correction procedure, much like that employed in the gradient projection method, is used to return to the constraint surface and the magnitude bounds on the dependent variables are checked for feasibility. As with the gradient projection method, a feasibility tolerance must be introduced to acknowledge the impossibility of returning exactly to the constraint surface. An example corresponding to $n = 3,\ m = 1,\ a = 0,\ b = +\infty$ is shown in Fig. 12.8.

To return to the surface once a tentative move along the tangent is made, an iterative scheme is employed. If the point $\mathbf{x}_k$ was the point at the previous step, then from any point $\mathbf{x} = (\mathbf{v},\ \mathbf{w})$ near $\mathbf{x}_k$ one gets back to the constraint surface by solving the nonlinear equation

$$\mathbf{h}(\mathbf{y}, \mathbf{w}) = \mathbf{0} \tag{12.64}$$

for $\mathbf{y}$ (with $\mathbf{w}$ fixed). This is accomplished through the iterative process

$$\mathbf{y}_{j+1} = \mathbf{y}_j - [\boldsymbol{\nabla}_{\mathrm{y}}\mathbf{h}(\mathbf{x}_k)]^{-1}\mathbf{h}(\mathbf{y}_j,\ \mathbf{w}), \tag{12.65}$$

which, if started close enough to $\mathbf{x}_k$, will produce $\{\mathbf{y}_j\}$ with $\mathbf{y}_j \to \mathbf{y}$, solving (12.64).

The reduced gradient method suffers from the same basic difficulties as the gradient projection method, but as with the latter method, these difficulties can all be more or less successfully resolved. Computation is somewhat less complex in the case of the reduced gradient method, because rather than compute with $[\boldsymbol{\nabla}\mathbf{h}(\mathbf{x})\boldsymbol{\nabla}\mathbf{h}(\mathbf{x})^T]^{-1}$ at each step, the matrix $[\boldsymbol{\nabla}_{\mathbf{y}}\mathbf{h}(\mathbf{y},\ \mathbf{z})]^{-1}$ is used.

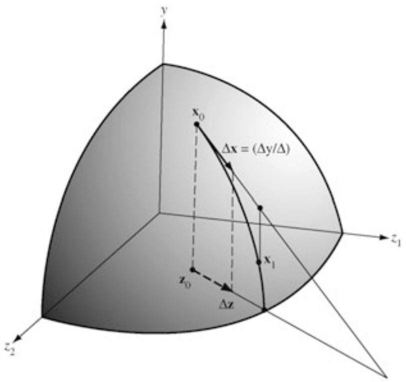

Fig. 12.8 Reduced gradient method

12.7 Convergence Rate of the Reduced Gradient Method

As argued before, for purposes of analyzing the rate of convergence, it is sufficient to consider the problem having only equality constraints

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}. \end{aligned} \tag{12.66}$$

We then regard the problem as being defined over a surface $\boldsymbol{\Omega}$ of dimension $n - m$. At this point it is again timely to consider the view of our bug, who lives on this constraint surface. Invariably, he continues to regard the problem as extremely elementary, and indeed would have little appreciation for the complexity that seems to face us. To him the problem is an unconstrained problem in $n-m$ dimensions not, as we see it, a constrained problem in n dimensions. The bug will tenaciously hold to the method of steepest descent. We can emulate him provided that we know how he measures distance on his surface and thus how he calculates gradients and what he considers to be straight lines.

Rather than imagine that the measure of distance on his surface is the one that would be inherited from us in n dimensions, as we did when studying the gradient projection method, we, in this instance, follow the construction shown in Fig. 12.9. In our n-dimensional space, $n - m$ coordinates are selected as independent variables in such a way that, given their values, the values of the remaining (dependent) variables are determined by the surface. There is already a coordinate system in the space of independent variables, and it can be used on the surface by projecting it parallel to the space of the remaining dependent variables. Thus, an arc on the surface is considered to be straight if its projection onto the space of independent variables is a segment of a straight line. With this method for inducing a geometry on the surface, the bug's notion of steepest descent exactly coincides with an idealized version of the reduced gradient method.

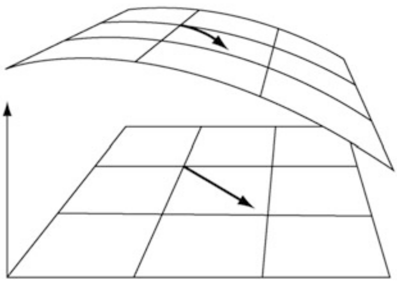

Fig. 12.9 Induced coordinate system

In the idealized version of the reduced gradient method for solving (12.66), the vector $\mathbf{x}$ is partitioned as $\mathbf{x} = (\mathbf{y},\ \mathbf{z})$ where $\mathbf{y} \in E^m,\ \mathbf{z} \in E^{n-m}$. It is assumed that the $m \times m$ matrix $\boldsymbol{\nabla}_{\mathbf{y}}\mathbf{h}(\mathbf{y}, \mathbf{z})$ is nonsingular throughout a given region of interest. (With respect to the more general problem, this region is a small neighborhood around the solution where it is not necessary to change the partition.) The vector $\mathbf{y}$ is regarded as an implicit function of $\mathbf{z}$ through the equation

$$\mathbf{h}(\mathbf{y}(\mathbf{z}),\ \mathbf{z}) = \mathbf{0}. \tag{12.67}$$

The ordinary method of steepest descent is then applied to the function $q(\mathbf{z}) = f(\mathbf{y}(\mathbf{z}),\ \mathbf{z})$. We note that the gradient $\mathbf{r}^T$ of this function is given by (12.63).

Since the method is really just the ordinary method of steepest descent with respect to $\mathbf{z}$, the rate of convergence is determined by the eigenvalues of the Hessian of the function q at the solution. We therefore turn to the question of evaluating this Hessian.

Denote by $\mathbf{Y}(\mathbf{z})$ the first derivatives of the implicit function $\mathbf{y}(\mathbf{z})$, that is, $\mathbf{Y}(\mathbf{z}) \equiv \nabla_{\mathbf{z}}\mathbf{y}(\mathbf{z})$. Explicitly,

$$\mathbf{Y}(\mathbf{z}) = -[\nabla_{\mathbf{y}}\mathbf{h}(\mathbf{y},\ \mathbf{z})]^{-1}\nabla_{\mathbf{z}}\mathbf{h}(\mathbf{y},\ \mathbf{z}). \tag{12.68}$$

For any $\boldsymbol{\lambda} \in E^m$ we have

$$q(\mathbf{z}) = f(\mathbf{y}(\mathbf{z}),\ \mathbf{z}) = f(\mathbf{y}(\mathbf{z}),\ \mathbf{z}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{y}(\mathbf{z}),\ \mathbf{z}). \tag{12.69}$$

Thus

$$\nabla q(\mathbf{z}) = [\nabla_{\mathbf{y}} f(\mathbf{y},\ \mathbf{z}) + \boldsymbol{\lambda}^T\nabla_{\mathbf{y}}\mathbf{h}(\mathbf{y},\ \mathbf{z})]\mathbf{Y}(\mathbf{z}) + \nabla_{\mathbf{z}} f(\mathbf{y},\ \mathbf{z}) + \boldsymbol{\lambda}^T\nabla_{\mathbf{z}}\mathbf{h}(\mathbf{y},\ \mathbf{z}). \tag{12.70}$$

Now if at a given point $\mathbf{x}^* = (\mathbf{y}^*,\ \mathbf{z}^*) = (\mathbf{y}(\mathbf{z}^*),\ \mathbf{z}^*)$, we let $\boldsymbol{\lambda}$ satisfy

$$\nabla_{\mathbf{y}} f(\mathbf{y}^*,\ \mathbf{z}^*) + \boldsymbol{\lambda}^T\nabla_{\mathbf{y}}\mathbf{h}(\mathbf{y}^*,\ \mathbf{z}^*) = \mathbf{0}; \tag{12.71}$$

then introducing the Lagrangian $l(\mathbf{y},\ \mathbf{z},\ \boldsymbol{\lambda}) = f(\mathbf{y},\ \mathbf{z}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{y},\ \mathbf{z})$, we obtain by differentiating (12.70)

$$\begin{aligned}\nabla^2 q(\mathbf{z}^*) = \mathbf{Y}(\mathbf{z}^*)^T\nabla^2_{\mathbf{yy}} l(\mathbf{y}^*,\ \mathbf{z}^*)\mathbf{Y}(\mathbf{z}^*) + \nabla^2_{\mathbf{zy}} l(\mathbf{y}^*,\ \mathbf{z}^*)\mathbf{Y}(\mathbf{z}^*)\\ +\mathbf{Y}(\mathbf{z}^*)^T\nabla^2_{yz} l(\mathbf{y}^*,\ \mathbf{z}^*) + \nabla^2_{zz} l(\mathbf{y}^*,\ \mathbf{z}^*).\end{aligned} \tag{12.72}$$

Or defining the $n \times (n-m)$ matrix

$$\mathbf{C} = \left[\frac{\mathbf{Y}(\mathbf{z}^*)}{\mathbf{I}}\right], \tag{12.73}$$

where $\mathbf{I}$ is the $(n-m) \times (n-m)$ identity, we have

$$\mathbf{Q} \equiv \nabla^2 q(\mathbf{z}^*) = \mathbf{C}^T\mathbf{L}(\mathbf{x}^*)\mathbf{C}. \tag{12.74}$$

The matrix $\mathbf{L}(\mathbf{x}^*)$ is the $n \times n$ Hessian of the Lagrangian at $\mathbf{x}^*$, and $\nabla^2 q(\mathbf{z}^*)$ is an $(n-m) \times (n-m)$ matrix that is a restriction of $\mathbf{L}(\mathbf{x}^*)$ to the tangent subspace M, but it is not the usual restriction. We summarize our conclusion with the following theorem.

Theorem. *Let $\mathbf{x}^*$ be a local solution of problem (12.66). Suppose that the idealized reduced gradient method produces a sequence $\{\boldsymbol{x}_k\}$ converging to $\mathbf{x}^*$ and that the partition $\mathbf{x} = (\boldsymbol{y},\ \boldsymbol{z})$ is used throughout the tail of the sequence. Let $\mathbf{L}$ be the Hessian of the Lagrangian at $\mathbf{x}^*$ and define the matrix $\boldsymbol{C}$ by (12.73) and (12.68). Then the sequence of objective values $\{f(\boldsymbol{x}_k)\}$ converges to $f(\boldsymbol{x}^*)$ linearly with a ratio no greater than $[(B-b)/(B+b)]^2$ where b and B are, respectively, the smallest and largest eigenvalues of the matrix $\mathbf{Q} = \boldsymbol{C}^T\boldsymbol{LC}$.*

To compare the matrix $\mathbf{C}^T\mathbf{LC}$ with the usual restriction of $\mathbf{L}$ to M that determines the convergence rate of most methods, we note that the $n \times (n-m)$ matrix $\mathbf{C}$ maps $\Delta\mathbf{z} \in E^{n-m}$ into $(\Delta\mathbf{y},\ \Delta\mathbf{z}) \in E^n$ lying in the tangent subspace M; that is, $\nabla_{\mathbf{y}}\mathbf{h}\Delta\mathbf{y} + \nabla_{\mathbf{z}}\mathbf{h}\Delta\mathbf{z} = \mathbf{0}$. Thus the columns of $\mathbf{C}$ form a basis for the subspace M. Next note that the columns of the matrix

$$\mathbf{E} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1/2} \tag{12.75}$$

form an orthonormal basis for M, since each column of $\mathbf{E}$ is just a linear combination of columns of $\mathbf{C}$ and by direct calculation we see that $\mathbf{E}^T\mathbf{E} = \mathbf{I}$. Thus by the procedure described in Sect. 11.6 we see that a representation for the usual restriction of $\mathbf{L}$ to M is

$$\mathbf{L}_M = (\mathbf{C}^T\mathbf{C})^{-1/2}\mathbf{C}^T\mathbf{L}\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1/2}. \tag{12.76}$$

Comparing (12.76) with (12.74) we deduce that

$$\mathbf{Q} = (\mathbf{C}^T\mathbf{C})^{1/2}\mathbf{L}_M(\mathbf{C}^T\mathbf{C})^{1/2}. \tag{12.77}$$

This means that the Hessian matrix for the reduced gradient method is the restriction of $\mathbf{L}$ to M but pre- and post-multiplied by a positive definite symmetric matrix.

The eigenvalues of $\mathbf{Q}$ depend on the exact nature of $\mathbf{C}$ as well as $\mathbf{L}_M$. Thus, the rate of convergence of the reduced gradient method is not coordinate independent but depends strongly on just which variables are declared as independent at the final stage of the process. The convergence rate can be either faster or slower than that of the gradient projection method. In general, however, if $\mathbf{C}$ is well-behaved (that is, well-conditioned), the ratio of eigenvalues for the reduced gradient method can be expected to be the same order of magnitude as that of the gradient projection method. If, however, $\mathbf{C}$ should be ill-conditioned, as would arise in the case where the implicit equation $\mathbf{h}(\mathbf{y}, \mathbf{z}) = \mathbf{0}$ is itself ill-conditioned, then it can be shown that the eigenvalue ratio for the reduced gradient method will most likely be considerably worsened. This suggests that care should be taken to select a set of basic variables $\mathbf{y}$ that leads to a well-behaved $\mathbf{C}$ matrix.

Example (The Hanging Chain Problem). Consider again the hanging chain problem discussed in Sect. 11.4. This problem can be used to illustrate a wide assortment of theoretical principles and practical techniques. Indeed, a study of this example clearly reveals the predictive power that can be derived from an interplay of theory and physical intuition.

The problem is

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n}(n - i + 0.5)y_i \\
\text{subject to} \quad & \sum_{i=1}^{n} y_i = 0 \\
& \sum_{i=1}^{n} \sqrt{1 - y_i^2} = 16,
\end{aligned}$$

where in the original formulation $n = 20$.

This problem has been solved numerically by the reduced gradient method.* An initial feasible solution was the triangular shape shown in Fig. 12.10a with

* The exact solution is obviously symmetric about the center of the chain, and hence the problem could be reduced to having ten links and only one constraint. However, this symmetry disappears if the first constraint value is specified as nonzero. Therefore for generality we solve the full chain problem.

$$y_i = \begin{cases} -0.6, & 1 \leqslant i \leqslant 10 \\ 0.6, & 11 \leqslant i \leqslant 20. \end{cases}$$

Table 12.1 Results of original chain problem

Iteration	Value	Solution (1/2 of chain)
0	−60.00000	$y_1 = -0.8148260$
10	−66.47610	$y_2 = -0.7826505$
20	−66.52180	$y_3 = -0.7429208$
30	−66.53595	$y_4 = -0.6930959$
40	−66.54154	$y_5 = -0.6310976$
50	−66.54537	$y_6 = -0.5541078$
60	−66.54628	$y_7 = -0.4597160$
69	−66.54659	$y_8 = -0.3468334$
70	−66.54659	$y_9 = -0.2169879$
		$y_{10} = -0.07492541$
Lagrange multipliers −9.993817, −6.763148		

The results obtained from a reduced gradient package are shown in Table 12.1. Note that convergence is obtained in approximately 70 iterations.

The Lagrange multipliers of the constraints are a by-product of the solution. These can be used to estimate the change in solution value if the constraint values are changed slightly. For example, suppose we wish to estimate, without resolving the problem, the change in potential energy (the objective function) that would result if the separation between the two supports were increased by, say, one inch. The change can be estimated by the formula $\Delta_u = -\lambda_2/12 = 0.0833 \times (6.76) = 0.563$. (When solved again numerically the change is found to be 0.568.)

Let us now pose some more challenging questions. Consider two variations of the original problem. In the first variation the chain is replaced by one having twice as many links, but each link is now half the size of the original links. The overall chain length is therefore the same as before. In the second variation the original chain is replaced by one having twice as many links, but each link is the same size as the original links. The chain length doubles in this case. If these problems are solved by the same method as the original problem, approximately how many iterations will be required—about the same number, many more, or substantially less?

These questions can be easily answered by using the theory of convergence rates developed in this chapter. The Hessian of the Lagrangian is

$$\mathbf{L} = \mathbf{F} + \lambda_1 \mathbf{H}_1 + \lambda_2 \mathbf{H}_2.$$

However, since the objective function and the first constraint are both linear, the only nonzero term in the above equation is $\lambda_2 \mathbf{H}_2$. Furthermore, since convergence rates depend only on eigenvalue ratios, the λ_2 can be ignored. Thus the eigenvalues of $\mathbf{H}_2$ determine the canonical convergence rate.

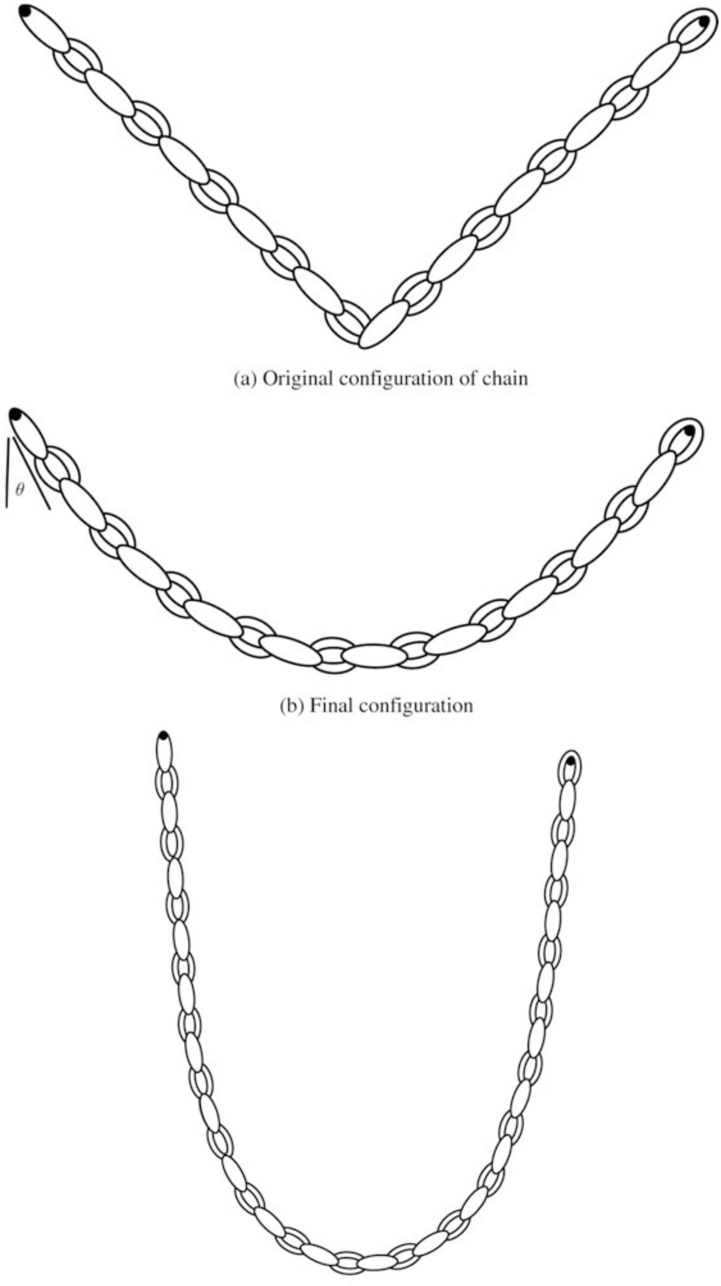

Fig. 12.10 The chain example. (**a**) Original configuration of chain. (**b**) Final configuration. (**c**) Long chain

It is easily seen that $\mathbf{H}_2$ is diagonal with ith diagonal term,

$$(\mathbf{H}_2)_{ii} = -(1 - y_i^2)^{-3/2},$$

and these values are the eigenvalues of $\mathbf{H}_2$. The canonical convergence rate is defined by the eigenvalues of $\mathbf{H}_{22}$ in the $(n-2)$-dimensional tangent subspace M. We cannot exactly determine these eigenvalues without a lot of work, but we can assume that they are close to the eigenvalues of $\mathbf{H}_{22}$. (Indeed, a version of the Interlocking Eigenvalues Lemma states that the $n-2$ eigenvalues are interlocked with the eigenvalues of $\mathbf{H}_{22}$.) Then the convergence rate of the gradient projection method will be governed by these eigenvalues. The reduced gradient method will most likely be somewhat slower.

The eigenvalue of smallest absolute value corresponds to the center links, where $y_i \simeq 0$. Conversely, the eigenvalue of largest absolute value corresponds to the first or last link, where y_i is largest in absolute value. Thus the relevant eigenvalue ratio is approximately

$$r = \frac{1}{(1 - y_1^2)^{3/2}} = \frac{1}{(\sin\theta)^{3/2}},$$

where θ is the angle shown in Fig. 12.10b.

For very little effort we have obtained a powerful understanding of the chain problem and its convergence properties. We can use this to answer the questions posed earlier. For the first variation, with twice as many links but each of half size, the angle θ will be about the same (perhaps a little smaller because of increased flexibility of the chain). Thus the number of iterations should be slightly larger because of the increase in θ and somewhat larger again because there are more variables (which tends to increase the condition number of $\mathbf{C}^T\mathbf{C}$). Note in Table 12.2 that about 122 iterations were required, which is consistent with this estimate.

For the second variation the chain will hang more vertically; hence y_1 will be larger, and therefore convergence will be fundamentally slower. To be more specific it is necessary to substitute a few numbers in our simple formula. For the original case we have $y_1 \simeq -.81$. This yields

$$r = (1 - .81^2)^{-3/2} = 4.9$$

and a convergence factor of

$$R = \left(\frac{r-1}{r+1}\right)^2 \simeq .44.$$

This is a modest value and quite consistent with the observed result of 70 iterations for a reduced gradient method. For the long chain we can estimate that $y_1 \simeq 98$. This yields

Table 12.2 Results of modified chain problems

Short links		Long chain	
Iteration	Value	Iteration	Value
0	−60.00000	0	−366.6061
10	−66.45499	10	−375.6423
20	−66.56377	20	−375.9123
40	−66.58443	50	−376.5128
60	−66.59191	100	−377.1625
80	−66.59514	200	−377.8983
100	−66.59656	500	−378.7989
120	−66.59825	1000	−379.3012
121	−66.59827	1500	−379.4994
122	−66.59827	2000	−379.5965
		2500	−379.6489
$y_1 = 0.4109519$		$y_1 = 0.9886223$	

$$r = (1 - .98^2)^{-3/2} \simeq 127$$
$$R = \left(\frac{r-1}{r+1}\right)^2 \simeq .969.$$

This last number represents extremely slow convergence. Indeed, since $(0.969)^{25} \simeq 0.44$, we expect that it may easily take 25 times as many iterations for the long chain problem to converge as the original problem (although quantitative estimates of this type are rough at best). This again is verified by the results shown in Table 12.2, where it is indicated that over 2,500 iterations were required by a version of the reduced gradient method.

*12.8 *Variations

It is possible to modify either the gradient projection method or the reduced gradient method so as to move in directions that are determined through additional considerations. For example, analogs of the conjugate gradient method, **PARTAN**, or any of the quasi-Newton methods can be applied to constrained problems by handling constraints through projection or reduction. The corresponding asymptotic rates of convergence for such methods are easily determined by applying the results for unconstrained problems on the $(n-m)$-dimensional surface of constraints, as illustrated in this chapter.

Although such generalizations can sometimes lead to substantial improvement in convergence rates, one must recognize that the detailed logic for a complicated generalization can become lengthy. If the method relies on the use of an approximate inverse Hessian restricted to the constraint surface, there must be an effective procedure for updating the approximation when the iterative process progresses from one set of active constraints to another. One would also like to insure that the poor eigenvalue structure sometimes associated with quasi-Newton methods does not dominate the short-term convergence characteristics of the extended method when the active constraint set changes. In other words, one would like to be able to achieve simultaneously both superlinear convergence and a guarantee of fast single step progress. There has been some work in this general area and it appears to be one of potential promise.

**Convex Simplex Method*

A popular modification of the reduced gradient method, termed the *convex simplex method*, most closely parallels the highly effective simplex method for solving linear programs. The major difference between this method and the reduced gradient method is that instead of moving all (or several) of the independent variables in the direction of the negative reduced gradient, only one independent variable is changed at a time. The selection of the one independent variable to change is made much as in the ordinary simplex method.

At a given feasible point, let $\mathbf{x} = (\mathbf{y},\ \mathbf{z})$ be the partition of $\mathbf{x}$ into dependent and independent parts, and assume for simplicity that the bounds on $\mathbf{x}$ are $\mathbf{x} \geqslant \mathbf{0}$. Given the reduced gradient $\mathbf{r}^T$ at the current point, the component z_i to be changed is found from:

1. Let $r_{i1} = \min_i \{r_i\}$.
2. Let $r_{i2}z_{i2} = \max_i \{r_i z_i\}$

   If $\quad r_{i1} = r_{i2}z_{i2} = 0$, terminate. Otherwise

   If $\quad r_{i1} \leqslant -|r_{i2}z_{i2}|$, increase z_{i1}

   If $\quad r_{i1} \geqslant -|r_{i2}z_{i2}|$, decrease z_{i2}.

The rule in Step 2 amounts to selecting the variable that yields the best potential decrease in the cost function. The rule accounts for the non-negativity constraint on the independent variables by weighting the cost coefficients of those variables that are candidates to be decreased by their distance from zero. This feature ensures global convergence of the method.

The remaining details of the method are identical to those of the reduced gradient method. Once a particular component of $\mathbf{z}$ is selected for change, according to the above criterion, the corresponding $\mathbf{y}$ vector is computed as a function of the change in that component so as to continuously satisfy the constraints. The component of $\mathbf{z}$ is continuously changed until either a local minimum with respect to that component is attained or the boundary of one nonnegativity constraint is reached.

Just as in the discussion of the reduced gradient method, it is convenient, for purposes of convergence analysis, to view the problem as unconstrained with respect to the independent variables. The convex simplex method is then seen to be a coordinate descent procedure in the space of these $n - m$ variables. Indeed, since the component selected is based on the magnitude of the components of the reduced gradient, the method is merely an adaptation of the Gauss-Southwell scheme discussed in Sect. 8.6 to the constrained situation. Hence, although it is difficult to pin down precisely, we expect that it would take approximately $n - m$ steps of this coordinate descent method to make the progress of a single reduced gradient step. To be competitive with the reduced gradient method; therefore, the difficulties associated with a single step—line searching and constraint evaluation—must be approximately $n - m$ times simpler when only a single component is varied than when all $n - m$ are varied simultaneously. This is indeed the case for linear programs and for some quadratic programs but not for nonlinear problems that require the full line search machinery. Hence, in general, the convex simplex method may not be a bargain.

12.9 Summary

The concept of feasible direction methods is a straightforward and logical extension of the methods used for unconstrained problems but leads to some subtle difficulties. These methods are susceptible to *jamming* (lack of global convergence) because many simple direction finding mappings and the usual line search mapping are not closed.

Problems with inequality constraints can be approached with an active set strategy. In this approach certain constraints are treated as active and the others are treated as inactive. By systematically adding and dropping constraints from the working set, the correct set of active constraints is determined during the search process. In general, however, an active set method may require that several constrained problems be solved exactly.

The most practical primal methods are the gradient projection methods and the reduced gradient method. Both of these basic methods can be regarded as the method of steepest descent applied on the surface defined by the active constraints. The rate of convergence for the two methods can be expected to be approximately equal and is determined by the eigenvalues of the Hessian of the Lagrangian restricted to the subspace tangent to the active constraints. Of the two methods, the reduced gradient method seems to be best. It can be easily modified to ensure against jamming and it requires fewer computations per iterative step and therefore, for most problems, will probably converge in less time than the gradient projection method.

12.10 Exercises

1. Show that the Frank-Wolfe method is globally convergent if the intersection of the feasible region and the objective level set $\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$ is bounded.
2. Sometimes a different normalizing term is used in (12.4). Show that the problem of finding $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ to

$$\begin{aligned}&\text{minimize} \quad \mathbf{c}^T\mathbf{d}\\&\text{subject to} \quad \mathbf{Ad} \leqslant \mathbf{0}, \ (\sum_i |d_i|^p)^{1/p} = 1\end{aligned}$$

for $p = 1$ or $p = \infty$ can be converted to a linear program.

3. Perhaps the most natural normalizing term to use in (12.4) is one based on the Euclidean norm. This leads to the problem of finding $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ to

$$\begin{aligned}&\text{minimize} \quad \mathbf{c}^T\mathbf{d}\\&\text{subject to} \quad \mathbf{Ad} \leqslant \mathbf{0}, \ \sum_{i=1}^n d_i^2 = 1.\end{aligned}$$

Find the Karush-Kuhn–Tucker necessary conditions for this problem and show how they can be solved by a modification of the simplex procedure.

4. Let $\Omega \subset E^n$ be a given feasible region. A set $\Gamma \subset E^{2n}$ consisting of pairs $(\mathbf{x}, \mathbf{d})$, with $\mathbf{x} \in \Omega$ and $\mathbf{d}$ a feasible direction at $\mathbf{x}$, is said to be a set of *uniformly feasible direction vectors* if there is a $\delta > 0$ such that $(\mathbf{x}, \mathbf{d}) \in \boldsymbol{\Gamma}$ implies that $\mathbf{x} + \alpha\mathbf{d}$ is feasible for all α, $0 \leqslant \alpha \leqslant \delta$. The number δ is referred to as the feasibility constant of the set $\boldsymbol{\Gamma}$.
Let $\Gamma \subset E^{2n}$ be a set of uniformly feasible direction vectors for Ω, with feasibility constant δ. Define the mapping

$$\begin{aligned}\mathbf{M}_\delta(\mathbf{x}, \mathbf{d}) = \{\mathbf{y} : f(\mathbf{y}) \leqslant f(\mathbf{x} + \tau\mathbf{d}) \text{ for all } \tau,\ 0 \leqslant \tau \leqslant \delta; \mathbf{y} = \mathbf{x} + \alpha\mathbf{d},\\ \text{for some } \alpha,\ 0 \leqslant \alpha \leqslant \infty,\ \mathbf{y} \in \Omega\}.\end{aligned}$$

Show that if $\mathbf{d} \neq \mathbf{0}$, the map $\mathbf{M}_\delta$ is closed at $(\mathbf{x}, \mathbf{d})$.

5. Let $\Gamma \subset E^{2n}$ be a set of uniformly feasible direction vectors for Ω with feasibility constant δ. For $\varepsilon > 0$ define the map ${}^\varepsilon\mathbf{M}_\delta$ or Γ by

$$\begin{aligned}{}^\varepsilon\mathbf{M}_\delta(\mathbf{x}, \mathbf{d}) = \{\mathbf{y} : f(\mathbf{y}) \leqslant f(\mathbf{x} + \tau\mathbf{d}) + \varepsilon \text{ for all } \tau, 0 \leqslant \tau \leqslant \delta; \mathbf{y} = \mathbf{x} + \alpha\mathbf{d},\\ \text{for some } \alpha,\ 0 \leqslant \alpha \leqslant \infty,\ \mathbf{y} \in \Omega\}.\end{aligned}$$

The map ${}^\varepsilon\mathbf{M}_\delta$ corresponds to an "inaccurate" constrained line search. Show that this map is closed if $\mathbf{d} \neq \mathbf{0}$.

6. For the problem

$$\begin{aligned}&\text{minimize} \quad f(\mathbf{x})\\&\text{subject to} \quad \mathbf{a}_i^T\mathbf{x} \leqslant b_i,\ i = 1, 2, \ldots, m\end{aligned}$$

consider selecting $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ at a feasible point $\mathbf{x}$ by solving the problem

$$\begin{aligned}&\text{minimize} \quad \boldsymbol{\nabla} f(\mathbf{x})\mathbf{d}\\&\text{subject to} \quad \mathbf{a}_i^T\mathbf{d} \leqslant (b_i - \mathrm{a}_i^T\mathbf{x})M,\ i = 1, 2, \ldots, m\\&\qquad\qquad\quad \sum_{i=1}^n |d_i| = 1,\end{aligned}$$

where M is some given positive constant. For large M the ith inequality of this subsidiary problem will be active only if the corresponding inequality in the original problem is nearly active at $\mathbf{x}$ (indeed, note that $M \to \infty$ corresponds to Zoutendijk's method). Show that this direction finding mapping is closed and generates uniformly feasible directions with feasibility constant $1/M$.

7. Generalize the method of Exercise 6 so that it is applicable to nonlinear inequalities.
8. An alternate, but equivalent, definition of the projected gradient $\mathbf{p}$ is that it is the vector solving
$$\begin{aligned} &\text{minimize} \quad |\mathbf{g} - \mathbf{p}|^2 \\ &\text{subject to} \quad \mathbf{A}_q\mathbf{p} = \mathbf{0}. \end{aligned}$$
Using the Karush-Kuhn–Tucker necessary conditions, solve this problem and thereby derive the formula for the projected gradient.
9. Show that finding the $\mathbf{d}$ that solves
$$\begin{aligned} &\text{minimize} \quad \mathbf{g}^T\mathbf{d} \\ &\text{subject to} \quad \mathbf{A}_q\mathbf{d} = \mathbf{0}, \ |\mathbf{d}|^2 = 1 \end{aligned}$$
gives a vector $\mathbf{d}$ that has the same direction as the negative projected gradient.
10. Let $\mathbf{P}$ be a projection matrix. Show that $\mathbf{P}^T = \mathbf{P}$, $\mathbf{P}^2 = \mathbf{P}$.
11. Suppose $\mathbf{A}_q = [\mathbf{a}^T,\ \mathbf{A}_{\bar{q}}]$ so that $\mathbf{A}_q$ is the matrix $\mathbf{A}_{\bar{q}}$ with the row $\mathbf{a}^T$ adjoined. Show that $(\mathbf{A}_q\mathbf{A}_q^T)^{-1}$ can be found from $(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1}$ from the formula
$$(\mathbf{A}_q\mathbf{A}_q^T)^{-1} = \begin{bmatrix} \varepsilon & -\varepsilon\mathbf{a}^T\mathbf{A}_{\bar{q}}^T(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1} \\ -\varepsilon(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1}\mathbf{A}_{\bar{q}}\mathbf{a} & (\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1}[\mathbf{I} + \mathbf{A}_{\bar{q}}\mathbf{a}\mathbf{a}^T\mathbf{A}_{\bar{q}}^T(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1}] \end{bmatrix},$$
where
$$\varepsilon = \frac{1}{\mathbf{a}^T\mathbf{a} - \mathbf{a}^T\mathbf{A}_{\bar{q}}^T(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}}^T)^{-1}\mathbf{A}_{\bar{q}}\mathbf{a}}.$$
Develop a similar formula for $(\mathbf{A}_{\bar{q}}\mathbf{A}_{\bar{q}})^{-1}$ in terms of $(\mathbf{A}_q\mathbf{A}_q)^{-1}$.
12. Show that the gradient projection method will solve a linear program in a finite number of steps.
13. Suppose that the projected negative gradient $\mathbf{d}$ is calculated satisfying
$$-\mathbf{g} = \mathbf{d} + \mathbf{A}_q^T\boldsymbol{\lambda}$$
and that some component λ_i of $\boldsymbol{\lambda}$, corresponding to an inequality, is negative. Show that if the ith inequality is dropped, the projection $\mathbf{d}_i$ of the negative gradient onto the remaining constraints is a feasible direction of descent.
14. Using the result of Exercise 13, it is possible to avoid the discontinuity at $\mathbf{d} = \mathbf{0}$ in the direction finding mapping of the simple gradient projection method. At a given point let $\gamma = -\min\{0,\ \lambda_i\}$, with the minimum taken with respect to the indices i corresponding the active inequalities. The direction to be taken at this point is $\mathbf{d} = -\mathbf{P}\mathbf{g}$ if $|\mathbf{P}\mathbf{g}| \geqslant \gamma$, or $\bar{\mathbf{d}}$, defined by dropping the inequality i for which $\lambda_i = -\gamma$, if $|\mathbf{P}\mathbf{g}| \leqslant \gamma$. (In case of equality either direction is selected.) Show that this direction finding map is closed over a region where the set of active inequalities does not change.

15. Consider the problem of maximizing entropy discussed in Example 3, Sect. 14.4. Suppose this problem were solved numerically with two constraints by the gradient projection method. Derive an estimate for the rate of convergence in terms of the optimal p_i's.
16. Find the geodesics of

    (a) a two-dimensional plane
    (b) a sphere.

17. Suppose that the problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{array}$$

is such that every point is a regular point. And suppose that the sequence of points $\{\mathbf{x}_k\}_{k=0}^{\infty}$ generated by geodesic descent is bounded. Prove that every limit point of the sequence satisfies the first-order necessary conditions for a constrained minimum.
18. Show that, for linear constraints, if at some point in the reduced gradient method $\mathbf{\Delta z}$ is zero, that point satisfies the Karush-Kuhn–Tucker first-order necessary conditions for a constrained minimum.
19. Consider the problem

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geqslant \mathbf{0}, \end{array}$$

where $\mathbf{A}$ is $m \times n$. Assume $f \in C^1$, that the feasible set is bounded, and that the nondegeneracy assumption holds. Suppose a "modified" reduced gradient algorithm is defined following the procedure in Sect. 12.6 but with two modifications: (1) the basic variables are, at the beginning of an iteration, always taken as the m largest variables (ties are broken arbitrarily); (2) the formula for $\mathbf{\Delta z}$ is replaced by

$$\Delta z_i = \begin{cases} -r_i & \text{if } r_i \leqslant 0 \\ -x_i r_i & \text{if } r_i > 0 \end{cases}$$

Establish the global convergence of this algorithm.
20. Find the exact solution to the example presented in Sect. 12.4.
21. Find the direction of movement that would be taken by the gradient projection method if in the example of Sect. 12.4 the constraint $x_4 = 0$ were relaxed. Show that if the term $-3x_4$ in the objective function were replaced by $-x_4$, then both the gradient projection method and the reduced gradient method would move in identical directions.
22. Show that in terms of convergence characteristics, the reduced gradient method behaves like the gradient projection method applied to a scaled version of the problem.

23. Let r be the condition number of $\mathbf{L}_M$ and s the condition number of $\mathbf{C}^T\mathbf{C}$. Show that the rate of convergence of the reduced gradient method is no worse than $[(sr - 1)/(sr + 1)]^2$.
24. Formulate the symmetric version of the hanging chain problem using a single constraint. Find an explicit expression for the condition number of the corresponding $\mathbf{C}^T\mathbf{C}$ matrix (assuming y_1 is basic). Use Exercise 23 to obtain an estimate of the convergence rate of the reduced gradient method applied to this problem, and compare it with the rate obtained in Table 12.1, Sect. 12.7. Repeat for the two-constraint formulation (assuming y_1 and y_n are basic).
25. Referring to Exercise 19 establish a global convergence result for the convex simplex method.

References

12.2 Feasible direction methods of various types were originally suggested and developed by Zoutendijk [Z4]. The systematic study of the global convergence properties of feasible direction methods was begun by Topkis and Veinott [T8] and by Zangwill [Z2]. The Frank-Wolfe method was initially proposed in [98].

12.3–12.4 The gradient projection method was proposed and developed (more completely than discussed here) by Rosen [R5, R6], who also introduced the notion of an active set strategy. See Gill, Murray, and Wright [G7] for a discussion of working sets and active set strategies.

12.5 This material is taken from Luenberger [L14].

12.6–12.7 The reduced gradient method was originally proposed by Wolfe [W5] for problems with linear constraints and generalized to nonlinear constraints by Abadie and Carpentier [A1]. Wolfe [W4] presents an example of jamming in the reduced gradient method. The convergence analysis given in this section is new.

12.8 The convex simplex method, for problems with linear constraints, together with a proof of its global convergence is due to Zangwill [Z2].