Constrained Optimization 5

Most problems in structural optimization must be formulated as constrained minimization problems. In a typical structural design problem the objective function is a fairly simple function of the design variables (e.g., weight), but the design has to satisfy a host of stress, displacement, buckling, and frequency constraints. These constraints are usually complex functions of the design variables available only from an analysis of a finite element model of the structure. This chapter offers a review of methods that are commonly used to solve such constrained problems.

The methods described in this chapter are for use when the computational cost of evaluating the objective function and constraints is small or moderate. In these methods the objective function or constraints these are calculated exactly (e.g., by a finite element program) whenever they are required by the optimization algorithm. This approach can require hundreds of evaluations of objective function and constraints, and is not practical for problems where a single evaluation is computationally expensive. For these more expensive problems we go through an intermediate stage of constructing approximations for the objective function and constraints, or at least for the more expensive functions. The optimization is then performed on the approximate problem. This approximation process is described in the next chapter.

The basic problem that we consider in this chapter is the minimization of a function subject to equality and inequality constraints

$$
\begin{aligned}
\textbf{minimize} \quad & f(\mathbf{x}) \\
\textbf{such that} \quad & h_i(\mathbf{x}) = 0, \qquad i = 1, \ldots, n_e\,, \\
& g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g\,.
\end{aligned}
\tag{5.1}
$$

The constraints divide the design space into two domains, the feasible domain where the constraints are satisfied, and the infeasible domain where at least one of the constraints is violated. In most practical problems the minimum is found on the boundary between the feasible and infeasible domains, that is at a point where $g_j(\mathbf{x}) = 0$ for at least one j. Otherwise, the inequality constraints may be removed without altering the solution. In most structural optimization problems the inequality constraints prescribe limits on sizes, stresses, displacements, etc. These limits have great impact on the design, so that typically several of the inequality constraints are active at the minimum.

While the methods described in this section are powerful, they can often perform poorly when design variables and constraints are scaled improperly. To prevent ill-conditioning, all the design variables should have similar magnitudes, and all constraints should have similar values when they are at similar levels of criticality. A common practice is to normalize constraints such that $g(\mathbf{x}) = 0.1$ correspond to a ten percent margin in a response quantity. For example, if the constraint is an upper limit σ_a on a stress measure σ, then the constraint may be written as

$$g = 1 - \frac{\sigma}{\sigma_a} \geq 0 \ . \tag{5.2}$$

Some of the numerical techniques offered in this chapter for the solution of constrained nonlinear optimization problems are not able to handle equality constraints, but are limited to inequality constraints. In such instances it is possible to replace the equality constraint of the form $h_i(\mathbf{x}) = 0$ with two inequality constraints $h_i(\mathbf{x}) \leq 0$ and $h_i(\mathbf{x}) \geq 0$. However, it is usually undesirable to increase the number of constraints. For problems with large numbers of inequality constraints, it is possible to construct an equivalent constraint to replace them. One of the ways to replace a family of inequality constraints ($g_i(\mathbf{x}) \geq 0, i = 1 \ldots m$) by an equivalent constraint is to use the Kreisselmeier-Steinhauser function [1] (KS-function) defined as

$$KS[g_i(\mathbf{x})] = -\frac{1}{\rho} \ln[\sum_i e^{-\rho g_i(\mathbf{x})}] \ , \tag{5.3}$$

where ρ is a parameter which determines the closeness of the KS-function to the smallest inequality $min[g_i(\mathbf{x})]$. For any positive value of the ρ, the KS-function is always more negative than the most negative constraint, forming a lower bound envelope to the inequalities. As the value of ρ is increased the KS-functions conforms with the minimum value of the functions more closely. The value of the KS-function is always bounded by

$$g_{\min} \leq KS[g_i(\mathbf{x})] \leq g_{\min} - \frac{\ln(m)}{\rho} \ . \tag{5.4}$$

For an equality constraint represented by a pair of inequalities, $h_i(\mathbf{x}) \leq 0$ and $-h_i(\mathbf{x}) \leq 0$, the solution is at a point where both inequalities are active, $h_i(\mathbf{x}) = -h_i(\mathbf{x}) = 0$, Figure 5.1 . Sobieski [2] shows that for a KS-function defined by such a positive and negative pair of h_i, the gradient of the KS-function at the solution point $h_i(\mathbf{x}) = 0$ vanishes regardless of the ρ value, and its value approaches to zero as the value of ρ tends to infinity, Figure 5.1 . Indeed, from Eq. (5.4) at $\mathbf{x}$ where $h_i = 0$, the KS-function has the property

$$0 \geq KS(h, -h) \geq -\frac{\ln(2)}{\rho} \ . \tag{5.5}$$

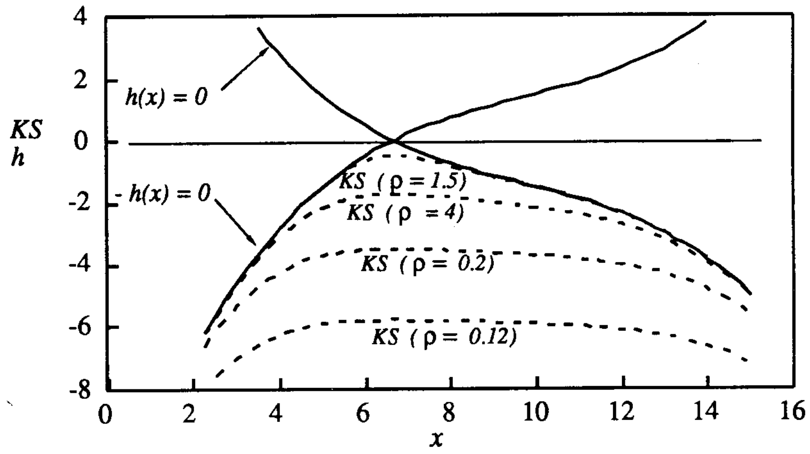

Figure 5.1 Kreisselmeier-Steinhauser function for replacing $h(\mathbf{x}) = 0$.

Consequently, an optimization problem

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && h_k(\mathbf{x}) = 0, \qquad k = 1, \ldots, n_e\,, \end{aligned} \tag{5.6}$$

may be reformulated as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && KS(h_1, -h_1, h_2, -h_2, \ldots, h_{n_e}, -h_{n_e}) \geq -\epsilon\ . \end{aligned} \tag{5.7}$$

where ϵ is a small tolerance.

5.1 The Kuhn-Tucker conditions

5.1.1 *General Case*

In general, problem (5.1) may have several local minima. Only under special circumstances are sure of the existence of single global minimum. The necessary conditions for a minimum of the constrained problem are obtained by using the Lagrange multiplier method. We start by considering the special case of equality constraints only. Using the Lagrange multiplier technique, we define the Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_{j=1}^{n_e} \lambda_j h_j(\mathbf{x})\,, \tag{5.1.1}$$

where λ_j are unknown Lagrange multipliers. The necessary conditions for a stationary point are

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} \lambda_j \frac{\partial h_j}{\partial x_i} = 0, \qquad i = 1, \ldots, n\,, \tag{5.1.2}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = h_j(\mathbf{x}) = 0, \qquad j = 1, \ldots, n_e\ . \tag{5.1.3}$$

These conditions, however, apply only at a *regular point*, that is at a point where the gradients of the constraints are linearly independent. If we have constraint gradients that are linearly dependent, it means that we can remove some constraints without affecting the solution. At a regular point, Eqs. (5.1.2) and (5.1.3) represent $n + n_e$ equations for the n_e Lagrange multipliers and the n coordinates of the stationary point.

The situation is somewhat more complicated when inequality constraints are present. To be able to apply the Lagrange multiplier method we first transform the inequality constraints to equality constraints by adding slack variables. That is, the inequality constraints are written as

$$g_j(\mathbf{x}) - t_j^2 = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.1.4}$$

where t_j is a slack variable which measures how far the jth constraint is from being critical. We can now form a Lagrangian function

$$\mathcal{L}(\mathbf{x}, \mathbf{t}, \boldsymbol{\lambda}) = f - \sum_{j=1}^{n_g} \lambda_j (g_j - t_j^2)\ . \tag{5.1.5}$$

Differentiating the Lagrangian function with respect to $\mathbf{x}$, $\boldsymbol{\lambda}$ and $\mathbf{t}$ we obtain

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1, \ldots, n\,, \tag{5.1.6}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = -g_j + t_j^2 = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.1.7}$$

$$\frac{\partial \mathcal{L}}{\partial t_j} = 2\lambda_j t_j = 0, \qquad j = 1, \ldots, n_g\ . \tag{5.1.8}$$

Equations (5.1.7) and (5.1.8) imply that when an inequality constraint is not critical (so that the corresponding slack variable is non-zero) then the Lagrange multiplier associated with the constraint is zero. Equations (5.1.6) to (5.1.8) are the necessary conditions for a stationary regular point. Note that for inequality constraints a regular point is one where the gradients of the *active* constraints are linearly independent. These conditions are modified slightly to yield the necessary conditions for a minimum and are known as the Kuhn-Tucker conditions. The Kuhn-Tucker conditions may be summarized as follows:

A point $\mathbf{x}$ is a local minimum of an inequality constrained problem only if a set of nonnegative λ_j's may be found such that:

1. Equation (5.1.6) is satisfied
2. The corresponding λ_j is zero if a constraint is not active.

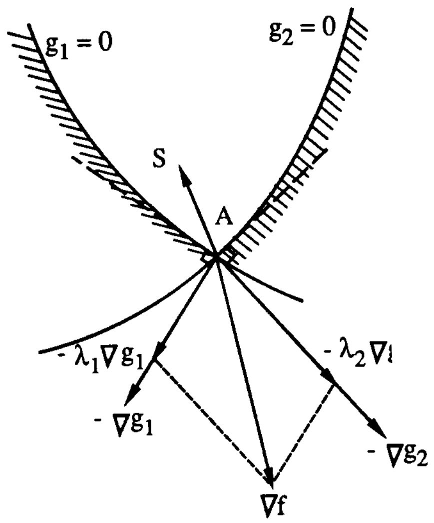

Figure 5.1.1 A geometrical interpretation of Kuhn-Tucker condition for the case of two constraints.

A geometrical interpretation of the Kuhn-Tucker conditions is illustrated in Fig. (5.1.1) for the case of two constraints. ∇g_1 and ∇g_2 denote the gradients of the two constraints which are orthogonal to the respective constraint surfaces. The vector $\mathbf{s}$ shows a typical feasible direction which does not lead immediately to any constraint violation. For the two-constraint case Eq. (5.1.6) may be written as

$$-\nabla f = -(\lambda_1 \nabla g_1 + \lambda_2 \nabla g_2) \ . \tag{5.1.9}$$

Assume that we want to determine whether point A is a minimum or not. To improve the design we need to proceed from point A in a direction $\mathbf{s}$ that is usable and feasible. For the direction to be usable, a small move along this direction should decrease the objective function. To be feasible, $\mathbf{s}$ should form an obtuse angle with $-\nabla g_1$ and $-\nabla g_2$. To be a direction of decreasing f it must form an acute angle with $-\nabla f$. Clearly from Figure (5.1.1), any vector which forms an acute angle with $-\nabla f$ will also form and acute angle with either $-\nabla g_1$ or $-\nabla g_2$. Thus the Kuhn-Tucker conditions mean that no feasible design with reduced objective function is to be found in the neighborhood of A. Mathematically, the condition that a direction $\mathbf{s}$ be feasible is written as

$$\mathbf{s}^T \nabla g_j \geq 0, \qquad j \in I_A \,, \tag{5.1.10}$$

where I_A is the set of active constraints Equality in Eq. (5.1.10) is permitted only for linear or concave constraints (see Section 5.1.2 for definition of concavity). The condition for a usable direction (one that decreases the objective function) is

$$\mathbf{s}^T \nabla f < 0 \ . \tag{5.1.11}$$

Multiplying Eq. (5.1.6) by s_i and summing over i we obtain

$$\mathbf{s}^T \nabla f = \sum_{j=1}^{n_g} \lambda_j \mathbf{s}^T \nabla g_j \ . \tag{5.1.12}$$

In view of Eqs. (5.1.10) and (5.1.11), Eq. (5.1.12) is impossible if the λ_j's are positive.

If the Kuhn-Tucker conditions are satisfied at a point it is impossible to find a direction with a negative slope for the objective function that does not violate the constraints. In some cases, though, it is possible to move in a direction which is tangent to the active constraints and perpendicular to the gradient (that is, has zero slope), that is

$$\mathbf{s}^T \nabla f = \mathbf{s}^T \nabla g_j = 0, \qquad j \in I_A \ . \tag{5.1.13}$$

The effect of such a move on the objective function and constraints can be determined only from higher derivatives. In some cases a move in this direction could reduce the objective function without violating the constraints even though the Kuhn-Tucker conditions are met. Therefore, the Kuhn-Tucker conditions are necessary but not sufficient for optimality.

The Kuhn-Tucker conditions are sufficient when the number of active constraints is equal to the number of design variables. In this case Eq. (5.1.13) cannot be satisfied with $\mathbf{s} \neq 0$ because ∇g_j includes n linearly independent directions (in n dimensional space a vector cannot be orthogonal to n linearly independent vectors).

When the number of active constraints is not equal to the number of design variables sufficient conditions for optimality require the second derivatives of the objective function and constraints. A sufficient condition for optimality is that the Hessian matrix of the Lagrangian function is positive definite in the subspace tangent to the active constraints. If we take, for example, the case of equality constraints, the Hessian matrix of the Lagrangian is

$$\nabla^2 \mathcal{L} = \nabla^2 f - \sum_{j=1}^{n_e} \lambda_j \nabla^2 h_j \ . \tag{5.1.14}$$

The sufficient condition for optimality is that

$$\mathbf{s}^T (\nabla^2 \mathcal{L}) \mathbf{s} > 0, \qquad \text{for all } \mathbf{s} \text{ for which } \mathbf{s}^T \nabla h_j = 0, \quad j = 1 \ldots, n_e \ . \tag{5.1.15}$$

When inequality constraints are present, the vector $\mathbf{s}$ also needs to be orthogonal to the gradients of the active constraints with positive Lagrange multipliers. For active constraints with zero Lagrange multipliers, $\mathbf{s}$ must satisfy

$$\mathbf{s}^T \nabla g_j \geq 0, \qquad \text{when } g_j = 0 \text{ and } \lambda_j = 0 \ . \tag{5.1.16}$$

Example 5.1.1

Find the minimum of

$$f = -x_1^3 - 2x_2^2 + 10x_1 - 6 - 2x_2^3\,,$$

subject to

$$\begin{aligned} g_1 &= 10 - x_1x_2 \geq 0,\\ g_2 &= x_1 \geq 0,\\ g_3 &= 10 - x_2 \geq 0\ . \end{aligned}$$

The Kuhn-Tucker conditions are

$$\begin{aligned} &-3x_1^2 + 10 + \lambda_1 x_2 - \lambda_2 = 0,\\ &-4x_2 - 6x_2^2 + \lambda_1 x_1 + \lambda_3 = 0\ . \end{aligned}$$

We have to check for all possibilities of active constraints.

The simplest case is when no constraints are active, $\lambda_1 = \lambda_2 = \lambda_3 = 0$. We get $x_1 = 1.826$, $x_2 = 0$, $f = 6.17$. The Hessian matrix of the Lagrangian,

$$\nabla^2\mathcal{L} = \begin{bmatrix} -6x_1 & \lambda_1 \\ \lambda_1 & -4 - 12x_2 \end{bmatrix},$$

is clearly negative definite, so that this point is a maximum. We next assume that the first constraint is active, $x_1x_2 = 10$, so that $x_1 \neq 0$ and g_2 is inactive and therefore $\lambda_2 = 0$. We have two possibilities for the third constraint. If it is active we get $x_1 = 1$, $x_2 = 10$, $\lambda_1 = -0.7$, and $\lambda_3 = 639.3$, so that this point is neither a minimum nor a maximum. If the third constraint is not active $\lambda_3 = 0$ and we obtain the following three equations

$$\begin{aligned} -3x_1^2 + 10 + \lambda_1 x_2 &= 0,\\ -4x_2 - 6x_2^2 + \lambda_1 x_1 &= 0,\\ x_1x_2 &= 10\ . \end{aligned}$$

The only solution for these equations that satisfies the constraints on x_1 and x_2 is

$$x_1 = 3.847, \qquad x_2 = 2.599, \qquad \lambda_1 = 13.24, \qquad f = -73.08\ .$$

This point satisfies the Kuhn-Tucker conditions for a minimum. However, the Hessian of the Lagrangian at that point

$$\nabla^2\mathcal{L} = \begin{bmatrix} -23.08 & 13.24 \\ 13.24 & -35.19 \end{bmatrix},$$

is negative definite, so that it cannot satisfy the sufficiency condition. In fact, an examination of the function f at neighboring points along $x_1x_2 = 10$ reveals that the point is not a minimum.

Next we consider the possibility that g_1 is not active, so that $\lambda_1 = 0$, and

$$-3x_1^2 + 10 - \lambda_2 = 0,$$
$$-4x_2 - 6x_2^2 + \lambda_3 = 0\ .$$

We have already considered the possibility of both λ's being zero, so we need to consider only three possibilities of one of these Lagrange multipliers being nonzero, or both being nonzero. The first case is $\lambda_2 \neq 0$, $\lambda_3 = 0$, then $g_2 = 0$ and we get $x_1 = 0$, $x_2 = 0$, $\lambda_2 = 10$, and $f = -6$, or $x_1 = 0$, $x_2 = -2/3$, $\lambda_2 = 10$, and $f = -6.99$. Both points satisfy the Kuhn-Tucker conditions for a minimum, but not the sufficiency condition. In fact, the vectors tangent to the active constraints ($x_1 = 0$ is the only one) have the form $\mathbf{s}^T = (0, a)$, and it is easy to check that $\mathbf{s}^T \nabla^2 \mathcal{L} \mathbf{s} < 0$. It is also easy to check that these points are indeed no minima by reducing x_2 slightly.

The next case is $\lambda_2 = 0$, $\lambda_3 \neq 0$, so that $g_3 = 0$. We get $x_1 = 1.826$, $x_2 = 10$, $\lambda_3 = 640$ and $f = -2194$. this point satisfies the Kuhn-Tucker conditions, but it is not a minimum either. It is easy to check that $\nabla^2 \mathcal{L}$ is negative definite in this case so that the sufficiency condition could not be satisfied. Finally, we consider the case $x_1 = 0$, $x_2 = 10$, $\lambda_2 = 10$, $\lambda_3 = 640$, $f = -2206$. Now the Kuhn-Tucker conditions are satisfied, and the number of active constraints is equal to the number of design variables, so that this point is a minimum.• • •

5.1.2 *Convex Problems*

There is a class of problems, namely convex problems, for which the Kuhn-Tucker conditions are not only necessary but also sufficient for a global minimum. To define convex problems we need the notions of convexity for a set of points and for a function. A set of points S is convex whenever the entire line segment connecting two points that are in S is also in S. That is

$$\text{if } \mathbf{x}_1, \mathbf{x}_2 \in S, \quad \text{then } \alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in S, \qquad 0 < \alpha < 1\ . \tag{5.1.17}$$

A function is convex if

$$f[\alpha \mathbf{x}_2 + (1-\alpha)\mathbf{x}_1] \leq \alpha f(\mathbf{x}_2) + (1-\alpha) f(\mathbf{x}_1), \qquad 0 < \alpha < 1\ . \tag{5.1.18}$$

This is shown pictorially for a function of a single variable in Figure (5.1.2). The straight segment connecting any two points on the curve must lie above the curve. Alternatively we note that the second derivative of f is non-negative $f''(x) \geq 0$. It can be shown that a function of n variables is convex if its matrix of second derivatives is positive semi-definite.

A convex optimization problem has a convex objective function and a convex feasible domain. It can be shown that the feasible domain is convex if all the inequality constraints g_j are *concave* (that is, $-g_j$ are convex) and the equality constraints are linear. A convex optimization problem has only one minimum, and the Kuhn-Tucker conditions are sufficient to establish it. Most optimization problems encountered in practice cannot be shown to be convex. However, the theory of convex programming is still very important in structural optimization, as we often approximate optimization problems by a series of convex approximations (see Chapter 9). The simplest such approximation is a linear approximation for the objective function and constraints– this produces a linear programming problem.

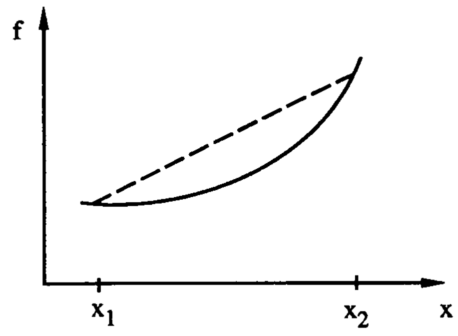

Figure 5.1.2 Convex function.

Example 5.1.2

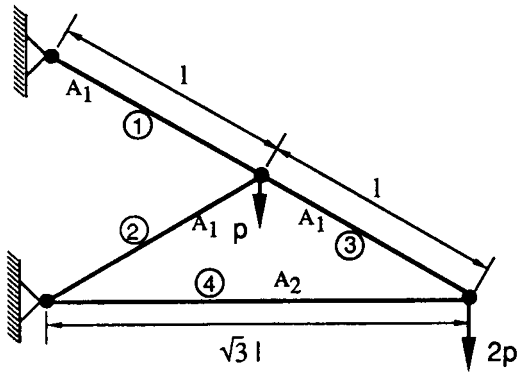

Figure 5.1.3 Four bar statically determinate truss.

Consider the minimum weight design of the four bar truss shown in Figure (5.1.3). For the sake of simplicity we assume that members 1 through 3 have the same area A_1 and member 4 has an area A_2. The constraints are limits on the stresses in the members and on the vertical displacement at the right end of the truss. Under the specified loading the member forces and the vertical displacement δ at the end are found to be

$$f_1 = 5p, \quad f_2 = -p, \quad f_3 = 4p, \quad f_4 = -2\sqrt{3}p\,,$$
$$\delta = \frac{6pl}{E}\left(\frac{3}{A_1} + \frac{\sqrt{3}}{A_2}\right)\,.$$

We assume the allowable stresses in tension and compression to be $8.74 \times 10^{-4}E$ and $4.83 \times 10^{-4}E$, respectively, and limit the vertical displacement to be no greater than $3 \times 10^{-3}l$. The minimum weight design subject to stress and displacement constraints can be formulated in terms of nondimensional design variables

$$x_1 = 10^{-3}\frac{A_1E}{p}, \quad x_2 = 10^{-3}\frac{A_2E}{p}\,,$$

as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0, \\ & g_2 = x_1 - 5.73 \geq 0, \\ & g_3 = x_2 - 7.17 \geq 0\ . \end{aligned}$$

The Kuhn-Tucker conditions are

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{3} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1, 2\,,$$

or

$$3 - \frac{18}{x_1^2}\lambda_1 - \lambda_2 = 0\,,$$

$$\sqrt{3} - \frac{6\sqrt{3}}{x_2^2}\lambda_1 - \lambda_3 = 0\ .$$

Consider first the possibility that $\lambda_1 = 0$. Then clearly $\lambda_2 = 3$, $\lambda_3 = \sqrt{3}$ so that $g_2 = 0$ and $g_3 = 0$, and then $x_1 = 5.73$, $x_2 = 7.17$, $g_1 = -1.59$, so that this solution is not feasible. We conclude that $\lambda_1 \neq 0$, and the first constraint must be active at the minimum. Consider now the possibility that $\lambda_2 = \lambda_3 = 0$. We have the two Kuhn-Tucker equations and the equation $g_1 = 0$ for the unknowns λ_1, x_1, x_2. The solution is

$$x_1 = x_2 = 9.464, \quad \lambda_1 = 14.93, \quad f = 44.78\ .$$

The Kuhn-Tucker conditions for a minimum are satisfied. If the problem is convex the Kuhn-Tucker conditions are sufficient to guarantee that this point is the global minimum. The objective function and the constraint functions g_2 and g_3 are linear, so that we need to check only g_1. For convexity g_1 has to be concave or $-g_1$ convex; this holds if the second derivative matrix $-\mathbf{A}_1$ of $-g_1$ is positive semi-definite

$$-\mathbf{A}_1 = \begin{bmatrix} 36/x_1^3 & 0 \\ 0 & 12\sqrt{3}/x_2^3 \end{bmatrix}\ .$$

Clearly, for $x_1 > 0$ and $x_2 > 0$, $-\mathbf{A}_1$ is positive definite so that the minimum that we found is a global minimum.• • •

5.2 Quadratic Programming Problems

One of the simplest form of nonlinear constrained optimization problems is in the form of *Quadratic Programming* (QP) problem. A general QP problem has a quadratic objective function with linear equality and inequality constraints. For the sake of simplicity we consider only an inequality problem with n_g constraints stated as

$$\begin{aligned} &\textbf{minimize} \quad && f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} \\ &\textbf{such that} \quad && \mathbf{A}\mathbf{x} \geq \mathbf{b}\,, \\ & && x_i \geq 0, \qquad i = 1, \ldots, n\,. \end{aligned} \tag{5.2.1}$$

The linear constraints form a convex feasible domain. If the objective function is also convex, then we have a convex optimization problem in which, as discussed in the previous section, the Kuhn-Tucker conditions become sufficient for the optimality of the problem. Hence, having a positive semi-definite or positive definite $\mathbf{Q}$ matrix assures a global minimum for the solution of the problem, if one exists. For many optimization problems the quadratic form $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ is either positive definite or positive semi-definite. Therefore, one of the methods for solving QP problems relies on solving the Kuhn-Tucker conditions.

We start by writing the Lagrange function for the Problem (5.2.1)

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{t}, \mathbf{s}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \boldsymbol{\lambda}^T(\mathbf{A}\mathbf{x} - \{t_j^2\} - \mathbf{b}) - \boldsymbol{\mu}^T(\mathbf{x} - \{s_i^2\})\,, \tag{5.2.2}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the vectors of Lagrange multipliers for the inequality constraints and the nonnegativity constraints, respectively, and $\{t_j^2\}$ and $\{s_i^2\}$ are the vectors of positive slack variables for the same. The necessary conditions for a stationary point are obtained by differentiating the Lagrangian with respect to the $\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{t}$, and $\mathbf{s}$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \mathbf{c} - \mathbf{Q}\mathbf{x} - \mathbf{A}^T\boldsymbol{\lambda} - \boldsymbol{\mu} = 0\,, \tag{5.2.3}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \mathbf{A}\mathbf{x} - \{t_j^2\} - \mathbf{b} = 0\,, \tag{5.2.4}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \mathbf{x} - \{s_i^2\} = 0\,, \tag{5.2.5}$$

$$\frac{\partial \mathcal{L}}{\partial t_j} = 2\lambda_j t_j = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.2.6}$$

$$\frac{\partial \mathcal{L}}{\partial s_i} = 2\mu_i s_i = 0, \qquad i = 1, \ldots, n\,. \tag{5.2.7}$$

where n_g is the number of inequality constraints, and n is the number of design variables. We define a new vector $\{q_j\} = \{t_j^2\}, \quad j = 1, \ldots, n_g \quad (\mathbf{q} \geq 0)$. After multiplying Eqs. (5.2.6) and (5.2.7) by $\{t_j\}$ and $\{s_i\}$, respectively, and eliminating

$\{s_i\}$ from the last equation by using Eq. (5.2.5), we can rewrite the Kuhn-Tucker conditions

$$\mathbf{Qx} + \mathbf{A}^T\boldsymbol{\lambda} + \boldsymbol{\mu} = \mathbf{c}\,, \tag{5.2.8}$$

$$\mathbf{Ax} - \mathbf{q} = \mathbf{b}\,, \tag{5.2.9}$$

$$\lambda_j q_j = 0, \qquad j = 1, \ldots, n_g\,, \tag{5.2.10}$$

$$\mu_i x_i = 0, \qquad i = 1, \ldots, n\,, \tag{5.2.11}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \text{and} \quad \boldsymbol{\mu} \geq \mathbf{0}\,. \tag{5.2.12}$$

Equations (5.2.8) and (5.2.9) form a set of $n+n_g$ linear equations for the solution of unknowns $x_i, \lambda_j, \mu_i,$ and q_j which also need to satisfy Eqs. (5.2.10) and (5.2.11). Despite the nonlinearity of the Eqs. (5.2.10) and (5.2.11), this problem can be solved as proposed by Wolfe [3] by using the procedure described in 3.6.3 for generating a basic feasible solution through the use of artificial variables. Introducing a set of artificial variables, $y_i, \quad i = 1, \ldots, n$, we define an artificial cost function to be minimized,

$$\textbf{minimize} \quad \sum_{i=1}^{n} y_i \tag{5.2.13}$$

$$\textbf{subject to} \quad \mathbf{Qx} + \mathbf{A}^T\boldsymbol{\lambda} + \boldsymbol{\mu} + \mathbf{y} = \mathbf{c}\,, \tag{5.2.14}$$

$$\mathbf{Ax} - \mathbf{q} = \mathbf{b}\,, \tag{5.2.15}$$

$$\mathbf{x} \geq \mathbf{0}, \quad \boldsymbol{\lambda} \geq \mathbf{0}, \quad \boldsymbol{\mu} \geq \mathbf{0}, \quad \text{and} \quad \mathbf{y} \geq \mathbf{0}\,. \tag{5.2.16}$$

Equations (5.2.13) through (5.2.16) can be solved by using the standard simplex method with the additional requirement that (5.2.10) and (5.2.11) be satisfied. These requirements can be implemented during the simplex algorithm by simply enforcing that the variables λ_j and q_j (and μ_i and x_i) not be included in the basic solution simultaneously. That is, we restrict a non-basic variable μ_i from entering the basis if the corresponding x_i is already among the basic variables.

Other methods for solving the quadratic programming problem are also available, and the reader is referred to Gill et al. ([4], pp. 177–180) for additional details.

5.3 Computing the Lagrange Multipliers

As may be seen from example 5.1.1, trying to find the minimum directly from the Kuhn-Tucker conditions may be difficult because we need to consider many combinations of active and inactive constraints, and this would in general involve the solution of highly nonlinear equations. The Kuhn-Tucker conditions are, however, often used to check whether a candidate minimum point satisfies the necessary conditions. In such a case we need to calculate the Lagrange multipliers (also called the Kuhn-Tucker multipliers) at a given point $\mathbf{x}$. As we will see in the next section, we may also want to calculate the Lagrange multipliers for the purpose of estimating the sensitivity of the optimum solution to small changes in the problem definition. To calculate the Lagrange multipliers we start by writing Eq. (5.1.6) in matrix notation as

$$\nabla f - \mathbf{N}\boldsymbol{\lambda} = 0\,, \tag{5.3.1}$$

where the matrix $\mathbf{N}$ is defined by

$$n_{ij} = \frac{\partial g_j}{\partial x_i}, \qquad j = 1, \ldots, r\,, \text{ and } \qquad i = 1, \ldots, n\,. \tag{5.3.2}$$

We consider only the active constraints and associated lagrange multipliers, and assume that there are r of them.

Typically, the number, r, of active constraints is less than n, so that with n equations in terms of r unknowns, Eq. (5.3.1) is an overdetermined system. We assume that the gradients of the constraints are linearly independent so that $\mathbf{N}$ has rank r. If the Kuhn-Tucker conditions are satisfied the equations are consistent and we have an exact solution. We could therefore use a subset of r equations to solve for the Lagrange multipliers. However, this approach may be susceptible to amplification of errors. Instead we can use a least-squares approach to solve the equations. We define a residual vector $\mathbf{u}$

$$\mathbf{u} = \mathbf{N}\boldsymbol{\lambda} - \nabla f\,, \tag{5.3.3}$$

A least squares solution of Eq. (5.3.1) will minimize the square of the Euclidean norm of the residual with respect to $\boldsymbol{\lambda}$

$$\|\mathbf{u}\|^2 = (\mathbf{N}\boldsymbol{\lambda} - \nabla f)^T(\mathbf{N}\boldsymbol{\lambda} - \nabla f) = \boldsymbol{\lambda}^T\mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T\mathbf{N}^T\nabla f + \nabla f^T\nabla f\,. \tag{5.3.4}$$

To minimize $\|\mathbf{u}\|^2$ we differentiate it with respect to each one of the Lagrange multipliers and get

$$2\mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} - 2\mathbf{N}^T\nabla f = 0\,, \tag{5.3.5}$$

or

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f\,. \tag{5.3.6}$$

This is the best solution in the least square sense. However, if the Kuhn-Tucker conditions are satisfied it should be the exact solution of Eq. (5.3.1). Substituting from Eq. (5.3.6) into Eq. (5.3.1) we obtain

$$\mathbf{P}\nabla f = 0\,, \tag{5.3.7}$$

where

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\,. \tag{5.3.8}$$

$\mathbf{P}$ is called the projection matrix. It will be shown in Section 5.5 that it projects a vector into the subspace tangent to the active constraints. Equation (5.3.7) implies that for the Kuhn-Tucker conditions to be satisfied the gradient of the objective function has to be orthogonal to that subspace.

In practice Eq. (5.3.6) is no longer popular for the calculation of the Lagrange multipliers. One reason is that the method is ill-conditioned and another is that it is not efficient. An efficient and better conditioned method for least squares calculations is based on the QR factorization of the matrix $\mathbf{N}$. The QR factorization of the matrix $\mathbf{N}$ consists of an $r \times r$ upper triangular matrix $\mathbf{R}$ and an $n \times n$ orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{QN} = \begin{pmatrix} \mathbf{Q}_1\mathbf{N} \\ \mathbf{Q}_2\mathbf{N} \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} . \tag{5.3.9}$$

Here $\mathbf{Q}_1$ is a matrix consisting of the first r rows of $\mathbf{Q}$, $\mathbf{Q}_2$ includes the last $n-r$ rows of $\mathbf{Q}$, and the zero represents an $(n-r) \times r$ zero matrix (for details of the QR factorization see most texts on numerical analysis, e.g., Dahlquist and Bjorck [5]). Because $\mathbf{Q}$ is an orthogonal matrix, the Euclidean norm of $\mathbf{Qu}$ is the same as that of $\mathbf{u}$, or

$$\|\mathbf{u}\|^2 = \|\mathbf{Qu}\|^2 = \|\mathbf{QN}\boldsymbol{\lambda} - \mathbf{Q}\nabla f\|^2 = \left\| \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix} \boldsymbol{\lambda} - \mathbf{Q}\nabla f \right\|^2 = \left\| \begin{pmatrix} \mathbf{R}\boldsymbol{\lambda} - \mathbf{Q}_1\nabla f \\ -\mathbf{Q}_2\nabla f \end{pmatrix} \right\|^2 . \tag{5.3.10}$$

From this form it can be seen that $\|\mathbf{u}\|^2$ is minimized by choosing $\boldsymbol{\lambda}$ so that

$$\mathbf{R}\boldsymbol{\lambda} = \mathbf{Q}_1\nabla f \ . \tag{5.3.11}$$

The last $n-r$ rows of the matrix $\mathbf{Q}$ denoted $\mathbf{Q}_2$ are also important in the following. They are orthogonal vectors which span the null space of $\mathbf{N}^T$. That is $\mathbf{N}^T$ times each one of these vectors is zero.

Example 5.3.1

Check whether the point $(-2,-2,4)$ is a local minimum of the problem

$$\begin{aligned} f &= x_1 + x_2 + x_3, \\ g_1 &= 8 - x_1^2 - x_2^2 \geq 0, \\ g_2 &= x_3 - 4 \geq 0, \\ g_3 &= x_2 + 8 \geq 0 \ . \end{aligned}$$

Only the first two constraints are critical at $(-2,-2,4)$

$$\frac{\partial g_1}{\partial x_1} = -2x_1 = 4, \quad \frac{\partial g_1}{\partial x_2} = -2x_2 = 4, \quad \frac{\partial g_1}{\partial x_3} = 0\,,$$

$$\frac{\partial g_2}{\partial x_1} = 0, \quad \frac{\partial g_2}{\partial x_2} = 0, \quad \frac{\partial g_2}{\partial x_3} = 1\,,$$

$$\frac{\partial f}{\partial x_1} = \frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial x_3} = 1\ .$$

So

$$\mathbf{N} = \begin{bmatrix} 4 & 0 \\ 4 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \nabla f = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix},$$

$$\mathbf{N}^T\mathbf{N} = \begin{bmatrix} 32 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{N}^T\nabla f = \begin{Bmatrix} 8 \\ 1 \end{Bmatrix},$$

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f = \begin{Bmatrix} 1/4 \\ 1 \end{Bmatrix},$$

also

$$\left[\mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right]\nabla f = 0\ .$$

Equation (5.3.7) is satisfied, and all the Lagrange multipliers are positive, so the Kuhn-Tucker conditions for a minimum are satisfied. • • •

5.4 Sensitivity of Optimum Solution to Problem Parameters

The Lagrange multipliers are not only useful for checking optimality, but they also provide information about the sensitivity of the optimal solution to problem parameters. In this role they are extremely valuable in practical applications. In most engineering design optimization problems we have a host of parameters such as material properties, dimensions and load levels that are fixed during the optimization. We often need the sensitivity of the optimum solution to these problem parameters, either because we do not know them accurately, or because we have some freedom to change them if we find that they have a large effect on the optimum design.

We assume now that the objective function and constraints depend on a parameter p so that the optimization problem is defined as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}, p) \\ &\textbf{such that} && g_j(\mathbf{x}, p) \geq 0, \qquad j = 1, \ldots, n_g\ . \end{aligned} \tag{5.4.1}$$

The solution of the problem is denoted $\mathbf{x}^*(p)$ and the corresponding objective function $f^*(p) = f(\mathbf{x}^*(p), p)$. We want to find the derivatives of $\mathbf{x}^*$ and f^* with respect to p. The equations that govern the optimum solution are the Kuhn-Tucker conditions, Eq. (5.3.1), and the set of active constraints

$$\mathbf{g}_a = 0. \tag{5.4.2}$$

where $\mathbf{g}_a$ denotes the vector of r active constraint functions. Equations (5.3.1) and (5.4.2) are satisfied by $\mathbf{x}^*(p)$ for all values of p that do not change the set of active constraints. Therefore, the derivatives of these equations with respect to p are zero, provided we consider the implicit dependence of $\mathbf{x}$ and $\boldsymbol{\lambda}$ on p. Differentiating Eq. (5.3.1) and (5.4.2) with respect to p we obtain

$$(\mathbf{A} - \mathbf{Z})\frac{d\mathbf{x}^*}{dp} - \mathbf{N}\frac{d\boldsymbol{\lambda}}{dp} + \frac{\partial}{\partial p}(\nabla f) - \left(\frac{\partial \mathbf{N}}{\partial p}\right)\boldsymbol{\lambda} = 0\,, \tag{5.4.3}$$

$$\mathbf{N}^T\frac{d\mathbf{x}^*}{dp} + \frac{\partial \mathbf{g}_a}{\partial p} = 0\,, \tag{5.4.4}$$

where $\mathbf{A}$ is the Hessian matrix of the objective function f, $a_{ij} = \partial^2 f/\partial x_i \partial x_j$, and $\mathbf{Z}$ is a matrix whose elements are

$$z_{kl} = \sum_j \frac{\partial^2 g_j}{\partial x_k \partial x_l} \lambda_j \ . \tag{5.4.5}$$

Equations (5.4.3) and (5.4.4) are a system of simultaneous equations for the derivatives of the design variables and of the Lagrange multipliers. Different special cases of this system are discussed by Sobieski et al. [6].

Often we do not need the derivatives of the design variables or of the Lagrange multipliers, but only the derivatives of the objective function. In this case the sensitivity analysis can be greatly simplified. We can write

$$\frac{df}{dp} = \frac{\partial f}{\partial p} + \sum_{l=1}^{n} \frac{\partial f}{\partial x_l}\frac{dx_l^*}{dp} = \frac{\partial f}{\partial p} + (\nabla f)^T \frac{d\mathbf{x}^*}{dp} \ . \tag{5.4.6}$$

Using Eq. (5.3.1) and (5.4.4) we get

$$\frac{df}{dp} = \frac{\partial f}{\partial p} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{g}_a}{\partial p} \ . \tag{5.4.7}$$

Equation (5.4.7) shows that the Lagrange multipliers are a measure of the effect of a change in the constraints on the objective function. Consider, for example, a constraint of the form $g_j(\mathbf{x}) = G_j(\mathbf{x}) - p \geq 0$. By increasing p we make the constraint more difficult to satisfy. Assume that many constraints are critical, but that p affects only this single constraint. We see that $\partial g_j/\partial p = -1$, and from Eq. (5.4.7) $df/dp = \lambda_j$, that is λ_j is the 'marginal price' that we pay in terms of an increase in the objective function for making g_j more difficult to satisfy.

The interpretation of Lagrange multipliers as the marginal prices of the constraints also explains why at the optimum all the Lagrange multipliers have to be non-negative. A negative Lagrange multiplier would indicate that we can reduce the objective function by making a constraint more difficult to satisfy— an absurdity.

Example 5.4.1

Consider the optimization problem

$$\begin{aligned} f &= x_1 + x_2 + x_3, \\ g_1 &= p - x_1^2 - x_2^2 \geq 0, \\ g_2 &= x_3 - 4 \geq 0, \\ g_3 &= x_2 + p \geq 0 \ . \end{aligned}$$

This problem was analyzed for $p = 8$ in Example 5.3.1, and the optimal solution was found to be $(-2, -2, 4)$. We want to find the derivative of this optimal solution with respect to p. At the optimal point we have $f = 0$ and $\boldsymbol{\lambda}^T = (0.25, 1.0)$, with the first two constraints being critical. We can calculate the derivative of the objective function from Eq. (5.4.7)

$$\frac{\partial f}{\partial p} = 0, \qquad \frac{\partial \mathbf{g}_a}{\partial p} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix},$$

so

$$\frac{df}{dp} = -0.25 .$$

To calculate the derivatives of the design variables and constraints we need to set up Eqs. (5.4.3) and (5.4.4). We get

$$\mathbf{A} = \mathbf{0}, \qquad \frac{\partial \nabla f}{\partial p} = \mathbf{0}, \qquad \frac{\partial \mathbf{N}}{\partial p} = \mathbf{0} .$$

Only g_1 has nonzero second derivatives $\partial^2 g_1/\partial x_1^2 = \partial^2 g_1/\partial x_2^2 = -2$ so from Eq. (5.4.5)

$$z_{11} = -2\lambda_1 = -0.5, \qquad z_{22} = -2\lambda_1 = -0.5, \qquad \mathbf{Z} = \begin{bmatrix} -.5 & 0 & 0 \\ 0 & -.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

With $\mathbf{N}$ from Example 5.3.1, Eq. (5.4.3) gives us

$$\begin{aligned} .5\dot{x}_1 - 4\dot{\lambda}_1 &= 0 , \\ .5\dot{x}_2 - 4\dot{\lambda}_1 &= 0 , \\ \dot{\lambda}_2 &= 0 , \end{aligned}$$

where a dot denotes derivative with respect to p. From Eq. (5.4.4) we get

$$\begin{aligned} 4\dot{x}_1 + 4\dot{x}_2 + 1 &= 0 , \\ \dot{x}_3 &= 0 . \end{aligned}$$

The solution of these five coupled equations is

$$\dot{x}_1 = \dot{x}_2 = -0.125, \quad \dot{x}_3 = 0, \quad \dot{\lambda}_1 = -0.0156, \quad \dot{\lambda}_2 = 0 .$$

We can check the derivatives of the objective function and design variables by changing p from 8 to 9 and re-optimizing. It is easy to check that we get $x_1 = x_2 = -2.121$, $x_3 = 4$, $f = -0.242$. These values compare well with linear extrapolation based on the derivatives which gives $x_1 = x_2 = -2.125$, $x_3 = 4$, $f = -0.25$.• • •

5.5 Gradient Projection and Reduced Gradient Methods

Rosen's gradient projection method is based on projecting the search direction into the subspace tangent to the active constraints. Let us first examine the method for the case of linear constraints [7]. We define the constrained problem as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && g_j(\mathbf{x}) = \sum_{i=1}^{n} a_{ji}x_i - b_j \geq 0, \qquad j = 1, \ldots, n_g \ . \end{aligned} \tag{5.5.1}$$

In vector form

$$g_j = \mathbf{a}_j^T \mathbf{x} - b_j \geq 0 \ . \tag{5.5.2}$$

If we select only the r active constraints ($j \in I_A$), we may write the constraint equations as

$$\mathbf{g}_a = \mathbf{N}^T \mathbf{x} - \mathbf{b} = 0 \,, \tag{5.5.3}$$

where $\mathbf{g}_a$ is the vector of active constraints and the columns of the matrix $\mathbf{N}$ are the gradients of these constraints. The basic assumption of the gradient projection method is that $\mathbf{x}$ lies in the subspace tangent to the active constraints. If

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{s} \,, \tag{5.5.4}$$

and both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ satisfy Eq. (5.5.3), then

$$\mathbf{N}^T \mathbf{s} = 0 \ . \tag{5.5.5}$$

If we want the steepest descent direction satisfying Eq. (5.5.5), we can pose the problem as

$$\begin{aligned} \textbf{minimize} \quad & \mathbf{s}^T \nabla f \\ \textbf{such that} \quad & \mathbf{N}^T \mathbf{s} = 0 \,, \\ \textbf{and} \quad & \mathbf{s}^T \mathbf{s} = 1 \ . \end{aligned} \tag{5.5.6}$$

That is, we want to find the direction with the most negative directional derivative which satisfies Eq. (5.5.5). We use Lagrange multipliers $\boldsymbol{\lambda}$ and μ to form the Lagrangian

$$\mathcal{L}(\mathbf{s}, \boldsymbol{\lambda}, \mu) = \mathbf{s}^T \nabla f - \mathbf{s}^T \mathbf{N} \boldsymbol{\lambda} - \mu(\mathbf{s}^T \mathbf{s} - 1) \ . \tag{5.5.7}$$

The condition for $\mathcal{L}$ to be stationary is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \nabla f - \mathbf{N} \boldsymbol{\lambda} - 2\mu \mathbf{s} = 0 \ . \tag{5.5.8}$$

Premultiplying Eq. (5.5.8) by $\mathbf{N}^T$ and using Eq. (5.5.5) we obtain

$$\mathbf{N}^T \nabla f - \mathbf{N}^T \mathbf{N} \boldsymbol{\lambda} = 0 \,, \tag{5.5.9}$$

or

$$\boldsymbol{\lambda} = (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \nabla f \ . \tag{5.5.10}$$

So that from Eq. (5.5.8)

$$\mathbf{s} = \frac{1}{2\mu}[I - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T]\nabla f = \frac{1}{2\mu}\mathbf{P}\nabla f \ . \tag{5.5.11}$$

$\mathbf{P}$ is the projection matrix defined in Eq. (5.3.8). The factor of $1/2\mu$ is not significant because $\mathbf{s}$ defines only the direction of search, so in general we use $\mathbf{s} = -\mathbf{P}\nabla f$. To show that $\mathbf{P}$ indeed has the projection property, we need to prove that if $\mathbf{w}$ is an arbitrary vector, then $\mathbf{Pw}$ is in the subspace tangent to the active constraints, that is $\mathbf{Pw}$ satisfies

$$\mathbf{N}^T\mathbf{Pw} = 0 \ . \tag{5.5.12}$$

We can easily verify this by using the definition of $\mathbf{P}$.

Equation (5.3.8) which defines the projection matrix $\mathbf{P}$ does not provide the most efficient way for calculating it. Instead it can be shown that

$$\mathbf{P} = \mathbf{Q}_2^T\mathbf{Q}_2 \,, \tag{5.5.13}$$

where the matrix $\mathbf{Q}_2$ consists of the last $n - r$ rows of the $\mathbf{Q}$ factor in the QR factorization of $\mathbf{N}$ (see Eq. (5.3.9)).

A version of the gradient projection method known as the *generalized reduced gradient* method was developed by Abadie and Carpentier [8]. As a first step we select r linearly independent rows of $\mathbf{N}$, denote their transpose as $\mathbf{N}_1$ and partition $\mathbf{N}^T$ as

$$\mathbf{N}^T = [\mathbf{N}_1 \quad \mathbf{N}_2] \,. \tag{5.5.14}$$

Next we consider Eq. (5.5.5) for the components s_i of the direction vector. The r equations corresponding to $\mathbf{N}_1$ are then used to eliminate r components of $\mathbf{s}$ and obtain a reduced order problem for the direction vector.

Once we have identified $\mathbf{N}_1$ we can easily obtain $\mathbf{Q}_2$ which is given as

$$\mathbf{Q}_2^T = \begin{bmatrix} -\mathbf{N}_1^{-1}\mathbf{N}_2 \\ \mathbf{I} \end{bmatrix} \ . \tag{5.5.15}$$

Equation (5.5.15) can be verified by checking that $\mathbf{N}^T\mathbf{Q}_2^T = 0$, so that $\mathbf{Q}_2\mathbf{N} = 0$, which is the requirement that $\mathbf{Q}_2$ has to satisfy (see discussion following Eq. (5.3.11)).

After obtaining $\mathbf{s}$ from Eq. (5.5.11) we can continue the search with a one dimensional minimization, Eq. (5.5.4), unless $\mathbf{s} = 0$. When $\mathbf{s} = 0$ Eq. (5.3.7) indicates that the Kuhn-Tucker conditions may be satisfied. We then calculate the Lagrange multipliers from Eq. (5.3.6) or Eq. (5.3.11). If all the components of $\boldsymbol{\lambda}$ are non-negative, the Kuhn-Tucker conditions are indeed satisfied and the optimization can be terminated. If some of the Lagrange multipliers are negative, it is an indication that while no progress is possible with the current set of active constraints, it may be possible to proceed by removing some of the constraints associated with negative Lagrange multipliers. A common strategy is to remove the constraint associated with the most negative Lagrange multiplier and repeat the calculation of $\mathbf{P}$ and $\mathbf{s}$. If $\mathbf{s}$

is now non-zero, a one-dimensional search may be started. If $\mathbf{s}$ remains zero and there are still negative Lagrange multipliers, we remove another constraint until all Lagrange multipliers become positive and we satisfy the Kuhn-Tucker conditions.

After a search direction has been determined, a one dimensional search must be carried out to determine the value of α in Eq. (5.5.4). Unlike the unconstrained case, there is an upper limit on α set by the inactive constraints. As α increases, some of them may become active and then violated. Substituting $\mathbf{x} = \mathbf{x}_i + \alpha\mathbf{s}$ into Eq. (5.5.2) we obtain

$$g_j = \mathbf{a}_j^T(\mathbf{x}_i + \alpha\mathbf{s}) - b_j \geq 0\,, \tag{5.5.16}$$

or

$$\alpha \leq -(\mathbf{a}_j^T\mathbf{x}_i - b_j)/\mathbf{a}_j^T\mathbf{s} = -g_j(\mathbf{x}_i)/\mathbf{a}_j^T\mathbf{s}\ . \tag{5.5.17}$$

Equation (5.5.17) is valid if $\mathbf{a}_j^T\mathbf{s} < 0$. Otherwise, there is no upper limit on α due to the jth constraint. From Eq. (5.5.17) we get a different α, say α_j for each constraint. The upper limit on α is the minimum

$$\bar{\alpha} = \min_{\alpha_j>0,\ j\ni I_A} \alpha_j\ . \tag{5.5.18}$$

At the end of the move, new constraints may become active, so that the set of active constraints may need to be updated before the next move is undertaken.

The version of the gradient projection method presented so far is an extension of the steepest descent method. Like the steepest descent method, it may have slow convergence. The method may be extended to correspond to Newton or quasi-Newton methods. In the unconstrained case, these methods use a search direction defined as

$$\mathbf{s} = -\mathbf{B}\nabla f\,, \tag{5.5.19}$$

where $\mathbf{B}$ is the inverse of the Hessian matrix of f or an approximation thereof. The direction that corresponds to such a method in the subspace tangent to the active constraints can be shown [4] to be

$$\mathbf{s} = -\mathbf{Q}_2^T(\mathbf{Q}_2^T\mathbf{A}_L\mathbf{Q}_2)^{-1}\mathbf{Q}_2\nabla f\,, \tag{5.5.20}$$

where $\mathbf{A}_L$ is the Hessian of the Lagrangian function or an approximation thereof.

The gradient projection method has been generalized by Rosen to nonlinear constraints [9]. The method is based on linearizing the constraints about $\mathbf{x}_i$ so that

$$\mathbf{N} = [\nabla g_1(\mathbf{x}_i),\ \nabla g_2(\mathbf{x}_i), \ldots, \nabla g_r(\mathbf{x}_i)]\ . \tag{5.5.21}$$

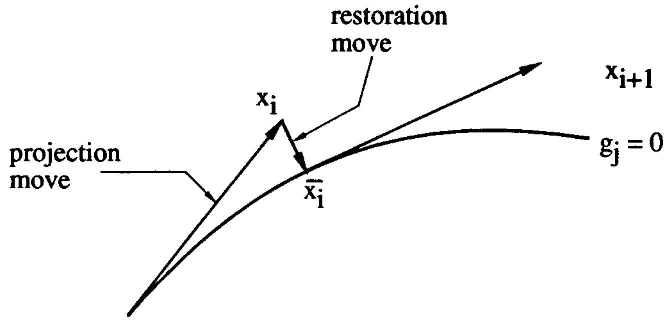

Figure 5.5.1 Projection and restoration moves.

The main difficulty caused by the nonlinearity of the constraints is that the one-dimensional search typically moves away from the constraint boundary. This is because we move in the tangent subspace which no longer follows exactly the constraint boundaries. After the one-dimensional search is over, Rosen prescribes a restoration move to bring $\mathbf{x}$ back to the constraint boundaries, see Figure 5.5.1.

To obtain the equation for the restoration move, we note that instead of Eq. (5.5.2) we now use the linear approximation

$$g_j \approx g_j(\mathbf{x}_i) + \nabla g_j^T(\bar{\mathbf{x}}_i - \mathbf{x}_i) \ . \tag{5.5.22}$$

We want to find a correction $\bar{\mathbf{x}}_i - \mathbf{x}_i$ in the tangent subspace (i.e. $\mathbf{P}(\bar{\mathbf{x}}_i - \mathbf{x}_i) = 0$) that would reduce g_j to zero. It is easy to check that

$$\bar{\mathbf{x}}_i - \mathbf{x}_i = -\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a(\mathbf{x}_i) \,, \tag{5.5.23}$$

is the desired correction, where $\mathbf{g}_a$ is the vector of active constraints. Equation (5.5.23) is based on a linear approximation, and may therefore have to be applied repeatedly until $\mathbf{g}_a$ is small enough.

In addition to the need for a restoration move, the nonlinearity of the constraints requires the re-evaluation of $\mathbf{N}$ at each point. It also complicates the choice of an upper limit for α which guarantees that we will not violate the presently inactive constraints. Haug and Arora [10] suggest a procedure which is better suited for the nonlinear case. The first advantage of their procedure is that it does not require a one-dimensional search. Instead, α in Eq. (5.5.4) is determined by specifying a desired specified reduction γ in the objective function. That is, we specify

$$f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \approx \gamma f(\mathbf{x}_i) \ . \tag{5.5.24}$$

Using a linear approximation with Eq. (5.5.4) we get

$$\alpha^* = -\frac{\gamma f(\mathbf{x}_i)}{\mathbf{s}^T\nabla f} \ . \tag{5.5.25}$$

The second feature of Haug and Arora's procedure is the combination of the projection and the restoration moves as

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha^*\mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a \,, \tag{5.5.26}$$

where Eqs. (5.5.4), (5.5.23) and (5.5.25) are used.

Example 5.5.1

Use the gradient projection method to solve the following problem

$$
\begin{aligned}
\textbf{minimize} \quad & f = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\
\textbf{subject to} \quad & g_1 = 2x_1 + x_2 + x_3 + 4x_4 - 7 \geq 0 \,, \\
& g_2 = x_1 + x_2 + x_3^2 + x_4 - 5.1 \geq 0 \,, \\
& x_i \geq 0, \qquad i = 1, \ldots, 4 \;.
\end{aligned}
$$

Assume that as a result of previous moves we start at the point $\mathbf{x}_0^T = (2, 2, 1, 0)$, $f(\mathbf{x}_0) = 5.0$, where the nonlinear constraint g_2 is slightly violated. The first constraint is active as well as the constraint on x_4. We start with a combined projection and restoration move, with a target improvement of 10% in the objective function. At $\mathbf{x}_0$

$$
\mathbf{N} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \\ 4 & 1 & 1 \end{bmatrix} , \qquad \mathbf{N}^T\mathbf{N} = \begin{bmatrix} 22 & 9 & 4 \\ 9 & 7 & 1 \\ 4 & 1 & 1 \end{bmatrix} ,
$$

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

$$
\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T = \frac{1}{11} \begin{bmatrix} 1 & -3 & 1 & 0 \\ -3 & 9 & -3 & 0 \\ 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} , \qquad \nabla f = \begin{Bmatrix} 2 \\ 4 \\ 2 \\ -3 \end{Bmatrix} .
$$

The projection move direction is $\mathbf{s} = -\mathbf{P}\nabla f = [8/11, -24/11, 8/11, 0]^T$. Since the magnitude of a direction vector is unimportant we scale $\mathbf{s}$ to $\mathbf{s}^T = [1, -3, 1, 0]$. For a 10% improvement in the objective function $\gamma = 0.1$ and from Eq. (5.5.25)

$$
\alpha^* = -\frac{0.1f}{\mathbf{s}^T\nabla f} = -\frac{0.1 \times 5}{-8} = 0.0625 \;.
$$

For the correction move we need the vector $\mathbf{g}_a$ of constraint values , $\mathbf{g}_a^T = (0, -0.1, 0)$, so the correction is

$$
-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \frac{-1}{110} \begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix} .
$$

Combining the projection and restoration moves, Eq. (5.5.26)

$$
\mathbf{x}_1 = \begin{Bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{Bmatrix} + 0.0625 \begin{Bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{Bmatrix} - \frac{1}{110} \begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 2.026 \\ 1.822 \\ 1.126 \\ 0 \end{Bmatrix} ,
$$

we get $f(\mathbf{x}_1) = 4.64$, $g_1(\mathbf{x}_1) = 0$, $g_2(\mathbf{x}_1) = 0.016$. Note that instead of 10% reduction we got only 7% due to the nonlinearity of the objective function. However, we did satisfy the nonlinear constraint.• • •

Example 5.5.2

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned}
\textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\
\textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\
& g_2 = x_1 - 5.73 \geq 0\,, \\
& g_3 = x_2 - 7.17 \geq 0\,,
\end{aligned}$$

where the x_i are non-dimensional areas

$$x_i = \frac{A_i E}{1000P}, \qquad i = 1, 2\ .$$

The first constraint represents a limit on the vertical displacement, and the other two represent stress constraints.

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$, where $x_1 = 11.61$, $x_2 = 7.17$, and $f = 47.25$. The gradients of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}, \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix}, \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}, \qquad \mathbf{N} = \begin{bmatrix} 0.1335 & 0 \\ 0.2021 & 1 \end{bmatrix}.$$

Because $\mathbf{N}$ is nonsingular, Eq. (5.3.8) shows that $\mathbf{P} = 0$. Also since the number of linearly independent active constraints is equal to the number of design variables the tangent subspace is a single point, so that there is no more room for progress. Using Eqs. (5.3.6) or (5.3.11) we obtain

$$\boldsymbol{\lambda} = \begin{Bmatrix} 22.47 \\ -2.798 \end{Bmatrix}.$$

The negative multiplier associated with g_3 indicates that this constraint can be dropped from the active set. Now

$$\mathbf{N} = \begin{bmatrix} 0.1335 \\ 0.2021 \end{bmatrix}.$$

The projection matrix is calculated from Eq. (5.3.8)

$$\mathbf{P} = \begin{bmatrix} 0.6962 & -0.4600 \\ -0.4600 & 0.3036 \end{bmatrix}, \qquad \mathbf{s} = -\mathbf{P}\nabla f = \begin{Bmatrix} -1.29 \\ 0.854 \end{Bmatrix}.$$

We attempt a 5% reduction in the objective function, and from Eq. (5.5.25)

$$\alpha^* = \frac{0.05 \times 47.25}{[-1.29\ 0.854]\begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}} = 0.988\ .$$

Since there was no constraint violation at $\mathbf{x}_0$ we do not need a combined projection and correction step, and

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha^* \mathbf{s} = \begin{Bmatrix} 11.61 \\ 7.17 \end{Bmatrix} + 0.988 \begin{Bmatrix} -1.29 \\ 0.854 \end{Bmatrix} = \begin{Bmatrix} 10.34 \\ 8.01 \end{Bmatrix} .$$

At $\mathbf{x}_1$ we have $f(\mathbf{x}_1) = 44.89$, $g_1(\mathbf{x}_1) = -0.0382$. Obviously g_2 is not violated. If there were a danger of that we would have to limit α^* using Eq. (5.5.17). The violation of the nonlinear constraint is not surprising, and its size indicates that we should reduce the attempted reduction in f in the next move. At x_1, only g_1 is active so

$$\mathbf{N} = \nabla \mathbf{g}_1 = \begin{Bmatrix} 0.1684 \\ 0.1620 \end{Bmatrix} .$$

The projection matrix is calculated to be

$$\mathbf{P} = \begin{bmatrix} 0.4806 & -0.4996 \\ -0.4996 & 0.5194 \end{bmatrix}, \qquad \mathbf{s} = -\mathbf{P}\nabla f = \begin{Bmatrix} -0.5764 \\ 0.5991 \end{Bmatrix} .$$

Because of the violation we reduce the attempted reduction in f to 2.5%, so

$$\alpha^* = -\frac{0.025 \times 44.89}{[-0.567 \; 0.599] \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}} = 1.62 .$$

We need also a correction due to the constraint violation ($\mathbf{g}_a = -0.0382$)

$$-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \begin{Bmatrix} 0.118 \\ 0.113 \end{Bmatrix} .$$

Altogether

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha^* \mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \begin{Bmatrix} 10.34 \\ 8.01 \end{Bmatrix} - 1.62 \begin{Bmatrix} 0.576 \\ -0.599 \end{Bmatrix} + \begin{Bmatrix} 0.118 \\ 0.113 \end{Bmatrix} = \begin{Bmatrix} 9.52 \\ 9.10 \end{Bmatrix} .$$

We obtain $f(\mathbf{x}_2) = 44.32$, $g_1(\mathbf{x}_2) = -0.0328$.

The optimum design is actually $\mathbf{x}^T = (9.464, 9.464)$, $f(\mathbf{x}) = 44.78$, so after two iterations we are quite close to the optimum design.• • •

5.6 The Feasible Directions Method

The feasible directions method [11] has the opposite philosophy to that of the gradient projection method. Instead of following the constraint boundaries, we try to stay as far away as possible from them. The typical iteration of the feasible direction method starts at the boundary of the feasible domain (unconstrained minimization techniques are used to generate a direction if no constraint is active).

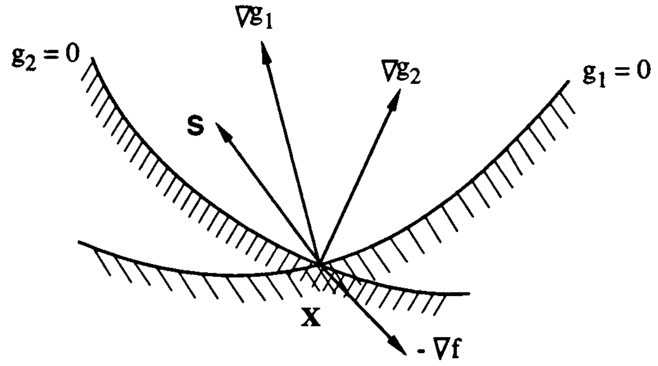

Figure 5.6.1 Selection of search direction using the feasible directions method.

Consider Figure 5.6.1. As a result of a previous move the design is at point $\mathbf{x}$ and we look for a direction $\mathbf{s}$ which keeps $\mathbf{x}$ in the feasible domain and improves the objective function. A vector $\mathbf{s}$ is defined as a feasible direction if at least a small step can be taken along it that does not immediately leave the feasible domain. If the constraints are smooth, this is satisfied if

$$\mathbf{s}^T \nabla g_j > 0, \qquad j \in I_A\,, \tag{5.6.1}$$

where I_A is the set of critical constraints at $\mathbf{x}$. The direction $\mathbf{s}$ is called a usable direction at the point $\mathbf{x}$ if in addition

$$\mathbf{s}^T \nabla f = \mathbf{s}^T \mathbf{g} < 0\ . \tag{5.6.2}$$

That is, $\mathbf{s}$ is a direction which reduces the objective function.

Among all possible choices of usable feasible directions we seek the direction which is best in some sense. We have two criteria for selecting a direction. On the one hand we want to reduce the objective function as much as possible. On the other hand we want to keep away from the constraint boundary as much as possible. A compromise is defined by the following maximization problem

$$\begin{aligned} \textbf{maximize} \quad & \beta \\ \textbf{such that} \quad & -\mathbf{s}^T \nabla g_j + \theta_j \beta \le 0, \qquad && j \in I_A\,, \\ & \mathbf{s}^T \nabla f + \beta \le 0, && \theta_j \ge 0\,, \\ & |s_i| \le 1\ . \end{aligned} \tag{5.6.3}$$

The θ_j are positive numbers called "push-off" factors because their magnitude determines how far $\mathbf{x}$ will move from the constraint boundaries. A value of $\theta_j = 0$ will result in a move tangent to the boundary of the the jth constraint, and so may be appropriate for a linear constraint. A large value of θ_j will result in a large angle between the constraint boundary and the move direction, and so is appropriate for a highly nonlinear constraint.

The optimization problem defined by Eq. (5.6.3) is linear and can be solved using the simplex algorithm. If $\beta_{max} > 0$, we have found a usable feasible direction. If we get $\beta_{max} = 0$ it can be shown that the Kuhn-Tucker conditions are satisfied.

Once a direction of search has been found, the choice of step length is typically based on a prescribed reduction in the objective function (using Eq. (5.5.25)). If at the end of the step no constraints are active, we continue in the same direction as long as $\mathbf{s}^T\nabla f$ is negative. We start the next iteration when $\mathbf{x}$ hits the constraint boundaries, or use a direction based on unconstrained technique if $\mathbf{x}$ is inside the feasible domain. Finally, if some constraints are violated after the initial step we make $\mathbf{x}$ retreat based on the value of the violated constraints. The method of feasible directions is implemented in the popular CONMIN program [12].

Example 5.6.1

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned}
\textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\
\textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\
& g_2 = x_1 - 5.73 \geq 0\,, \\
& g_3 = x_2 - 7.17 \geq 0\,,
\end{aligned}$$

where the x_i are non-dimensional areas

$$x_i = \frac{A_i E}{1000P}\,, \qquad i = 1, 2\,.$$

The first constraint represents a limit on the vertical displacement, and the other two constraints represent stress constraints.

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$ where $\mathbf{x}_0^T = (11.61, 7.17)$ and $f = 47.25$. The gradient of the objective function and two active constraints are

$$\nabla f = \left\{ \begin{matrix} 3 \\ \sqrt{3} \end{matrix} \right\}\,, \qquad \nabla g_1 = \left\{ \begin{matrix} 0.1335 \\ 0.2021 \end{matrix} \right\}\,, \qquad \nabla g_3 = \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}\,.$$

Selecting $\theta_1 = \theta_2 = 1$, we find that Eq. (5.6.3) becomes

$$\begin{aligned}
\textbf{maximize} \quad & \beta \\
\textbf{subject to} \quad & -0.1335s_1 - 0.2021s_2 + \beta \leq 0\,, \\
& -s_2 + \beta \leq 0\,, \\
& 3s_1 + \sqrt{3}s_2 + \beta \leq 0\,, \\
& -1 \leq s_1 \leq 1\,, \\
& -1 \leq s_2 \leq 1\,.
\end{aligned}$$

The solution of this linear program is $s_1 = -0.6172$, $s_2 = 1$, and we now need to execute the one dimensional search

$$\mathbf{x}_1 = \left\{ \begin{array}{c} 11.61 \\ 7.17 \end{array} \right\} + \alpha \left\{ \begin{array}{c} -0.6172 \\ 1 \end{array} \right\} .$$

Because the objective function is linear, this direction will remain a descent direction indefinitely, and α will be limited only by the constraints. The requirement that g_2 is not violated will lead to $\alpha = 9.527$, $x_1 = 5.73$, $x_2 = 16.7$ which violates g_1. We see that because g_1 is nonlinear, even though we start the search by moving away from it we still bump into it again (see Figure 5.6.2). It can be easily checked that for $\alpha > 5.385$ we violate g_1. So we take $\alpha = 5.385$ and obtain $x_1 = 8.29$, $x_2 = 12.56$, $f = 46.62$.

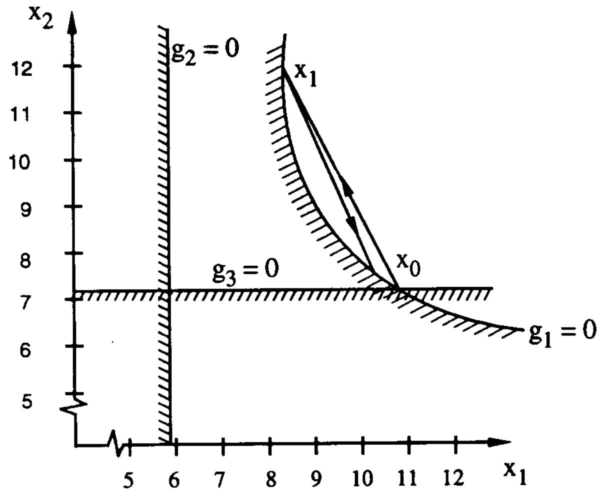

Figure 5.6.2 Feasible direction solution of 4 bar truss example.

For the next iteration we have only one active constraint

$$\nabla g_1 = \left\{ \begin{array}{c} 0.2619 \\ 0.0659 \end{array} \right\}, \qquad \nabla f = \left\{ \begin{array}{c} 3 \\ \sqrt{3} \end{array} \right\} .$$

The linear program for obtaining $\mathbf{s}$ is

$$\begin{aligned} \textbf{maximize} \quad & \beta \\ \textbf{subject to} \quad & -0.2619 s_1 - 0.0659 s_2 + \beta \le 0 , \\ & 3 s_1 + \sqrt{3} s_2 + \beta \le 0 , \\ & -1 \le s_1 \le 1 , \\ & -1 \le s_2 \le 1 . \end{aligned}$$

The solution of the linear program is $s_1 = 0.5512$, $s_2 = -1$, so that the one-dimensional search is

$$\mathbf{x} = \begin{Bmatrix} 8.29 \\ 12.56 \end{Bmatrix} + \alpha \begin{Bmatrix} 0.5512 \\ -1 \end{Bmatrix} .$$

Again α is limited only by the constraints. The lower limit on x_2 dictates $\alpha \leq 5.35$. However, the constraint g_1 is again more critical. It can be verified that for $\alpha > 4.957$ it is violated, so we take $\alpha = 4.957$, $x_1 = 11.02$, $x_2 = 7.60$, $f = 46.22$. The optimum design found in Example 5.1.2 is $x_1 = x_2 = 9.464$, $f = 44.78$. The design space and the two iterations are shown in Figure 5.6.2. • • •

5.7 Penalty Function Methods

When the energy crisis erupted in the middle seventies, the United States Congress passed legislation intended to reduce the fuel consumption of American cars. The target was an average fuel consumption of 27.5 miles per gallon for new cars in 1985. Rather than simply legislate this limit Congress took a gradual approach, with a different limit set each year to bring up the average from about 14 miles per gallon to the target value. Thus the limit was set at 26 for 1984, 25 for 1983, 24 for 1982, and so on. Furthermore, the limit was not absolute, but there was a fine of $50 per 0.1 miles per gallon violation per car.

This approach to constraining the automobile companies to produce fuel efficient cars has two important aspects. First, by legislating a penalty proportional to the violation rather than an absolute limit, the government allowed the auto companies more flexibility. That meant they could follow a time schedule that approximated the government schedule without having to adhere to it rigidly. Second, the gradual approach made enforcement easier politically. Had the government simply set the ultimate limit for 1985 only, nobody would have paid attention to the law in the 1970's. Then as 1985 moved closer there would have been a rush to develop fuel efficient cars. The hurried effort could mean both non-optimal car designs and political pressure to delay the enforcement of the law.

The fuel efficiency law is an example in which constraints on behavior or economic activities are imposed via penalties whose magnitude depends on the degree of violation of the constraints. It is no wonder that this simple and appealing approach has found application in constrained optimization. Instead of applying constraints we replace them by penalties which depend on the degree of constraint violations. This approach is attractive because it replaces a constrained optimization problem by an unconstrained one.

The penalties associated with constraint violation have to be high enough so that the constraints are only slightly violated. However, just as there are political problems associated with imposing abrupt high penalties in real life, so there are numerical difficulties associated with such a practice in numerical optimization. For this reason we opt for a gradual approach where we start with small penalties and increase them gradually.

5.7.1 *Exterior Penalty Function*

The exterior penalty function associates a penalty with a violation of a constraint. The term 'exterior' refers to the fact that penalties are applied only in the exterior of the feasible domain. The most common exterior penalty function is one which associates a penalty which is proportional to the square of a violation. That is, the constrained minimization problem, Eq. (5.1)

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) \\ \textbf{such that} \quad & h_i(\mathbf{x}) = 0, \quad i = 1, \ldots, n_e\,, \\ & g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, n_g\,, \end{aligned} \tag{5.7.1}$$

is replaced by

$$\begin{aligned} \textbf{minimize} \quad & \phi(\mathbf{x}, r) = f(\mathbf{x}) + r\sum_{i=1}^{n_e} h_i^2(\mathbf{x}) + r\sum_{j=1}^{n_g} < -g_j >^2 \\ & r = r_1, r_2, \ldots, \qquad r_i \to \infty\,, \end{aligned} \tag{5.7.2}$$

where $< a >$ denote the positive part of a or $\max(a, 0)$. The inequality terms are treated differently from the equality terms because the penalty applies only for constraint violation. The positive multiplier r controls the magnitude of the penalty terms. It may seem logical to choose a very high value of r to ensure that no constraints are violated. However, as noted before, this approach leads to numerical difficulties illustrated later in an example. Instead the minimization is started with a relatively small value of r, and then r is gradually increased. A typical value for r_{i+1}/r_i is 5. A typical plot of $\phi(\mathbf{x}, r)$ as a function of r is shown in Figure 5.7.1 for a simple example.

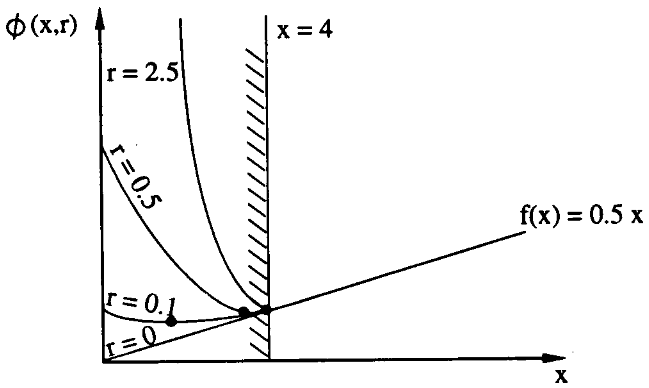

Figure 5.7.1 Exterior penalty function for $f = 0.5x$ subject to $x - 4 \geq 0$.

We see that as r is increased, the minimum of ϕ moves closer to the constraint boundary. However, the curvature of ϕ near the minimum also increases. It is the high values of the curvature associated with large values of r which often lead to numerical difficulties. By using a sequence of values of r, we use the minima obtained for small values of r as starting points for the search with higher r values. Thus the ill-conditioning associated with the large curvature is counterbalanced by the availability of a good starting point.

Based on the type of constraint normalization given by Eq. (5.2) we can select a reasonable starting value for the penalty multiplier r. A rule of thumb is that one should start with the total penalty being about equal to the objective function for typical constraint violation of 50% of the response limits. In most optimization problems the total number of active constraints is about the same as or just slightly lower than the number of design variables. Assuming we start with one quarter of the eventual active constraints being violated by about 50% (or $g = -0.5$) then we have

$$f(\mathbf{x}_0) \approx r_0 \frac{n}{4}(0.5)^2, \qquad \text{or } r_0 = 16\frac{f(\mathbf{x}_0)}{n} \ . \tag{5.7.3}$$

It is also important to obtain a good starting point for restarting the optimization as r is increased. The minimum of the optimization for the previous value of r is a reasonable starting point, but one can do better. Fiacco and McCormick [13] show that the position of the minimum of $\phi(\mathbf{x}, r)$ has the asymptotic form

$$\mathbf{x}^*(r) = \mathbf{a} + \mathbf{b}/r, \qquad \text{as } r \to \infty \ . \tag{5.7.4}$$

Once the optimum has been found for two values of r, say r_{i-1}, and r_i, the vectors $\mathbf{a}$ and $\mathbf{b}$ may be estimated, and the value of $\mathbf{x}^*(r)$ predicted for subsequent values of r. It is easy to check that in order to satisfy Eq. (5.7.4), $\mathbf{a}$ and $\mathbf{b}$ are given as

$$\begin{aligned} \mathbf{a} &= \frac{c\mathbf{x}^*(r_{i-1}) - \mathbf{x}^*(r_i)}{c-1} \ , \\ \mathbf{b} &= [\mathbf{x}^*(r_{i-1}) - \mathbf{a}]\, r_{i-1} \ , \end{aligned} \tag{5.7.5}$$

where

$$c = r_{i-1}/r_i \ . \tag{5.7.6}$$

In addition to predicting a good value of the design variables for restarting the optimization for the next value of r, Eq. (5.7.4) provides us with a useful convergence criterion, namely

$$\|\mathbf{x}^* - \mathbf{a}\| \le \epsilon_1 \ , \tag{5.7.7}$$

where $\mathbf{a}$ is estimated from the last two values of r, and ϵ_1 is a specified tolerance chosen to be small compared to a typical value of $\|\mathbf{x}\|$.

A second convergence criterion is based on the magnitude of the penalty terms, which, as shown in Example 5.7.1, go to zero as r goes to infinity. Therefore, a reasonable convergence criterion is

$$\left|\frac{\phi - f}{f}\right| \le \epsilon_2 \ . \tag{5.7.8}$$

Finally, a criterion based on the change in the value of the objective function at the minimum f^* is also used

$$\left| \frac{f^*(r_i) - f^*(r_{i-1})}{f^*(r_i)} \right| \leq 0 \ . \tag{5.7.9}$$

A typical value for ϵ_2 or ϵ_3 is 0.001.

Example 5.7.1

Minimize $f = x_1^2 + 10x_2^2$ such that $x_1 + x_2 = 4$. We have

$$\phi = x_1^2 + 10x_2^2 + r(4 - x_1 - x_2)^2 \ .$$

The gradient $\nabla\phi$ is given as

$$\mathbf{g} = \left\{ \begin{array}{c} 2x_1(1+r) + 2rx_2 - 8r \\ 2x_2(10+r) + 2rx_1 - 8r \end{array} \right\} \ .$$

Setting the gradient to zero we obtain

$$x_1 = \frac{40\,r}{10+11r}, \qquad x_2 = \frac{4\,r}{10+11r} \ .$$

The solution as a function of r is shown in Table 5.7.1.

Table 5.7.1 Minimization of ϕ for different penalty multipliers.

r	x_1	x_2	f	ϕ
1	1.905	0.1905	3.992	7.619
10	3.333	0.3333	12.220	13.333
100	3.604	0.3604	14.288	14.144
1000	3.633	0.3633	14.518	14.532

It can be seen that as r is increased the solution converges to the exact solution of $x^T = (3.636, 0.3636)$, $f = 14.54$. The convergence is indicated by the shrinking difference between the objective function and the augmented function ϕ. The Hessian of ϕ is given as

$$\mathbf{H} = \begin{bmatrix} 2+2r & 2r \\ 2r & 20+2r \end{bmatrix} \ .$$

As r increases this matrix becomes more and more ill-conditioned, as all four components become approximately $2r$. This ill-conditioning of the Hessian matrix for large values of r often occurs when the exterior penalty function is used, and can cause numerical difficulties for large problems.

We can use Table 5.7.1 to test the extrapolation procedure, Eq. (5.7.4). For example, with the values of $r = 1$ and $r = 10$, Eq. (5.7.5) gives

$$\mathbf{a} = \frac{0.1\mathbf{x}^*(1) - \mathbf{x}^*(10)}{-0.9} = \left\{ \begin{array}{c} 3.492 \\ 0.3492 \end{array} \right\} ,$$

$$\mathbf{b} = \mathbf{x}^*(1) - \mathbf{a} = \begin{Bmatrix} -0.159 \\ -0.0159 \end{Bmatrix} .$$

We can now use Eq. (5.7.4) to find a starting point for the optimization for $r = 100$ to get

$$\mathbf{a} + \mathbf{b}/100 = (3.490, 0.3490)^T ,$$

which is substantially closer to $\mathbf{x}^*(100) = (3.604, 0.3604)^T$ than to $\mathbf{x}^*(10) = (3.333, 0.3333)^T$. • • •

5.7.2 *Interior and Extended Interior Penalty Functions*

With the exterior penalty function, constraints contribute penalty terms only when they are violated. As a result, the design typically moves in the infeasible domain. If the minimization is terminated before r becomes very large (for example, because of shortage of computer resources) the resulting designs may be useless. When only inequality constraints are present, it is possible to define an interior penalty function that keeps the design in the feasible domain. The common form of the interior penalty method replaces the inequality constrained problem

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g , \end{aligned} \tag{5.7.10}$$

by

$$\begin{aligned} &\textbf{minimize} \quad \phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) , \\ &\qquad r = r_1, r_2, \ldots, \qquad r_i \to 0, \quad r_i > 0 . \end{aligned} \tag{5.7.11}$$

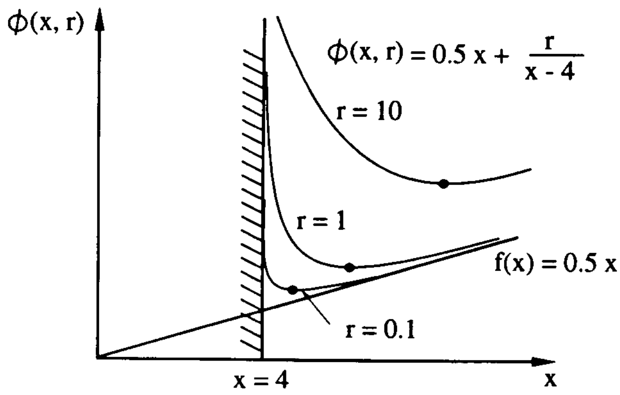

Figure 5.7.2 Interior penalty function for $f(x) = 0.5x$ subject to $x - 4 \geq 0$.

The penalty term is proportional to $1/g_j$ and becomes infinitely large at the boundary of the feasible domain creating a barrier there (interior penalty function methods are sometimes called barrier methods). It is assumed that the search is confined to the feasible domain. Otherwise, the penalty becomes negative which does not make any sense. Figure 5.7.2 shows the application of the interior penalty function to the simple example used for the exterior penalty function in Figure 5.7.1. Besides the inverse penalty function defined in Eq. (5.7.11), there has been some use of a logarithmic interior penalty function

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) - r \sum_{j=1}^{n_g} log(g_j(\mathbf{x})) \ . \tag{5.7.12}$$

While the interior penalty function has the advantage over the exterior one in that it produces a series of feasible designs, it also requires a feasible starting point. Unfortunately, it is often difficult to find such a feasible starting design. Also, because of the use of approximation (see Chapter 6), it is quite common for the optimization process to stray occasionally into the infeasible domain. For these reasons it may be advantageous to use a combination of interior and exterior penalty functions called an extended interior penalty function. An example is the quadratic extended interior penalty function of Haftka and Starnes [14]

$$\begin{gathered}\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} p(g_j) \,, \\ r = r_1, r_2, \ldots, \qquad r_i \to 0 \,,\end{gathered} \tag{5.7.13}$$

where

$$p(g_j) = \begin{cases} 1/g_j & \text{for } g_j \geq g_0 \\ 1/g_0[3 - 3(g_j/g_0) + (g_j/g_0)^2] & \text{for } g_i < g_0 \ . \end{cases} \tag{5.7.14}$$

It is easy to check that $p(g_j)$ has continuity up to second derivatives. The transition parameter g_0 which defines the boundary between the interior and exterior parts of the penalty terms must be chosen so that the penalty associated with the constraint, $rp(g_j)$, becomes infinite for negative g_j as r tends to zero. This results in the requirement that

$$r/g_0^3 \to \infty, \quad \text{as } r \to 0 \ . \tag{5.7.15}$$

This can be achieved by selecting g_0 as

$$g_0 = cr^{1/2} \,, \tag{5.7.16}$$

where c is a constant.

It is also possible to include equality constraints with interior and extended interior penalty functions. For example, the interior penalty function Eq. (5.7.11) is augmented as

$$\begin{gathered}\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) + r^{-1/2} \sum_{i=1}^{n_e} h_i^2(\mathbf{x}) \,, \\ r = r_1, r_2, \ldots, \qquad r_i \to 0 \ .\end{gathered} \tag{5.7.17}$$

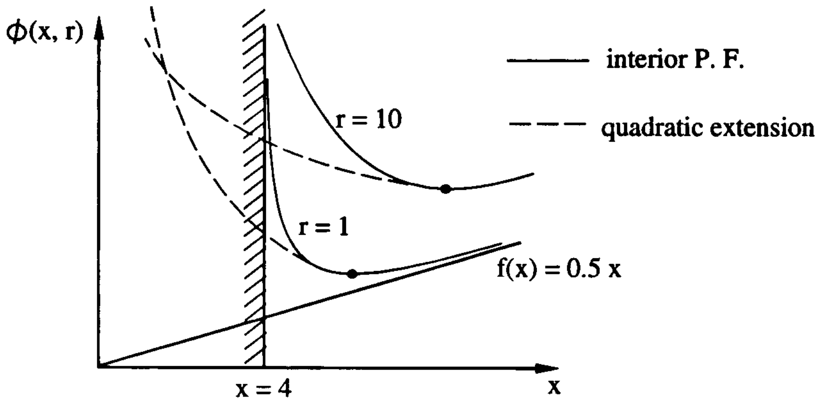

Figure 5.7.3 Extended interior penalty function for $f(x) = 0.5x$ subject to $g(x) = x - 4 \geq 0$.

The considerations for the choice of an initial value of r are similar to those for the exterior penalty function. A reasonable choice for the interior penalty function would require that $n/4$ active constraints at $g = 0.5$ (that is 50% margin for properly normalized constraints) would result in a total penalty equal to the objective function. Using Eq. (5.7.11) we obtain

$$f(\mathbf{x}) = \frac{n}{4}\frac{r}{0.5}, \qquad \text{or } \; r = 2f(\mathbf{x})/n \; .$$

For the extended interior penalty function it is more reasonable to assume that the $n/4$ constraints are critical ($g = 0$), so that from Eq. (5.7.13)

$$f(\mathbf{x}) = r\frac{n}{4}\frac{3}{g_0}, \qquad \text{or } \; r = \frac{4}{3}g_0 f(\mathbf{x})/n \; .$$

A reasonable starting value for g_0 is 0.1. As for the exterior penalty function, it is possible to obtain an expression for the asymptotic (as $r \to 0$) coordinates of the minimum of ϕ as [10]

$$\mathbf{x}^*(r) = \mathbf{a} + \mathbf{b}r^{1/2}, \quad \text{as } \; r \to 0\,, \tag{5.7.18}$$

and

$$f^*(r) = a + br^{1/2}, \quad \text{as } \; r \to 0 \; .$$

$\mathbf{a}$, $\mathbf{b}$, a and b may be estimated once the minimization has been carried out for two values of r. For example, the estimates for $\mathbf{a}$ and $\mathbf{b}$ are

$$\begin{aligned} \mathbf{a} &= \frac{c^{1/2}\mathbf{x}^*(r_{i-1}) - \mathbf{x}^*(r_i)}{c^{1/2} - 1}\,, \\ \mathbf{b} &= \frac{\mathbf{x}^*(r_{i-1}) - \mathbf{a}}{r_{i-1}^{1/2}}\,, \end{aligned} \tag{5.7.19}$$

where $c = r_i/r_{i-1}$. As in the case of exterior penalty function, these expressions may be used for convergence tests and extrapolation.

5.7.3 *Unconstrained Minimization with Penalty Functions*

Penalty functions convert a constrained minimization problem into an unconstrained one. It may seem that we should now use the best available methods for unconstrained minimization, such as quasi-Newton methods. This may not necessarily be the case. The penalty terms cause the function ϕ to have large curvatures near the constraint boundary even if the curvatures of the objective function and constraints are small. This effect permits an inexpensive approximate calculation of the Hessian matrix, so that we can use Newton's method without incurring the high cost of calculating second derivatives of constraints. This may be more attractive than using quasi-Newton methods (where the Hessian is also approximated on the basis of first derivatives) because a good approximation is obtained with a single analysis rather than with the n moves typically required for a quasi-Newton method. Consider, for example, an exterior penalty function applied to equality constraints

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{i=1}^{n_e} h_i^2(\mathbf{x}) \ . \tag{5.7.20}$$

The second derivatives of ϕ are given as

$$\frac{\partial^2 \phi}{\partial x_k \partial x_l} = \frac{\partial^2 f}{\partial x_k \partial x_l} + r \sum_{i=1}^{n_e} 2 \left(\frac{\partial h_i}{\partial x_k} \frac{\partial h_i}{\partial x_l} + h_i \frac{\partial^2 h_i}{\partial x_k \partial x_l} \right) \ . \tag{5.7.21}$$

Because of the equality constraint, h_i is close to zero, especially for the later stages of the optimization (large r), and we can neglect the last term in Eq. (5.7.21). For large values of r we can also neglect the first term, so that we can calculate second derivatives of ϕ based on first derivatives of the constraints. The availability of inexpensive second derivatives permits the use of Newton's method where the number of iterations is typically independent of the number of design variables. Quasi-Newton and conjugate gradient methods, on the other hand, require a number of iterations proportional to the number of design variables. Thus the use of Newton's method becomes attractive when the number of design variables is large. The application of Newton's method with the above approximation of second derivatives is known as the Gauss-Newton method.

For the interior penalty function we have a similar situation. The augmented objective function ϕ is given as

$$\phi(\mathbf{x}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} 1/g_j(\mathbf{x}) \, , \tag{5.7.22}$$

and the second derivatives are

$$\frac{\partial^2 \phi}{\partial x_k \partial x_l} = \frac{\partial^2 f}{\partial x_k \partial x_l} + r \sum_{j=1}^{n_g} \frac{1}{g_j^3} \left(2 \frac{\partial g_j}{\partial x_k} \frac{\partial g_j}{\partial x_l} - g_j \frac{\partial^2 g_j}{\partial x_k \partial x_l} \right) \ . \tag{5.7.23}$$

Now the argument for neglecting the first and last terms in Eq. (5.7.23) is somewhat lengthier. First we observe that because of the $1/g_j^3$ term, the second derivatives are dominated by the critical constraints (g_j small). For these constraints the last term in Eq. (5.7.23) is negligible compared to the first-derivative term because g_j is small. Finally, from Eq. (5.7.18) it can be shown that r/g_j^3 goes to infinity for active constraints as r goes to zero, so that the first term in Eq. (5.7.23) can be neglected compared to the second. The same argument can also be used for extended interior penalty functions [14].

The power of the Gauss-Newton method is shown in [14] for a high- aspect-ratio wing made of composite materials (see Figure 5.7.4) designed subject to stress and displacement constraints.

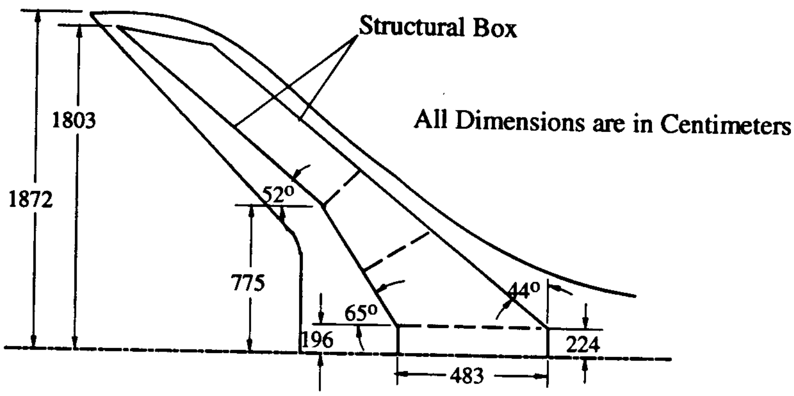

Figure 5.7.4 Aerodynamic planform and structural box for high-aspect ratio wing, from [14].

Table 5.7.2 Results of high-aspect-ratio wing study

Number of design variables	CDC 6600 CPU time sec	Total number of unconstrained minimizations	Total number of analyses	Final mass, kg
13	142	4	21	887.3
25	217	4	19	869.1
32	293	5	22	661.7
50	460	5	25	658.2
74	777	5	28	648.6
146	1708	5	26	513.0

The structural box of the wing was modeled with a finite element model with 67 nodes and 290 finite elements. The number of design variables controlling the thickness of the various elements was varied from 13 to 146. The effect of the number of design variables on the number of iterations (analyses) is shown in Table 5.7.2. It is seen that the number of iterations per unconstrained minimization is almost constant (about five). With a quasi-Newton method that number may be expected to be similar to the number of design variables.

Because of the sharp curvature of ϕ near the constraint boundary, it may also be appropriate to use specialized line searches with penalty functions [15].

5.7.4 *Integer Programming with Penalty Functions*

An extension of the penalty function approach has been implemented by Shin et al. [16] for problems with discrete-valued design variables. The extension is based on introduction of additional penalty terms into the augmented-objective function $\phi(\mathbf{x}, r)$ to reflect the requirement that the design variables take discrete values,

$$x_i \in X_i = \{d_{i1}, d_{i2}, \ldots, d_{il}\}, \qquad i \in I_d\,, \tag{5.7.24}$$

where I_d is the set of design variables that can take only discrete values, and X_i is the set of allowable discrete values. Note that several variables may have the same allowable set of discrete values. In this case the augmented objective function which includes the penalty terms due to constraints and the non-discrete values of the design variables is defined as

$$\phi(\mathbf{x}, r, s) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} p(g_j) + s \sum_{i \in I_d} \psi_d(x_i)\ , \tag{5.7.25}$$

where s is a penalty multiplier for non-discrete values of the design variables, and $\psi_d(x_i)$ the penalty term for non-discrete values of the ith design variable. Different forms for the discrete penalty function are possible. The penalty terms $\psi_d(x_i)$ are assumed to take the following sine-function form in Ref. [16],

$$\psi_d(x_i) = \frac{1}{2}\left(\sin \frac{2\pi[x_i - \frac{1}{4}(d_{i(j+1)} + 3d_{ij})]}{d_{i(j+1)} - d_{ij}} + 1\right)\ , \qquad d_{ij} \le x_i \le d_{i(j+1)}\ . \tag{5.7.26}$$

While penalizing the non-discrete valued design variables, the functions $\psi_d(x_i)$ assure the continuity of the first derivatives of the augmented function at the discrete values of the design variables. The response surfaces generated by Eq. (5.7.25) are determined according to the values of the penalty multipliers r and s. In contrast to the multiplier r, which initially has a large value and decreases as we move from one iteration to another, the value of the multiplier s is initially zero and increases gradually.

One of the important factors in the application of the proposed method is to determine when to activate s, and how fast to increase it to obtain discrete optimum design. Clearly, if the initial value of s is too big and introduced too early in the design process, the design variables will be trapped away from the global minimum, resulting in a sub-optimal solution. To avoid this problem, the multiplier s has to be activated after optimization of several response surfaces which include only constraint penalty terms. In fact, since sometimes the optimum design with discrete values is in the neighborhood of the continuous optimum, it may be desirable not to activate the penalty for the non-discrete design variables until reasonable convergence to the continuous solution is achieved. This is especially true for problems in which the intervals between discrete values are very small.

A criterion for the activation of the non-discrete penalty multiplier s is the same as the convergence criterion of Eq. (5.7.6), that is

$$\left|\frac{\phi - f}{f}\right| \le \epsilon_c \; . \qquad (5.7.27)$$

A typical value for ϵ_c is 0.01. The magnitude of the non-discrete penalty multiplier, s, at the first discrete iteration is calculated such that the penalty associated with the discrete-valued design variables that are not at their allowed values is of the order of 10 percent of the constraint penalty.

$$s \approx 0.1 r p(g) \; . \qquad (5.7.28)$$

As the iteration for discrete optimization proceeds, the non-discrete penalty multiplier for the new iteration is increased by a factor of the order of 10. It is also important to decide how to control the penalty multiplier for the constraints, r, during the discrete optimization process. If r is decreased for each discrete optimization iteration as in the continuous optimization process, the design can be stalled due to high penalties for constraint violation. Thus, it is suggested that the penalty multiplier r be frozen at the end of the continuous optimization process. However, the nearest discrete solution at this response surface may not be a feasible design, in which case the design must move away from the continuous optimum by moving back to the previous response surface. This can be achieved by increasing the penalty multiplier, r, by a factor of 10.

The solution process for the discrete optimization is terminated if the design variables are sufficiently close to the prescribed discrete values. The convergence criterion for discrete optimization is

$$\max_{\mathbf{i}\in\mathbf{I_i}}\left\{\mathbf{min}\left\{\frac{|x_i - d_{ij}|}{d_{i(j+1)} - d_{ij}}, \frac{|x_i - d_{i(j+1)}|}{d_{i(j+1)} - d_{ij}}\right\}\right\} \le \epsilon_d \, , \qquad (5.7.29)$$

where a typical value of the convergence tolerance ϵ_d is 0.001.

Example 5.7.2

Cross-sectional areas of members of a two-bar truss shown in the Figure 5.7.5 are to be selected from a discrete set of values, $A_i \in \{1.0, 1.5, 2.0\}, \; i = 1, 2$. Determine the minimum weight structure using the modified penalty function approach such that the horizontal displacement u at the point of application of the force does not exceed $2/3(Fl/E)$. Use a tolerance $\epsilon_c = 0.1$ for the activation of the penalty terms for non-discrete valued design variables, and a convergence tolerance for the design variables $\epsilon_d = 0.001$.

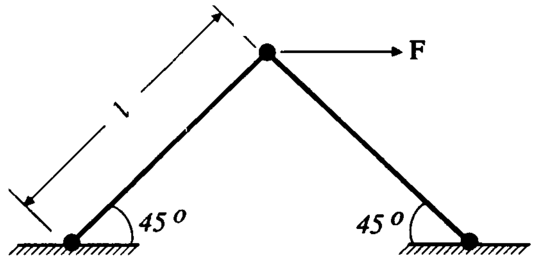

Figure 5.7.5 Two-bar truss.

Upon normalization, the design problem is posed as

$$\begin{aligned} \textbf{minimize} \quad & f = \frac{W}{\rho l} = x_1 + x_2 \\ \textbf{subject to} \quad & g = \frac{uE}{Fl} = 1.5 - 1/x_1 - 1/x_2 \geq 0\,, \\ & x_i = A_i \in \{1.0, 1.5, 2.0\}, \qquad i = 1, \ldots, 2\ . \end{aligned}$$

Using an initial design of $x_1 = x_2 = 5$ and transition parameter $g_0 = 0.1$, we have $g = 1.1 > g_0$, therefore, from Eq. (5.7.14) the penalty terms for the constraints are in the form of $p(g) = 1/g$. The augmented function for the extended interior penalty function approach is

$$\phi = x_1 + x_2 + \frac{r}{1.5 - 1/x_1 - 1/x_2}\ .$$

Setting the gradient to zero, we can show that the minimum of the augmented function as a function of the penalty multiplier r is

$$x_1 = x_2 = \frac{24 + \sqrt{576 - 36\,(16 - 4r)}}{18}\ .$$

The initial value of the penalty multiplier r is chosen so that the penalty introduced for the constraint is equal to the objective function value,

$$r\frac{1}{g(x_0)} = f(x_0), \qquad r = 11\ .$$

The minima of the augmented function as functions of the penalty multiplier r are shown in Table 5.7.3 . After four iterations the constraint penalty $(\phi - f)$ is within the desired range of the objective function to activate the penalty terms for the non-discrete values of the design variables.

From Eq. (5.7.25) the augmented function for the modified penalty function approach has the form

$$\begin{aligned} \phi = & x_1 + x_2 + \frac{r}{1.5 - 1/x_1 - 1/x_2} + \frac{s\,\{1 + \sin[4\pi\,(x_1 - 1.125)]\}}{2} \\ & + (s/2)\,\{1 + \sin[4\pi\,(x_2 - 1.125)]\}\ . \end{aligned}$$

Table 5.7.3 Minimization of ϕ without the discrete penalty

r	x_1	x_2	f	g	ϕ
-	5.000	5.000	10.00	1.100	-
11	3.544	3.544	7.089	0.9357	18.844
1.1	2.033	2.033	4.065	0.5160	6.197
0.11	1.554	1.554	3.109	0.2134	3.624
0.011	1.403	1.403	2.807	0.0747	2.954

The minimum of the augmented function can again be obtained by setting the gradient to zero

$$1 - \frac{r}{(1.5 - 2/x_1)^2 {x_1}^2} + 2\pi s \cos[4\pi\,(x_1 - 1.125)] \;= 0\,,$$

which can be solved numerically. The initial value of the penalty multiplier s is calculated from Eq. (5.7.28)

$$s = 0.1\ (0.011)\ \frac{1}{0.0747} = 0.0147\ .$$

The minima of the augmented function (which includes the penalty for the non-discrete valued variables) are shown in Table 5.7.4 as a function of s.

Table 5.7.4 Minimization of ϕ with the discrete penalty

r	s	x_1	x_2	f	ϕ
0.011	0.0147	1.406	1.406	2.813	2.963
	0.1472	1.432	1.432	2.864	3.021
	1.472	1.493	1.493	2.986	3.060
	14.72	1.499	1.499	2.999	3.065
	147.2	1.500	1.500	3.000	3.066

After four discrete iterations we obtain a minimum at $x_1 = x_2 = 3/2$. There are two more minima, $\mathbf{x} = (2, 1)$ and $\mathbf{x} = (1, 2)$, with the same value of the objective function of $f = 3.0$. • • •

5.8 Multiplier Methods

Multiplier methods combine the use of Lagrange multipliers with penalty functions. When only Lagrange multipliers are employed the optimum is a stationary point rather than a minimum of the Lagrangian function. When only penalty functions are employed we have a minimum but also ill-conditioning. By using both we may hope to get an unconstrained problem where the function to be minimized does not suffer from ill-conditioning. A good survey of multiplier methods was conducted by

Bertsekas [17]. We study first the use of multiplier methods for equality constrained problems.

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && h_j(\mathbf{x}) = 0, \qquad j = 1, \ldots, n_e \ . \end{aligned} \tag{5.8.1}$$

We define the augmented Lagrangian function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) - \sum_{j=1}^{n_e} \lambda_j h_j(\mathbf{x}) + r \sum_{j=1}^{n_e} h_j^2(\mathbf{x}) \ . \tag{5.8.2}$$

If all the Lagrange multipliers are set to zero, we get the usual exterior penalty function. On the other hand, if we use the correct values of the Lagrange multipliers, λ_j^*, it can be shown that we get the correct minimum of problem (5.8.1) for any positive value of r. Then there is no need to use the large value of r required for the exterior penalty function. Of course, we do not know what are the correct values of the Lagrange multipliers.

Multiplier methods are based on estimating the Lagrange multipliers. When the estimates are good, it is possible to approach the optimum without using large r values. The value of r needs to be only large enough so that $\mathcal{L}$ has a minimum rather than a stationary point at the optimum. To obtain an estimate for the Lagrange multipliers we compare the stationarity conditions for $\mathcal{L}$,

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} (\lambda_j - 2 r h_j) \frac{\partial h_j}{\partial x_i} = 0 \,, \tag{5.8.3}$$

with the exact conditions for the Lagrange multipliers

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_e} \lambda_j^* \frac{\partial h_j}{\partial x_i} = 0 \ . \tag{5.8.4}$$

Comparing Eqs. (5.8.3) and (5.8.4) we expect that

$$\lambda_j - 2 r h_j \rightarrow \lambda_j^* \,, \tag{5.8.5}$$

as the minimum is approached. Based on this relation, Hestenes [18] suggested using Eq. (5.8.5) as an estimate for λ_j^*. That is

$$\lambda_j^{(k+1)} = \lambda_j^{(k)} - 2 r^{(k)} h_j^{(k)} \,, \tag{5.8.6}$$

where k is an iteration number.

Example 5.8.1

We repeat Example 5.7.1 using Hestenes' multiplier method.

$$f(\mathbf{x}) = x_1^2 + 10x_2^2\,,$$
$$h(\mathbf{x}) = x_1 + x_2 - 4 = 0\ .$$

The augmented Lagrangian is

$$\mathcal{L} = x_1^2 + 10x_2^2 - \lambda(x_1 + x_2 - 4) + r(x_1 + x_2 - 4)^2\ .$$

To find the stationary points of the augmented Lagrangian we differentiate with respect to x_1 and x_2 to get

$$2x_1 - \lambda + 2r(x_1 + x_2 - 4) = 0\,,$$
$$20x_2 - \lambda + 2r(x_1 + x_2 - 4) = 0\,,$$

which yield

$$x_1 = 10x_2 = \frac{5\lambda + 40r}{10 + 11r}\ .$$

We want to compare the results with those of Example 5.7.1, so we start with the same initial r value $r_0 = 1$, the initial estimate of $\lambda = 0$ and get

$$\mathbf{x}_1 = (1.905, 0.1905)^T, \qquad h = -1.905\ .$$

So, using Eq. (5.8.6) we estimate $\lambda^{(1)}$ as

$$\lambda^{(1)} = -2 \times 1 \times (-1.905) = 3.81\ .$$

We next repeat the optimization with $r^{(1)} = 10$, $\lambda^{(1)} = 3.81$ and get

$$\mathbf{x}_2 = (3.492, 0.3492)^T, \qquad h = -0.1587\ .$$

For the same value of r, we obtained in Example 5.7.1 $\mathbf{x}_2 = (3.333, 0.3333)^T$, so that we are now closer to the exact solution of $\mathbf{x} = (3.636, 0, 3636)^T$. Now we estimate a new λ from Eq. (5.8.6)

$$\lambda^{(2)} = 3.81 - 2 \times 10 \times (-0.1587) = 6.984\ .$$

For the next iteration we may, for example, fix the value of r at 10 and change only λ. For $\lambda = 6.984$ we obtain

$$\mathbf{x}_3 = (3.624, 0.3624), \qquad h = -0.0136\,,$$

which shows that good convergence can be obtained without increasing r.• • •

There are several ways to extend the multiplier method to deal with inequality constraints. The formulation below is based on Fletcher's work [19]. The constrained problem that we examine is

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) \\ \textbf{such that} \quad & g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g \ . \end{aligned} \tag{5.8.7}$$

The augmented Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) + r \sum_{j=1}^{n_g} \langle \frac{\lambda_j}{2r} - g_j \rangle^2 \,, \tag{5.8.8}$$

where $< a >= max(a, 0)$. The condition of stationarity of $\mathcal{L}$ is

$$\frac{\partial f}{\partial x_i} - 2r \sum_{j=1}^{n_g} \langle \frac{\lambda_j}{2r} - g_j \rangle \frac{\partial g_j}{\partial x_i} = 0 \ . \tag{5.8.9}$$

The exact stationarity condition is

$$\frac{\partial f}{\partial x_i} - \sum_{j=1}^{n_g} \lambda_j^* \frac{\partial g_j}{\partial x_i} = 0 \,, \tag{5.8.10}$$

where it is also required that $\lambda_j^* g_j = 0$. Comparing Eqs (5.8.9) and (5.8.10) we expect an estimate for λ_j^* of the form

$$\lambda_j^* = max(\lambda_j - 2r g_j, 0) \ . \tag{5.8.11}$$

5.9 Projected Lagrangian Methods (Sequential Quadratic Programming)

The addition of penalty terms to the Lagrangian function by multiplier methods converts the optimum from a stationary point of the Lagrangian function to a minimum point of the augmented Lagrangian. Projected Lagrangian methods achieve the same result by a different method. They are based on a theorem that states that the optimum is a minimum of the Lagrangian function in the subspace of vectors orthogonal to the gradients of the active constraints (the tangent subspace). Projected Lagrangian methods employ a quadratic approximation to the Lagrangian in this subspace. The direction seeking algorithm is more complex than for the methods considered so far. It requires the solution of a quadratic programming problem, that is an optimization problem with a quadratic objective function and linear constraints. Projected Lagrangian methods are part of a class of methods known as sequential quadratic programming (SQP)methods. The extra work associated with the solution of the quadratic programming direction seeking problem is often rewarded by faster convergence.

The present discussion is a simplified version of Powell's projected Lagrangian method [20]. In particular we consider only the case of inequality constraints

$$\begin{aligned} &\textbf{minimize} \quad f(\mathbf{x}) \\ &\textbf{such that} \quad g_j(\mathbf{x}) \geq 0, \qquad j = 1, \ldots, n_g \; . \end{aligned} \tag{5.9.1}$$

Assume that at the ith iteration the design is at $\mathbf{x}_i$, and we seek a move direction $\mathbf{s}$. The direction $\mathbf{s}$ is the solution of the following quadratic programming problem

$$\begin{aligned} &\textbf{minimize} \quad \phi(\mathbf{s}) = f(\mathbf{x}_i) + \mathbf{s}^T \mathbf{g}(\mathbf{x}_i) + \frac{1}{2}\mathbf{s}^T \mathbf{A}(\mathbf{x}_i, \boldsymbol{\lambda}_i)\mathbf{s} \\ &\textbf{such that} \quad g_j(\mathbf{x}_i) + \mathbf{s}^T \nabla g_j(\mathbf{x}_i) \geq 0, \qquad j = 1, \ldots, n_g \, , \end{aligned} \tag{5.9.2}$$

where $\mathbf{g}$ is the gradient of f, and $\mathbf{A}$ is a positive definite approximation to the Hessian of the Lagrangian function discussed below. This quadratic programming problem can be solved by a variety of methods which take advantage of its special nature. The solution of the quadratic programming problem yields $\mathbf{s}$ and $\boldsymbol{\lambda}_{i+1}$. We then have

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{s} \, , \tag{5.9.3}$$

where α is found by minimizing the function

$$\psi(\alpha) = f(\mathbf{x}) + \sum_{j=1}^{n_g} \mu_j |min(0, g_j(\mathbf{x}))| \, , \tag{5.9.4}$$

and the μ_j are equal to the absolute values of the Lagrange multipliers for the first iteration, i.e.

$$\mu_j = max[|\lambda_j^{(i)}, \frac{1}{2}(\mu_j^{(i-1)} + |\lambda_j^{(i-1)}|)] \, , \tag{5.9.5}$$

with the superscript i denoting iteration number. The matrix $\mathbf{A}$ is initialized to some positive definite matrix (e.g the identity matrix) and then updated using a BFGS type equation (see Chapter 4).

$$\mathbf{A}_{new} = \mathbf{A} - \frac{\mathbf{A}\Delta\mathbf{x}\Delta\mathbf{x}^T\mathbf{A}}{\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}} + \frac{\Delta\mathbf{l}\Delta\mathbf{l}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} \, , \tag{5.9.6}$$

where

$$\Delta\mathbf{x} = \mathbf{x}_{i+1} - \mathbf{x}_i \; , \qquad \Delta\mathbf{l} = \nabla_x \mathcal{L}(\mathbf{x}_{i+1}, \boldsymbol{\lambda}_i) - \nabla_x \mathcal{L}(\mathbf{x}_i, \boldsymbol{\lambda}_i) \, , \tag{5.9.7}$$

where $\mathcal{L}$ is the Lagrangian function and ∇_x denotes the gradient of the Lagrangian function with respect to $\mathbf{x}$. To guarantee the positive definiteness of $\mathbf{A}$, $\Delta\mathbf{l}$ is modified if $\Delta\mathbf{x}^T\Delta\mathbf{l} \leq 0.2\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}$ and replaced by

$$\Delta\mathbf{l}' = \theta\Delta\mathbf{l} + (1-\theta)\mathbf{A}\Delta\mathbf{x} \, , \tag{5.9.8}$$

where

$$\theta = \frac{0.8\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}}{\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{x}^T\Delta\mathbf{l}} \; . \tag{5.9.9}$$

Example 5.9.1

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\ & g_2 = x_1 - 5.73 \geq 0\,, \\ & g_3 = x_2 - 7.17 \geq 0\ . \end{aligned}$$

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$ where $x_1 = 11.61$, $x_2 = 7.17$ and $f = 47.25$. The gradient of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}, \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix}, \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}, \qquad \mathbf{N} = \begin{bmatrix} 0.1335 & 0 \\ 0.2021 & 1 \end{bmatrix}.$$

We start with $\mathbf{A}$ set to the unit matrix so that

$$\phi(\mathbf{s}) = 47.25 + 3s_1 + \sqrt{3}s_2 + 0.5s_1^2 + 0.5s_2^2\,,$$

and the linearized constraints are

$$\begin{aligned} g_1(\mathbf{s}) &= 0.1335s_1 + 0.2021s_2 \geq 0\,, \\ g_2(\mathbf{s}) &= 5.88 + s_1 \geq 0\,, \\ g_3(\mathbf{s}) &= s_2 \geq 0\ . \end{aligned}$$

We solve this quadratic programming problem directly with the use of the Kuhn-Tucker conditions

$$\begin{aligned} 3 + s_1 - 0.1335\lambda_1 - \lambda_2 &= 0\,, \\ \sqrt{3} + s_2 - 0.2021\lambda_1 - \lambda_3 &= 0\ . \end{aligned}$$

A consideration of all possibilities for active constraints shows that the optimum is obtained when only g_1 is active, so that $\lambda_2 = \lambda_3 = 0$ and $\lambda_1 = 12.8$, $s_1 = -1.29$, $s_2 = 0.855$. The next design is

$$\mathbf{x}_1 = \begin{Bmatrix} 11.61 \\ 7.17 \end{Bmatrix} + \alpha \begin{Bmatrix} -1.29 \\ 0.855 \end{Bmatrix},$$

where α is found by minimizing $\psi(\alpha)$ of Eq. (5.9.4). For the first iteration $\mu_j = |\lambda_j|$ so

$$\psi = 3(11.61 - 1.29\alpha) + \sqrt{3}(7.17 + 0.855\alpha) + 12.8\left|3 - \frac{18}{11.61 - 1.29\alpha} - \frac{6\sqrt{3}}{7.17 + 0.855\alpha}\right|.$$

By changing α systematically we find that ψ is a minimum near $\alpha = 2.2$, so that

$$\mathbf{x}_1 = (8.77, 9.05)^T, \quad f(\mathbf{x}_1) = 41.98, \quad g_1(\mathbf{x}_1) = -0.201 \ .$$

To update $\mathbf{A}$ we need $\Delta\mathbf{x}$ and $\Delta\mathbf{l}$. We have

$$\mathcal{L} = 3x_1 + \sqrt{3}x_2 - 12.8(3 - 18/x_1 + 6\sqrt{3}/x_2)\,,$$

so that

$$\nabla_x\mathcal{L} = (3 - 230.4/x_1^2,\ \sqrt{3} - 133.0/x_2^2)^T\,,$$

and

$$\Delta\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0 = \begin{Bmatrix} -2.84 \\ 1.88 \end{Bmatrix}, \quad \Delta\mathbf{l} = \nabla_x\mathcal{L}(\mathbf{x}_1) - \nabla_x\mathcal{L}(\mathbf{x}_0) = \begin{Bmatrix} -1.31 \\ 0.963 \end{Bmatrix} \ .$$

With $\mathbf{A}$ being the identity matrix we have $\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x} = 11.6$, $\Delta\mathbf{x}^T\Delta\mathbf{l} = 5.53$. Because $\Delta\mathbf{x}^T\Delta\mathbf{l} > 0.2\Delta\mathbf{x}^T\mathbf{A}\Delta\mathbf{x}$ we can use Eq. (5.9.5) to update $\mathbf{A}$

$$A_{new} = I - \frac{\Delta\mathbf{x}\Delta\mathbf{x}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} + \frac{\Delta\mathbf{l}\Delta\mathbf{l}^T}{\Delta\mathbf{x}^T\Delta\mathbf{x}} = \begin{bmatrix} 0.453 & 0.352 \\ 0.352 & 0.775 \end{bmatrix} \ .$$

For the second iteration

$$\begin{aligned}
\phi(\mathbf{s}) &= 41.98 + 3s_1 + \sqrt{3}s_2 + 0.5(0.453s_1^2 + 0.775s_2^2 + 0.704s_1s_2)\,,\\
g_1(\mathbf{s}) &= -0.201 + 0.234s_1 + 0.127s_2 \geq 0\,,\\
g_2(\mathbf{s}) &= 3.04 + s_1 \geq 0\,,\\
g_3(\mathbf{s}) &= 1.88 + s_2 \geq 0 \ .
\end{aligned}$$

We can again solve the quadratic programming directly with the use of the Kuhn-Tucker conditions

$$\begin{aligned}
3 + 0.453s_1 + 0.352s_2 - 0.234\lambda_1 - \lambda_2 &= 0\,,\\
\sqrt{3} + 0.352s_1 + 0.775s_2 - 0.127\lambda_1 - \lambda_3 &= 0 \ .
\end{aligned}$$

The solution is

$$\lambda_1 = 14.31, \quad \lambda_2 = \lambda_3 = 0, \quad s_1 = 1.059, \quad s_2 = -0.376 \ .$$

The one dimensional search seeks to minimize

$$\psi(\alpha) = f(\alpha) + \mu_1 g_1(\alpha)\,,$$

where

$$\mu_1 = max(\lambda_1, \frac{1}{2}(|\lambda_1| + \mu_1^{old})) = 14.31 \ .$$

The one-dimensional search yields approximately $\alpha = 0.5$, so that

$$\mathbf{x}_2 = (9.30, 8.86)^T, \quad f(\mathbf{x}_2) = 43.25, \quad g_1(\mathbf{x}_2) = -0.108\,,$$

so that we have made good progress towards the optimum $\mathbf{x}^* = (9.46, 9.46)^T$. • • •

5.10 Exercises

1. Check the nature of the stationary points of the constrained problem

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = x_1^2 + 4x_2^2 + 9x_3^2 \\ \textbf{such that} \quad & x_1 + 2x_2 + 3x_3 \geq 30\,, \\ & x_2 x_3 \geq 2\,, \\ & x_3 \geq 4\,, \\ & x_1 x_2 \geq 0\;. \end{aligned}$$

2. For the problem

$$\begin{aligned} \textbf{minimize} \quad & f(\mathbf{x}) = 3x_1^2 - 2x_1 - 5x_2^2 + 30x_2 \\ \textbf{such that} \quad & 2x_1 + 3x_2 \geq 8\,, \\ & 3x_1 + 2x_2 \leq 15\,, \\ & x_2 \leq 5\;. \end{aligned}$$

Check for a minimum at the following points: (a) (5/3, 5.00) (b) (1/3, 5.00) (c) (3.97,1.55).

3. Calculate the derivative of the solution of Example 5.1.2 with respect to a change in the allowable displacement. First use the Lagrange multiplier to obtain the derivative of the objective function, and then calculate the derivatives of the design variables and Lagrange multipliers and verify the derivative of the objective function. Finally, estimate from the derivatives of the solution how much we can change the allowable displacement without changing the set of active constraints.

4. Solve for the minimum of problem 1 using the gradient projection method from the point (17, 1/2, 4).

5. Complete two additional moves in Example 5.5.2.

6. Find a feasible usable direction for problem 1 at the point (17, 1/2, 4).

7. Use an exterior penalty function to solve Example 5.1.2.

8. Use an interior penalty function to solve Example 5.1.2.

9. Consider the design of a box of maximum volume such that the surface area is equal to S and there is one face with an area of $S/4$. Use the method of multipliers to solve this problem, employing three design variables.

10. Complete two more iterations in Example 5.9.1.

5.11 References

[1] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index,"Proceedings of IFAC Symposium on Computer Aided Design of Control Systems, Zurich, Switzerland, pp. 113-117, 1979.

[2] Sobieszczanski-Sobieski, J., "A Technique for Locating Function Roots and for Satisfying Equality Constraints in Optimization," NASA TM-104037, NASA LaRC, 1991.

[3] Wolfe, P.. "The Simplex Method for Quadratic Programming," Econometrica, 27 (3), pp. 382–398, 1959.

[4] Gill, P.E., Murray, W., and Wright, M.H., Practical Optimization, Academic Press, 1981.

[5] Dahlquist, G., and Bjorck, A., Numerical Methods, Prentice Hall, 1974.

[6] Sobieszczanski-Sobieski, J., Barthelemy, J.F., and Riley, K.M., "Sensitivity of Optimum Solutions of Problem Parameters", AIAA Journal, 20 (9), pp. 1291–1299, 1982.

[7] Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming—Part I: Linear Constraints", The Society for Industrial and Appl. Mech. Journal, 8 (1), pp. 181– 217, 1960.

[8] Abadie, J., and Carpentier, J., "Generalization of the Wolfe Reduced Gradient Method for Nonlinear Constraints", in: Optimization (R. Fletcher, ed.), pp. 37–49, Academic Press, 1969.

[9] Rosen, J.B., "The Gradient Projection Method for Nonlinear Programming—Part II: Nonlinear Constraints", The Society for Industrial and Appl. Mech. Journal, 9 (4), pp. 514–532, 1961.

[10] Haug, E.J., and Arora, J.S., Applied Optimal Design: Mechanical and Structural Systems, John Wiley, New York, 1979.

[11] Zoutendijk, G., Methods of Feasible Directions, Elsevier, Amsterdam, 1960.

[12] Vanderplaats, G.N., "CONMIN—A Fortran Program for Constrained Function Minimization", NASA TM X-62282, 1973.

[13] Fiacco, V., and McCormick, G.P., Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley, New York, 1968.

[14] Haftka, R.T., and Starnes, J.H., Jr., "Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization", AIAA Journal, 14 (6), pp.718–724, 1976.

[15] Moe, J., "Penalty Function Methods in Optimum Structural Design—Theory and Applications", in: Optimum Structural Design (Gallagher and Zienkiewicz, eds.), pp. 143–177, John Wiley, 1973.

[16] Shin, D.K, Gürdal, Z., and Griffin, O. H. Jr., "A Penalty Approach for Nonlinear Optimization with Discrete Design Variables," Engineering Optimization, 16, pp. 29–42, 1990.

[17] Bertsekas, D.P., "Multiplier Methods: A Survey," Automatica, 12, pp. 133–145, 1976.

[18] Hestenes, M.R., "Multiplier and Gradient Methods," Journal of Optimization Theory and Applications, 4 (5), pp. 303–320, 1969.

[19] Fletcher, R., "An Ideal Penalty Function for Constrained Optimization," Journal of the Institute of Mathematics and its Applications, 15, pp.319–342, 1975.

[20] Powell, M.J.D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", Proceedings of the 1977 Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics, Vol. 630, pp. 144–157, Springer-Verlag, Berlin, 1978.