REPORT NO.
UCB/SESM-86/09

STRUCTURAL ENGINEERING AND
STRUCTURAL MECHANICS

OPTIMIZATION ALGORITHMS FOR STRUCTURAL RELIABILITY ANALYSIS

by

PEI-LING LIU
and
ARMEN DER KIUREGHIAN

Report to Sponsor:
National Science Foundation

JULY 1986

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

OPTIMIZATION ALGORITHMS FOR STRUCTURAL RELIABILITY ANALYSIS

By

Pei-Ling Liu
Graduate Research Assistant

and

Armen Der Kiureghian
Professor

A report on research sponsored by
the National Science Foundation
under Grants No. CEE-8205049 and ECE-8416518

Report No. UCB/SESM-86/09
Structural Engineering and Structural Mechanics
Department of Civil Engineering
University of California, Berkeley

July 1986

Abstract

In certain applications of structural reliability, it is required to find points on the limit-state surface with minimal distance to the origin of a standard space. This problem can be formulated as a constrained optimization program and can be solved by several standard algorithms. In this report, six constrained optimization methods are investigated to determine their applicability for solving the structural reliability problem. These include primal methods, penalty methods, dual methods, Lagrange methods, and two algorithms previously used in reliability analysis. The underlying concepts of each method are introduced and its performance is investigated with due consideration to the properties of the algorithm and the structure of the reliability problem. Four criteria are proposed to serve as the bases of comparison: generality, robustness, efficiency, and capacity.

Among the algorithms studied, the gradient projection method (a primal method) is found to satisfy all four criteria. One widely used algorithm is shown to lack robustness in certain situations. A modification to this algorithm is introduced to enhance its stability. This modified version, although not globally convergent, is found to be more efficient than the gradient projection method in several examples studied.

Acknowledgement

The work presented in this report has been supported by the National Science Foundation under Grants No. CEE-8205049 and ECE-8416518. This support is gratefully acknowledged. Opinions, findings, and conclusions expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

1. Introduction

In the analysis of structural reliability, it is common to define a set of basic random variables, $\mathbf{X} = (X_1, X_2, \cdots, X_n)^T$, to describe uncertain quantities of a structure, such as loads, properties of materials, and dimensions. For each specific failure mode of the structure, a performance function $g(\mathbf{X})$ is formulated such that for an outcome $\mathbf{X} = \mathbf{x}$, the structure fails if $g(\mathbf{x}) < 0$ and it survives if $g(\mathbf{x}) > 0$. The boundary between the two sets, $g(\mathbf{x}) = 0$, is known as the limit-state surface. Accordingly, the probability of failure of the structure in the specified mode is

$$p_f = \int_{g(\mathbf{x})<0} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} \tag{1}$$

where $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint probability density function of $\mathbf{X}$. For n greater than 2 or 3, it is impractical to use numerical integration to evaluate Eq. 1. Therefore, approximate reliability methods have been developed.

In the most widely used reliability method [10], approximations are made in the space of standard, uncorrelated, normal variates, $\mathbf{Y}$, obtained from a transformation of the basic variables

$$\mathbf{Y} = \mathbf{T}(\mathbf{X}) \tag{2}$$

where the transformation $\mathbf{T}$ is determined in terms of the distribution of $\mathbf{X}$ [7]. In the space of $\mathbf{Y}$, denoted as the standard space, approximations to p_f are obtained by replacing the limit-state surface

$$g(\mathbf{T}^{-1}(\mathbf{y})) = G(\mathbf{y}) = 0 \tag{3}$$

with first or second-order approximating surfaces. These surfaces are fitted to the limit-state surface at point(s) with minimal distance to the origin, known as the design point(s). In particular, in the first-order reliability method, the limit-state surface is replaced with its tangent hyperplane at the globally minimal distance point and the first-order approximation to p_f is

$$p_{f1} = \Phi^{-1}(-\beta) \tag{4}$$

where $\Phi(.)$ is the standard normal cumulative probability, and β, known as the reliability index, is the distance from the origin to the design point, as shown in Fig. 1. Better approximations are obtained by fitting a higher-order surface at the design point or by multiple fitting at the locally minimal distance points. See Ref. [10] for further details.

Although the concept of the above approximate reliability method is simple, it may not be easy to find the minimal distance points on the limit-state surface. This is because in real applications the size of $\mathbf{X}$ can be very large and the performance function often is difficult to compute, as it may require algorithmic routines, such as finite element analysis, eigenvalue solutions, or numerical integrations. The determination of the design point involves the solution of the following constrained optimization problem:

$$\begin{aligned} P1: \quad & \textit{minimize} \quad F(\mathbf{y}) \\ & \textit{subject to} \quad G(\mathbf{y}) = 0 \end{aligned} \tag{5}$$

where $F(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{y}$ is the objective function. There are many algorithms available in the literature that can solve this problem. Each algorithm has its advantages and disadvantages, which depend on the attributes of the algorithm and the structure of the problem. The purpose of this report is to examine and compare several existing algorithms and determine their relative merits for use in structural reliability applications. For the purpose of this report, it is assumed that the constraint, $G(\mathbf{y}) = 0$, is continuous and twice differentable.

Before introducing the algorithms, some well-known definitions in optimization are reviewed for the convenience of later discussion.

2. Necessary Conditions for Local Minima

Consider the optimization problem $P1$ defined in Eq. 5. The Lagrangian associated with this optimization program is defined as [9],

$$l(\mathbf{y},\lambda) = F(\mathbf{y}) + \lambda G(\mathbf{y}) \tag{6}$$

where λ is a constant. The tangent plane M of the limit-state surface $G(\mathbf{y}) = 0$ at the optimal point $\mathbf{y}^*$ is given by

$$\mathrm{M} = \{\mathbf{y} : \nabla G(\mathbf{y}^*)(\mathbf{y}-\mathbf{y}^*) = 0\} \tag{7}$$

Using these definitions, the necessary conditions for an optimal point of $P1$ can be stated as follows [9]:

Necessary Conditions: -- Suppose $\mathbf{y}^*$ is a local minimum point of $P1$ subject to the constraint $G(\mathbf{y}) = 0$. Then, there is a λ^*, called the Lagrange multiplier of $P1$, such that

$$\nabla l(\mathbf{y}^*,\lambda^*) = \mathbf{y}^{*T} + \lambda^* \nabla G(\mathbf{y}^*) = 0 \tag{8}$$

Also the Hessian of the Lagrangian

$$\nabla^2 l(\mathbf{y}^*,\lambda^*) = \mathbf{I} + \nabla^2 G(\mathbf{y}^*) \tag{9}$$

is positive semidefinite on M, i.e., $\mathbf{y}^T \nabla^2 l(\mathbf{y}^*,\lambda^*)\mathbf{y} \geq 0$ for all $\mathbf{y}$ on M.

The above conditions are called the second-order necessary conditions for a local minimum point. Eq. 8 alone is known as the first-order necessary condition for a local minimum point.

3. Line Search

Almost all iterative algorithms discussed in this report have the same structure: From a starting point, $\mathbf{y}_k$, one determines a direction of search, $\mathbf{d}_k$ and then searches for a new point along the direction,

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \alpha_k \mathbf{d}_k \tag{10}$$

which minimizes the objective function. Although the new point is the optimal point along $\mathbf{d}_k$, it may not be the optimal point of the entire feasible set. Hence, at the new point this process is repeated until the point satisfies the optimality conditions. The main difference between various optimization algorithms is the rule by which the search direction is selected.

The process of searching for the minimum point along a direction is called a line search. Important line search schemes include the Newton method, quadratic fitting, cubic fitting, the False Position method, and the Armijo rule [9]. The first four methods are based on curve fitting. In these methods, the values or gradients of the objective function at selected points along the direction are calculated. A curve complying with these data is then obtained and its minimum determined. If this minimum point is not the true minimum of the objective function along the direction of movement, the value or gradient of the objective function at this point is computed and a new curve is fitted using the new data.

In practice, in order to reduce the total computation time, an ideal line search is seldom carried out. Instead, a criterion is used to terminate the line search before the exact minimum is found. The criterion ensures that the step size α_k is neither too large nor too small. Armijo's rule [9] is a popular criterion for terminating the line search. According to this rule, a step size α_k is acceptable if it satisfies the following conditions:

$$F(\mathbf{y}_k + \alpha_k \mathbf{d}_k) \leq F(\mathbf{y}_k) + \varepsilon \alpha_k \nabla F(\mathbf{y}_k) \mathbf{d}_k \tag{11}$$

$$F(\mathbf{y}_k + \eta \alpha_k \mathbf{d}_k) > F(\mathbf{y}_k) + \varepsilon \eta \alpha_k \nabla F(\mathbf{y}_k) \mathbf{d}_k \tag{12}$$

where $0 < \varepsilon < 1$ and $\eta > 1$, and $\varepsilon = 0.2$ and $\eta = 2$ are often used. A very simple line search scheme can be constructed using this rule. First, an arbitrary α_k is selected. If the new point satisfies Eq. 11, the value of α_k is replaced with $\eta \alpha_k$ and the process is repeated until the largest α_k satisfying both Eqs. 11 and 12 is found. If the new point does not satisfy Eq. 11, α_k is replaced by α_k / η and the process is repeated until Eqs. 11 and 12 are both satisfied. Although Armijo's rule may not be as efficient as the curve fitting schemes, it is very easy to implement.

4. Global and Local Convergence Properties

An iterative algorithm is said to be globally convergent if for any starting point it is guaranteed that the sequence of points will converge to a solution. Zangwill [17] developed four convergence theorems which define the conditions under which the global convergence of an algorithm is ensured. These conditions will not be introduced here since the proof of global convergence is not the focus of this report.

Local convergence properties are a measure of the ultimate speed of convergence and are generally used to determine the effectiveness of an algorithm. The speed of convergence of an iterative scheme is usually measured by its order of convergence. Let the sequence of vectors $\{\mathbf{y}_k\}$ converge to $\mathbf{y}^*$ and E be a non-negative error function satisfying $E(\mathbf{y}^*) = 0$. The order of convergence of $\{\mathbf{y}_k\}$ is usually defined as the supermum of the non-negative number p satisfying [9]

$$\lim_{k \to \infty} \frac{E(\mathbf{y}_{k+1})}{E(\mathbf{y}_k)^p} < \infty \tag{13}$$

If for the same sequence $\{\mathbf{y}_k\}$,

$$\lim_{k \to \infty} \frac{E(\mathbf{y}_{k+1})}{E(\mathbf{y}_k)} = \rho < 1 \tag{14}$$

the sequence is said to converge linearly to $\mathbf{y}^*$ with a convergence ratio ρ. A linearly convergent algorithm is efficient if the convergence ratio is small. Most algorithms discussed in this report converge linearly. Hence, the convergence ratio ρ will be the basic measure of the convergence rate of each algorithm.

5. Program Structure

In addition to the global and local convergence properties of an algorithm, the structure of the problem also plays an important role in the performance of the algorithm.

The optimization program $P1$ consists of two parts: the objective function and the equality constraint. The objective function is half the inner product of the vector $\mathbf{y}$. This objective function has several advantageous properties. For instance, it is of pure quadratic form, it is convex, and its gradient is simply $\mathbf{y}$. Because of the simplicity of this function, it does not dominate the choice of the minimization algorithm.

As for the constraint part, this problem is simple since it has only one equality constraint, $G(\mathbf{y}) = 0$. This constraint is assumed to be smooth and twice differentiable, but not necessarily convex. However, the value of $G(\mathbf{y})$ is usually difficult to obtain for two reasons: First, the inverse transformation $\mathbf{T}^{-1}$ often is not available in closed form and, consequently, the performance function $G(\mathbf{Y})$ is not an explicit function of $\mathbf{Y}$. Second, even when the inverse transformation is available in closed form, the performance function $g(\mathbf{X})$ may not be an explicit function of $\mathbf{X}$. For example, $g(\mathbf{X})$ may be defined in terms of stresses, which themselves are implicit functions of loads. Hence, for most problems of interest, $G(\mathbf{y})$ is not an explicit function and its computation requires considerable effort. For the same reasons, the gradient vector $\nabla G(\mathbf{y})$ usually cannot be obtained analytically. If a finite difference method is used to approximate $\nabla G(\mathbf{y})$, at least $n+1$ computations of $G(\mathbf{y})$ are necessary. Since in real applications the size of $\mathbf{X}$ can be as large as several hundred, the computation work needed to obtain $\nabla G(\mathbf{y})$ can be overwhelming. Obviously, algorithms that require the computation of the Hessian matrix $\nabla^2 G(\mathbf{y})$ should be considered impractical for this application.

6. Comparison Criteria

The criteria to test the performance of an algorithm should be formulated such that both the problem structure and the characteristics of the optimization scheme are taken into account. Following Lootsma [8], four criteria are

proposed here to evaluate the performance of various nonlinear optimization algorithms for application to structural reliability problems:

Generality -- The generality of a method refers to the types of problems that can be solved by the method. Certain constrained optimization algorithms are restricted to specific types of problems. For example, some methods require that the feasible set of the problem be convex. These methods are rejected by the generality criterion, since $\{\mathbf{y} : G(\mathbf{y}) = 0\}$ is usually not a convex set. Other methods require the Hessian of the constraint, which may be difficult to compute in structural reliability applications.

Robustness -- Robustness refers to the power of a method to solve problems with required accuracy. It incorporates the global convergence properties of an algorithm.

Efficiency -- The efficiency of a method is measured by the effort that it requires to solve a problem. This criterion requires that the algorithm have as high a convergence rate as possible. Also the number of computations of $G(\mathbf{y})$ should be as few as possible.

Capacity -- The capacity of a method refers to the maximum size of basic random variables that can be solved by the method. This criterion demands that the storage needed be as small as possible.

Six nonlinear optimization algorithms are discussed in the following sections; namely, primal methods, penalty methods, dual methods, Lagrange methods, and two methods specifically designed for the reliability problem. The basic ideas of these algorithms are introduced and their performance is evaluated using the four criteria proposed above.

7. Primal Methods

A primal method is an iterative algorithm that solves the original problem

directly (in contrast to other types of methods) by generating a sequence of points which converge to the optimal solution. Each point in the sequence lies in the feasible domain, and the values of the objective function associated with these points decrease monotonically.

There are three commonly used primal methods; namely, the feasible direction method, the gradient projection method, and the reduced gradient method. It can be shown that, for the present problem, the gradient projection method is actually a special case of the reduced gradient method. In spite of this, the two methods are introduced separately, since their underlying concepts are different.

7.1. The Feasible Direction Method

The feasible direction method generates a sequence of points according to the rule [9]

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \alpha_k \mathbf{d}_k \tag{15}$$

where $\mathbf{d}_k$ is a feasible direction at $\mathbf{y}_k$, i.e., there exists an $\bar{\alpha}$ such that $\mathbf{y}_k + \alpha \mathbf{d}_k$ is feasible for all α, $0 \leq \alpha \leq \bar{\alpha}$. The positive scalar α_k is chosen to minimize the objective function along the direction of search.

Since the constraint $G(\mathbf{y}) = 0$ is usually nonlinear, the feasible direction generally does not exist. Hence, this method is not applicable to the reliability problem.

7.2. The Gradient Projection Method

The gradient projection method, originally developed by Rosen [14], is the modified version of the widely known steepest descent method for unconstrained optimization. In the unconstrained case, the steepest descent method generates a new point by a line search along the negative gradient direction at each iterative step. For constrained problems, since each point must remain in

the feasible domain, the moving direction $\mathbf{d}_k$ is taken to be the projection of the negative gradient of the objective function onto the tangent plane of the feasible set (see Fig. 2). Hence, for program $P1$, the new search direction $\mathbf{d}_k$ should satisfy $\nabla G(\mathbf{y}_k)\mathbf{d}_k = 0$ and $\nabla F(\mathbf{y}_k)\mathbf{d}_k < 0$, where $\nabla F(\mathbf{y}_k) = \mathbf{y}_k^T$. From these conditions, $\mathbf{d}_k$ is expressed as

$$\mathbf{d}_k = -\left[\mathbf{I} - \frac{\nabla G(\mathbf{y}_k)^T \nabla G(\mathbf{y}_k)}{|\nabla G(\mathbf{y}_k)|^2}\right]\mathbf{y}_k \tag{16}$$

Because the constraint may not be linear, the new search point could be infeasible. Therefore, a Newton-type correction is often used to pull the point back to the feasible set (see Fig. 2).

A cycle of the gradient projection method can be summarized as follows:

(1) Choose a feasible initial point $\mathbf{y}_0$ and set k = 0.

(2) Compute $\nabla G(\mathbf{y}_k)$.

(3) Calculate $\mathbf{d}_k$ according to Eq. 16.

(4) If $\mathbf{d}_k = \mathbf{0}$, stop. Otherwise, set $i = 0$, then

i) calculate

$$\mathbf{y}_{k+1}^0 = \mathbf{y}_k + \alpha_k \mathbf{d}_k \tag{17}$$

ii) Repeatedly use the following formula until $G(\mathbf{y}_{k+1}^{i+1}) = 0$ is satisfied within the required accuracy:

$$\mathbf{y}_{k+1}^{i+1} = \mathbf{y}_{k+1}^i - \frac{G(\mathbf{y}_{k+1}^i)}{|\nabla G(\mathbf{y}_k)|^2}\nabla G(\mathbf{y}_k)^T \tag{18}$$

(5) Set $\mathbf{y}_k$ to $\mathbf{y}_{k+1}$ and k to $k+1$, and return to (2).

Theoretically speaking, the step size α_k in Eq. 17 should be obtained by exact line search such that $F(\mathbf{y}_{k+1})$ (after the Newton-type correction) is a minimum along that direction. However, it is extremely time-consuming to do exact line search in a nonlinearly constrained problem. Hence, the step size is usually selected based on a simple rule, such as Armijo's rule [9] described earlier.

The gradient projection method is applicable to the structural reliability problem since it solves general constrained optimization problems. When applied to this problem, the gradient projection method is globally convergent, since only one equality constraint exists. Hence it satisfies both the *generality* and *robustness* criteria. In addition, because the gradient projection method uses only the first derivative information of $P1$, only a few n-dimensional vectors, such as $\mathbf{y}_k$ and $\nabla G(\mathbf{y}_k)$, should be stored during the solution phase. Compared with methods that require the storage of the $n \times n$ Hessian matrix, the gradient projection method has a much larger capacity.

To investigate the efficiency of this method, the local convergence properties should be examined. Since this method is the constrained version of the steepest descent method, it can be shown that this method converges linearly and its asymptotic convergence ratio is [9]

$$\rho \leq \left[\frac{r_M - 1}{r_M + 1}\right]^2 \tag{19}$$

where r_M is the condition number, i.e., the ratio of the largest to the smallest eigenvalue, of the Hessian of the Lagrangian at the optimal point restricted to the tangent subspace M. It is clear that the convergence is slowed as r_M increases. The asymptotic convergence ratio, though usually not computed, serves as a theoretical tool to compare various methods. According to the study of Lootsma [8], the gradient projection method is rated as the most efficient method among several general constrained optimization techniques. Hence, the gradient projection method satisfies all the proposed criteria and is appropriate for solving the structural reliability problem.

7.3. The Reduced Gradient Method

The reduced gradient method was originally proposed by Wolfe [16] to solve problems with linear constraints. It was later extended by Abadie and Carpen-

tier [2] to handle nonlinear constrained problems. Its main idea follows from the simplex method [11] where the variables $\mathbf{y}$ are partitioned into dependent variables $\mathbf{y}_1$ of dimension m and independent variables $\mathbf{y}_2$ of dimension $n-m$, where m is the number of constraints. Here, $m = 1$. The definitions of dependent and independent variables are generalized in the reduced gradient method. A vector $\mathbf{t}$, similarly partitioned, is defined at point $\mathbf{y}$ as

$$\begin{aligned} t_1 &= u\, G(\mathbf{y}) \\ \mathbf{t}_2 &= \mathbf{V}^T\mathbf{y} - \mathbf{v} \end{aligned} \tag{20}$$

where u is a constant, $\mathbf{v}$ is an arbitrary $n-1$ vector, and $\mathbf{V}$ is an $n-1 \times n$ matrix so selected that its row vectors and $\nabla G(\mathbf{y})$ span the whole space at point $\mathbf{y}$. Thus, t_1 is the residual of the constraint $G(\mathbf{y}) = 0$, which must be zero if the constraint is satisfied, and $\mathbf{t}_2$ is the reduced set of (n-1) unconstrained, or independent, variables.

Starting from a feasible point, the reduced gradient method first defines $\mathbf{t}$, then takes a steepest-descent step in the reduced space while keeping t_1 fixed. This process is repeated until convergence is achieved. Note that $\mathbf{t}$ is introduced here to illustrate the concept of the algorithm; it is not really calculated. The entire procedure is carried out in the $\mathbf{y}$-space rather than the $\mathbf{t}$-space.

There are several ways of defining matrix $\mathbf{V}$. Each leads to a different version of the reduced gradient method. One of the more popular ways is to select $\mathbf{V}$ such that the $\mathbf{t}$-coordinates form an orthonormal system. If the orthonormal transformation is adopted, it can be shown that the reduced gradient method is in fact identical to the gradient projection method. Even for a different choice of $\mathbf{V}$, these two methods have similar properties. Hence, the performance analysis will not be repeated here.

8. Penalty Methods

Penalty methods are a class of optimization algorithms which transform a

constrained problem into an unconstrained problem by adding a penalty term $c\,P(\mathbf{y})$ to the original objective function, where c is a positive penalty parameter and $P(\mathbf{y})$ is a penalty function which satisfies $P(\mathbf{y}) = 0$ in the feasible region and $P(\mathbf{y}) > 0$ elsewhere. If the standard quadratic penalty function is adopted, the unconstrained penalty problem takes the form

$$P2: \quad \textit{minimize} \quad q(\mathbf{y}) \tag{21}$$

where

$$q(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}c\,G(\mathbf{y})^2 \tag{22}$$

Suppose the penalty parameter c approaches infinity. The minimization process will force the solution to satisfy $G(\mathbf{y}) = 0$ and minimize $\frac{1}{2}\mathbf{y}^T\mathbf{y}$ at the same time. Thus, $P1$ and $P2$ should yield exactly the same solutions. In real implementation, where it is not possible to have $c = \infty$, the solution of $P2$ may still be a good approximation to that of $P1$ if c is sufficiently large.

Once the penalty problem is set up, one can use any unconstrained optimization techniques to solve the problem. The only difference between the penalty problem and general unconstrained problems is that c does not necessarily remain constant throughout the solution phase. The value of c is often chosen to be a small number at the beginning of the analysis and then increased in the subsequent steps.

If the second-order Newton's method is used to solve $P2$, the sequence $\{\mathbf{y}_k\}$ converges quadratically. However, it is very difficult to calculate the Hessian of the objective function in reliability analysis. Therefore, only first-order algorithms can be used. Here, first-order algorithms refer to those methods in which only the first derivatives of the objective function and the constraint are computed. If the steepest descent method is applied, the convergence ratio is governed by [9]

$$\rho \leq \left[\frac{r_p - 1}{r_p + 1}\right]^2 \tag{23}$$

where r_p is the condition number of the Hessian of the Lagrangian at the optimal point,

$$\nabla^2 q(\mathbf{y}^*) = \mathbf{I} + c\, G(\mathbf{y}^*)\nabla^2 G(\mathbf{y}^*) + c\, \nabla G(\mathbf{y}^*)^T \nabla G(\mathbf{y}^*) \tag{24}$$

Note that in Eq. 24, the third term on the right-hand side is a rank one matrix, and the second term is approximately zero at the solution point. Accordingly, $\nabla^2 q(\mathbf{y}^*)$ has $n-1$ unity (or nearly unity) eigenvalues and one large eigenvalue whose magnitude is proportional to c. This matrix is, thus, ill-conditioned since the condition number r_p is very large. It follows that the steepest descent method converges very slowly. To avoid the poor convergence rate, an accelerating technique should be used.

Luenberger [9] proposed an algorithm which combines the penalty function method with the idea of the gradient projection method. To illustrate the combined method, two subspaces are introduced first. Let $M(\mathbf{y}_k)$ be the subspace tangent to the surface $S_k = \{\mathbf{y}: G(\mathbf{y}) = G(\mathbf{y}_k)\}$ at $\mathbf{y}_k$ and $N(\mathbf{y}_k)$ be the orthogonal complement of $M(\mathbf{y}_k)$. In the present case, $N(\mathbf{y}_k)$ is the rank one subspace spanned by $\nabla G(\mathbf{y}_k)$ and $M(\mathbf{y}_k) = \{\mathbf{y}: \nabla G(\mathbf{y}_k)(\mathbf{y}-\mathbf{y}_k) = 0\}$.

The combined method contains two steps in each cycle. The first step is to apply Newton's method over the subspace $N(\mathbf{y}_k)$ to obtain a point of the form

$$\mathbf{w}_k = \mathbf{y}_k + \gamma_k \nabla G(\mathbf{y}_k) \tag{25}$$

where γ_k is the step size. The second step is to take an ordinary steepest descent step from $\mathbf{w}_k$ to obtain $\mathbf{y}_{k+1}$. These two steps are illustrated in Fig. 3. If $\mathbf{w}_k$ is a local minimum point of q in the subspace $N(\mathbf{y}_k)$, $\nabla q(\mathbf{w}_k)$ would be orthogonal to $\nabla G(\mathbf{y}_k)$. And if $G(\mathbf{y})$ is not highly nonlinear, $\nabla G(\mathbf{w}_k)$ is approximately parallel to $\nabla G(\mathbf{y}_k)$. Consequently, one could expect $\nabla q(\mathbf{w}_k)$ to be approximately orthogonal to $\nabla G(\mathbf{w}_k)$. Furthermore, from Eq. 22, $\mathbf{w}_k^T = \nabla q(\mathbf{w}_k) - c\, G(\mathbf{w}_k)\nabla G(\mathbf{w}_k)$. It fol-

lows that $\nabla q(\mathbf{w}_k)$ is approximately equal to the projection of the gradient of $\frac{1}{2}\mathbf{y}^T\mathbf{y}$ onto $M(\mathbf{w}_k)$. Therefore, the second step is similar to a step in the gradient projection method except that $\mathbf{w}_k$ is not a feasible point.

To execute Newton's step, $q(\mathbf{w}_k)$ is first approximated by its Taylor expansion around $\mathbf{y}_k$ up to the quadratic term,

$$q(\mathbf{w}_k) \approx q(\mathbf{y}_k) + \gamma_k \nabla q(\mathbf{y}_k)\nabla G(\mathbf{y}_k) + \frac{1}{2}\gamma_k^2 \nabla G(\mathbf{y}_k)\nabla^2 q(\mathbf{y}_k)\nabla G(\mathbf{y}_k)^T \tag{26}$$

Then, $\mathbf{w}_k$ is calculated by setting $\frac{dq(\mathbf{w}_k)}{d\gamma_k}$ to zero. For large values of c, $\nabla^2 q(\mathbf{y}_k)$ can be closely approximated by $c\,\nabla G(\mathbf{y}_k)^T \nabla G(\mathbf{y}_k)$. Hence, $\nabla^2 q(\mathbf{y}_k)$ need not to be calculated in the implementation. In summary, a cycle of the combined method is as follows:

(1) Calculate

$$\mathbf{d}_k = -\frac{1}{c\,|\nabla G(\mathbf{y}_k)|^4}\nabla G(\mathbf{y}_k)\nabla q(\mathbf{y}_k)^T \nabla G(\mathbf{y}_k)^T \tag{27}$$

(2) Perform a line search to find γ_k which minimizes $q(\mathbf{y}_k + \gamma_k \mathbf{d}_k)$; then set $\mathbf{w}_k = \mathbf{y}_k + \gamma_k \mathbf{d}_k$.

(3) Calculate $\nabla q(\mathbf{w}_k)$.

(4) Perform a line search to find α_k which minimizes $q(\mathbf{w}_k - \alpha_k \nabla q(\mathbf{w}_k))$, and set $\mathbf{y}_{k+1} = \mathbf{w}_k - \alpha_k \nabla q(\mathbf{w}_k)$.

The combined method eliminates the effect of the ill-conditioned Hessian matrix and converges at the same rate as the gradient projection method. It also avoids the computational difficulty associated with the requirement of remaining feasible. Furthermore, as far as the *capacity* criterion is concerned, the combined method is attractive since it does not require the storage of an $n \times n$ matrix, which is necessary in Newton's method or its variants. Since the combined penalty method is applicable to general nonlinear optimization problems, it satisfies the *generality* requirement. Finally, the combined method is

globally convergent. Having all these advantages, this method is suitable for the reliability problem.

Nevertheless, this method is not perfect. In first-order reliability analysis, the failure probability is sensitive to the distance between the origin and the design point. Unfortunately, if the penalty method is adopted, the solution is never exact unless the penalty parameter c approaches infinity. Hence, one may not obtain a solution with required accuracy by the penalty method, and this method may not be as robust as the gradient projection method.

9. Dual Methods

As in linear optimization problems, the constrained problem $P1$ can also be solved by solving its associated dual problem. Algorithms which solve the dual problem instead of the original problem are called dual methods. The dual problem corresponding to $P1$ is [9]

$$D1: \quad \textit{maximze} \quad \varphi(\lambda) \tag{28}$$

where

$$\varphi(\lambda) = \textit{minimum}\,[\,l(\mathbf{y},\lambda)\,] \tag{29}$$

According to the Local Duality Theorem [9], the dual problem $D1$ and the primal problem $P1$ would have the same local solution if the primal problem has a local solution at $\mathbf{y}^*$ with Lagrange multiplier λ^*, and the Hessian of the Lagrangian, $\nabla^2 l(\mathbf{y}^*,\lambda^*)$, is positive definite. There is no guarantee, however, that in reliability problems the Lagrangian is convex near the solution of $P1$. Hence, unless some modification is made, the dual method cannot be applied to the reliability problem.

The augmented Lagrangian method [9,5], or the multiplier method, is a dual method which incorporates the concept of the penalty method to eliminate the limitation on the dual method. Instead of solving $D1$, the augmented Lagrangian method solves the associated dual problem of the following

program:

$$P3: \quad \textit{minimze} \quad F(\mathbf{y}) + \frac{1}{2} c\, G(\mathbf{y})^2$$
$$\textit{subject to} \quad G(\mathbf{y}) = 0 \tag{30}$$

Note that this problem is equivalent to $P1$, since the optimal solution is not altered by the addition of the penalty term. It can be shown that there exists a c^* such that for all $c \geq c^*$, the augmented Lagrangian

$$l_c(\mathbf{y}, \lambda^*) = F(\mathbf{y}) + \lambda^* G(\mathbf{y}) + \frac{1}{2} c\, G(\mathbf{y})^2 \tag{31}$$

has a local minimum at $\mathbf{y}^*$. In other words, if c is sufficiently large, the Lagrangian associated with $P3$ is made convex by the addition of the penalty term, and hence the dual method can be applied. Another important feature of this method is that it only requires c to be greater than c^* to obtain an exact solution. Accordingly, the ill-conditioning of the Hessian of the Lagrangian is resolved in comparison with the standard penalty method.

To use the augmented Lagrangian method, first choose an initial point $\mathbf{y}_0$, an initial penalty parameter c, and an initial multiplier λ_1. Then, set k to 1 and execute the following steps.

(1) Solve the unconstrained problem

$$\varphi_c(\lambda_k) = \textit{minimum}\,[\,l_c(\mathbf{y}, \lambda_k)\,] \tag{32}$$

and set $\mathbf{y}_k$ to be the associated optimal point.

(2) Modify λ_k according to the following formula

$$\lambda_{k+1} = \lambda_k + c\, G(\mathbf{y}_k) \tag{33}$$

(3) Increase c if the constraint violation has not decreased sufficiently from $\mathbf{y}_{k-1}$ to $\mathbf{y}_k$.

(4) Set k to $k+1$ and repeat steps (1) - (4) until optimality is achieved.

Note that the updating process in Eq. 33 is simply a steepest ascent iteration with a constant step size c for maximizing the augmented dual function φ_c.

The augmented Lagrangian method is globally convergent in application to the reliability problem. It is more efficient than the standard penalty method, as mentioned earlier, and if the steepest descent method is used to solve Eq. 32, its capacity is about the same as the combined penalty method. Therefore, it is applicable to the reliability problem. However, this method is difficult to apply, because users have to make initial estimates on c and λ, and these initial values may influence the performance of the algorithm. Hence, for users that are not familiar with optimization schemes, the augmented Lagrangian method may not be appropriate.

10. Lagrange Methods

Consider again the constrained problem $P1$ defined in Eq. 5. The optimal point $\mathbf{y}^*$ must satisfy the first-order necessary conditions as well as the equality constraint. In other words, $\mathbf{y}^*$ must be a solution of the following simultaneous equations:

$$\begin{aligned} \mathbf{y}^T + \lambda \nabla G(\mathbf{y}) &= 0 \\ G(\mathbf{y}) &= 0 \end{aligned} \tag{34}$$

The idea of the Lagrange method is to find $\mathbf{y}^*$ by solving the above equations instead of the original optimization problem. The use of this method for solving the reliability problem was suggested by Shinozuka [15].

Many standard algorithms for solving systems of nonlinear equations are available to solve the Lagrange equations. Unfortunately, as pointed out by Luenberger [9], all these methods have their disadvantages. For instance, the first-order method may converge to an infeasible point; Newton's method requires second-order information; the quasi-Newton method and the modified Newton method require the solution of n simultaneous linear equations in each step to determine the direction of search. All these difficulties suggest that the Lagrange method may not be a good optimization technique for the reliability

problem.

11. The HL-RF Method

A method, originally proposed by Hasofer and Lind [6] for second-moment reliability analysis and later extended by Rackwitz and Fiessler [13] to include distribution information, is currently used in most applications to solve the optimization problem in structural reliability. For brevity, this method is denoted herein as HL-RF method. Unlike the previous methods, the HL-RF method is a specific iterative scheme rather than a class of algorithms, and it only solves problems having the form $P1$.

The best way to explain this method is through a geometrical interpretation. First consider the special case in which $G(\mathbf{y}) = b + \mathbf{a}^T\mathbf{y}$ is a linear function of $\mathbf{y}$. The optimization program $P1$ is now equivalent to computing the shortest distance from the hyperplane $b + \mathbf{a}^T\mathbf{y} = 0$ to the origin. Using the conditions that the solution $\mathbf{y}^*$ lies on the hyperplane and the vector $\mathbf{y}^*$ is orthogonal to the hyperplane, the optimal point $\mathbf{y}^*$ is expressed as

$$\mathbf{y}^* = \frac{1}{|\mathbf{a}|^2}[\mathbf{a}^T\mathbf{y}_0 - G(\mathbf{y}_0)]\mathbf{a} \tag{35}$$

where $\mathbf{y}_0$ is the starting point. Fig. 4 shows how the optimal point is obtained in the bivariate case. In this figure, the vertical axis represents the value of $G(\mathbf{y})$.

Now extend this concept to the general case in which $G(\mathbf{y})$ is nonlinear. The HL-RF method approximates the hypersurface $z = G(\mathbf{y})$ by its tangent plane at the trial point $\mathbf{y}_k$, as shown in Fig. 5. Then an improved point is obtained in parallel to the linear case as if $G(\mathbf{y}_k) + \nabla G(\mathbf{y}_k)(\mathbf{y}-\mathbf{y}_k) = 0$ were the constraint. Hence, one step of the nonlinear version of the HL-RF method is expressed as follows:

$$\mathbf{y}_{k+1} = \frac{1}{|\nabla G(\mathbf{y}_k)|^2}[\nabla G(\mathbf{y}_k)\mathbf{y}_k - G(\mathbf{y}_k)]\nabla G(\mathbf{y}_k)^T \tag{36}$$

In the nonlinear case, the above formula is applied repeatedly until the sequence $\{\mathbf{y}_k\}$ converges.

The HL-RF method has the advantage that, in comparison with other methods, it requires the least amount of storage and computation in each step. In addition, experience shows that for most situations this method not only converges but also converges fast. However, this method may fail to converge in some situations. This is best illustrated for a single-variable case. In that case, using the HL-RF method to find the design point on the limit-state surface amounts to using the Newton-Raphson method to find the root of $G(y) = 0$. It is well known that the Newton-Raphson method may fail to find the roots of a function in certain circumstances. Fig. 6 shows one example in which the sequence of points generated by the Newton-Raphson method get further and further away from the solution point. It is obvious that the HL-RF method will never converge in this case. In the multi-variate case, there are other situations in which the HL-RF method breaks down. For example, suppose the performance function is $G(\mathbf{y}) = y_1 y_2 - d$, and the starting point is $[a, b]^T$. Then, the HL-RF method will generate points that go back and forth between $[a, b]^T$ and $[b, a]^T$, if $[a, b]^T$ falls on either of the two ellipses $y_1^2 + y_2^2 + y_1 y_2 + d = 0$ and $y_1^2 + y_2^2 - y_1 y_2 - d = 0$, as shown in Fig. 7. These facts indicate that the HL-RF method may not be a suitable algorithm for the optimization problem.

There are several variations of the HL-RF method. Parkinson [12] derived a scheme for second-moment reliability analysis which is similar to the method by Hasofer and Lind [6] except that Eq. 36 is expressed in the original space, i.e., in terms of $\mathbf{x}$ and $\nabla g(\mathbf{x})$, and each new point is adjusted along one coordinate to lie on the limit-state surface. It can be shown that this method under certain conditions is not stable. Rackwitz, et al. [1] proposed two modified algorithms to improve the stability of the HL-RF method. In one algorithm the

modified point is obtained between points $\mathbf{w}_k$ and $\mathbf{y}_{k+1}$ in Fig. 5 according to the rule $\bar{\mathbf{y}}_{k+1} = v\,\mathbf{y}_{k+1} + (1-v)\mathbf{w}_k$, where v, $v \le 1$, is a preselected step size. The stability of this modified algorithm for a selected v is not guaranteed. The second modified algorithm involves two sub-iterations: one that uses a pure Newton search to locate a point on the limit-state surface, and another to search along a tangent direction to find a point whose position vector is as parallel as possible to its gradient. No indication as to the convergence properties of this algorithm is given in Ref. [1]. However, it is expected to require more computations than the gradient projection method which is globally convergent.

One possible way to improve the HL-RF method is to introduce a merit function $m(\mathbf{y})$ to monitor the convergence of the sequence. The merit function should be chosen such that it has a global minimum at the solution point of $P1$ and its value decreases in each iteration step. Recall that a local minimum point of $P1$ must satisfy $G(\mathbf{y}^*) = 0$ and $\mathbf{y}^* + \lambda \nabla G(\mathbf{y}^*)^T = 0$, where $\lambda = -\frac{\nabla G(\mathbf{y}^*)\mathbf{y}}{|\nabla G(\mathbf{y}^*)|^2}$. From these conditions, a non-negative merit function can be constructed as

$$m(\mathbf{y}) = \frac{1}{2}\left|\mathbf{y} - \frac{\nabla G(\mathbf{y})\mathbf{y}}{|\nabla G(\mathbf{y})|^2}\nabla G(\mathbf{y})^T\right|^2 + \frac{1}{2}c\,G(\mathbf{y})^2 \tag{37}$$

where c is a positive constant. Obviously, all minimum points of $P1$ are global minimum points of $m(\mathbf{y})$. With this merit function incorporated, a cycle of the modified HL-RF method is as follows:

(1) Compute the direction vector

$$\mathbf{d}_k = \frac{1}{|\nabla G(\mathbf{y}_k)|^2}[\nabla G(\mathbf{y}_k)\mathbf{y}_k - G(\mathbf{y}_k)]\nabla G(\mathbf{y}_k)^T - \mathbf{y}_k \tag{38}$$

(2) Perform line search along $\mathbf{d}_k$ until a sufficient decrease in $m(\mathbf{y})$ is achieved.

(3) Check if $\mathbf{y}_{k+1}$ satisfies the optimality conditions. If not, repeat steps 1-3.

The above merit function is a convenient guide to decide the step size at each iteration, as it is in terms of quantities which are already known. However,

it may not behave well. Specifically, the function may have local minima which are not the solution points of $P1$, and $\mathbf{d}_k$ may not be a descent direction of $m(\mathbf{y})$. Therefore, the global convergence of this modified algorithm may not be guaranteed. Nevertheless, the modification is expected to greatly improve the stability of the original HL-RF method.

12. The Linear Regression Method

The linear regression method, proposed by Ditlevsen [4], is a specific algorithm which solves only the optimization problem in reliability analysis. This method was constructed by observing that $\mathbf{y}^*$ is a design point if and only if

$$\mathbf{y}^* = \mathrm{E}[\mathbf{Y} \mid \nabla G(\mathbf{y}^*)(\mathbf{Y}-\mathbf{y}^*) = 0] \tag{39}$$

where $\mathrm{E}[\mathbf{Y} \mid .]$ denotes the conditional mean of $\mathbf{Y}$. In order to find a point $\mathbf{y}^*$ that satisfies Eq. 39, the linear regression method iterates as follows:

(1) Start from a point $\mathbf{y}_k$ on the limit-state surface.

(2) Determine the value of u such that

$$\mathbf{y}_{k+1} = \mathrm{E}[\mathbf{Y} \mid \nabla G(\mathbf{y}_k)(\mathbf{Y}-\mathbf{y}_k) = u] \tag{40}$$

lies on the limit-state surface.

(3) Check if $\mathbf{y}_{k+1}$ satisfies Eq. 39. If not, repeat the process from $\mathbf{y}_{k+1}$.

This iterative scheme is illustrated in Fig. 8. It should be mentioned that the scheme proposed in Ref. [4] works in the original space. The advantage of expressing this scheme in the standard space is that one can see the parallelism between this method and the gradient projection method by comparing Figs. 2 and 8. In fact, the linear regression method is equivalent to the special case of the gradient projection method in which the step size is always taken to be unity, i.e., no line search is done. Because of this, one could expect the linear regression method to fail to converge in some cases. Take, for example, $G(\mathbf{y}) = y_1 y_2 - d$. It can easily be shown that the linear regression algorithm will

never converge for this example.

13. Numerical Examples

Based on the analysis in the previous sections, the gradient projection method, the combined penalty method, and the augmented Lagrangian method appear to be rather promising in solving the reliability problem. The modified HL-RF method still needs test to verify its usefulness. The original HL-RF method, though not very stable, is also compared in the examples, since it is widely used to solve the reliability problem. These five algorithms are coded into a computer program so that their performance can be further investigated through numerical examples.

Three examples are used to examine the performance of the above algorithms. Because of the limited number and type of examples, the results may not be representative of the overall performance of the algorithms. However, by way of this limited comparison, one still gains some insight into their relative merits.

13.1. Example 1

This example is taken from the reliability analysis of a pipeline where the limit-state surface was generated by least-square fitting. Four statistically independent basic random variables S, W, P, and E are included. The statistical properties of these variables are listed in Table 1. The performance function is

$$\begin{aligned} g(S,W,P,E) = {} & 1.1 - 0.00115\,S\,W + 0.001572\,W^2 + 0.001175\,S^2 \\ & + 0.01347\,WP - 0.07047\,W - 0.005340\,S - 0.01495\,SP \\ & - 0.06105\,WE + 0.07172\,SE - 0.2259\,P + 0.03335\,P^2 \\ & - 0.5585\,PE + 0.9976\,E - 1.339\,E^2 \end{aligned} \tag{41}$$

Starting from the mean point, all algorithms converged, except the HL-RF method which after 100 steps exhibited no trend of convergence. The other four algorithms converged to the same point, $\mathbf{y}^* = [1.318,\ 0.0137,\ 0.3252,$

$0.04376]^T$, and the first-order failure probabilities of the pipeline calculated by these methods are identically $p_{f1} = 0.087$. Table 2 lists the number of iteration steps, the number of computations of the performance function (including those required for computing the gradient vectors with central finite difference formulation), and the CPU time (on VAX 11/750 computer) spent by these four methods. The results in Table 2 are based on a value of $c = 2\times10^5$ for the combined penalty method, an initial value of $c = 10$ augmented Lagrangian method, and $c = 1$ for the modified HL-RF method. However, all three methods were also tested for other values of c. The combined penalty method was very efficient for c less than 2×10^5; however, it yielded solutions with required accuracy only for c greater than 10^4. The augmented Lagrangian method was efficient when the initial value of c was taken to be less than 400. The efficiency and accuracy of the modified HL-RF method only varied slightly for c ranging from 1 to 10^{10}. It is apparent that the performance of the combined penalty and the augmented Lagrangian methods might be sensitive to the selected value of c. This is very undesirable since the optimal value of c varies from problem to problem and is usually unknown.

Table 2 shows that, for the present example, the modified HL-RF method is more efficient than the other three methods. It is noted that the solution time listed in Table 2 may not be meaningful because of the small size of the problem.

13.2. Example 2

In some applications, the performance function $g(\mathbf{x})$ may contain noise, which may arise from errors in numerical routines, such as integration, eigensolutions, or finite element analysis. To examine the performance of the five algorithms for such an unfavorable situation, the performance function

$$g(\mathbf{X}) = X_1 + 2X_2 + 2X_3 + X_4 - 5X_5 - 5X_6$$

$$+ 0.001[\sin(100X_1) + \sin(100X_2) + \sin(100X_3) + \sin(100X_4) + \sin(100X_5) + \sin(100X_6)] \quad (42)$$

is selected which has high-frequency, artificial noise. The six random variables are statistically independent with the properties listed in Table 3.

Starting from the mean point, only the gradient projection method and the modified HL-RF method converged to a solution, $\mathbf{y}^* = [-0.189, -0.353, -0.353, -0.189, 1.90, 1.26, 1.26]$. The other three algorithms failed to converge after 100 iteration steps. The reliability index and the first-order estimate of the failure probability are $\beta = 2.348$ and $p_{f1} = 0.00943$. To check whether the solution obtained is a globally minimum point, the problem was reanalyzed with the noise terms removed. All solutions using different starting points converged to the same point as the case with noise. Hence, it is believed that the solution obtained is a global minimum point.

The required computations by the gradient projection and the modified HL-RF methods for solving the above problem are shown in Table 4. For the latter method, $c = 10$ was selected. The modified HL-RF method appears to be more efficient than the gradient projection method for this problem. However, this result is dependent on the selected value of c. Further investigation with the modified HL-RF method revealed that the efficiency of the method was best for $6 \leq c \leq 35$, which by coincidence included the selected value of c. For values of c outside this range, the convergence of the modified HL-RF method slowed down considerably. For example, for $c = 5$, the method required 15 steps to converge. For this reason, a definitive statement as to the relative efficiency of the two methods cannot be given for this problem.

13.3. Example 3

The reliability of a three-span, five-story, frame structure subjected to horizontal loads is examined (see Fig. 9 and Table 5). There are a total of 21 basic

variables: 3 applied loads, 2 Young's moduli, 8 moments of inertia, and 8 cross-sectional areas. Tables 6 and 7 list the statistical properties of these variables. It is assumed that the structure fails if the horizontal displacement u_1 of the top floor exceeds 0.2 foot. Thus,

$$g(\mathbf{X}) = 0.2 - u_1(\mathbf{X}) \tag{43}$$

The relation for u_1 in terms of X is given in matrix form using standard structural analysis techniques. In contrast to the first two examples, where the gradients were computed using a finite-difference scheme, explicit matrix expressions for the gradients were used in this example [3].

The five methods were used to analyze this problem. Starting from the mean point, the combined penalty method and the augmented Lagrangian method failed to converge after 500 calculations of the performance function. The other three methods (with $c = 50$ for the modified HL-RF method) converged within 10 steps to the same point, as shown in Tables 8 and 9. The required computations by the three methods are shown in Table 8. The gradient projection method appears to be less efficient than the other two methods in terms of the CPU time or the number of computations of the performance function. Further investigation revealed that the modified HL-RF method converges very slowly for $c \leq 1$; but for $c > 11$ its efficiency is almost invariant to the value of c.

14. Summary

In certain applications of structural reliability, it is required to find points on the limit-state surface with minimal distance to the origin of a standard space. This problem can be formulated as a constrained optimization program and can be solved by several standard algorithms. The objective of this report is to examine and compare the attributes of several existing algorithms, considering the specific structure of the reliability problem.

Six constrained optimization methods are discussed in this report: primal methods, penalty methods, dual methods, Lagrange methods, and two algorithms specifically designed for the reliability problem; namely, the HL-RF method and its modifications, and the linear regression method. The underlying concepts of each method are introduced and its performance is investigated with due consideration to the properties of the algorithm and the structure of the reliability problem. Four criteria are proposed to serve as the bases of comparison: generality, robustness, efficiency, and capacity.

Among the algorithms studied, the gradient projection method (a primal method), the combined penalty method, and the augmented Lagrangian method (a dual method) all theoretically satisfy the stated criteria. The HL-RF method, although lacking robustness, possesses several desirable properties. A modified version, therefore, is proposed to improve its stability. The linear regression method is shown to be a special case of the gradient projection method.

The three general optimization algorithms and the HL-RF method and its modification are examined by way of three numerical examples. The results indicate that the gradient projection method and the modified HL-RF method are superior techniques for use in structural reliability analysis. Furthermore, for a smooth limit-state surface, the modified HL-RF method appears to be more efficient than the gradient projection method, and its performance is almost invariant to the value of parameter c.

Because a limited number of examples are tested in this study, the results reported might be biased. The best algorithm will stand out only through continued application of the proposed algorithms.

References

1. in *User's Manual for Structural Reliability Programs CUTALG-FORM-SORM-SYSREL*, Laboratorium fur den Konstruktiven Ingenieurbau, Technische Universitat Munchen, Munich, Germany, 1985.

2. J. Abadie and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in *Optimization*, ed. by R. Fletcher, pp. 37-47, Academic Press, London, 1969.

3. A. Der Kiureghian and J-B. Ke, "Finite-Element Based Reliability Analysis of Frame Structures," in *Proceedings*, Fourth International Conference on Structural Safety and Reliability, Kobe, Japan, May 1985.

4. O. Ditlevsen, "Principle of Normal Tail Approximation," *Journal of the Engineering Mechanics Division*, vol. 107, no. EM6, pp. 1191-1208, ASCE, Dec. 1981.

5. P. E. Gill, W. Murray, and M. H. Wright, in *Practical Optimization*, Academic Press, New York, 1981.

6. A. M. Hasofer and N. C. Lind, "Exact and Invariant Second-Moment Code Format," *Journal of the Engineering Mechanics Division*, vol. 100, no. EM1, pp. 111-121, ASCE, Feb. 1974.

7. M. Hohenbichler and R. Rackwitz, "Non-Normal Dependent Vectors in Structural Safety," *Journal of the Engineering Mechanics Division*, vol. 107, no. EM6, pp. 1227-1238, ASCE, Dec. 1981.

8. F. A. Lootsma, "Comparative Performance Evaluation, Experimental Design, and Generation of Test Problems," in *Computational Mathematical Programming*, NATO ASI series, Series F, vol. 15, Springer-Verlag, Berlin, 1984.

9. D. G. Luenberger, in *Linear and Nonlinear Programming, 2nd Edition*, Addison-Wesley, Reading, Massachusetts, 1984.

10. H. O. Madsen, S. Krenk, and N. C. Lind, in *Methods of Structural Safety*, pp. 66-84, Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

11. K. G. Murty, in *Linear Programming*, John Wiley, New York, 1983.

12. D. B. Parkinson, "Solution for Second Moment Reliability Index," *Journal of Engineering Mechanics*, vol. 104, no. EM5, pp. 1267-1275, ASCE, October 1978.

13. R. Rackwitz and B. Fiessler, "Structural Reliability Under Combined Load Sequences," *Computers and Structures*, vol. 9, pp. 489-494, 1978.

14. J. Rosen, "The Gradient Projection method for Nonlinear Programming, Part II. Nonlinear Constraints," *J. Soc. indust. Appl. Math.*, vol. 9, pp. 514-532, 1961.

15. M. Shinozuka, "Basic Analysis of Structural Safety," *Journal of Structural Engineering*, vol. 109, no. ST3, pp. 721-740, ASCE, March 1983.

16. P. Wolfe, "Methods of Nonlinear Programming," in *Nonlinear Programming*, ed. by J. Abadie, Chapter 6, pp. 97-131, John Wiley, New York, 1967.

17. W. I. Zangwill, in *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Englewood Cliffs, N.J., 1969.

Table 1 Statistical Data of Example 1

Variable	Distribution	Mean	Standard Deviation
S	type-II largest value	10.	5.
W	normal	25.	5.
P	normal	0.8	0.2
E	lognormal	0.0625	0.0625

Table 2 Comparison of Methods for Example 1

Method	No. of steps	CPU (sec)	No. of $G(\mathbf{y})$
Gradient Projection	21	2.4	335
Combined Penalty	21	2.7	452
Augmented Lagrangian	22	2.4	338
Modified HL-RF	15	2.2	261

Table 3 Statistical Data of Example 2

Variable	Distribution	Mean	Standard Deviation
X_1	lognormal	120.	12.
X_2	lognormal	120.	12.
X_3	lognormal	120.	12.
X_4	lognormal	120.	12.
X_5	lognormal	50.	15.
X_6	lognormal	40.	12.

Table 4 Comparison of Methods for Example 2

Method	No. of steps	CPU (sec)	No. of $G(\mathbf{y})$
Gradient Projection	14	3.1	354
Modified HL-RF	7	2.5	234

Table 5 Frame Element Properties

Element	Young's Modulus	Moment of Inertia	Cross Section Area
B_1	E_4	I_{10}	A_{18}
B_2	E_4	I_{11}	A_{19}
B_3	E_4	I_{12}	A_{20}
B_4	E_4	I_{13}	A_{21}
C_1	E_5	I_6	A_{14}
C_2	E_5	I_7	A_{15}
C_3	E_5	I_8	A_{16}
C_4	E_5	I_9	A_{17}

Table 6 Statistical Data of Example 3

Variable	Distribution	Mean	Standard Deviation
P_1	rayleigh	30.00	9.00
P_2	rayleigh	20.00	8.00
P_3	rayleigh	16.00	6.40
E_4	normal	454000.00	40000.00
E_5	normal	497000.00	40000.00
I_6	normal	0.94	0.12
I_7	normal	1.33	0.15
I_8	normal	2.47	0.30
I_9	normal	3.00	0.35
I_{10}	normal	1.25	0.30
I_{11}	normal	1.63	0.40
I_{12}	normal	2.69	0.65
I_{13}	normal	3.00	0.75
A_{14}	normal	3.36	0.60
A_{15}	normal	4.00	0.80
A_{16}	normal	5.44	1.00
A_{17}	normal	6.00	1.20
A_{18}	normal	2.72	1.00
A_{19}	normal	3.13	1.10
A_{20}	normal	4.01	1.30
A_{21}	normal	4.50	1.50

Note: The units of P_i, E_i, I_i, and A_i are kips, kips/ft^2, ft^4, and ft^2, respectively.

Table 7 Correlation Coefficients of the Random Variables

	P_1	P_2	P_3	E_4	E_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}	A_{21}
P_1	1.0																				
P_2	0.5	1.0																			
P_3	0.5	0.5	1.0																		
E_4	0.	0.	0.	1.0																	
E_5	0.	0.	0.	0.9	1.0																
I_6	0.	0.	0.	0.	0.	1.0															
I_7	0.	0.	0.	0.	0.	0.13	1.0														
I_8	0.	0.	0.	0.	0.	0.13	0.13	1.0													
I_9	0.	0.	0.	0.	0.	0.13	0.13	0.13	1.0												
I_{10}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	1.0											
I_{11}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	1.0										
I_{12}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	0.13	1.0									
I_{13}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0								
A_{14}	0.	0.	0.	0.	0.	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0							
A_{15}	0.	0.	0.	0.	0.	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0						
A_{16}	0.	0.	0.	0.	0.	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0					
A_{17}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0				
A_{18}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0			
A_{19}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0		
A_{20}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0	
A_{21}	0.	0.	0.	0.	0.	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.95	0.13	0.13	0.13	0.13	0.13	0.13	0.13	1.0

Table 8 Comparison of Methods for Example 3

Method	No. of steps	CPU (sec)	No. of $G(\mathbf{y})$	No. of $\nabla G(\mathbf{y})$
Gradient Projection	6	65.6	15	6
HL-RF	7	38.4	7	7
Modified HL-RF	7	38.8	7	7

Table 9 Solutions of Example 3

Basic Variables	Design Point	Basic Variables	Design Point
P_1	7.665e+01	I_{12}	2.102e+00
P_2	3.637e+01	I_{13}	2.690e+00
P_3	2.842e+01	A_{14}	3.166e+00
E_4	4.208e+05	A_{15}	3.691e+00
E_5	4.650e+05	A_{16}	4.991e+00
I_6	9.010e-01	A_{17}	5.549e+00
I_7	1.271e+00	A_{18}	2.190e+00
I_8	2.333e+00	A_{19}	2.118e+00
I_9	2.867e+00	A_{20}	2.880e+00
I_{10}	1.088e+00	A_{21}	3.890e+00
I_{11}	1.248e+00		
β	2.968	P_f	0.0015

Note: The units of P_i, E_i, I_i, and A_i are kips, kips/ft^2, ft^4, and ft^2, respectively.

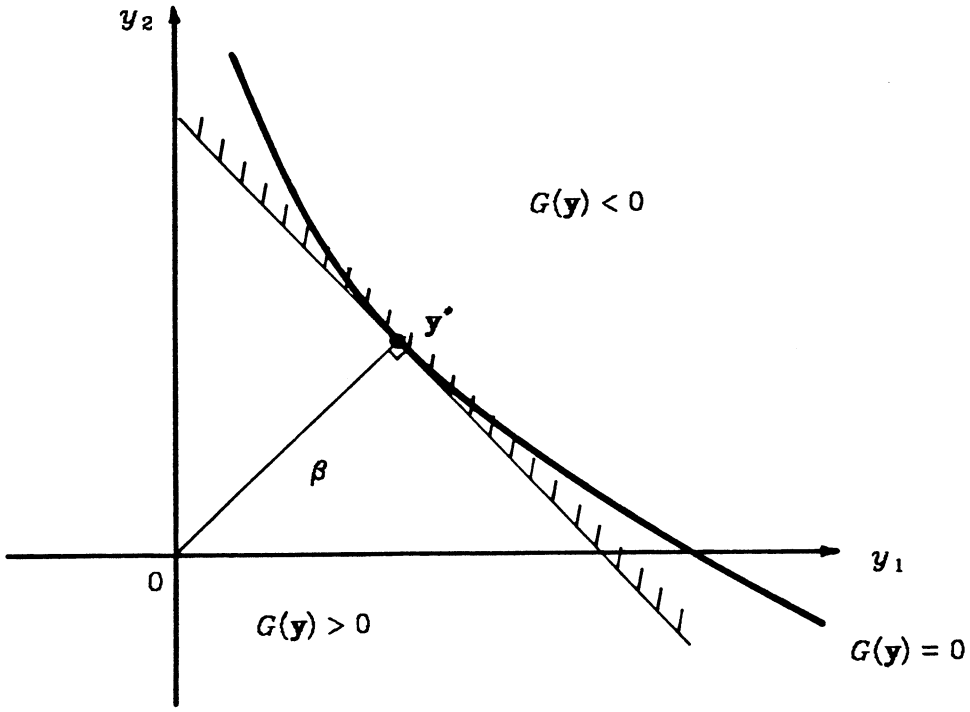

Fig. 1 First-Order Reliability Analysis

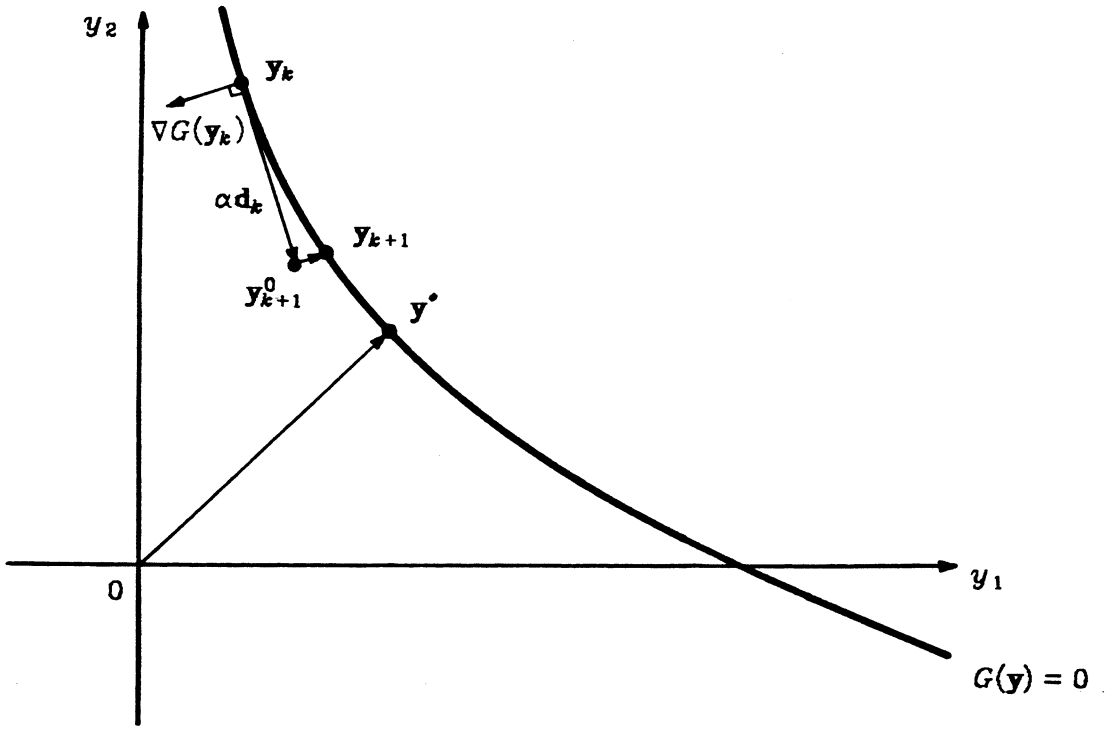

Fig. 2 Gradient Projection Method

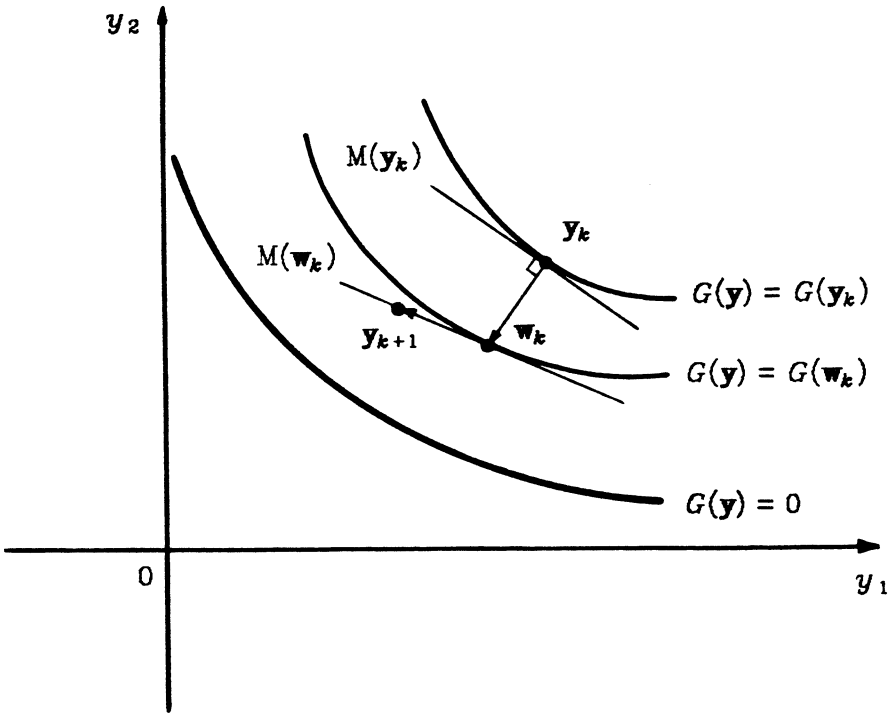

Fig. 3 Combined Penalty Method

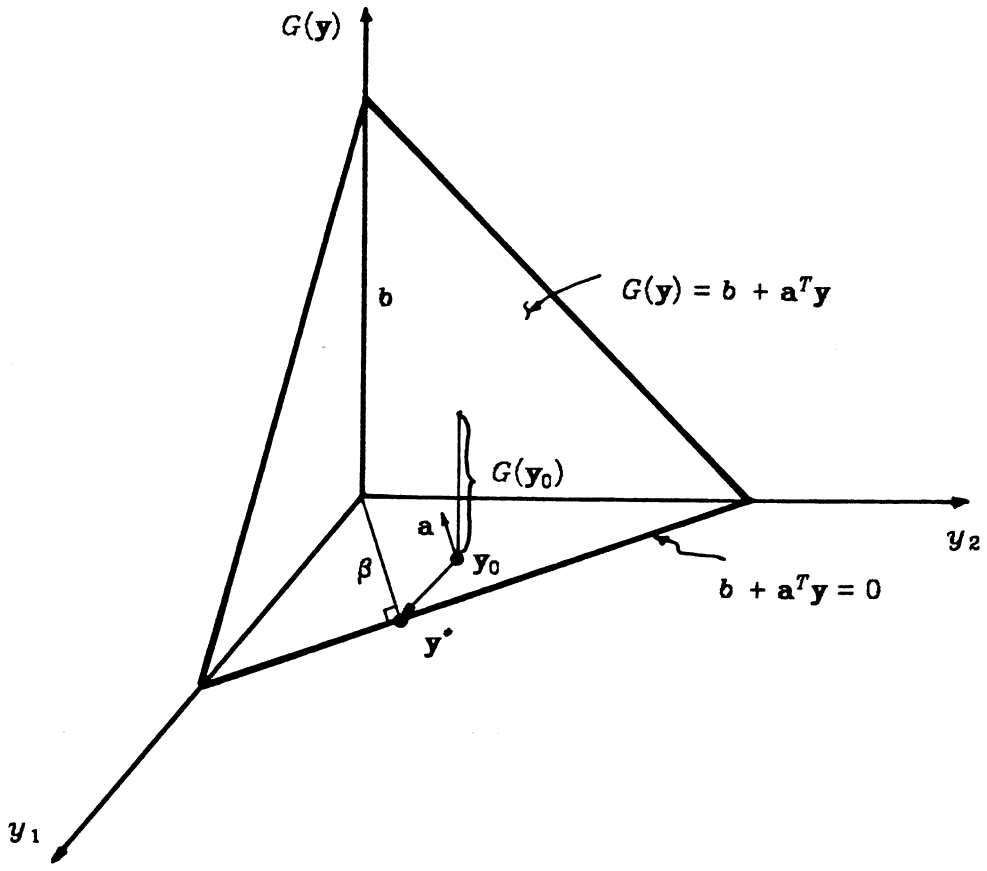

Fig. 4 HL-RF Method - Linear Performance Function

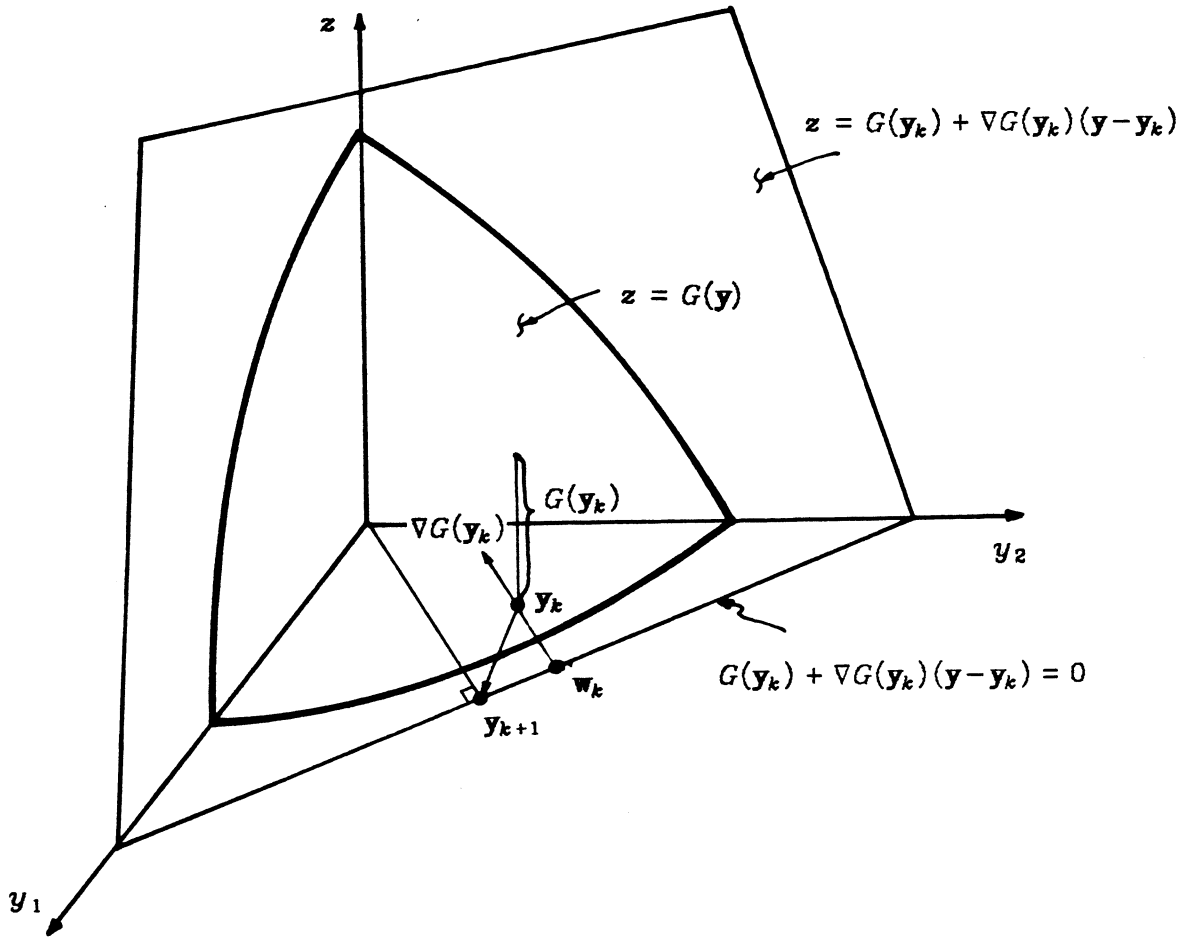

Fig. 5 HL-RF Method - Nonlinear Performance Function

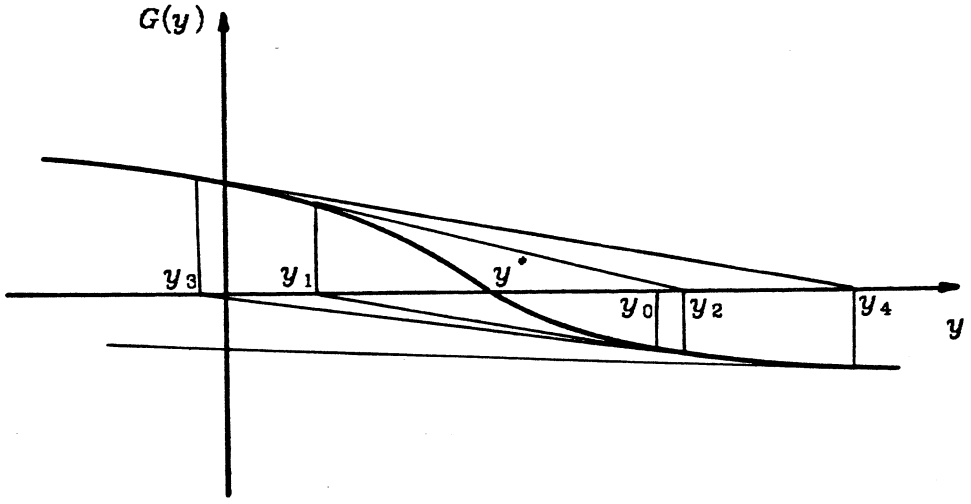

Fig. 6 Example in Which the Newton-Raphson Method Diverges

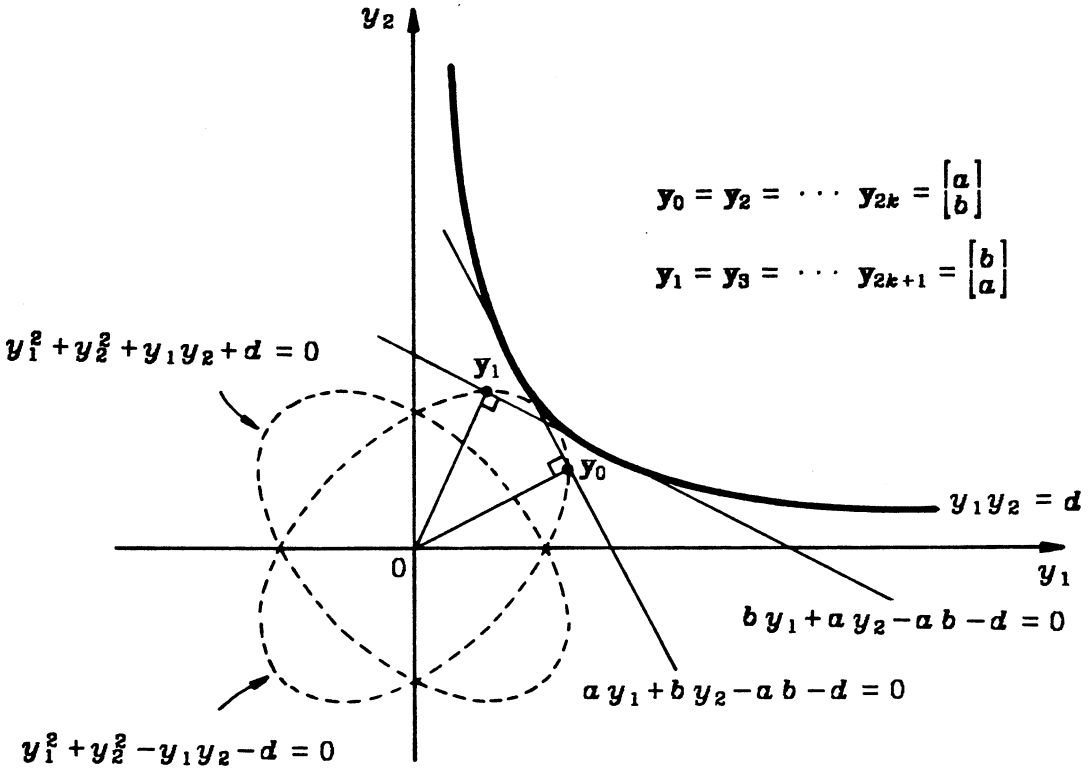

Fig. 7 Example in Which the HL-RF Method Fails

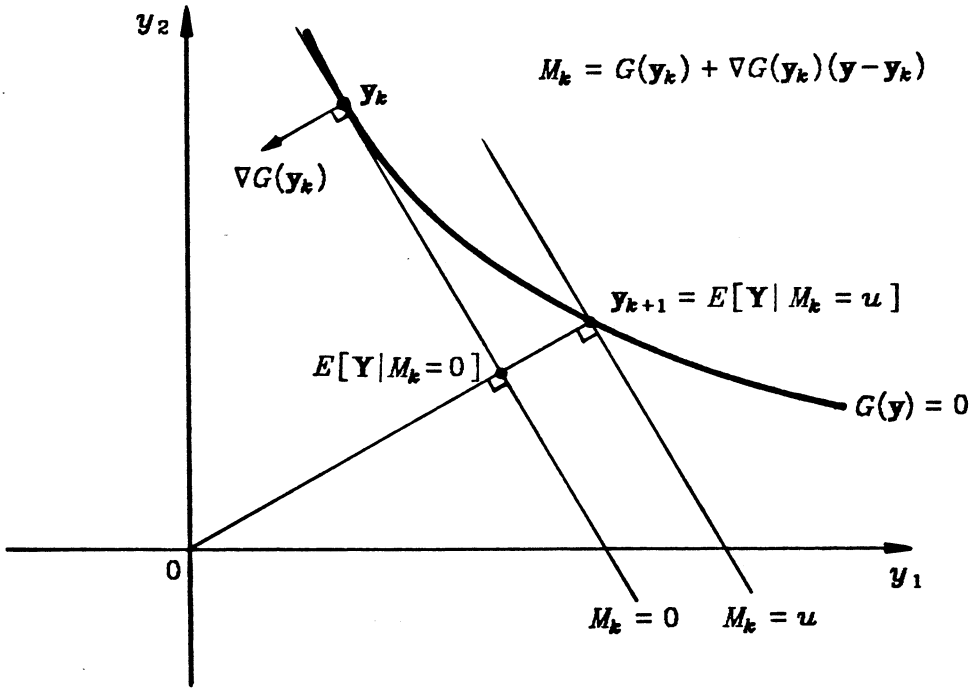

Fig. 8 Linear Regression Method

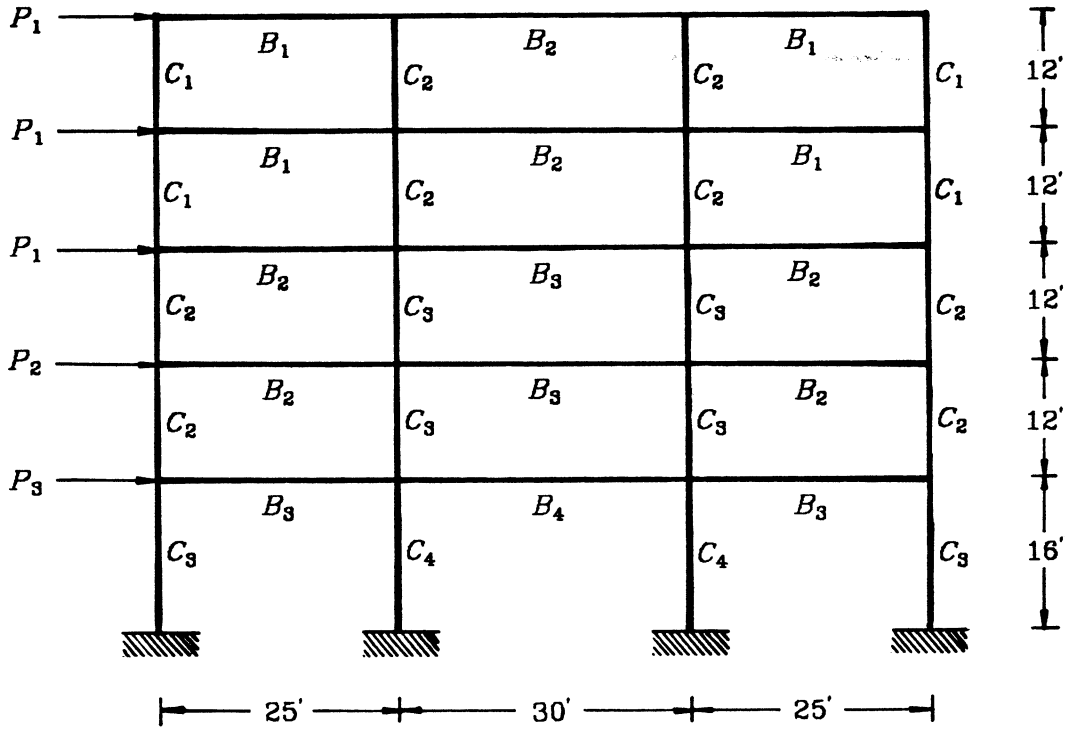

Fig. 9 Frame Configuration