REPORT NO.
UCB/SEMM-89/05

**STRUCTURAL ENGINEERING,
MECHANICS AND MATERIALS**

FINITE-ELEMENT RELIABILITY METHODS FOR GEOMETRICALLY NONLINEAR STOCHASTIC STRUCTURES

by

PEI-LING LIU
and
ARMEN DER KIUREGHIAN

Report to Sponsor:
National Science Foundation

JANUARY 1989

**DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA AT BERKELEY
BERKELEY, CALIFORNIA**

FINITE-ELEMENT RELIABILITY METHODS FOR GEOMETRICALLY NONLINEAR STOCHASTIC STRUCTURES

By

Pei-Ling Liu

and

Armen Der Kiureghian

A report on research supported by
the National Science Foundation under
Grants No. CEE-8205049 and CES-8618905

Report No. UCB/SEMM-89/05
Structural Engineering, Mechanics, and Materials
Department of Civil Engineering
University of California, Berkeley

January 1989

ABSTRACT

A general framework for first- and second-order reliability analysis of structures with geometrical nonlinearity is presented. The structure is either linear or nonlinear elastic, and is subjected to static loads. The material properties, geometry, and external loads of the structure are considered as random variables or random fields. The failure criteria of the structure is expressed in terms of limit-state functions.

Four major steps are involved in the first- and second-order reliability methods: (1) selection of probability models for random variables and random fields, and representation of the latter in terms of random variables; (2) transformation of the random variables into a set of independent, standard normal variates; (3) iterative solution of a constrained optimization problem to find the nearest point on the limit-state surface to the origin in the standard normal space; and (4) integration of the failure probability using a first- or second-order approximation of the limit-state surface. In the course of the optimization programming, the structural response and its gradient with respect to the basic random variables are required at each iteration step. The finite element method is used to compute these two quantities. Analytical formulas for the gradient of the response are derived to improve the efficiency and accuracy of the reliability computation. The formulas are in terms of the tangent stiffness matrix and the initial load stiffness matrix, which are readily available if Newton's method is used to solve for the structural responses. No iterations are involved in the computation of the gradient.

A general-purpose reliability code, CALREL-FEAP, is developed to perform the finite-element reliability analysis. The reliabilities of a built-up column and a stochastic plate with a random hole are studied using this code. Sensitivities of the failure probabilities with respect to parameters in the probability distribution functions or in the limit-state functions are examined. The usefulness of these sensitivity measures in structural design process is demonstrated.

ACKNOWLEDGMENTS

This research work was supported by the National Science Foundation under Grants No. CEE-8205049 and CES-8618905. Support was also received from the Mitsubishi Research Institute, Japan. These supports are gratefully acknowledged.

The first author is also grateful for having received the Genevieve McEnerney Fellowship, William H. and Helena S. Popert Fellowship, Wilson and Albert M. Flagg Scholarship, and Stanley M. Tasheira Scholarship during the course of her studies at the University of California, Berkeley.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 General Remarks

There are various sources of uncertainty in structural design. External loads, environmental factors, material properties and geometry of structures all usually possess inherent variability. Incomplete statistical data, lack of understanding of structural behavior, and simplified statistical and mechanical models give rise to further uncertainties. For a proper assessment of structural safety, it is essential to take these uncertainties into account in the analysis of structures.

The finite element method is undoubtedly the most powerful tool for the analysis of complex structures. It has been proven to be effective in many areas of structural engineering. However, despite its popularity, the traditional finite element method is limited to dealing with deterministic problems. As such, it is inadequate for the assessment of structural safety.

Methods of structural reliability analysis, employing concepts of probability and statistics, account for the uncertain nature of structures and their environments. In recent years, robust and accurate techniques for computing probabilities of failure have been developed. However, the application of these techniques to structures is severely limited without a tool to efficiently analyze the behavior of complex structures.

The main objective of this study is to combine the finite element method and the reliability method into a general framework for the reliability assessment of structures. The combined methodology is denoted here as *the finite-element reliability method.* Special attention is given to the application of this methodology to geometrically nonlinear, elastic structures. Another goal of this study is to develop a general-purpose computer code to facilitate the application of the finite-element reliability method to real problems.

1.2 Approach of Analysis

The first- and second-order reliability methods (FORM and SORM) and the finite element method are the main constituents of the proposed finite-element reliability method. In the analysis, the FORM/SORM reliability methods provide the basis for modeling and analysis of uncertainties and computation of probabilities, and the finite element method provides the necessary computational framework for analyzing complex structures.

There are five major components in the finite-element reliability method: selection of proper probability models and the associated probability transformations; discretization of random fields (used in describing structural properties or external loads with spatial variability); search of the design point; computation of structural responses and their gradients; and evaluation of the failure probability. These issues are discussed individually in the present report. Existing approaches addressing these problems are described and compared. For some problems, new techniques are proposed to improve the efficiency, stability, and feasibility of FORM and SORM in the context of the finite element application.

An equally important issue in the finite-element reliability is the interpretation of analysis results. In two example applications, failure probabilities are evaluated for a range of mean axial forces, displacement thresholds, and correlation lengths of material-property random fields. Measures of sensitivity of the reliability with respect to various parameters are obtained and their use in structural design are discussed.

1.3 Literature Review

Structural analyses which combine the finite element method and the theory of probability or statistics were initiated in 1970's. Such analysis techniques are usually denoted as the probabilistic or stochastic finite element analysis. Basically, there are three types of stochastic finite element methods; namely, simulation methods, perturbation methods, and reliability methods.

The direct Monte Carlo simulation method was used in many early works in stochastic finite element analysis [e.g., 5, 62]. In this brute-force method, deterministic analysis is

carried out for a series of parameters generated in accordance with their probability distribution. The desired statistics of the response quantities, such as the mean, variance, and exceedance probabilities, are then evaluated based on the generated sample. The Monte Carlo simulation method has the advantage that it is adaptable to all types of problems and the results can be obtained to desired accuracy. However, for practical problems with many random variables or small probabilities this procedure is usually too expensive, since a large number of solutions are needed to obtain reliable results. To reduce the computation cost, Shinozuka and Dasgupta [61] and Yamazaki et al. [76] introduced the expansion Monte Carlo simulation method, in which the Neumann expansion technique is used in deriving the finite element solution for the response variation. Since only the mean stiffness matrix needs to be decomposed with this formulation, computation time is reduced significantly. This method was used in Refs. 61, 76, and 63 to compute the means and covariance matrix of the static and dynamic response quantities for structures whose material properties possessed spatial variability. Although this method is claimed to be applicable to nonlinear problems in Ref. 61, such application is doubtful.

The perturbation method involves the first- or second-order Taylor series expansion of the terms in the governing equation of the structure around the mean values of the random variables. The variation of the response is then obtained by solving a set of deterministic equations. In this method, random variables are characterized by the first and second statistical moments. No distribution information is required. The aim of this method is to compute the first and second moments of the response quantities. The first-order perturbation method was used by Collins and Thomson [8] to obtain the statistics for eigenvalues and eigenvectors of dynamic problems. Dendrou and Houstis [11,12] applied the method to field problems for estimation of the statistical properties of the solution process. Baecher and Ingra [6] and Righetti and Harrop-Williams [54] applied the same method to geotechnical problems. Handa and Anderson [27] applied a similar approach to a beam and a truss structure to estimate the first two statistical moments of structural displacements and stresses. Hisada et al. applied the first- and second-order perturbation methods to

linear and nonlinear problems [29,30], structures with uncertain foundations [51], stresses caused by structural misfits [32], and eigenvalue and dynamic problems [31]. Similar approaches were adopted by Liu et al. with applications to dynamic analysis of nonlinear continua [41,42]. A stochastic finite-element technique was developed by Lawrence [36] using an energy-based approach. Similar to the perturbation method, the Taylor series expansion is used in this method to derive the variation of response quantities. The perturbation methods have the advantage that they can be applied to a large variety of problems. However, these methods yield satisfactory results only when the variations of the random variables are small. Furthermore, the perturbation methods are not suitable for use in the safety assessment of structures. This is because the failure probability is usually sensitive to the tail of the probability distribution. Since the perturbation methods expand the governing equation at the mean values of the random variables, the error in the estimation of the failure probability can be very large.

The reliability methods aim at evaluating the failure probabilities of structures. Failure criteria of structures are specified in terms of limit-state functions, which define surfaces separating the safe and failure sets. The first-order reliability method was used by Der Kiureghian et al. [13,15,16,19] for static analysis of linear structures with random properties, and by Igusa and Der Kiureghian [35] for dynamic analysis. Recently, Arnbjerg-Nielsen and Bjerager [3] applied the same method to nonlinear frames and used omission sensitivity factors to improve the efficiency of the computation. The first-order method involves transforming the random variables into the standard normal space and replacing the limit-state surface by a hyperplane at the point with the minimum distance to the origin. Liu and Der Kiureghian [39] used the first- and second-order reliability methods for static analysis of geometrically nonlinear trusses. The second-order method is similar to the first-order method except that the limit-state surface is replaced by a second-order surface. As such, the second-order method gives a better estimate of the failure probability, but it is more costly. A detailed description of these two methods is given in Chapter 2. Tani et al. [65] applied the first-order second-moment reliability method to

laminated plate, and Liu et al. [43] used the same method for fracture mechanics. This method is similar to the first-order method mentioned above except that the random variables are characterized only by their first two statistical moments instead of their probability distributions. As a result, the failure probability by this method is ad hoc and crude.

An important issue in first- and second-order finite-element reliability is the computation of the response gradients. For linear structures under static loads, several investigators [e.g., 4,19] derived analytical expressions for the gradient by taking the derivative of the equilibrium equation with respect to the design parameter or the basic random variable. Dems and Mróz derived the gradients of displacement, stress, and strain responses of linear structures in term of the solution of the primary and an adjoint structures [9,10]. This method, however, is difficult to implement. To avoid explicit differentiation of the tangent stiffness matrix, Dias and Nagtegaal [20] and Dias and Nakazawa [21] proposed an iterative strategy to compute the gradient. This method may not be as efficient as the analytical approach, and it may fail to converge if the variation of the random variable is too large. Nevertheless, it has the advantage that it is applicable to the case where a consistent tangent stiffness matrix is not available. Although this method has only been applied to linear structures, it is claimed to be applicable to mildly nonlinear problems as well. For geometrically nonlinear, elastic structures, Wu and Arora [74] and Ryu et al. [57] derived formulas for the response gradient in terms of the secant stiffness matrix. Since for nonlinear structures the secant stiffness matrix is not readily available, these formulas are of little practical use. Ryu et al. [57] made an interpretation of these formulas in terms of the tangent stiffness matrix. Their results are consistent with the formulas derived in this study.

Other relevant issues, such as optimization schemes for finding the design point, the discretization of random fields, and reduction of random variables are reviewed at length in subsequent chapters.

1.4 Outline of Report

This report is organized as follows:

Chapter 2. First and Second Order Reliability Methods

The chapter begins with a general formulation of the structural reliability problem. The first- and second-order reliability methods (FORM and SORM) for approximate estimation of the failure probability are described. In these methods, the basic random variables need to be transformed into the standard normal space. Hence, probability transformations for different distributions of the basic random variables are described. An important feature of FORM is that it provides sensitivities of the first-order failure probability with respect to a given set of parameters. The formulas for these sensitivity measures are given and their applications are briefly discussed.

Chapter 3. Optimization Algorithms for Reliability Analysis

FORM and SORM require the solution of a constrained optimization algorithm to find the point with the minimum distance from the limit-state surface to the origin in a standard normal space. In this chapter, several gradient-based optimization schemes are compared for their suitability in this application, and a new algorithm is proposed.

Chapter 4. Finite Element Formulation of Geometrically Nonlinear Elastic Structures

As mentioned in Section 1.2, the finite element method serves as a means for analyzing structures in the finite-element reliability method. In this chapter, continuum and finite element formulations of geometrically nonlinear problems are described, and analytical expressions for the response gradient are derived.

Chapter 5. Computational Aspects of Finite Element Reliability Analysis

Several computational issues are discussed in this chapter. First, the adjoint method is introduced to improve the efficiency in the computation of the gradient of the limit-state function. Then, three issues regarding random fields are discussed: a convenient model for

describing non-Gaussian random fields; general guidelines for selecting random field mesh size; and six methods for describing random fields in terms of random variables. Finally, two methods are presented for reducing the number of basic random variables in the optimization process of FORM and SORM.

Chapter 6. Computer Implementation

This chapter introduces the general-purpose finite-element reliability code CALREL-FEAP. The features of the two constituent routines, CALREL and FEAP, are presented and the linkage between the two is described. More details on the code are provided in Appendix B.

Chapter 7. Application Examples

Two application examples are presented in this chapter to illustrate the proposed finite-element reliability method and the capabilities of the CALREL-FEAP code. Special attention is given to reliability sensitivity analysis and its potential use in structural design.

CHAPTER 2

FIRST AND SECOND ORDER RELIABILITY METHODS

2.1 Introduction

The structural reliability problems of interest here are based on two fundamental assumptions. First, the uncertainties in the structure and its environment are assumed to be modeled by random variables. These may include the variabilities in the material properties, the structural shape, and the external loads. The set of basic random variables describing these uncertainties are represented by a vector $\mathbf{V} = [V_1,...,V_n]^T$. Second, the structure may fail in any of a finite number of modes, and with respect to each mode it is either in a safe state or in a failure state. For each mode, the state of the structure is determined by the value of a limit-state function.

Structural limit states are usually defined in terms of structural responses and response thresholds. The response thresholds may be included in vector $\mathbf{V}$ if they are random in nature. The structural responses, denoted by a vector $\mathbf{S}$, are functions of the basic random variables, i.e.,

$$\mathbf{S} = \mathbf{S}(\mathbf{V}) \tag{2.1}$$

The mapping from $\mathbf{V}$ to $\mathbf{S}$ is denoted the *mechanical transformation* of the structure. An analytical expression for this transformation is available only in trivial cases. Other than that, the transformation is in an algorithmic form, such as a finite element code.

The limit-state function for a given failure mode is expressed as an explicit function of $\mathbf{V}$ and $\mathbf{S}$, i.e., $g(\mathbf{V},\mathbf{S})$. In view of the mechanical transformation in Eq. 2.1, this is an implicit function of the basic variables $\mathbf{V}$. In general, it is possible to formulate the limit-state function such that the set $g(\mathbf{v},\mathbf{s}) > 0$ defines the safe state and the set $g(\mathbf{v},\mathbf{s}) \leq 0$ defines the failure state. The boundary between the two sets, $g(\mathbf{v},\mathbf{s}) = 0$, is called the limit-state surface. Thus, the probability of failure in the considered mode is given by

$$P_f = \int_{g(\mathbf{v},\mathbf{s})\leq 0} f_{\mathbf{V}}(\mathbf{v})\, d\mathbf{v} \tag{2.2}$$

where $f_{\mathbf{V}}(\mathbf{v})$ is the joint probability density function (PDF) of **V**. The evaluation of the failure probability for a single failure mode is called component reliability analysis. The probability due to the combination of failure modes requires a system reliability analysis, for which the component analysis is the basic element. This presentation will focus on component reliability analysis. Details on system reliability analysis can be found in Refs. 22 and 49.

Although the expression for the failure probability in Eq. 2.2 appears simple, in practice it is almost impossible to perform the multi-fold integral directly, either analytically or numerically. This is because the number of basic random variables is often very large (say several tens or hundred) and the integration domain is implicit in **V**. Hence, alternative methods are needed. The difference between various reliability methods lies in the approach used for evaluating the multi-fold integral in Eq. 2.2. This study adopts the first- and second-order reliability methods to assess the reliability of geometrically nonlinear, elastic structures within a finite element framework. These two methods are reviewed in detail in the following sections.

2.2 The First-Order Reliability Method

In the first-order reliability method (FORM), the basic random variables **V** are transformed into a set of statistically independent, standard normal variates

$$\mathbf{Y} = \mathbf{Y}(\mathbf{V}) \tag{2.3}$$

This mapping, denoted the *probability transformation*, exists for continuous random variables and is invertible. Details of this transformation are discussed in Section 2.4. In terms of the standard normal variates, the limit-state function is denoted

$$G(\mathbf{Y}) = g(\mathbf{V}(\mathbf{Y}), \mathbf{S}(\mathbf{V}(\mathbf{Y}))) \tag{2.4}$$

Since the probability transformation is a one-to-one mapping, the probability content in the failure domain is preserved in the standard normal space. Hence, in this space the failure probability may be expressed by

$$P_f = \int_{G(\mathbf{y})\le 0} \phi(\mathbf{y})\, d\mathbf{y} \tag{2.5}$$

where $\phi(\mathbf{y})$ denotes the standard normal density of $\mathbf{Y}$.

The standard normal space has three useful properties: (1) The probability density function is rotationally symmetric about the origin, (2) it decays exponentially with the square of the distance from the origin, and (3) the probability content of some simple sets in this space are available for arbitrary dimensions. From the first two properties it follows that the contribution to the integral in Eq. 2.5 comes mostly from the region around the point on the limit-state surface that is nearest to the origin. The third property is used to construct simple approximations to the probability integral. These approximations are achieved by replacing the limit-state surface with an approximating surface fitted at the nearest point for which the probability content is known.

In FORM, the limit-state surface in the standard normal space is replaced by its tangent hyperplane at the point nearest to the origin (see Fig. 2.1). This point, denoted $\mathbf{y}^*$, is known as the design point, and the distance from the origin to the point, denoted β, is known as the reliability index. The first-order estimate of the failure probability is given by

$$P_{f1} = \int_{\nabla G(\mathbf{y}^*)(\mathbf{y}-\mathbf{y}^*)\le 0} \phi(\mathbf{y})\, d\mathbf{y} = \Phi(-\beta) \tag{2.6}$$

where $\nabla G(\mathbf{y}^*)$ is the gradient of $G(\mathbf{y})$ computed at $\mathbf{y}^*$, $\Phi(.)$ is the cumulative distribution function (CDF) of the standard normal variate, and $\Phi(-\beta)$ represents the probability content on the failure side of the tangent hyperplane. By the chain rule of differentiation, the gradient $\nabla G(\mathbf{y})$ is

$$\nabla G(\mathbf{y}) = \left[\frac{\partial g(\mathbf{v},\mathbf{s})}{\partial \mathbf{v}}\bigg|_{\mathbf{s}} + \frac{\partial g(\mathbf{v},\mathbf{s})}{\partial \mathbf{s}}\bigg|_{\mathbf{v}} \mathbf{J}_{\mathbf{s},\mathbf{v}} \right] \mathbf{J}_{\mathbf{v},\mathbf{y}} \tag{2.7}$$

in which $\mathbf{J}_{\mathbf{s},\mathbf{v}} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{v}}$ is the Jacobian of the mechanical transformation, and $\mathbf{J}_{\mathbf{v},\mathbf{y}} = \mathbf{J}_{\mathbf{y},\mathbf{v}}^{-1}$, where $\mathbf{J}_{\mathbf{y},\mathbf{v}} = \dfrac{\partial \mathbf{y}}{\partial \mathbf{v}}$ is the Jacobian of the probability transformation. The partial derivatives of $g(\mathbf{v},\mathbf{s})$ in the above expression are easily computed for any limit-state function explicit

in v and s. The Jacobian of the probability transformation and its inverse are also easily computed, as described in Section 2.4. For elastic structures, analytical expressions for $J_{s,v}$ are derived in Chapter 4.

It is possible that the limit-state surface has multiple minimum distance points. In that case, the surface is approximated by a polyhedron and system reliability techniques are employed to improve the first-order probability estimate (see Ref. 49).

The main challenge in FORM is in finding the nearest point on the limit-state surface to the origin. This is a constrained optimization problem. Optimization algorithms suitable for solving this problem have been investigated in an earlier study [38,40] and are described in Chapter 3.

The first-order reliability method provides a good approximation of the exact failure probability if the limit-state surface in the standard normal space is flat or nearly flat. Nonlinearity in this surface may arise from the nonlinearity in the limit-state function, $g(V,S)$, or the nonlinearity in the probability transformation when V has a nonnormal distribution. A higher order approximating surface, such as a quadratic surface, may be used to improve the accuracy of the approximation in such cases. Such a method is described in the following section.

2.3 The Second-Order Reliability Method

In the second-order reliability method (SORM), the limit-state surface is replaced by a second-order surface fitted to the design point. Such an approximation method was first investigated by Fiessler et al. [23]. However, their results, which are derived for general quadratic surfaces, are too cumbersome for practical use. In subsequent studies [e.g., 7,17,67,68], paraboloid approximations were used.

Two types of paraboloid approximations are used in SORM: the curvature-fitted paraboloid [67,68], and the point-fitted paraboloid [17]. To construct these paraboloids, the coordinates of the standard normal space are first rotated through an orthogonal transformation

$$\mathbf{Y}' = \mathbf{Q}\mathbf{Y} \tag{2.8}$$

such that the new y'_n axis coincides with the design point (see Fig. 2.2). This is achieved by selecting the n-th row of $\mathbf{Q}$ as $-\nabla G(\mathbf{y}^*)/|\nabla G(\mathbf{y}^*)|$. The remaining rows of $\mathbf{Q}$ are determined by a suitable orthogonalization scheme, such as the Gram-Schmidt algorithm [66]. Thus, in the rotated space the limit-state surface is defined by

$$y'_n = G'(\mathbf{y}'_{n-1}) \tag{2.9}$$

where $\mathbf{y}'_{n-1} = (y'_1, y'_2, \cdots, y'_{n-1})$.

The curvature-fitted paraboloid is determined by fitting its curvatures at the apex to the main curvatures of the limit-state surface at the design point. The paraboloid is defined by

$$y'_n = \beta + \frac{1}{2}\mathbf{y}'^{\mathrm{T}}_{n-1}\mathbf{A}\mathbf{y}'_{n-1} \tag{2.10}$$

in which $\mathbf{A}$ is the second-derivative matrix of $G'(\mathbf{y}'_{n-1})$. The elements of this matrix are computed more conveniently from [67]

$$A_{ij} = \frac{[\mathbf{Q}\nabla^2 G(\mathbf{y}^*)\mathbf{Q}^{\mathrm{T}}]_{ij}}{|\nabla G(\mathbf{y}^*)|} \qquad i,j = 1,2,\ldots,n-1 \tag{2.11}$$

In the special case where the coordinate axes y'_i, $i = 1, \cdots, n-1$, coincide with the principal axes of the limit-state surface, the matrix $\mathbf{A}$ is diagonal and the paraboloid is defined by

$$y'_n = \beta + \frac{1}{2}\sum_{i=1}^{n-1} \kappa_i y'^2_i \tag{2.12}$$

where κ_i are the principal curvatures. More generally, this equation can be obtained by a suitable rotation of the $\mathbf{y}'_{n-1}$ axes, in which case κ_i represent the eigenvalues of $\mathbf{A}$. These eigenvalues are needed to compute the probability content of the paraboloid, as is described later in this section.

Since only $n-1$ unknowns are involved in Eq. 2.12, a paraboloid tangent at the design point can be defined given the coordinates of $n-1$ points on the limit-state surface. However, the principal axes of the limit-state surface are unknown unless the eigenvalue

problem for the second-derivative matrix **A** is solved. This can be a rather tedious task, particularly for the finite-element application involving large number of random variables. The motivation behind the point fitted paraboloid [17] is to avoid the eigensolution by assuming that the y'_i axes coincide with the principal axes of the paraboloid regardless of the true orientation of the principal axes of the limit-state surface. This assumption eliminates not only the requirement of an eigensolution but also the second-derivative computation. In addition, the form in Eq. 2.12 greatly simplifies the probability computation.

Theoretically, only one point along each coordinate axis y'_i is needed to compute the principal curvatures κ_i of a paraboloid which has y'_i as the principal axes. Nevertheless, in the point fitting method two points are used along each axis to improve the surface fitting. The fitting points on the surface are selected according to the following ad hoc rules [2, 17]: Along the positive(negative) direction of axis y'_i, if the surface curves towards the origin, a fitting point is selected such that its abscissa is $k\beta(-k\beta)$, where k is a predetermined coefficient (see Fig. 2.2). On the other hand, if the surface curves away from the origin, a fitting point is selected such that it is at a distance $\sqrt{\beta^2 + (k\beta)^2}$ from the origin. The coefficient k is selected as follows [17]:

$$k = \begin{cases} 1 & \text{for } \beta \le 3 \\ \dfrac{3}{\beta} & \text{for } \beta > 3 \end{cases} \tag{2.13}$$

The preceding rules ensure that a fitting point is found under all circumstances, and that a consistent approximation is achieved in all situations.

The two fitting points located in the plane of y'_i and y'_n are used to define two semi-parabolas with principal curvatures $\kappa_{\pm i}$, where the subscripts $-i$ and $+i$ denote the negative and positive directions of y'_i (see Fig. 2.2). The principal curvature κ_i of the fitting paraboloid along axis y'_i is, then, taken as the weighted average of κ_{-i} and κ_{+i} according to the following rule [17]:

$$\frac{1}{\sqrt{1+\beta\kappa_i}} = \frac{1}{2}\left(\frac{1}{\sqrt{1+\beta\kappa_{-i}}} + \frac{1}{\sqrt{1+\beta\kappa_{+i}}}\right) \tag{2.14}$$

This rule defines a parabola whose probability content equals the sum of the probability contents of the two semi-parabolas. It also has the effect of accounting for higher-order effects, such as the case where the design point is an inflection point of the surface.

Mathematically, the assumption made in the point fitting method amounts to neglecting the off-diagonal terms of the second-derivative matrix. The error due to this neglect is analyzed in Ref. 17. If the limit-state surface is indeed a second-order surface, the curvature fitting method gives an exact result. However, the limit-state surface in general may contain higher order terms. While these terms are entirely neglected in the curvature fitting method, the point fitting method includes their contribution in an approximate manner by fitting to the points away from the origin and by the use of the semi-parabolas just mentioned. The point-fitting method also has the important advantage of being insensitive to the noise on the limit-state surface, which may arise when the limit-state function is in an algorithmic form. This advantage is of particular interest in the finite-element reliability methods, which is the main concern of the present study.

Several formulas are available for computing the probability content of a parabolic set. The first formula was proposed by Breitung in 1984 [7] and is

$$P_{f2} \approx \Phi(-\beta) \prod_{i=1}^{n-1} (1 + \beta\kappa_i)^{-1/2} \tag{2.15}$$

The above formula is asymptotically exact as β approaches infinity while $\beta\kappa_i$ remain fixed. This approximation is slightly improved for small values of β if the term $\beta\kappa_i$ inside the parenthesis is replaced by $[\phi(-\beta)/\Phi(-\beta)]\kappa_i$ [33]. A much more accurate formula is Tvedt's so-called double-integral formula [68]

$$P_{f2} \approx \frac{2}{\sqrt{\pi}} \phi(\beta) \mathrm{Re} \int_0^\infty \int_0^\infty \prod_{j=1}^{n-1} (r_p\{[1 + (\beta^2 + 2s)^{1/2}\kappa_j + i\sqrt{2u}\kappa_j]^{-1/2}\})$$
$$(\beta^2 + 2s)^{-1/2} \exp(-s - u^2)\, ds du \tag{2.16}$$

where $r_p\{.\}$ denotes the root with positive real part, κ_j are the principal curvatures, and $i = \sqrt{-1}$. This equation is computed numerically by Gauss quadratures. Very recently,

Tvedt [69] has reported an exact single-integral formula for the probability content of the parabolic set. An evaluation of this formula in the present study was not possible because of time constraint.

The above formulas for P_{f2} require the computation of the eigenvalues of A. (It is noted that an alternative form of Breitung's formula is available [67] which only requires the determinant of A). When the point-fitted paraboloid is used, A is diagonal and its eigenvalues are readily available. This greatly reduces the required computation.

2.4 Probability Transformation

When the basic random variables V_i are statistically independent, a transformation to the standard normal space is given by

$$y_i = \Phi^{-1}[F_{V_i}(v_i)] \qquad i = 1,2,\cdots,n \tag{2.17}$$

where $F_{V_i}(v_i)$ is the CDF of V_i. This mapping is one-to-one, provided each $F_{V_i}(v_i)$ is a strictly increasing function.

If the basic random variables are jointly normal, the transformation to the standard normal space is given by

$$\mathbf{y} = \Gamma(\mathbf{v} - \mathbf{m}) \tag{2.18}$$

in which m is the mean vector of V, and $\Gamma = \mathbf{L}^{-1}$, where L is the lower triangular matrix obtained from the Cholesky decomposition [64] of the covariance matrix of V.

In the most general case of dependent nonnormal variables, the choice of the transformation depends on the form of the joint distribution of V. Hohenbichler and Rackwitz [34] have suggested the use of the following transformation due to Segal [59] and Rosenblatt [56]:

$$y_1 = \Phi^{-1}[F_{V_1}(v_1)]$$
$$y_2 = \Phi^{-1}[F_{V_2|V_1}(v_2|v_1)]$$
$$\cdot$$
$$y_i = \Phi^{-1}[F_{V_i|V_1V_2\cdots V_{i-1}}(v_i|v_1,v_2,\cdots,v_{i-1})]$$
$$\cdot$$

$$y_n = \Phi^{-1}[F_{V_n|V_1V_2\cdots V_{n-1}}(v_n|v_1, v_2, \cdots, v_{n-1})] \tag{2.19}$$

in which $F_{V_i|V_1V_2\cdots V_{i-1}}(v_i|v_1, v_2, \cdots, v_{i-1})$ denotes the conditional CDF of V_i given $V_1 = v_1$, $V_2 = v_2$, ..., and $V_{i-1} = v_{i-1}$. This transformation, which is one-to-one provided each conditional CDF is a strictly increasing function of its arguments, is applicable to all types of joint distribution models. However, its use is convenient only when the conditional distributions are directly available or can be easily obtained. Another drawback of this transformation is that it is dependent on the ordering of the basic random variables. As a result, FORM/SORM approximations employing this transformation are dependent on this ordering.

In an earlier study [18], a joint distribution model, originally introduced by Nataf [52], with prescribed marginal distributions and covariances was proposed. The joint PDF of V is defined such that the variables $\mathbf{Z} = (Z_1, \cdots, Z_n)$ obtained from the marginal transformations

$$z_i = \Phi^{-1}[F_{V_i}(v_i)] \qquad i = 1, 2, \cdots, n \tag{2.20}$$

are jointly normal. Using the rules of random variable transformation, it was shown that the joint density of V that reproduces the prescribed marginals and covariances is given by

$$f_{\mathbf{V}}(\mathbf{v}) = \phi_n(\mathbf{z}, \mathbf{C}_0) \frac{f_{V_1}(v_1) f_{V_2}(v_2) \cdots f_{V_n}(v_n)}{\phi(z_1)\phi(z_2)\cdots\phi(z_n)} \tag{2.21}$$

where $f_{V_i}(v_i) = dF_{V_i}(v_i)/dv_i$ is the PDF of V_i, and $\phi_n(\mathbf{z}, \mathbf{C}_0)$ is the n-dimensional normal density of zero means, unit standard deviations, and correlation matrix $\mathbf{C}_0$. The elements $\rho_{0,ij}$ of the latter matrix are given in terms of the correlation coefficients ρ_{ij} of V_i and V_j through the integral relation [18]

$$\rho_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\frac{v_i - \mu_i}{\sigma_i}\right]\left[\frac{v_j - \mu_j}{\sigma_j}\right] \phi_2(z_i, z_j, \rho_{0,ij})\, dz_i\, dz_j \tag{2.22}$$

where μ_i and σ_j are the mean and standard deviation of V_i, respectively, and $\phi_2(z_i, z_j, \rho_{0,ij})$ is the bivariate normal density of zero means, unit standard deviations, and correlation coefficient $\rho_{0,ij}$. For each pair of marginal distributions with known ρ_{ij}, the

preceding equation can be solved iteratively for $\rho_{0,ij}$. Since this calculation is tedious, a set of semi-empirical formulae relating $\rho_{0,ij}$ to ρ_{ij} for selected marginal distributions are developed in Ref. 18.

The distribution model in Eq. 2.21 is valid under two conditions: (1) The mappings in Eq. 2.20 are one to one. This is true if each marginal $F_{V_i}(v_i)$ is strictly increasing. (2) The correlation matrix $\mathbf{C}_0$ is positive definite. This condition is normally satisfied if V_i's are not too strongly correlated.

The transformation to the standard normal space for the above distribution model is given by

$$\mathbf{y} = \Gamma_0 \mathbf{z} = \Gamma_0 \begin{Bmatrix} \Phi^{-1}[F_{V_1}(v_1)] \\ \vdots \\ \Phi^{-1}[F_{V_n}(v_n)] \end{Bmatrix} \tag{2.23}$$

in which $\Gamma_0 = \mathbf{L}_0^{-1}$, where $\mathbf{L}_0$ is the lower triangular matrix obtained from the Cholesky decomposition of $\mathbf{C}_0$. This transformation is invariant of the ordering of the random variables V_i.

The above joint distribution model has several useful features. First, it is applicable to an arbitrary number of random variables with prescribed marginals and covariances. Second, using the transformation in Eq. 2.23, the resulting FORM/SORM approximations are invariant of the ordering of the basic variables. Third, the required transformation in Eq. 2.23 is computationally much simpler than the transformation in Eq. 2.19 when the conditional distributions are specified. Most importantly, this distribution can be used to model nonnormal random fields [14]. This feature is particularly useful in modeling stochastic material variability in finite-element reliability applications (see Chapter 5).

It is noted that all three transformations in Eqs. 2.18, 2.19, and 2.23 for dependent variables have triangular forms. That is, each Y_i is only a function of V_1, V_2, ... , and V_i and each V_i is only a function of Y_1, Y_2, ..., and Y_i. Hence, both the forward and back-

ward transformations are easily done. In particular, no decomposition is needed for the inverse transformation from Y to V.

2.5 Parameter Sensitivity Analysis

An important feature of FORM is that it provides measures of sensitivity of the reliability index and the first-order estimate of the failure probability with respect to the basic random variables as well as the parameters defining the probability distributions or the limit-state function. The first set of such sensitivity measures is with respect to the variations in the coordinates of the design point, $\mathbf{y}^*$, in the standard normal space. This sensitivity measure is given by the gradient vector

$$\nabla_{y^*}\beta = \alpha(y^*)^T \tag{2.24}$$

in which $\alpha(y^*)^T = -\nabla G(y^*)/|\nabla G(y^*)|$ is the unit vector normal to the limit-state surface at the design point and directed towards the failure set. This vector provides a measure of relative importance of the standard variates Y_i.

The sensitivity of β with respect to the coordinates of the design point, $\mathbf{v}^*$, in the original space is given by [16]

$$\nabla_{v^*}\beta = -\alpha(y^*)^T J_{y,v}|_{y=y^*} \tag{2.25}$$

It is noted, however, that the values of $\nabla_{v^*}\beta$ are dependent on the units of V. Therefore, to compare the relative importance of the basic random variables, Der Kiureghian and Ke [16] defined the unit vector

$$\gamma(v) = \frac{\mathbf{D}\,\alpha(y)^T J_{y,v}}{|\mathbf{D}\,\alpha(y)^T J_{y,v}|} \tag{2.26}$$

where **D** is the diagonal matrix of standard deviations of V. The components of this dimensionless vector are scaled by the corresponding standard deviations to make the respective variations statistically equivalent. Hence, the unit vector $\gamma(v^*)$ at the design point represents the scaled and normalized sensitivities of β with respect to the variations in the coordinates of $\mathbf{v}^*$. As such, it provides a measure of relative importance of the basic variables V_i. A useful application of the measures α and γ is in reducing the

number of random variables in the optimization process for finding the design point. The details are discussed in Section 5.4.

Other measures of reliability sensitivity are with respect to the parameters in the distribution function of V and the limit-state function $g(.)$. Let θ_d denote the parameters defining the distribution function of V (such as means, standard deviations, etc.) and θ_l denote the parameters defining the limit-state function, i.e., $F_V(v,\theta_d)$ and $g(V,S,\theta_l)$. It can be shown [49] that the sensitivities of β with respect to θ_d and θ_l are given by

$$\nabla_{\theta_d}\beta = \alpha(y^*)^T \frac{\partial y^*(v^*,\theta_d)}{\partial \theta_d}\Big|_{v^*} \tag{2.27}$$

$$\nabla_{\theta_l}\beta = \frac{1}{|\nabla G(y^*)|}\frac{\partial g(v^*,s^*,\theta_l)}{\partial \theta_l}\Big|_{v^*,s^*} \tag{2.28}$$

For either set of parameters θ, the sensitivities of the first-order failure probability are obtained by applying the chain rule to Eq. 2.6 and are

$$\nabla_\theta P_{f1} = -\phi(\beta)\nabla_\theta\beta \tag{2.29}$$

The above sensitivity measures can be extremely useful in the structural design process. They can be used to identify the variables or parameters which have major influences on the failure probability, thus providing an effective means for improving the design. These measures also help in gaining insight into the complex behavior of structural systems. Example applications in Chapter 7 illustrate these facts.

2.6 Random Field Modeling

In finite-element reliability analysis, it is often the case that the material properties, structural geometry, and external loads have random spatial variabilities and, hence, are modeled by random fields rather than random variables. For an application within the framework of the reliability theory described in this chapter, it is necessary that such random fields be represented in terms of random variables. This subject is addressed in Chapter 5.

FORM/SORM applications involving random processes fluctuating in time have been studied by Igusa and Der Kiureghian [35], Guers and Rackwitz [26], Madsen [47], and Wen and Chen [73]. This topic is beyond the scope of this report.

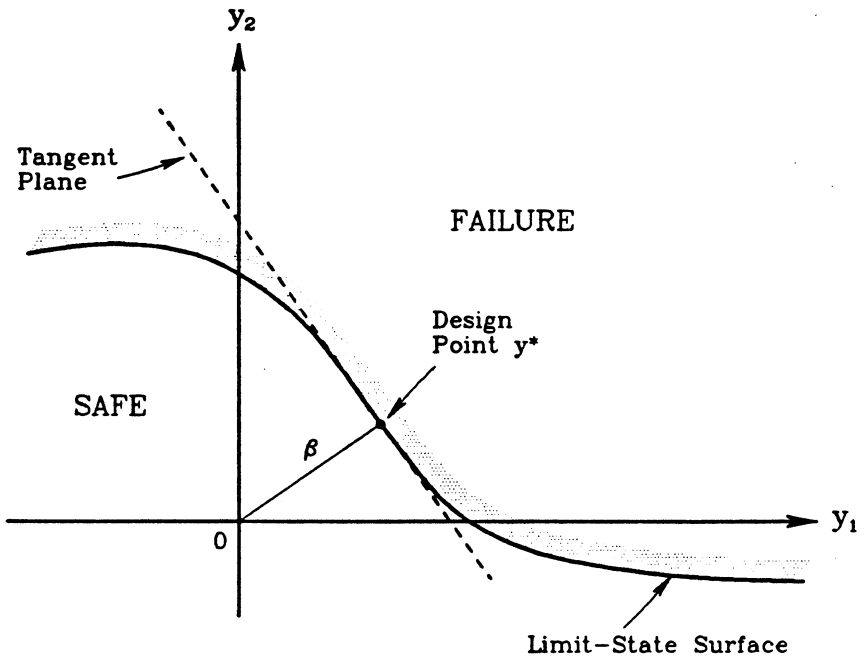


Figure 2.1 The First-Order Reliability Method

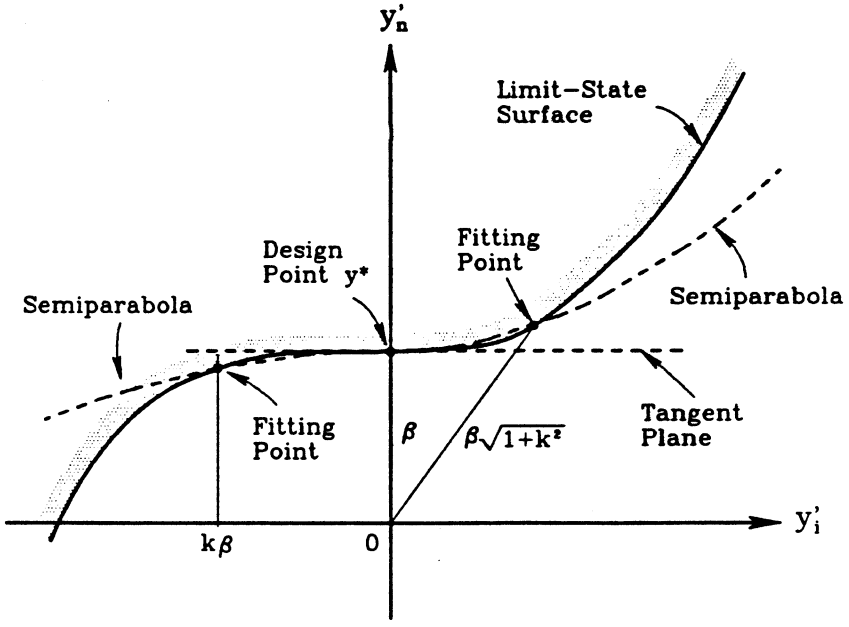


Figure 2.2 The Point-Fitting Method

CHAPTER 3

OPTIMIZATION ALGORITHMS FOR RELIABILITY ANALYSIS

3.1 Introduction

The main effort in the first-order reliability method is in finding the minimum-distance point from the limit-state surface to the origin. This is formulated as a constrained optimization problem:

$$\begin{aligned} P1: \quad & \textit{minimize} \quad F(\mathbf{y}) \\ & \textit{subject to} \quad G(\mathbf{y}) = 0 \end{aligned} \tag{3.1}$$

where $F(\mathbf{y}) = \mathbf{y}^T\mathbf{y}/2$ is the objective function and $G(\mathbf{y})$ is the limit-state function in the standard normal space. It is assumed here that the constraint, $G(\mathbf{y}) = 0$, is continuous and twice differentable.

This chapter examines and compares several optimization algorithms for the solution of the above problem. Of special interest are the properties of various algorithms in connection with their application to the finite-element reliability problem.

3.2 Basic Concepts of Optimization Programming

Necessary Conditions for Local Minima. The Lagrangian associated with the optimization program $P1$ in Eq. 3.1 is defined by [46]

$$l(\mathbf{y}, \lambda) = F(\mathbf{y}) + \lambda G(\mathbf{y}) \tag{3.2}$$

where λ is a constant. If $\mathbf{y}^*$ is a local minimum point of $P1$, then there is a λ^*, denoted the Lagrange multiplier, such that [46]

$$\nabla l(\mathbf{y}^*, \lambda^*) = \mathbf{y}^{*T} + \lambda^* \nabla G(\mathbf{y}^*) = 0 \tag{3.3}$$

In addition, the Hessian of the Lagrangian, $\nabla^2 l(\mathbf{y}^*, \lambda^*) = \mathbf{I} + \nabla^2 G(\mathbf{y}^*)$, is positive semidefinite on the tangent plane $\mathbf{M} = \{\mathbf{y} : \nabla G(\mathbf{y}^*)(\mathbf{y}-\mathbf{y}^*) = 0\}$, i.e., $\mathbf{y}^T\nabla^2 l(\mathbf{y}^*, \lambda^*)\mathbf{y} \geq 0$ for all $\mathbf{y}$ on $\mathbf{M}$.

Line Search. Most iterative algorithms discussed in this chapter 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,

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \xi_k \mathbf{d}_k \tag{3.4}$$

by adjusting the step length, ξ_k, such that the objective function along the direction is minimized. 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 algorithms is in 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 [46]. 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 ξ_k is neither too large nor too small. Armijo's rule [46] is a popular criterion for terminating the line search.

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. Conditions under which global convergence is ensured were developed by Zangwill [77]. 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 [46]

$$\lim_{k \to \infty} \frac{E(\mathbf{y}_{k+1})}{E(\mathbf{y}_k)^p} < \infty \tag{3.5}$$

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

$$\lim_{k \to \infty} \frac{E(\mathbf{y}_{k+1})}{E(\mathbf{y}_k)} = \rho < 1 \tag{3.6}$$

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. The convergence is called superlinear if $\rho = 0$. Most algorithms discussed in this chapter converge linearly.

3.3 Program Structure

The structure of the optimization program plays an important role in the performance of the algorithm. $P1$ consists of two parts: the objective function and the equality constraint. The objective function is of pure quadratic form, is convex, and its gradient is simply $\mathbf{y}$. Because of this simplicity, the objective function does not influence the performance of the algorithm. The constraint $G(\mathbf{y}) = 0$ is usually smooth and twice differentiable, but it is not necessarily convex. Furthermore, in the context of the finite element reliability method, where the mechanical transformation in Eq. 2.1 is in an algorithmic form and the size of $\mathbf{y}$ can be very large (e.g., several hundred), the computations of $G(\mathbf{y})$ and $\nabla G(\mathbf{y})$ are difficult since the former is not an explicit function of $\mathbf{y}$. Therefore, the constraint strongly influences the performance of the algorithm. Obviously, algorithms that require the computation of the Hessian matrix $\nabla^2 G(\mathbf{y})$ would be impractical for such applications. In recent years, efficient algorithms for computing the gradient $\nabla G(\mathbf{y})$ for the finite-element reliability method have been developed [15, 39]. This subject is dealt with in Chapters 4 and 5 of this report.

3.4 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 [45], four criteria are proposed here to evaluate the performance of various nonlinear optimization algorithms for application to finite element reliability problems: (1) *Generality* -- The generality of an algorithm refers to the types of problems that it can solve. Certain algorithms are restricted to specific types of problems. For example, some methods require that the feasiole 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; (2) *Robustness* -- Robustness refers to the power of a method to solve problems with a required accuracy. It incorporates the global convergence properties of the algorithm; (3) *Efficiency* -- Efficiency is measured by the effort required to solve a problem. It is dependent on the convergence rate of the algorithm. In finite element reliability applications, efficiency can be measured by the numbers of computations of $G(\mathbf{y})$ and $\nabla G(\mathbf{y})$ that are needed to converge to the design point; and (4) *Capacity* -- The capacity of a method refers to the maximum size of the problem that can be solved by the method; it depends on the required and available computer storage.

Five nonlinear optimization algorithms are discussed in the following sections; namely, the gradient projection method, the penalty method, the augmented Lagrangian method, the sequential quadratic programming method, and a widely used method which was specifically designed for the reliability problem. The basic ideas of these algorithms are introduced and their performances are evaluated using the four criteria proposed above.

3.5 The Gradient Projection Method

The gradient projection method, originally developed by Rosen [55], is a primal method. A primal method is an iterative algorithm that solves the original problem directly (in contrast to indirect 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.

The gradient projection method 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 iteration step. For constrained problems, since each point must remain in the feasible domain, the search 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. 3.1). 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 $\nabla F(\mathbf{y}_k) = \mathbf{y}_k^{\mathrm{T}}$. From these conditions, $\mathbf{d}_k$ is expressed as

$$\mathbf{d}_k = -\left[\mathbf{I} - \frac{\nabla G(\mathbf{y}_k)^{\mathrm{T}} \nabla G(\mathbf{y}_k)}{|\nabla G(\mathbf{y}_k)|^2}\right] \mathbf{y}_k \tag{3.7}$$

Because the constraint may not be linear, the new search point could be infeasible. Therefore, a Newton-type correction is used to pull the point back to the feasible set (see Fig. 3.1).

A cycle of the gradient projection method may proceed as follows:

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

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

(3) Calculate $\mathbf{d}_k$ from Eq. 3.7.

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

i) for the selected step size ξ_k calculate

$$\mathbf{y}_{k+1}^0 = \mathbf{y}_k + \xi_k \mathbf{d}_k \tag{3.8}$$

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

$$\mathbf{y}_{k+1}^{i+1} = \mathbf{y}_{k+1}^i - \frac{G(\mathbf{y}_{k+1}^i)}{|\nabla G(\mathbf{y}_k)|^2} \nabla G(\mathbf{y}_k)^{\mathrm{T}} \tag{3.9}$$

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

Theoretically speaking, the step size ξ_k in Eq. 3.8 should be obtained by an 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 [46].

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.

Since this method is the constrained version of the steepest descent method, it can be shown that it converges linearly and its asymptotic convergence ratio is [46]

$$\rho \leq \left(\frac{r_M - 1}{r_M + 1} \right)^2 \tag{3.10}$$

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 slower for larger r_M. The asymptotic convergence ratio, though usually not computed, serves as a theoretical tool to compare various methods. According to the study of Lootsma [45], the gradient projection method is rated as the most efficient method among several general constrained optimization techniques. Hence, the gradient projection method appears to satisfy all the proposed criteria and is appropriate for solving the structural reliability problem.

3.6 The Penalty Method

The penalty method is 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 set and $P(\mathbf{y}) > 0$ elsewhere. If the standard quadratic penalty function is adopted, the unconstrained penalty problem takes the form

$$P2: \quad \textit{minimize} \quad \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}c\,G(\mathbf{y})^2 \tag{3.11}$$

Suppose the penalty parameter c approaches infinity. The minimization process will force the solution to satisfy $G(\mathbf{y}) = 0$ and minimize $\mathbf{y}^T\mathbf{y}/2$ at the same time. Thus, $P1$ and $P2$ should yield exactly the same solutions. Once the penalty problem is set up, one can use any unconstrained optimization technique to solve the problem.

The properties of the penalty method depend on the technique used to solve $P2$. As mentioned earlier, it is impractical to compute the Hessian of the constraint. Therefore, only methods using the first-order derivatives are acceptable. The performance of the first-order schemes depend on the condition number of the Hessian of the objective function. Unfortunately, because of the penalty term, the Hessian is ill-conditioned. As a result, the first-order schemes converge very slowly [38, 40]. Furthermore, in first-order reliability analysis, the failure probability is sensitive to the distance between the origin and the design point. 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. The penalty method is thus considered inappropriate because of the robustness and efficiency concerns.

3.7 The Augmented Lagrangian Method

As in linear optimization problems, the constrained problem $P1$ can 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 [46]

$$D1: \quad \textit{maximze} \quad \psi(\lambda) \tag{3.12}$$

$$\psi(\lambda) = \textit{minimum}\,[l(\mathbf{y}, \lambda)] \tag{3.13}$$

According to the Local Duality Theorem [46], the dual problem $D1$ and the primal problem $P1$ 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 [24, 46], 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:

$$\begin{aligned} P3: \quad & \textit{minimze} \quad F(\mathbf{y}) + \frac{1}{2} c\, G(\mathbf{y})^2 \\ & \textit{subject to} \quad G(\mathbf{y}) = 0 \end{aligned} \tag{3.14}$$

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 [46] that there exists a c^* such that for all $c \geq c^*$, the augmented Lagrangian

$$l_c(\mathbf{y}, \lambda^*) = F(\mathbf{y}) + \lambda^* G(\mathbf{y}) + \frac{1}{2} c\, G(\mathbf{y})^2 \tag{3.15}$$

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. An 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

$$\phi_c(\lambda_k) = \textit{minimum}\,[l_c(\mathbf{y}, \lambda_k)] \tag{3.16}$$

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

(2) Modify λ_k according to the following formula

$$\lambda_{k+1} = \lambda_k + c\, G(\mathbf{y}_k) \tag{3.17}$$

(3) Increase c if the constraint violation has not decreased sufficiently in going 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. 3.17 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. 3.13, its capacity is about the same as the gradient projection method. Therefore, it is applicable to the reliability problem. However, this method is difficult to implement, because the user has to make initial estimates on c and λ, and these initial values may influence the performance of the algorithm.

3.8 The Sequential Quadratic Programming Method

Considering Eqs. 3.1 and 3.3, the optimal point $\mathbf{y}^*$ must be a solution of the following simultaneous equations:

$$\begin{aligned} \mathbf{y}^T + \lambda \nabla G(\mathbf{y}) &= 0 \\ G(\mathbf{y}) &= 0 \end{aligned} \tag{3.18}$$

Lagrange methods 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 first suggested by Shinozuka [60].

Many standard algorithms for solving systems of nonlinear equations are available. The Newton method solves the above equations by the following recursive formulas [46]:

$$\begin{bmatrix} \mathbf{y}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_k \\ \lambda_k \end{bmatrix} + \xi_k \begin{bmatrix} \mathbf{d}_k \\ \kappa_k \end{bmatrix} \tag{3.19}$$

where ξ_k is the step length, and

$$\begin{bmatrix} \mathbf{d}_k \\ \kappa_k \end{bmatrix} = \begin{bmatrix} \nabla^2 l(\mathbf{y}_k, \lambda_k) & \nabla G(\mathbf{y}_k)^T \\ \nabla G(\mathbf{y}_k) & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{y}_k \\ -G(\mathbf{y}_k) \end{bmatrix} \tag{3.20}$$

It can be shown [46] that $\mathbf{d}_k$ and κ_k are the solution and the Lagrange multiplier of the following quadratic program:

$$P4: \quad \textit{minimze} \quad \frac{1}{2}\mathbf{d}^T\nabla^2 l(\mathbf{y}_k, \lambda_k)\mathbf{d} + \mathbf{y}^T\mathbf{d}$$
$$\textit{subject to} \quad G(\mathbf{y}_k) + \nabla G(\mathbf{y}_k)\mathbf{d} = 0 \tag{3.21}$$

For this reason, The algorithm defined in Eqs. 3.19 and 3.20 is referred to as the recursive quadratic programming method, or the sequential quadratic programming method.

Schittkowski [58] proposed a sequential quadratic programming scheme which has become popular in the field of reliability analysis. The essential ideas of this method are to replace the Hessian of the Lagrangian by an approximate matrix which is updated at each step, and to use the augmented Lagrangian function in Eq. 3.15 to determine the step length ξ_k.

Without describing the derivations, the algorithm consists of the following steps:

(1) Select some positive tolerances $\bar{c}$, $\bar{e}$, ζ, and υ with $\bar{c}>1$, $\bar{e}<1$, $\zeta<1$, and $\upsilon<\frac{1}{2}$.

Choose initial values

$$\mathbf{z}_0 = \begin{bmatrix} \mathbf{y}_0 \\ \lambda_0 \end{bmatrix}, \quad \mathbf{B}_0 = \mathbf{I} \tag{3.22}$$

Define $\delta_{-1} = 1$ and $c_{-1} = \bar{c}$.

(2) Solve the quadratic subproblem $P4$ or Eq. 3.20 and denote the optimal solution and multiplier by $\mathbf{d}_k$ and κ_k, respectively.

(3) Determine the new penalty parameter c_{k+1}:

Compute

$$\delta_k = \min(\mathbf{d}_k^T\mathbf{B}_k\mathbf{d}_k / |\mathbf{d}_k|^2, \delta_{k-1}) \tag{3.23}$$

$$e_k = \begin{cases} |\mathbf{d}_k|^2 / |\kappa_k - \lambda_k|^2, & \text{if } \lambda_k \neq \kappa_k \\ \bar{e}, & \textit{otherwise} \end{cases} \tag{3.24}$$

and let i be the smallest positive integer with

$$\frac{1}{\bar{c}^i} < \frac{1}{4} e_k \delta_k \left(1 - \frac{\delta_k}{4}\right). \tag{3.25}$$

Define

$$c_k = \max\left(c_{k-1}, \bar{c}^i\right) \tag{3.26}$$

(4) Do line search. Evaluate the smallest nonnegative integer j which satisfies

$$\begin{aligned} & l_{c_k}\left(\mathbf{y}_k + \zeta^j \mathbf{d}_k, \lambda_k + \zeta^j (\kappa_k - \lambda_k)\right) \\ & \le l_{c_k}(\mathbf{y}_k, \lambda_k) + \upsilon \zeta^j \nabla l_{c_k}(\mathbf{y}_k, \lambda_k) \begin{bmatrix} \mathbf{d}_k \\ \kappa_k - \lambda_k \end{bmatrix} \end{aligned} \tag{3.27}$$

where $l_{c_k}(.)$ is the augmented Lagrangian function with penalty parameter c_k. The step length ξ_k is then equal to ζ^j.

(5) Let $\mathbf{y}_{k+1} = \mathbf{y}_k + \xi_k \mathbf{d}_k$ and $\lambda_{k+1} = \lambda_k + \xi_k (\kappa_k - \lambda_k)$.

(6) Update the approximate Hessian, $\mathbf{B}$, by the BFGS scheme [58],

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{1}{\xi_k \mathbf{q}^{\mathrm{T}} \mathbf{d}_k} \mathbf{q}\mathbf{q}^{\mathrm{T}} - \frac{1}{\mathbf{d}_k^{\mathrm{T}} \mathbf{B}_k \mathbf{d}_k} \mathbf{B}_k \mathbf{d}_k \mathbf{d}_k^{\mathrm{T}} \mathbf{B}_k \tag{3.28}$$

where

$$\mathbf{q} = \theta \mathbf{q}' + (1 - \theta) \xi_k \mathbf{B}_k \mathbf{d}_k \tag{3.29}$$

with

$$\mathbf{q}' = \nabla l(\mathbf{y}_{k+1}, \lambda_{k+1}) - \nabla l(\mathbf{y}_k, \lambda_k), \tag{3.30}$$

$$\theta = \begin{cases} 1 & \text{if } \mathbf{d}_k^{\mathrm{T}} \mathbf{q}' \ge 0.2 \xi_k \mathbf{d}_k^{\mathrm{T}} \mathbf{B}_k \mathbf{d}_k \\ \dfrac{0.8 \xi_k \mathbf{d}_k^{\mathrm{T}} \mathbf{B}_k \mathbf{d}_k}{\xi_k \mathbf{d}_k^{\mathrm{T}} \mathbf{B}_k \mathbf{d}_k - \mathbf{d}_k^{\mathrm{T}} \mathbf{q}'} & \textit{otherwise} \end{cases} \tag{3.31}$$

(7) Repeat steps 2 to 6 until convergence is achieved.

This algorithm has several attractive features: it is applicable to general reliability problems, it is globally convergent under mild conditions, and, best of all, the local convergence is superlinear. There are, of course, trade-offs for the high efficiency. Namely, it requires the storage of the $(n + 1) \times (n + 1)$ matrix on the right-hand side of Eq. 3.20 and the solution of the associated $(n + 1)$ equations at each step. Nevertheless, this algorithm is appealing, especially when it is applied to finite-element reliability analysis of nonlinear

structures. Compared with the computations required for the solution of the nonlinear structural response, the solution of the $(n + 1)$ equations is remunerative. Therefore, although this method does not have as large a capacity as the other methods, it is considered suitable for the reliability problem.

3.9 The HL-RF Method

This method, originally proposed by Hasofer and Lind [28] for second-moment reliability analysis and later extended by Rackwitz and Fiessler [53] to include distribution information, is currently the most widely used method for solving the optimization problem in structural reliability [49]. For brevity, this method is denoted herein as the 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 $P1$. This method is based on the following recursive formula:

$$\mathbf{y}_{k+1} = \frac{1}{|\nabla G(\mathbf{y}_k)|^2}[\nabla G(\mathbf{y}_k)\,\mathbf{y}_k - G(\mathbf{y}_k)]\nabla G(\mathbf{y}_k)^{\mathrm{T}} \tag{3.32}$$

It can be shown that this method is a special case of the sequential quadratic programming method, in which the Hessian of the Lagrangian is approximated by an identity matrix and the step length is $\xi_k = 1$.

In comparison with other methods, the HL-RF method 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, as shown in an earlier study [38], this method may fail to converge. Therefore, modifications have been suggested to improve its robustness. Rackwitz, et al. [1] proposed two modified procedures. One uses $\mathbf{y}_k$ and $\mathbf{y}_{k+1}$ from Eq. 3.32 to define a new iteration point such that it satisfies the linearized constraint. The other involves two sub-iterations, one using a pure Newton search to locate a point on the feasible set, and the other searching along a tangent direction to find a point whose position vector is as parallel to its gradient as possible. Both modifications appear to be ad hoc. The robustness of the former procedure is not

guaranteed. The second procedure would require more computations than the gradient projection method.

Another way to improve the robustness of the HL-RF method is to introduce a merit function $m(\mathbf{y})$ to monitor the convergence of the sequence. The merit function should have a global minimum at the solution point of $P1$ and it should decrease in each iteration step. Since the solution point of $P1$ satisfies $G(\mathbf{y}^*) = 0$ and $\mathbf{y}^* + \lambda^* \nabla G(\mathbf{y}^*) = 0$, where $\lambda^* = -\frac{\nabla G(\mathbf{y}^*)\mathbf{y}^*}{|\nabla G(\mathbf{y}^*)|^2}$, a non-negative merit function can be constructed as

$$m(\mathbf{y}) = \frac{1}{2}\left|\mathbf{y} - \frac{\nabla G(\mathbf{y})\mathbf{y}}{|\nabla G(\mathbf{y})|^2}\nabla G(\mathbf{y})\right|^2 + \frac{1}{2}c\,G(\mathbf{y})^2 \tag{3.33}$$

where c is a positive constant. Obviously, all minimum points of $P1$ are global minimum points of $m(\mathbf{y})$. In the modified HL-RF method, the new iteration point is selected by a line search along the direction vector

$$\mathbf{d}_k = \frac{1}{|\nabla G(\mathbf{y}_k)|^2}[\nabla G(\mathbf{y}_k)\mathbf{y}_k - G(\mathbf{y}_k)]\nabla G(\mathbf{y}_k)^{\mathrm{T}} - \mathbf{y}_k \tag{3.34}$$

until a sufficient decrease in $m(\mathbf{y})$ is achieved.

The merit function in Eq. 3.33 is a convenient guide for selecting the step size, since it is in terms of quantities which are already known. However, 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 is not guaranteed. Nevertheless, the modification greatly improves the robustness of the original HL-RF method, as numerical examples in the next section will show.

3.10 Comparison of Algorithms

Based on the analysis in the previous sections, the gradient projection method (GP), the augmented Lagrangian method (AL), and the sequential quadratic programming method (SQP) appear to be promising methods in solving the reliability problem. The modified HL-RF method (MHL-RF) requires further testing to examine its robustness. The original HL-RF method, though not robust, is of interest since it is presently widely

used to solve the reliability problem. These five algorithms are coded into a computer program so that their performances can be further investigated through numerical examples.

Five examples are used to examine the performance of the above algorithms. Three measures of comparison are selected: the numbers of computations of $G(\mathbf{y})$ and $\nabla G(\mathbf{y})$, and the CPU time required by each method to converge. In all the examples the gradient vector $\nabla G(\mathbf{y})$ is computed analytically, and the initial values of the parameter λ in the AL method is taken to be zero. The tolerance used for checking convergence is identical for the various 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, useful insight into the relative merits of the algorithms is gained.

Example 3.1 This example is taken from the reliability analysis of a pipeline where the limit-state surface was generated by response-surface fitting. The limit-state function is

$$\begin{aligned} g(\mathbf{V}) = {} & 1.1 - 0.00115 V_1 V_2 + 0.00157 V_2^2 + 0.00117 V_1^2 \\ & + 0.0135 V_2 V_3 - 0.0705 V_2 - 0.00534 V_1 - 0.0149 V_1 V_3 \\ & - 0.0611 V_2 V_4 + 0.0717 V_1 V_4 - 0.226 V_3 + 0.0333 V_3^2 \\ & - 0.558 V_3 V_4 + 0.998 V_4 - 1.339 V_4^2 \end{aligned} \tag{3.35}$$

where V_1, V_2, V_3, and V_4 are statistically independent random variables; V_1 has a type-II largest value distribution with a mean 10 and a standard deviation 5; V_2 and V_3 are both normal with means 25 and 0.8 and standard deviations 5 and 0.2, respectively; and V_4 has the lognormal distribution with a mean 0.0625 and a standard deviation 0.0625.

Starting from the mean point, all algorithms converged to the point $\mathbf{v}^* = [15.09, 25.07, 0.8653, 0.04582]^T$ with $\beta = 1.36$, except the HL-RF method which exhibited a seesaw behavior after 25 iterations. Table 3.1 lists the number of computations of $G(\mathbf{y})$ and $\nabla G(\mathbf{y})$ and the CPU time (on MicroVax II/GPX) spent by these four methods. These results are based on an initial value of $c = 2\times 10^3$ for the AL method, and $c = 10$ for the MHL-RF method. These two methods were also tested for other values of c. The efficiency and accuracy of the MHL-RF method only varied slightly for c ranging from 1

to 10^6. The AL method did not converge to the exact solution point for initial values of c less than 2×10^3, and it became inefficient when c was taken greater than 10^4. For instance, for $c = 10^6$, 929 computations of $G(\mathbf{y})$ were performed before convergence was achieved. Thus, the performance of the AL method is sensitive to the selected value of c. This property is undesirable since the optimal value of c varies from problem to problem and is usually unknown. It is apparent from Table 3.1 that, for the present example, the SQP method is the most efficient among the various methods. However, one should note that the solution time for this example may not be a meaningful measure of efficiency because of the small size of the problem.

Example 3.2 In some applications the limit-state function may contain noise, which may arise from errors in numerical routines, such as numerical integration, equation solution, or eigenvalue solution required as a part of the finite element analysis. To examine the performances of the five algorithms for such an unfavorable situation, the limit-state function

$$g(\mathbf{V}) = V_1 + 2V_2 + 2V_3 + V_4 - 5V_5 - 5V_6 + 0.001\sum_{i=1}^{6}\sin(100V_i) \tag{3.36}$$

is selected which has high-frequency, artificial noise. The six random variables in this function are statistically independent with lognormal distributions. V_1 to V_4 have means 120 and standard deviations 12; V_5 has a mean 50 and a standard deviation 15; and V_6 has a mean 40 and a standard deviation 12.

Starting from the mean point, the GP, SQP, and MHL-RF methods converged to the solution point $\mathbf{v}^* = [117.3, 115.3, 115.3, 117.3, 83.62, 55.54]^T$ with $\beta = 2.3482$. The HL-RF method exhibited an unstable behavior, and the AL method failed to converge to a solution because of the noise. To check whether the solution obtained with the first three methods 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 in the case with noise. Hence, it is believed that the solution obtained is a global minimum point.

The required computations by the GP, SQP, and MHL-RF methods (the latter with c = 10) are listed in Table 3.1. Since the computations of $G(\mathbf{y})$ and $\nabla G(\mathbf{y})$ for this problem take equal time, from the results in Table 3.1 one may conclude that the GP and MHL-RF methods are slightly more efficient than the SQP method for this example. However, this result is dependent on the selected value of c for the MHL-RF method. Further investigation with the MHL-RF method revealed that the efficiency of the method was best for $10 \leq c \leq 2000$, which included the selected value of c. For values of c outside this range, the convergence of the MHL-RF method slowed down considerably.

Example 3.3 The reliability of the three-bay, five-story, linear elastic frame structure in Fig. 3.2 is examined. Taken from Ref. 38, This problem has 21 basic random variables: 3 applied loads, 2 Young's moduli, 8 moments of inertia, and 8 cross-sectional areas. Statistical dependence between loads, material properties, and member dimensions is considered. Of interest is the probability that the horizontal displacement at node 1 exceeds 0.2 *ft*. Thus, the limit-state function is expressed as

$$g(u_1(\mathbf{v})) = 0.2 - u_1(\mathbf{v}) \tag{3.37}$$

The mechanical transformation $u_1(\mathbf{v})$ and its gradient ∇u_1 are computed based on the approach described in Chapters 4 and 5. The effect of geometrical nonlinearity is not included in this example.

Starting from the mean point, all the methods (with c = 10 for the AL method, and c = 50 for the MHL-RF method) converged to the same point (see Ref. 38 for the coordinates of the solution point). The required computations are listed in Table 3.1. In this example, the gradient ∇u_1 is computed using the adjoint method, which is described in Chapter 5. For this linear structure, the computations required for ∇u_1 include the assembly of an $m \times n$ matrix and the solution of m simultaneous equations, where m = 60 is the number of degrees of freedom of the structure, and n = 21 is the number of basic random variables. Since no decomposition is required in the equation solving, the time required for computing the gradient for this problem is less than that required for computing the

limit-state function. It is seen in Table 3.1 that the HL-RF method is the most efficient and the AL method is the least efficient among the five methods. Further investigations revealed that the MHL-RF method converged very slowly for $c \leq 10$, but for $c > 50$ its efficiency was almost invariant to the value of c.

Example 3.4 The reliability of a two-dimensional, built-up column shown in Fig. 7.1 of Chapter 7 is studied. The column is composed of elastic truss members. However, the constitutive law and the equilibrium of the members are formulated to account for the geometrical nonlinearity of the column. Twenty two basic random variables define this problem, including two correlated elastic moduli of the truss members, two external forces, and the x-coordinates of the nodes representing the random imperfections of the column. See Chapter 7 for other details of this problem. The column is considered failed if its horizontal displacement at the midspan exceeds 30 in. Hence, the limit-state function is defined as

$$g(u_{10}(\mathbf{v})) = 30 - u_{10}(\mathbf{v}) \tag{3.38}$$

where $u_{10}(\mathbf{v})$ is the horizontal displacement of node 10. This problem is solved by the use of the general-purpose, finite-element reliability code described in Chapter 6.

Starting from the mean point, the HL-RF and AL methods failed to converge because they generated points which yielded unstable columns. The other three methods (with $c = 10$ for the MHL-RF method) converged to the same solution point, which is described in Chapter 7. The required computations by the three methods are listed in Table 3.1. For this problem, the GP method is not as efficient as the other two methods. Further investigation showed that for this nonlinear structure the feasible set correction is the most time consuming part of the GP algorithm.

For geometrically nonlinear structures, each computation of $G(\mathbf{y})$ requires an iterative solution of the nonlinear response. On the other hand, once the response is obtained, the required effort for computing the gradient $\nabla G(\mathbf{y})$ by the adjoint method (see Chapter 5) essentially amounts to assembling an $m \times n$ matrix and solving a set of m equations, where

m and n respectively denote the number of degrees of freedom of the structure and the number of basic random variables. Therefore, for such problems, the gradient vector is much easier to compute than the structural response. This is unlike general optimization problems, in which the gradient is usually calculated by a finite difference scheme. Thus, in geometrically nonlinear finite-element reliability applications, one should choose an algorithm which takes full advantage of the gradient to speed up the convergence. For instance, the MHL-RF method may be preferred since it makes use of the response as well as the gradient at each search point.

Example 3.5 The reliability of an elastic plate with a random hole is studied (see Fig. 7.6 of Chapter 7). The plate is subjected to a uniformly distributed tensile load of random magnitude at the two opposite edges. The Young's modulus and Poisson's ratio of the plate are modeled as random fields, and the coordinates of the hole are assumed to be randomly perturbed from a mean circular shape. After random field discretization of a quarter of the plate, 85 random variables are considered, including 1 for the tensile load intensity, 24 for the nodal coordinates around the hole, 30 for the Young's modulus field, and 30 for the Poisson's ratio field. See Chapter 7 for other details of this problem. The plate is considered failed if the principal stress at the point of stress concentration, including the geometrical nonlinearity of the plate, exceeds 600 *psi*. Hence, the limit-state function is defined as

$$g(\tilde{T}^*(\mathbf{v})) = 600 - \tilde{T}^*(\mathbf{v}) \tag{3.39}$$

where $\tilde{T}^*(\mathbf{v})$ is the principal Cauchy stress at the point of interest.

Starting from the mean point, the AL method failed to converge after 1200 computations of the limit-state function, while exhibiting a seesaw behavior. The other four methods (with $c = 50$ for the MHL-RF method) converged to the same solution point (see Chapter 7 for the details of the solution). The required computations by the four methods are listed in Table 3.1. Similar to Example 3.4, the GP method is by far the most inefficient among the four algorithms. Furthermore, the HL-RF and MHL-RF methods con-

verged about three times faster than the SQP method.

3.11 Summary

Several constrained optimization algorithms are compared for their suitability in finite-element reliability analysis. The comparison is based on four criteria: generality, robustness, efficiency, and capacity. Five algorithms are found to be suitable for such analysis and are examined by way of five numerical examples. These schemes include the gradient projection method, the augmented Lagrangian method, the sequential quadratic programming method, the HL-RF method, and the modified HL-RF method. The results indicate that the sequential quadratic programming method, the gradient projection method, and the modified HL-RF method are robust techniques for use in structural reliability applications. Furthermore, for applications in nonlinear finite-element reliability analysis, the sequential quadratic programming method and the modified HL-RF method appear to be more efficient than the gradient projection method. Since the results reported are based on a limited number of examples, the findings should not be considered as definitive. The best algorithm will stand out only through continued application and testing.

Table 3.1 Comparison of Optimization Algorithms

Example	Method	CPU time *sec.*	Number of computations $G(\mathbf{y})$	$\nabla G(\mathbf{y})$
1	GP	14.0	64	64
	AL	14.0	220	61
	SQP	13.3	29	14
	HL-RF		not converged	
	MHL-RF	13.8	28	28
2	GP	7.7	154	82
	AL		not converged	
	SQP	9.0	239	36
	HL-RF		not converged	
	MHL-RF	8.1	121	121
3	GP	79.7	37	37
	AL	364.5	266	82
	SQP	36.1	13	7
	HL-RF	31.4	9	9
	MHL-RF	36.5	12	12
4	GP	965.3	72	72
	AL		not converged	
	SQP	218.1	16	8
	HL-RF		not converged	
	MHL-RF	255.3	13	13
5	GP	18248.9	119	119
	AL		not converged	
	SQP	3377.6	26	11
	HL-RF	1282.7	8	8
	MHL-RF	1292.3	8	8

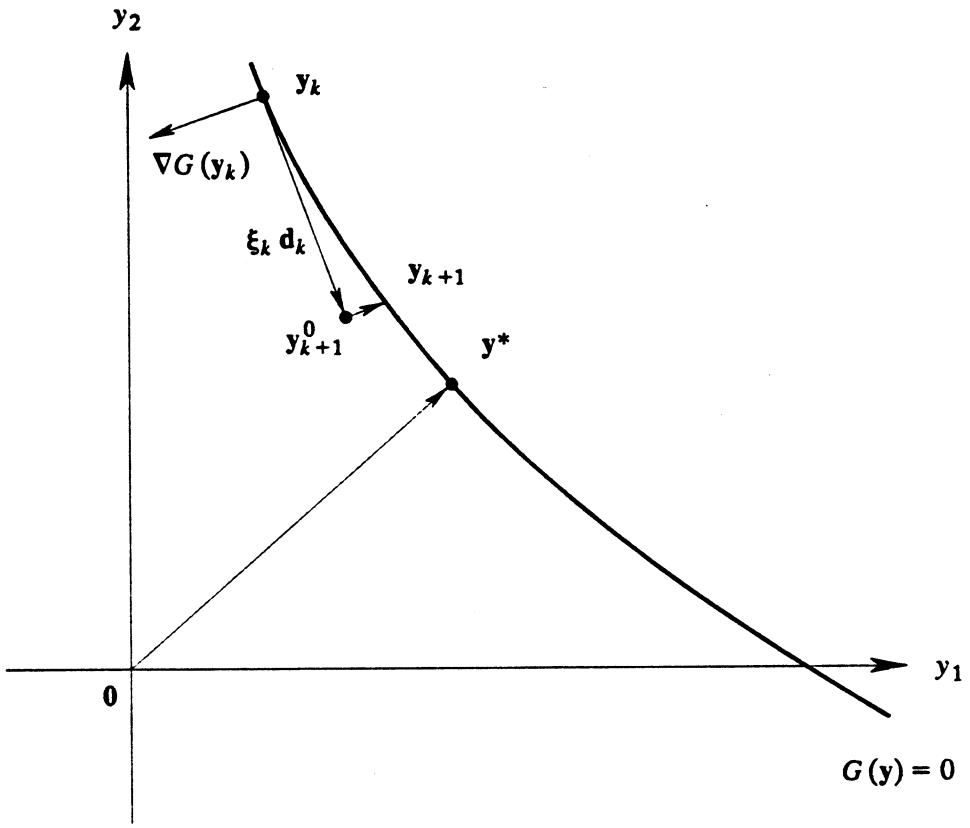


Figure 3.1 The Gradient Projection Method

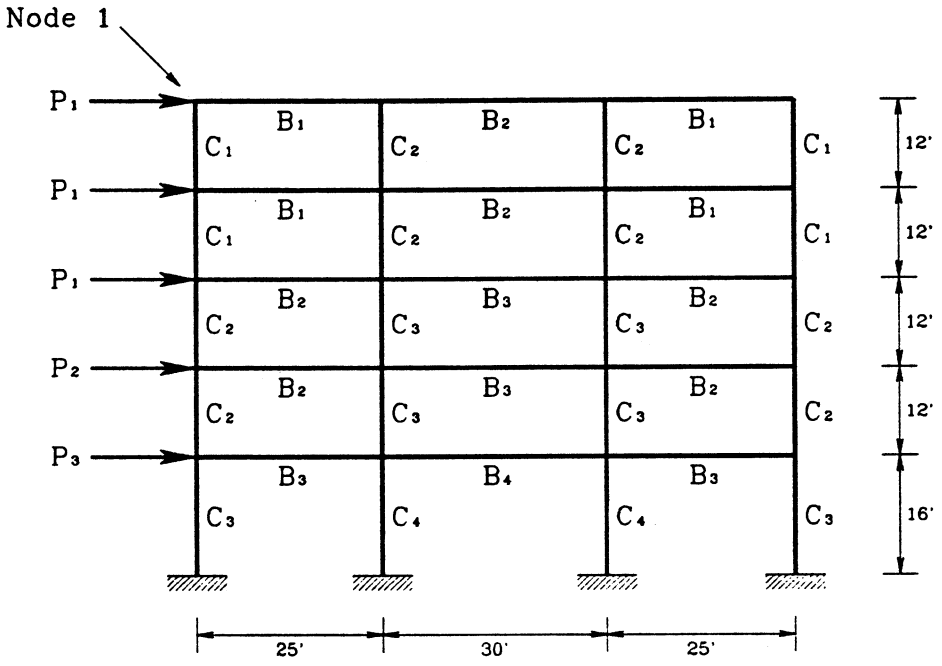


Figure 3.2 Example Frame Structure

CHAPTER 4

FINITE ELEMENT FORMULATION OF GEOMETRICALLY NONLINEAR ELASTIC STRUCTURES

4.1 Introduction

As was described in the previous two chapters, the first- and second-order structural reliability methods require the computation of the structural response, $\mathbf{s}$, as well as the response gradient, $\mathbf{J}_{\mathbf{s},\mathbf{v}}$. Conventional finite element methods for linear or nonlinear structures normally do not compute the response gradient. Thus, in order to develop a FORM/SORM finite element methodology, it is necessary to develop efficient algorithms for computing the response gradient. Of course, the response gradient can be computed by a finite difference scheme by repeatedly perturbing each variable. However, such an approach would be extremely time-consuming since in a problem with n random variables one would have to repeat the finite element solution at least $n+1$ times just to compute the gradient. Furthermore, the accuracy of a finite difference scheme is questionable for nonlinear structural problems where the solution is obtained through an iterative scheme. Since the iterative solution invariably involves numerical errors, the change in the computed response may not truly reflect the change due to the variable perturbation. Because of these efficiency and accuracy concerns, analytical expressions for computing the response gradients are of interest. As mentioned in Chapter 1, for geometrically nonlinear, elastic structures, analytical formulas for the response gradient have been derived by Wu and Arora [74] and Ryu et al. [57]. Both their derivations involve the derivative of the equilibrium equation which is in terms of the secant stiffness matrix of the structure. Since for nonlinear structures with multiple degrees of freedom the secant stiffness matrix is not available, these approaches are considered inappropriate. Ryu et al. [57] have made an interpretation of their formulas in terms of the tangent stiffness matrix. Their results are consistent with the formulas derived in Section 4.4.

This chapter first describes the continuum and finite element formulations for large deformation problems. The governing equations are given for three-dimensional continua in terms of the external and resisting forces. Then, general expressions for the response gradient of geometrically nonlinear, elastic structures are derived based on this formulation.

4.2 Continuum Formulation

For a continuum with large deformations, several alternative definitions of strains and stresses are available. The Green-Lagrangian strain **E** and the Almansi strain **E*** are the two most commonly used strain measures, which are defined as follows [50]:

$$\mathbf{E} = \frac{1}{2}[\mathbf{F}^T\mathbf{F} - \mathbf{I}] \tag{4.1}$$

$$\mathbf{E}^* = \frac{1}{2}[\mathbf{I} - (\mathbf{F}^{-1})^T\mathbf{F}^{-1}] \tag{4.2}$$

in which **F** is the deformation gradient of the continuum,

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \tag{4.3}$$

where **I** is the identity matrix, **u** denotes point displacements, and **X** and $\mathbf{x} = \mathbf{X} + \mathbf{u}$ respectively represent the coordinates of material points in the original and current configurations. In physical terms, the Green-Lagrangian strain is a measure of the deformation relative to the original configuration of the continuum, whereas the Almansi strain is a measure of the deformation relative to the current configuration.

The conjugate stress tensor associated with the Green-Lagrangian strain is the second Piola-Kirchhoff stress **T**, and that associated with the Almansi strain is the Cauchy stress **T***. These two stress measures are related by the following equation [50]:

$$\mathbf{T} = (\det \mathbf{F})\,\mathbf{F}^{-1}\mathbf{T}^*\,\mathbf{F}^{-T} \tag{4.4}$$

For elastic continua, stresses are functions of strains and material parameters $\mathbf{p}_m$, i.e.

$$\mathbf{T} = \mathbf{T}(\mathbf{E}, \mathbf{p}_m) \tag{4.5}$$

$$\mathbf{T}^* = \mathbf{T}^*(\mathbf{E}^*, \mathbf{p}_m) \tag{4.6}$$

Note that the components of $\mathbf{p}_m$ may be a subset of the basic random variables V if the material properties are random.

The equations of static equilibrium and boundary conditions with respect to the original and current configurations are as follows [50]:

Referential Formulation (original configuration):

$$\sum_L \frac{\partial (\mathbf{T}\mathbf{F}^T)_{Li}}{\partial X_L} + \rho_0 b_{0i} = 0 \qquad i = 1,2,3 \tag{4.7}$$

$\mathbf{u} = \bar{\mathbf{u}}$ on the displacement boundary

$\mathbf{F}\,\mathbf{T}\hat{\mathbf{n}}_0 = \bar{\mathbf{t}}_0$ on the traction boundary

Spatial Formulation (current configuration):

$$\sum_l \frac{\partial T^*_{li}}{\partial x_l} + \rho b_i = 0 \qquad i = 1,2,3 \tag{4.8}$$

$\mathbf{u} = \bar{\mathbf{u}}$ on the displacement boundary

$\mathbf{T}^* \hat{\mathbf{n}} = \bar{\mathbf{t}}$ on the traction boundary

where $\rho_0 b_{0i}$ and ρb_i are the body forces, $\bar{\mathbf{u}}$ is a known displacement field, $\hat{\mathbf{n}}_0$ and $\hat{\mathbf{n}}$ are the unit normals on the original and deformed traction boundaries, respectively, $\bar{\mathbf{t}}_0$ is the force acting on the current element per unit original area, and $\bar{\mathbf{t}}$ is the force acting on the current element per unit current area.

Either of the referential or spatial formulation can be used to solve the large deformation problem. In the following section, however, only the referential formulation is employed for the finite element discretization. This is because with the referential formulation the response gradient is easier to compute. Nevertheless, in deriving the expressions for the response gradient no assumption is made regarding the choice of the formulation. Hence, results derived for the gradient are applicable to both formulations.

4.3 Finite Element Formulation

Consider a standard finite element discretization of a continuum

$$\Omega \approx \bigcup_e \Omega_e , \qquad \bigcup_e \Omega_e = \varnothing \tag{4.9}$$

where Ω and Ω_e denote the continuum and finite elements, respectively. Within a finite element Ω_e, the displacement field $\mathbf{u}$ is approximated by

$$\mathbf{u} \approx \sum_{I=1}^{np} \mathbf{N}_I \mathbf{U}_I \tag{4.10}$$

where np is the number of nodal points of the element, $\mathbf{N}_I$ are the shape functions, and $\mathbf{U}_I$ are the nodal displacements of the element. For the convenience of the finite element formulation, the components of the Green-Lagrangian strain are arranged in the following vector form:

$$\mathbf{E} = \begin{bmatrix} E_{11} \\ E_{22} \\ E_{33} \\ 2E_{12} \\ 2E_{23} \\ 2E_{31} \end{bmatrix} = \mathbf{E}_0 + \mathbf{E}_l \tag{4.11}$$

where

$$\mathbf{E}_0 = \begin{bmatrix} \dfrac{\partial u_1}{\partial X_1} \\ \dfrac{\partial u_2}{\partial X_2} \\ \dfrac{\partial u_3}{\partial X_3} \\ \dfrac{\partial u_1}{\partial X_2} + \dfrac{\partial u_2}{\partial X_1} \\ \dfrac{\partial u_2}{\partial X_3} + \dfrac{\partial u_3}{\partial X_2} \\ \dfrac{\partial u_3}{\partial X_1} + \dfrac{\partial u_1}{\partial X_3} \end{bmatrix} \tag{4.12}$$

are the linear strain terms, and

$$\mathbf{E}_l = \frac{1}{2}\begin{bmatrix} \frac{\partial \mathbf{u}^T}{\partial X_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{u}^T}{\partial X_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \mathbf{u}^T}{\partial X_3} \\ \frac{\partial \mathbf{u}^T}{\partial X_2} & \frac{\partial \mathbf{u}^T}{\partial X_1} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{u}^T}{\partial X_3} & \frac{\partial \mathbf{u}^T}{\partial X_2} \\ \frac{\partial \mathbf{u}^T}{\partial X_3} & \mathbf{0} & \frac{\partial \mathbf{u}^T}{\partial X_1} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial X_1} \\ \frac{\partial \mathbf{u}}{\partial X_2} \\ \frac{\partial \mathbf{u}}{\partial X_3} \end{bmatrix} = \frac{1}{2}\mathbf{A}\,\boldsymbol{\theta} \tag{4.13}$$

are the nonlinear strain terms. If the approximate displacement field in Eq. 4.10 is adopted, the Green-Lagrangian strain in the element is

$$\mathbf{E} \approx \sum_{I=1}^{np} (\mathbf{B}_{0I} + \frac{1}{2}\mathbf{B}_{lI})\,\mathbf{U}_I \tag{4.14}$$

where

$$\mathbf{B}_{0I} = \begin{bmatrix} N_{I,x_1} & 0 & 0 \\ 0 & N_{I,x_2} & 0 \\ 0 & 0 & N_{I,x_3} \\ N_{I,x_2} & N_{I,x_1} & 0 \\ 0 & N_{I,x_3} & N_{I,x_2} \\ N_{I,x_3} & 0 & N_{I,x_1} \end{bmatrix} \tag{4.15}$$

$$\mathbf{B}_{lI} = \mathbf{A}\begin{bmatrix} N_{I,x_1}\mathbf{I} \\ N_{I,x_2}\mathbf{I} \\ N_{I,x_3}\mathbf{I} \end{bmatrix} = \mathbf{A}\,\mathbf{G}_I \tag{4.16}$$

in which $N_{I,x_L} = \partial N_I/\partial X_L$, $L = 1,2,3$. It can be verified using Eq. 4.14 that

$$d\mathbf{E} \approx \sum_{I=1}^{np} (\mathbf{B}_{0I} + \mathbf{B}_{lI})\,d\mathbf{U}_I = \sum_{I=1}^{np} \bar{\mathbf{B}}_I\,d\mathbf{U}_I \tag{4.17}$$

Using the principle of virtual displacements [50] with the above approximation, one obtains the following weak form of the equilibrium equations:

$$\mathbf{R} = \mathbf{P} \tag{4.18}$$

where

$$\mathbf{R} = \bigcup_e \mathbf{R}_e = \bigcup_e \int_{\Omega_e} \bar{\mathbf{B}}^{\mathrm{T}} \mathbf{T}\, d\Omega \tag{4.19}$$

is the resisting force vector, and

$$\mathbf{P} = \bigcup_e \mathbf{P}_e = \bigcup_e \left[\int_{\partial\Omega_e} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}}_0\, d(\partial\Omega) + \int_{\Omega_e} \mathbf{N}^{\mathrm{T}} \rho_0 \mathbf{b}_0\, d\Omega \right] \tag{4.20}$$

is the external load vector, in which $\bar{\mathbf{B}} = [\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \ldots, \bar{\mathbf{B}}_{np}]$, $\mathbf{N} = [\mathbf{N}_1, \mathbf{N}_2, \ldots, \mathbf{N}_{np}]$, and $\partial\Omega_e$ denotes the element boundary. Note that in Eq 4.19 the second Piola-Kirchhoff stress $\mathbf{T}$ is rearranged into a vector form $[T_{11}, T_{22}, T_{33}, T_{12}, T_{23}, T_{31}]^{\mathrm{T}}$ to be consistent with the definition of the strain vector.

A solution scheme for Eq. 4.18 based on the Newton method is as follows:

$$\mathbf{K}\,\Delta\mathbf{U}_{i+1} = \mathbf{P}_i - \mathbf{R}_i \tag{4.21}$$

$$\mathbf{U}_{i+1} = \mathbf{U}_i + \Delta\mathbf{U}_{i+1} \tag{4.22}$$

in which the subscripts i and $i+1$ denote the iterations at which the quantities are computed, $\mathbf{U}$ is the global nodal displacement vector, and $\mathbf{K} = \mathbf{K}_T - \mathbf{K}_L$ is the current stiffness matrix, where $\mathbf{K}_T = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right]_i$ is the tangent stiffness and $\mathbf{K}_L = \left[\frac{\partial \mathbf{P}}{\partial \mathbf{U}}\right]_i$ is the initial load stiffness. It can be shown that [78]

$$\mathbf{K}_T = \bigcup_e \mathbf{K}_{Te} = \bigcup_e \left[\int_{\Omega_e} \bar{\mathbf{B}}^{\mathrm{T}} \mathbf{D}_T \bar{\mathbf{B}}\, d\Omega + \int_{\Omega_e} \mathbf{G}^{\mathrm{T}} \mathbf{H} \mathbf{G}\, d\Omega \right] \tag{4.23}$$

where $\mathbf{D}_T = \frac{\partial \mathbf{T}}{\partial \mathbf{E}}$ is the tangent elasticity matrix of the material, $\mathbf{G} = [\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_{np}]^{\mathrm{T}}$, and

$$\mathbf{H} = \begin{bmatrix} S_{11}\mathbf{I} & S_{12}\mathbf{I} & S_{13}\mathbf{I} \\ S_{21}\mathbf{I} & S_{22}\mathbf{I} & S_{23}\mathbf{I} \\ S_{31}\mathbf{I} & S_{32}\mathbf{I} & S_{33}\mathbf{I} \end{bmatrix} \tag{4.24}$$

The initial load stiffness matrix $\mathbf{K}_L$ is nonzero only when follower forces exist, i.e., when the external loads are dependent on the continuum deformation. The large deformation problem is solved by repeatedly applying Eqs. 4.21 and 4.22 until convergence is achieved.

4.4 Formulation of the Response Gradient

Let $\mathbf{V}_m$, $\mathbf{V}_g$, and $\mathbf{V}_l$ denote the basic random variables respectively representing uncertain material properties, structural geometry, and external loads, and let $\mathbf{v}_m$, $\mathbf{v}_g$, and $\mathbf{v}_l$ represent their respective outcomes. Observe that the original nodal coordinates, $\mathbf{X}^d$, are functions of $\mathbf{v}_g$. In addition, for geometrically nonlinear, elastic structures under static loads, the nodal displacements $\mathbf{U}$ are functions of $\mathbf{X}^d$, $\mathbf{v}_m$, and $\mathbf{v}_l$; the resisting forces $\mathbf{R}$ are functions of $\mathbf{X}^d$, $\mathbf{U}$, and $\mathbf{v}_m$; and the external loads $\mathbf{P}$ are functions of $\mathbf{X}^d$, $\mathbf{U}$, and $\mathbf{v}_l$. That is,

$$\mathbf{R} = \mathbf{R}(\mathbf{X}^d(\mathbf{v}_g), \mathbf{U}(\mathbf{X}^d, \mathbf{v}_m, \mathbf{v}_l), \mathbf{v}_m) \tag{4.25}$$

$$\mathbf{P} = \mathbf{P}(\mathbf{X}^d(\mathbf{v}_g), \mathbf{U}(\mathbf{X}^d, \mathbf{v}_m, \mathbf{v}_l), \mathbf{v}_l) \tag{4.26}$$

The gradient of the structural response is obtained by taking the derivative of Eq. 4.18 with respect to $\mathbf{v}$, i.e.,

$$\nabla_{\mathbf{v}}\mathbf{R} = \nabla_{\mathbf{v}}\mathbf{P} \tag{4.27}$$

By the chain rule of differentiation, the derivatives with respect to the material property variables are

$$\nabla_{\mathbf{v}_m}\mathbf{R}(\mathbf{X}^d(\mathbf{v}_g), \mathbf{U}(\mathbf{X}^d, \mathbf{v}_m, \mathbf{v}_l), \mathbf{v}_m) = \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\bigg|_{\mathbf{v}_m} \nabla_{\mathbf{v}_m}\mathbf{U} + \frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\bigg|_{\mathbf{U}} \tag{4.28}$$

$$\nabla_{\mathbf{v}_m}\mathbf{P}(\mathbf{X}^d(\mathbf{v}_g), \mathbf{U}(\mathbf{X}^d, \mathbf{v}_m, \mathbf{v}_l), \mathbf{v}_l) = \frac{\partial \mathbf{P}}{\partial \mathbf{U}}\bigg|_{\mathbf{v}_m} \nabla_{\mathbf{v}_m}\mathbf{U} \tag{4.29}$$

By definition, $\dfrac{\partial \mathbf{R}}{\partial \mathbf{U}}\Big|_{\mathbf{v}_m} = \mathbf{K}_T$ and $\dfrac{\partial \mathbf{P}}{\partial \mathbf{U}}\Big|_{\mathbf{v}_m} = \mathbf{K}_L$. Substituting the preceding equations into Eq. 4.27 and rearranging terms, one obtains the following expression for the gradient of the displacement with respect to the material-property variables:

$$\nabla_{\mathbf{v}_m}\mathbf{U} = \mathbf{K}^{-1}\left[-\frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\bigg|_{\mathbf{U}}\right] \tag{4.30}$$

If the referential formulation in Eq. 4.19 is used,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\bigg|_{\mathbf{U}} = \bigcup_e \int_{\Omega_e} \bar{\mathbf{B}}^{\mathrm{T}} \frac{\partial \mathbf{T}}{\partial \mathbf{v}_m}\bigg|_{\mathbf{U}} d\Omega \tag{4.31}$$

in which the derivatives $\frac{\partial \mathbf{T}}{\partial \mathbf{v}_m}\Big|_{\mathbf{U}}$ are computed by using Eq. 4.5 with the strains fixed. Note that $\mathbf{v}_m$ contains the random material parameters in $\mathbf{p}_m$. Equation 4.30 is consistent with Ryu et al.'s interpretation of their gradient formula which is in terms of the secant stiffness matrix [57].

Once $\nabla_{\mathbf{v}_s}\mathbf{U}$ is obtained, one can apply the chain rule to Eqs. 4.14 and 4.5 (or 4.6) to determine the gradients of strains and stresses. Let ϵ and σ denote either E and T or E* and T*. Since ϵ are only functions of the nodal displacements $\mathbf{U}_I$ of the element,

$$\begin{aligned}\nabla_{\mathbf{v}_s}\epsilon &= \frac{\partial \epsilon}{\partial \mathbf{U}}\Big|_{\mathbf{v}_s}\nabla_{\mathbf{v}_s}\mathbf{U} \\ &= \sum_{I=1}^{np}\bar{\mathbf{B}}_I\nabla_{\mathbf{v}_s}\mathbf{U}_I\end{aligned} \tag{4.32}$$

Furthermore, σ are functions of $\mathbf{v}_m$ and ϵ. Therefore,

$$\nabla_{\mathbf{v}_s}\sigma = \frac{\partial \sigma}{\partial \mathbf{v}_m}\Big|_{\epsilon} + \frac{\partial \sigma}{\partial \epsilon}\Big|_{\mathbf{v}_s}\frac{\partial \epsilon}{\partial \mathbf{U}}\Big|_{\mathbf{v}_s}\nabla_{\mathbf{v}_s}\mathbf{U} \tag{4.33}$$

where $\frac{\partial \sigma}{\partial \epsilon}\Big|_{\mathbf{v}_s} = \mathbf{D}_T$.

In the above expressions for the gradients, the stiffness matrix **K**, its decomposition, and $\mathbf{D}_T$ are readily available if Newton's method is used to solve for the response. Therefore, only one new matrix, $\frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\Big|_{\mathbf{U}}$, needs to be formed for the computation of the gradients. In the formation of this matrix, the individual element terms may be computed either by their analytical expressions or by a finite difference scheme. The more efficient approach depends on the type of element.

The expressions for the gradients of the response with respect to $\mathbf{v}_g$ and $\mathbf{v}_l$ are derived in a similar manner. The gradients of the displacements, strains, and stresses with respect to the load variables $\mathbf{v}_l$ are

$$\nabla_{\mathbf{v}_l}\mathbf{U} = \mathbf{K}^{-1}\frac{\partial \mathbf{P}}{\partial \mathbf{v}_l}\Big|_{\mathbf{U}} \tag{4.34}$$

$$\nabla_{v_l}\epsilon = \frac{\partial \epsilon}{\partial U}\Big|_{v_l} \nabla_{v_l} U$$

$$= \sum_{I=1}^{np} \bar{B}_I \nabla_{v_l} U_I \tag{4.35}$$

$$\nabla_{v_l}\sigma = \frac{\partial \sigma}{\partial \epsilon}\Big|_{v_l} \frac{\partial \epsilon}{\partial U}\Big|_{v_l} \nabla_{v_l} U \tag{4.36}$$

where $\frac{\partial \sigma}{\partial \epsilon}\Big|_{v_l} = D_T$. If the referential formulation in Eq. 4.20 is used,

$$\frac{\partial P}{\partial v_l}\Big|_U = \bigcup_e \left[\int_{\partial \Omega_e} N^T \frac{\partial \bar{t}_0}{\partial v_l}\Big|_U d(\partial\Omega) + \int_{\Omega_e} N^T \frac{\partial(\rho_0 b_0)}{\partial v_l}\Big|_U d\Omega \right] \tag{4.37}$$

This is the only new matrix to form in the computation of the gradient with respect to the load variables.

The gradients of the responses with respect to the geometry variables v_g are

$$\nabla_{v_g} U = K^{-1} \left[\frac{\partial P}{\partial X^d}\Big|_U - \frac{\partial R}{\partial X^d}\Big|_U \right] \frac{\partial X^d}{\partial v_g} \tag{4.38}$$

$$\nabla_{v_g}\epsilon = \frac{\partial \epsilon}{\partial v_g}\Big|_U + \frac{\partial \epsilon}{\partial U}\Big|_{v_g} \nabla_{v_g} U \tag{4.39}$$

in which, noting that for fixed displacements strains are dependent on the original configuration,

$$\frac{\partial \epsilon}{\partial v_g}\Big|_U = \sum_{I=1}^{np}\sum_{J=1}^{np} \left[\frac{\partial B_{0I}}{\partial X_J^d}\Big|_U + \frac{1}{2}\frac{\partial B_{lI}}{\partial X_J^d}\Big|_U \right] \frac{\partial X_J^d}{\partial v_g} U_I \tag{4.40}$$

where X_J^d are the original nodal coordinates of the element, and

$$\frac{\partial \epsilon}{\partial U}\Big|_{v_g} \nabla_{v_g} U = \sum_{I=1}^{np} \bar{B}_I \nabla_{v_g} U_I \tag{4.41}$$

$$\nabla_{v_g}\sigma = \frac{\partial \sigma}{\partial \epsilon}\Big|_{v_g} \frac{\partial \epsilon}{\partial v_g}\Big|_U + \frac{\partial \sigma}{\partial \epsilon}\Big|_{v_g} \frac{\partial \epsilon}{\partial U}\Big|_{v_g} \nabla_{v_g} U \tag{4.42}$$

where $\frac{\partial \sigma}{\partial \epsilon}\Big|_{v_g} = D_T$. In the preceding equations, $\frac{\partial P}{\partial X^d}\Big|_U$, $\frac{\partial R}{\partial X^d}\Big|_U$, $\frac{\partial X^d}{\partial v_g}$, $\frac{\partial B_{0I}}{\partial X_J^d}\Big|_U$, and $\frac{\partial B_{lI}}{\partial X_J^d}\Big|_U$ are the new matrices involved in the gradient computation. The third term is easy to compute because X^d is usually an explicit function of v_g. The computation of

the last two terms is tedious but straightforward. The first two terms, on the other hand, are difficult to compute because the domains of the integrals in the expressions of **R** and **P** are dependent on $\mathbf{X}^d$. To carry out these derivatives, the integral domains in Eqs. 4.19 and 4.20 must be mapped onto fixed configurations so that one can take the derivative inside the integral. Such a mapping is a standard scheme in conventional finite element analysis, where the local element coordinate system is transformed onto a natural coordinate system for convenience in the Gauss quadrature computation. In any case, if the analytical expressions of $\frac{\partial \mathbf{P}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}}$ and $\frac{\partial \mathbf{R}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}}$ for a particular element become too complicated, one can always use a finite difference scheme to compute these matrices. Since only the elements next to the structural boundary whose nodal coordinates are random need to be included in this computation, a finite difference scheme normally would not require excessive time.

Before closing this section, it should be pointed out that the analytical expressions for $\nabla_{\mathbf{v}}\mathbf{U}$, $\nabla_{\mathbf{v}}\epsilon$, and $\nabla_{\mathbf{v}}\sigma$ are not used directly in computing the gradient of the limit-state function. An efficient scheme for evaluating $\nabla_{\mathbf{v}} g(\mathbf{v},\mathbf{s})$ that avoids the direct computation of these terms is introduced in the following chapter.

4.5 Concluding Remarks

In this chapter, a finite element formulation for computing the response as well as the response gradient of an elastic continuum under static loads and large deformation is developed. Although the formulation is for a three dimensional continuum, the expressions for the response gradient apply to all types of elements. For each type of element, appropriate expressions should be used for the terms appearing in the equations for **R** and **P**. For instance, for a truss element, one should substitute the axial force for **T** in Eq. 4.19. The essential matrices for the elements used in the example applications in Chapter 7 are derived in Appendix A.

In contrast to the structural response, no iterations are required for computing the

gradients. The gradients are computed in terms of the results at the end of the iterative solution for the response. Hence, it is much simpler to compute the gradients than the response itself. This feature should be exploited when choosing the optimization scheme for finding the design point in a reliability analysis. Specifically, one should choose an algorithm which makes as much use of the gradient as possible to speed up the convergence. In this respect, the modified HL-RF method is most desirable.

It is important to note that although in Section 4.3 the weak form of the equilibrium equation was formulated with respect to the original configuration, the expressions for the gradient hold for the spatial formulation as well. However, the terms involved in the expressions of the gradient become more difficult to compute for the spatial formulation. This is because the domains of the integrals in the expressions for **R** and **P** are then functions of the displacements and, therefore, are dependent on the basic random variables. Finally, although it was not emphasized, the expressions for the gradient derived in this chapter are valid for structures with nonlinear material, as long as the material is elastic.

CHAPTER 5

COMPUTATIONAL ASPECTS OF FINITE ELEMENT RELIABILITY ANALYSIS

5.1 General Remarks

This chapter is devoted to several issues concerning the computational efficiency and accuracy of the finite element reliability method. Several such issues were addressed in the preceding chapters, including the study of efficient optimization algorithms for finding the design point and the derivation of analytical expressions for the response gradient of geometrically nonlinear, elastic structures. In this chapter, three additional issues are discussed. These concern the numerical computation of the gradient, modeling and discretization of random fields, and reduction of the number of basic random variables.

5.2 Computation of the Gradient

As shown in Chapters 2 and 3, the gradient of the limit-state function with respect to the basic variables is an essential component of the FORM analysis. In the most general case, assume the limit-state function is expressed in terms of a set of displacements, $\mathbf{U}$, stresses, σ, or strains, ϵ, and a set of thresholds included in the vector of basic variables. Noting the dependences $\mathbf{U} = \mathbf{U}(\mathbf{v})$, $\epsilon = \epsilon(\mathbf{U}(\mathbf{v}),\mathbf{v})$, and $\sigma = \sigma(\epsilon(\mathbf{U}(\mathbf{v}),\mathbf{v}),\mathbf{v})$, the expression for the gradient, then, takes on the form

$$\begin{aligned}\nabla g(\mathbf{v},\mathbf{s}) &= \nabla g(\mathbf{v},\mathbf{U},\epsilon,\sigma) \\ &= \left[\frac{\partial g}{\partial \mathbf{v}} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial \mathbf{v}}\bigg|_{\mathbf{U}} + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \mathbf{v}}\bigg|_{\epsilon} + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\bigg|_{\mathbf{v}}\frac{\partial \epsilon}{\partial \mathbf{v}}\bigg|_{\mathbf{U}}\right] \\ &\quad + \left[\frac{\partial g}{\partial \mathbf{U}} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial \mathbf{U}}\bigg|_{\mathbf{v}} + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\bigg|_{\mathbf{v}}\frac{\partial \epsilon}{\partial \mathbf{U}}\bigg|_{\mathbf{v}}\right]\nabla_{\mathbf{v}}\mathbf{U} \qquad (5.1)\end{aligned}$$

The four partial derivatives of $g(.)$ in the right-hand side of this expression are easy to compute, since the limit-state function is usually an explicit function of the responses and the response thresholds. For geometrically nonlinear, elastic structures, the partial derivatives of ϵ and σ are computed by the expressions derived in Chapter 4. In particular, note

that $\frac{\partial \epsilon}{\partial v}\Big|_U$ involves the geometry variables, $\frac{\partial \sigma}{\partial v}\Big|_\epsilon$ involves the material variables, $\frac{\partial \sigma}{\partial \epsilon}\Big|_v = \mathbf{D}_T$ is the tangent elasticity matrix, and $\frac{\partial \epsilon}{\partial U}\Big|_v$ is given by Eqs. 4.32, 4.35, and 4.41. Finally, $\nabla_v \mathbf{U}$ is computed as the solution of

$$\mathbf{K} \nabla_v \mathbf{U} = \frac{\partial \mathbf{P}}{\partial v}\Big|_U - \frac{\partial \mathbf{R}}{\partial v}\Big|_U \tag{5.2}$$

The gradient $\nabla g(\mathbf{v}, \mathbf{s})$ can be obtained by solving the above equation and substituting $\nabla_v \mathbf{U}$ into Eq. 5.1. For a structure with m degrees of freedom and n basic random variables, this direct approach would require n solutions of m simultaneous equations. That can be rather time consuming since both m and n can be very large in real applications.

To avoid costly and unnecessary computations, the adjoint method suggested by Arora and Haug [4] is employed. Instead of computing $\frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v$ and $\nabla_v \mathbf{U}$ separately, in this method the product $\left[\frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v\right] \nabla_v \mathbf{U}$ is computed by solving an adjoint problem.

The procedure of the adjoint method is as follows:

1. Compute $\frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v$.
2. Solve the following equation for an auxiliary vector λ:

$$\lambda^T \mathbf{K} = \frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v \tag{5.3}$$

3. Postmultiply both sides of Eq. 5.3 by $\nabla_v \mathbf{U}$ and make use of the identity in Eq. 5.2 to obtain the following equation:

$$\lambda^T \mathbf{K} \nabla_v \mathbf{U} = \left[\frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v\right] \nabla_v \mathbf{U}$$
$$= \lambda^T \left[\frac{\partial \mathbf{P}}{\partial v}\Big|_U - \frac{\partial \mathbf{R}}{\partial v}\Big|_U\right] \tag{5.4}$$

4. Substitute $\lambda^T \left[\frac{\partial P}{\partial v}\Big|_U - \frac{\partial R}{\partial v}\Big|_U \right]$ for $\left[\frac{\partial g}{\partial U} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial U}\Big|_v + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial U}\Big|_v \right]$ $\nabla_v U$ in Eq. 5.1 and compute and add $\frac{\partial g}{\partial v} + \frac{\partial g}{\partial \epsilon}\frac{\partial \epsilon}{\partial v}\Big|_U + \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial v}\Big|_\epsilon +$ $\frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial \epsilon}\Big|_v \frac{\partial \epsilon}{\partial v}\Big|_U$ to obtain $\nabla g(\mathbf{v}, s)$.

In contrast to the direct method, the adjoint method solves the m simultaneous equations in Eq. 5.3 only once. Hence, this method saves considerable amount of computation time when the number of basic random variables is large.

5.3 Modeling and Discretization of Random Fields

5.3.1 Random Field Modeling

Let $W(\mathbf{X})$ denote a random field that models the random variability of a material property, structural geometry, or load quantity in space. In most applications, $W(\mathbf{X})$ is assumed to be Gaussian because of practical convenience and lack of alternative models. The Gaussian random field has a convenient property in that the field is completely defined by its mean function $\mu_W(\mathbf{X})$, variance function $\sigma_W^2(\mathbf{X})$, and autocorrelation coefficient function $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$. If it is further assumed that the random field is homogeneous, as is customarily done in practice, the mean and variance functions are constant, and the autocorrelation coefficient function $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j) = \rho_{WW}(\mathbf{X}_j - \mathbf{X}_i)$. These properties greatly simplify the parameter estimation and modeling of random fields.

Despite its convenience and popularity, the Gaussian model is not applicable in many situations. For example, some random quantities have bounded distributions either by nature or by definition, such as the Young's modulus of a material which is always positive. Other random quantities may exhibit skewness in their distributions, such as the extremes of certain load quantities. Non-Gaussian models are, thus, necessary to describe such quantities.

The Nataf distribution model defined by Eqs. 2.20 – 2.22 of Chapter 2 has been

used by Grigoriu [25] and Der Kiureghian [14] to model non-Gaussian processes with a prescribed marginal distribution and mean and autocorrelation functions. Let the random field $W(\mathbf{X})$ have the marginal CDF $F_W(w(\mathbf{X}))$. The random field is completely defined by assuming that the transformed process

$$Z(\mathbf{X}) = \Phi^{-1}[F_W(W(\mathbf{X}))] \tag{5.5}$$

is Gaussian with zero mean, unit variance and autocorrelation coefficient function $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$. For any set of $\mathbf{X}_i$ and $\mathbf{X}_j$, the relation between $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$ and $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$ is as in Eq. 2.22 with μ_i, σ_i, ρ_{ij}, and $\rho_{0,ij}$ replaced by $\mu_W(\mathbf{X}_i)$, $\sigma_W(\mathbf{X}_i)$, $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$, and $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$, respectively.

There are certain restrictions on $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$ for $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$ to be a valid autocorrelation coefficient function. Such restrictions for homogeneous and non-homogeneous fields with various types of marginal distributions are discussed in Ref. 14. In order to avoid such restrictions, Der Kiureghian [14] has proposed that instead of estimating $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$ from the observed data $w(\mathbf{X})$ and then computing $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$, one directly estimate $\rho_{ZZ}(\mathbf{X}_i, \mathbf{X}_j)$ by analyzing the transformed data $z(\mathbf{X}) = \Phi^{-1}[F_W(w(\mathbf{X}))]$. By doing so, the restrictions on $\rho_{WW}(\mathbf{X}_i, \mathbf{X}_j)$ are lifted, and the proposed model always works. This non-Gaussian model is used in Example 7.2 of Chapter 7 to describe the Young's modulus and Poisson's ratio of a plate as random fields.

Yamazaki and Shinozuka [75] also proposed a model for homogeneous non-Gaussian random fields which is in terms of the marginal distributions and power spectral density function of the field. Digital simulation of such a field using an iterative scheme is described in Ref. 75. It can be shown that their model is an approximation of the Nataf model described above.

5.3.2 The Random Field Mesh

As discussed in Chapter 2, when the uncertainties in a structure or its environment are modeled by random fields, for FORM/SORM analysis it is necessary to represent such random fields in terms of random variables. Several methods have been suggested for such

representation of random fields. One approach requires that the domain of the random field be discretized into adjacent and non-overlapping elements, denoted *random field elements*, similar to the discretization employed in the finite element method. The value of the material property or load in each random field element is then represented by a single random variable.

The selection of the finite element mesh and the random field mesh is an important task in finite element reliability analysis. For the finite element mesh, the proper element size is controlled by the expected stress gradient in each region of the structure. For the random field mesh, two factors need to be taken into account. One is the rate of fluctuation of the random field, as measured, for example, by the correlation length. The correlation length is defined here as the length over which the autocorrelation coefficient function drops to a small values, say e^{-1}. For a short correlation length, the rate of fluctuation of the random field is high and a fine mesh is required. Limited experience indicates that an element size one quarter to one half the correlation length is appropriate [15]. The second is the numerical stability of the transformation to the standard normal space, which was described in Section 2.4 of Chapter 2. This factor is important because if the random field mesh is excessively fine, the discretized element variables are highly correlated and their correlation matrix is nearly singular. The transformation to the standard normal space then may become numerically unstable. Hence, this second factor provides a lower bound on the element size.

Usually, it is not efficient to use the same mesh for both the finite element and the random field discretizations, since the selection criteria are entirely different. Hence, Der Kiureghian and Ke [15] suggested using separate meshes for the finite element discretization and each random field of properties or loads. They further suggested that the finite element mesh be selected such that it satisfies all the requirements arising from the stress gradient as well as the rate of fluctuation of each random field. This is because a region around which the structural property or load has a high rate of fluctuation is likely to have

a high stress gradient. A separate mesh for each random field of property or load is then considered, which is equal to or coarser than the finite element mesh, such that each random field element is a block of one or more finite elements. This approach reduces the number of basic random variables and avoids the numerical instability in the probability transformation.

All these rules provide useful guidelines regarding the mesh selection. Further studies and experience are needed before more specific and quantitative rules can be developed.

5.3.3 Representation of Random Fields

Six methods have been suggested for representation of random fields in terms of random variables: the spatial averaging method, the midpoint method, the nodal-point method, the interpolation method, and two series expansion methods. These methods are described in the following subsections.

5.3.3.1 The Spatial Averaging Method

The spatial averaging method, suggested by Vanmarcke [70] and Vanmarcke and Grigoriu [72], uses the local average of the field over a random field element to represent the random quantity for the element. For a random field $W(\mathbf{X})$, the discretized value for an element i is given by

$$W_i = \frac{1}{\Omega_i} \int_{\Omega_i} W(\mathbf{X})\, d\Omega \tag{5.6}$$

where Ω_i is the domain of the element. For homogeneous fields and rectangular elements whose edges are parallel to the coordinate axes, Vanmarcke [71] has derived expressions for the covariances of the discretized variables W_i in terms of the autocovariance function of $W(\mathbf{X})$.

According to Der Kiureghian and Ke [15], for Gaussian fields the spatial averaging method yields accurate results even for rather coarse meshes of the random field. Unfortunately, there are two difficulties with the application of this method in finite element reli-

ability analysis. First of all, for a 2D or 3D continuum of arbitrary shape it is not always possible to discretize the domain into rectangular elements. For non-rectangular elements, Vanmarcke [71] has suggested that the element be replaced by a collection of adjacent, non-overlapping rectangular elements so that the same formulas for rectangular elements can be used. Der Kiureghian and Ke [15] replaced the integral in Eq. 5.6 by a Gaussian quadrature for calculating the covariances of W_i. Righetti and Harrop-Williams [54] proposed another approximation approach for 2D problems, in which trapezoidal and triangular elements are mapped into their equivalent rectangular elements. The equivalent rectangular element is defined as the rectangle possessing the same centroid and area of the original element, and sides proportional to the maximum difference between the nodal coordinates in each direction. All these approximation methods introduce errors in the computed covariance matrix of W_i. Experience shows that even small numerical errors in this computation may lead to a non-positive definite covariance matrix. Consequently, the probability transformation in Eq. 2.3 and, in turn, FORM and SORM calculations may break down.

Another drawback of the spatial averaging method is that the robability distribution of W_i is difficult or impossible to obtain, except when the field is Gaussian, in which case W_i is also Gaussian. Thus, this method is appropriate only when the random field is Gaussian. This further restricts the use of the spatial averaging method to the Gaussian case in finite element reliability analysis.

5.3.3.2 The Midpoint Method and the Nodal-Point Method

The midpoint and nodal-point methods are two point discretization methods which represent the uncertainties of a random field by the values at some specific points. In the midpoint method, proposed by Dendrou and Houstis [11], Hisada and Nakagiri [31], Shinozuka and Dasgupta[61], and Der Kiureghian and Ke [15], the field value over an element is represented by the value at the centroid of the element. Hence, the randomness in a random field element i is represented by the random variable

$$W_i = W(\tilde{\mathbf{X}}_i) \tag{5.7}$$

in which $\tilde{\mathbf{X}}_i = \frac{1}{np}\sum_{j=1}^{np} \mathbf{X}_j^d$ are the coordinates of the centroid, where np is the number of nodes of the random field element, and $\mathbf{X}_j^d$ are the nodal coordinates.

The nodal-point method, proposed by Hisada and Nakagiri [30], represents the random field in terms of the values at the nodal points of the finite element mesh. This method was used to discretize a random field modeling the random geometry of a structure. In this method, the randomness of the field at node i is represented by

$$W_i = W(\tilde{\mathbf{X}}_i) \tag{5.8}$$

where $\tilde{\mathbf{X}}_i = \mathbf{X}_i^d$ are the coordinates of node i.

In both the midpoint method and the nodal-point method, the mean, variance, and marginal distribution of W_i are the same as those of the process at point $\tilde{\mathbf{X}}_i$. The correlation coefficient matrix of W_i is directly computed in terms of the autocorrelation coefficient function of the random field,

$$\rho_{W_i W_j} = \rho_{WW}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j) \tag{5.9}$$

and the joint distribution for any set of W_i is given by the specified distribution of the random field.

As pointed out by Der Kiureghian and Ke [15], the midpoint method tends to overrepresent the variability of the field within each element, and it does not provide as accurate a result as the spatial averaging method for a coarse random field mesh. However, these point discretization methods have three advantages: First, no complicated computations are required for the covariance matrix and the method is easy to implement. Second, the correlation coefficient matrix obtained by Eq. 5.9 is always positive-definite, provided a valid autocorrelation function is specified. Hence, the numerical stability problem arising in the spatial averaging method does not exist in this case. Most importantly, the distribution information on the discretized variables W_i is retained and the method is not restricted to Gaussian random fields.

5.3.3.3 The Interpolation Method

Liu et al. [42] suggested the discretization of the random field $W(\mathbf{X})$ into q random nodal values, W_i, $i = 1,...,q$. The value at an arbitrary point is obtained by the following interpolation rule:

$$\tilde{W}(\mathbf{X}) = \sum_{i=1}^{q} N_i(\mathbf{X}) W_i \tag{5.10}$$

where W_i is the value of $W(\mathbf{X})$ at node $\mathbf{X}_i$, and $N_i(\mathbf{X})$ are shape functions. The number q is not necessarily equal to the number of finite elements and the shape functions $N_i(\mathbf{X})$ need not be the same as the finite element interpolation functions for the displacement field. Since the choice of the q nodal points and shape functions is arbitrary, the interpolation method constitutes a class of random field discretization methods. In particular, if the nodes are chosen to be the centroids of the random field elements and the shape functions are assumed to be unity inside each element and zero elsewhere, the interpolation method becomes identical to the midpoint method described in the previous section.

Liu et al. [42] further suggested a method to reduce the number of random variables W_i. The random vector $\mathbf{W}$ is transformed into an uncorrelated random vector $\mathbf{C}$ by

$$\mathbf{C} = \psi^{\mathrm{T}} \mathbf{W} \tag{5.11}$$

such that $\mathrm{Cov}(\mathbf{C}, \mathbf{C})$ is diagonal. The matrix ψ is obtained by solving the eigenproblem

$$\mathrm{Cov}(\mathbf{W}, \mathbf{W})\,\psi = \psi \Lambda \tag{5.12}$$

where Λ is the eigenvalue matrix containing the variances of $\mathbf{C}$. Liu et al. [42] observed that a good approximation of the random field can be obtained by retaining only the C_i with large variances, thus reducing the number of random variables. It should be emphasized, however, that this reduction is only applicable to Gaussian random fields. This is because the distribution of $\mathbf{C}$ is generally unknown or difficult to obtain unless $W(\mathbf{X})$ is Gaussian.

5.3.3.4 Series Expansion Methods

Two series expansion methods have been suggested for second-moment stochastic finite-element analysis. One is the basis random variable method proposed by Lawrence [36], and the other is the kernel expansion method proposed by Spanos and Ghanem [63]. In the first method, the random field is expanded into a double series

$$W(\mathbf{X}) = \sum_{i=0}^{\infty} \sum_{j=1}^{\infty} w_{ij} e_i \phi_j(\mathbf{X}) \tag{5.13}$$

in which $\phi_j(\mathbf{X})$ are a set of linearly independent shape functions, e_i are independent basis random variables having the properties

$$\mathrm{E}[e_i] = \begin{cases} 1 & i = 0 \\ 0 & i = 1,2,\ldots \end{cases} \tag{5.14}$$

$$\mathrm{E}[e_i e_j] = \delta_{ij} \tag{5.15}$$

in which δ_{ij} is the Kronecker delta, and the coefficients w_{ij} are determined by least-square fitting to the moment functions of the random field. In Ref. 36, Legendre polynomials were suggested for the shape functions, $\phi_j(\mathbf{X})$. In applications, only a few dominant terms are included in the expansion.

The kernel expansion method [63] employs the Karhunen-Loeve orthogonal expansion [44] to decompose a one-dimensional random field. The random field is expanded into the sum of its mean function and a single series

$$W(X) = \mu_W(X) + \sum_{i=0}^{\infty} W_i \sqrt{\lambda_i}\, \phi_i(X) \tag{5.16}$$

where W_i are random coefficients independent of X, and λ_i and $\phi_i(X)$ are the eigenvalues and eigenfunctions of the covariance kernel, respectively. The latter are obtained as the solutions of the eigenvalue problem

$$\int \mathrm{Cov}(X, t)\, \phi_i(X)\, dt = \lambda_i \phi_i(X) \tag{5.17}$$

Similar to the basis random variable method, the series in Eq. 5.16 is truncated after the first few dominant terms. Since the series in Eq. 5.16 has a zero mean and the eigenfunctions are orthogonal, the random coefficients W_i have properties similar to the basis

random variables, i.e.,

$$E[W_i] = 0 \tag{5.18}$$

$$E[W_i W_j] = \delta_{ij} \tag{5.19}$$

One major obstacle of this method is the difficulty in solving the eigenvalue problem in Eq. 5.17 for arbitrary geometry and boundary conditions. This would be a particularly difficult task if the method is to be extended to 2D or 3D random fields.

By virtue of the central-limit theorem, both series expansion methods described above are strictly applicable only to Gaussian random fields. Therefore, they are appropriate for second-moment analysis, or for reliability analysis when the random fields are truly Gaussian.

5.3.3.5 Concluding Remarks

Six methods for representing random fields in terms of random variables were described in the preceding subsections. The spatial averaging method and the expansion methods are restricted to Gaussian random fields and, thus, are not appropriate for use in finite-element reliability analysis when the structural property or load random fields are non-Gaussian. The interpolation method with the scheme to reduce the number of random variables is also restricted to Gaussian random fields. The midpoint method and the nodal-point method are employed in the present study because they are numerically stable and are applicable to all types of random fields. Further research is needed in order to develop more efficient ways of representing non-Gaussian random fields.

5.4 Reduction of Basic Random Variables

Experience has shown that the number of iterations needed in the optimization algorithm to converge to the design point is independent of the number of basic variables. However, the required computation of the gradient vector at each iteration step is proportional to the number of basic variables. Hence, it is desirable to reduce the number of basic variables during the optimization phase, particularly if a costly method is used to

compute the gradients. Three sensitivity measures have been used in the literature [16,35,48] for this purpose. Basically, these measures examine the relative influences of the basic variables on the reliability index at each step of iteration. Basic variables found to have small influences in the first few iteration cycles are replaced by deterministic numbers in the subsequent iterations.

Der Kiureghian et al. [16] proposed the use of the sensitivity vector $\gamma(\mathbf{v})$ defined in Eq. 2.26 of Chapter 2 to determine the relative influences of the basic variables on β. It is shown in Ref. 16 that the ith component of $\gamma(\mathbf{v})$ represents the sensitivity of β with respect to a normalized variation in the ith basic variable, V_i. If this component is close to zero, then β is insensitive to the variability in V_i. Hence, V_i can be replaced by a deterministic value without much influence on the final result. The median value or the value at the latest iteration point were suggested as the substitute deterministic value.

An alternative sensitivity measure was introduced by Igusa and Der Kiureghian [35] for reliability analysis of uncertain structures subjected to stochastic excitation. This measure is designed for the case where the limit-state function is defined in terms of the extreme of a response process. This measure is not discussed here since the reliability of structures under stochastic excitation is beyond the scope of this study.

Madsen [48] proposed the use of the negative unit normal $\alpha(\mathbf{y})$ as the sensitivity measure. As mentioned in Chapter 2, the ith component of $\alpha(\mathbf{y})$ is a measure of the sensitivity of β with respect to the ith standard normal variate, Y_i. When the basic variables $\mathbf{V}$ are dependent, there is no one-to-one correspondence between V_i and Y_i. That is, a small α_i does not necessarily imply that β is insensitive to the variability in V_i. Thus, $\alpha(\mathbf{y})$ can only be used to determine which Y_i are unimportant and can be replaced by deterministic values in the subsequent iterations. This measure is suitable only when the gradient is computed in the standard normal space or when $\mathbf{V}$ are independent.

In finite-element reliability analysis, the basic random variables are usually dependent, since often they represent correlated random fields. Furthermore, as already men-

tioned, it is usually imperative to use analytical expressions of the response gradients both for reasons of accuracy and efficiency. In this context, the sensitivity vector proposed by Madsen for omitting the unimportant standard normal variates is not useful, since such omission does not result in reduced computations of the response gradients. (Note that perturbing the standard normal variate Y_i may generally result in perturbing values for all the basic variables.) The sensitivity vector $\gamma(\mathbf{v})$ introduced by Der Kiureghian and Ke [16], on the other hand, is usable in the context of the finite-element reliability method since it provides an importance measure for the original random variables. Thus, omitting each random variable V_i means that the corresponding gradient $\partial g/\partial v_i$ need not be computed in the subsequent analysis.

To determine the optimal deterministic substitute, Madsen [48] introduced a set of omission sensitivity factors. The omission sensitivity factor of a variable is defined as the ratio between the values of the first-order reliability index calculated with and without that variable replaced by a deterministic value. The deterministic substitute for the variable is then chosen such that the omission sensitivity factor is as close to unity as possible. Thus, for a variable Y_i with small $\alpha_{i,1}$ a substitute of $\beta_k \alpha_{i,1}/2$ is recommended at the k-th iteration step, where β_k is updated in each step, and $\alpha_{i,1}$ is the α-value computed at the first iteration.

Despite the fact that the amount of computation can be greatly reduced by omitting random variables, these approaches should be employed with caution. The variables which have little influence on β at the beginning of the iteration may turn out to be important at the final stage. Moreover, if a second-order reliability analysis is to be carried out, the omission of basic random variables is risky because no second-order information is taken into account in the above approaches. Hence, with these methods the computation time may be saved at the cost of accuracy. It is thus advisable to use the results from such analysis only as an initial estimate. If a more accurate assessment of the reliability is needed, the iteration should be continued with all basic variables included until the exact

design point is found.

CHAPTER 6

COMPUTER IMPLEMENTATION

6.1 Introduction

This chapter describes the development of a general-purpose, finite-element reliability code, which is based on the formulation and methods described in the preceding chapters. The code, denoted CALREL-FEAP, is developed by combining a reliability code, CALREL, and a finite-element code, FEAP. Three issues have been of main concern in this development; namely, the efficiency, the modularity, and the generality of the code. Efficiency in both computation time and data storage is essential if the code is to have practical applicability. Thus, special effort has been made to improve the efficiency of the code by using methods such as the adjoint method and the analytical expressions of the gradient. Special attention has also been given to the memory allocation in the code. This is described later in this chapter. The modularity of the code is essential for the continued maintenance and extension of the code. Research in the fields of finite element and structural reliability is still carried out in different groups. Therefore, it is important that the code allows a researcher with expertise in only one of the two fields to modify the program to implement new research results or to test new methods. This is accomplished by developing the code in a modular form and by keeping the finite element and reliability parts of the code, to the extent possible, independent of one another. Finally, the generality of the code is desirable since the ideas behind the finite-element reliability method are applicable to a large variety of problems. Thus, the code is structured in such a manner that a user may develop and implement a new element routine or a new reliability computation method without significantly altering the main code. This will allow the extension and application of CALREL-FEAP in a variety of civil engineering problems. In the following, a detailed description of the constituent parts of the CALREL-FEAP code is presented.

The CAL-Reliability Program (CALREL) is a general-purpose structural reliability program developed by the author, H. Z. Lin, and A. Der Kiureghian. An essential component of this program is the specification of the limit-state functions that define the failure criteria and their gradients with respect to the basic variables. In CALREL, these specifications, which are problem dependent, are provided by the user in two subroutines. In finite-element reliability analysis, the limit-state functions are in terms of load effects, which are themselves implicit functions of the basic variables through the mechanical transformation, Eq. 2.1. Therefore, in such analysis it is necessary to hook up a finite element code through these two user-defined subroutines. The finite element code that has been selected for the present study is the Finite Element Analysis Program (FEAP) developed by R. L. Taylor [78]. This code is selected primarily because of its great flexibility and its ability to handle all types of elements and structural behavior. This code not only meets the requirements for the present study of geometrically nonlinear structures, but provides opportunity for future research on other classes of problems, including the reliability of structures with material nonlinearity and structures under dynamic loads.

Originally, CALREL and FEAP were two independent programs. In order to be connected to CALREL, FEAP has been modified into a subroutine. The challenge here has been to make the connection such that to the extent possible the two codes are left undisturbed. This is important since these two programs have been developed and will be maintained by two different groups of researchers. With the structures of the two programs basically unchanged after the connection, an expert on reliability can simply implement a new reliability routine in CALREL without knowing FEAP, and an expert on finite elements wanting to perform reliability analysis for a new class of structural problems can modify or extend FEAP to provide the necessary mechanical transformation without altering CALREL.

The layout of the combined CALREL-FEAP code is shown in Fig. 6.1. The two programs basically have no overlap. CALREL does not call the subroutines in FEAP, and

FEAP does not call those in CALREL. Their interface, the user-defined routines, is problem dependent and is not affected by the changes in either CALREL or FEAP. Thus, CALREL-FEAP is highly flexible and easy to update with such an arrangement. The details of these two programs and their linkage are described in the following sections.

6.2 The CAL-Reliability Program (CALREL)

CALREL has been developed on a virtual memory computer, MicroVax. It has also been transferred to operate on PC compatible systems for educational purposes. CALREL is composed of four modules:

1. Program control;
2. Problem definition;
3. Reliability analysis routines;
4. User-defined routines.

The program control module reads and interprets macro commands in the input file and executes the associated analysis routines. The problem definition routine reads the problem data and the parameters used in the solution algorithms. The user-defined routines define the limit-state functions, their gradients, and user-provided probability distributions, if any.

CALREL contains a set of analysis routines which perform the following:

1. First-order reliability analysis (FORM) for a component;
2. Second-order reliability analysis (SORM) for a component;
3. Estimation of reliability bounds for a series system;
4. Estimation of reliability for a series system by the PNET method;
5. Directional simulation for a general system;
6. Monte Carlo simulation for a general system;
7. First-order sensitivity analysis.

The basic difference between the component and system reliability analysis is in the number of limit-state functions used to define the failure set. Component reliability analysis requires a single limit-state function (although several such functions can be analyzed with a single run), whereas system reliability requires a multitude of such functions. The execution of the various analyses is controlled by a sequence of macro commands issued in the input file. It is seen that CALREL has capabilities beyond those described in Chapter 2. Namely, it is capable of performing system reliability analysis and estimation of probability by the two simulation methods. All these capabilities are applicable for the finite-element reliability analysis. However, the computation time in certain applications, such as those using the Monte Carlo simulation method, can be prohibitively long.

The input file of CALREL is composed of several macro command sections. The macro commands available in the current version are:

CALREL	BOUND	DATA	DIRS	END	EXIT
FORM	MONT	PNET	RESTART	SENS	SORM

The input file always starts with the commands CALREL or RESTART and terminates by the command EXIT. Most macro commands are followed by optional modifiers to indicate how that command is to be carried out. For example, if the following command is issued,

```
FORM NPR=3
```

CALREL will perform a first-order reliability analysis and print results at every third iteration step of the optimization algorithm. With the exception of CALREL, RESTART, and EXIT, the commands can be repeated and their order in the input file is flexible. However, some commands are executable only after other analyses are completed. For instance, a second-order analysis is possible only after the results of a first-order analysis are obtained. Hence, the command SORM can be issued only after a command FORM.

The macro command DATA indicates the beginning of the data-input mode. The input data of CALREL consist of several data sections. Each section starts with a title describing the type of data to follow. The types of data sections included in the current version are:

TITLE	CUTSETS	FLAG	OPTIMIZATION
PARAMETER	STATISTICS	LIMIT	

There are default values for most of the data sections, except STATISTICS which reads the statistics of the basic random variables and must be provided at least once for each problem. The command END terminates the data-input mode and revokes the macro command mode.

The formats and definitions of the macro commands and the input data are given in detail in Appendix B. An example input file is also included in this appendix.

Three problem dependent and, therefore, user-defined subroutines are necessary for running CALREL. The first routine, UGFUN, defines the limit-state functions. The second routine, UDGX, defines the gradients of the limit-state functions if analytical expressions are to be used. CALREL provides an option for computing the gradients by the finite difference scheme. In that case, UDGX is left as a dummy subroutine. The third routine, UDD, is used to define the probability distributions that are not available in the CALREL distribution library. This routine can also be dummy if no user-provided distributions are necessary. The formats of these three subroutines are given in Appendix B.

In finite-element reliability analysis, the limit-state functions in UGFUN are expressed in terms of load effects. The dependence of the load effects on the basic variables is provided through the finite element code. Thus, appropriate call statements to the finite element code (here FEAP) are included in UGFUN in order to compute the limit-state functions. If the gradients are to be computed by a finite difference scheme, no other connection to the finite element code is necessary. However, if the gradients are to be

computed using analytical expressions, such as those developed in Chapter 4 for geometrically nonlinear structures, then proper call statements to the routines that extract the necessary arrays from the finite element analysis must be included in the subroutine UDGX. Thus, the subroutines UGFUN and UDGX are the only routines that connect to the finite element code. The details of the subroutines UGFUN and UDGX for finite element reliability analysis with FEAP are described in Appendix B.

6.3 The Finite Element Analysis Program (FEAP)

FEAP is a general-purpose finite element analysis program. It is designed such that one can easily make modifications in it to meet the requirements of different analysis schemes or problem areas. FEAP consists of five general modules:

1. Problem control;
2. Problem definition and mesh input;
3. Problem solution;
4. Element library;
5. Graphic output.

The problem control module reads the problem size and the input/output file names. The problem definition module reads the input data and a set of macro commands to specify the solution algorithm. The problem solution module executes the solution steps according to the macro commands specified in the input file. The graphic output module displays the mesh or the results of the analysis on the screen.

FEAP has an element library. The library elements in FEAP have been developed to solve large classes of problems in structural mechanics. If these elements cannot be used to model a particular problem, then the user can add an appropriate element to the library by writing a subroutine which computes the necessary quantities for the element, such as the stiffness matrix, the stresses and strains, etc..

FEAP can be executed both in interactive and batch modes. To run FEAP, the user

has to prepare an input file which contains the input data and the macro commands. Details on the macro commands and the format of the input file can be found in the user's manual of FEAP in Ref. 78.

To combine FEAP and CALREL, the following modifications have been made in FEAP:

1. FEAP has been converted from an independent program into a subroutine;
2. FEAP has been made to operate only in a batch mode;
3. The graphic output option has been suppressed;
4. Instead of being specified by the user, the input/output file names have been set to be feap.inp and feap.out;
5. A new macro command GRAD has been added;
6. The element subroutine has been extended to allow the formation of the matrices required in the gradient computation;
7. The input data have been extended to allow the specification of external loads, material properties, and structural geometry as random variables.

Modifications 2 and 3 have been made because the subprogram FEAP is executed repeatedly and automatically during the FORM and SORM operations and, hence, graphic output and interactive operations are not appropriate. A graphic capability at the end of the reliability analysis, however, is essential for a careful evaluation of the results and is planned in the future development of CALREL-FEAP. The new command GRAD has been added to compute the gradients of structural responses which are required in FORM. The element subroutine has been modified to form the matrices needed in the gradient computation. Details of the element subroutine are available in Appendix B.

The random nature of the external loads, material properties, and structural geometry is specified by the user in the input data file of the subprogram FEAP. The subprogram FEAP differentiates random quantities from deterministic quantities merely by the type of

the data entry. If the data is a real number, the entry is considered a deterministic quantity, and if the data is an integer number, the quantity is considered random and represented by the basic random variable identified by the integer number.

The input file of the subprogram FEAP consists of two parts. The first part is composed of data sections which define the control information and the mesh of the problem. The types of data sections included in the current version are:

FEAP	ANGL	BLOC	BOUN	COOR	CONS	EBOU
ELEM	FORC	LINK	MATE	NOPR	PAGE	POLA
PRIN	SLOA	SPHE	TEMP	TIE		

Each section has a specific function in defining the problem to be solved. The first line of the data section is always FEAP which specifies the problem size. Among the remaining sections, BLOC generates mesh blocks, COOR reads and generates nodal coordinates, FORC reads and generates nodal forces, SLOA reads boundary tractions, and MATE reads material properties. Since the nodal coordinates, nodal forces, surface loads, and material properties may be random, these sections have been modified to allow specification of random quantities. The formats of these sections remain unchanged, except that the entries corresponding to the nodal coordinates, loads, and material properties may be deterministic or random. As mentioned earlier, if the entry is real, the coordinate, load, or material property is equal to the entered value, and if the entry is an integer k, the coordinate, load, or material property is assigned the kth random variable. Additional information on the data sections are available in Ref. 78.

The second part of the input file for the subprogram FEAP contains a list of macro commands defining the solution algorithm. Currently available macro commands in CALREL-FEAP are:

MACRO	BETA	CHEC	CMAS	DATA	DISP	DT
END	FORM	IDEN	INIT	LMAS	LOOP	MESH
NEWF	NEXT	NOPR	PRIN	PROP	REAC	SOLV

STRE SUBS TANG TIME TOL UTAN GRAD

The above list does not include several commands of the original FEAP, which are related to the graphic and interactive modes and have been suppressed. This command section must start with the command MACRO and terminate with END.

The command STRE in the original version of FEAP has the format

STRE,node,n1,n2,n3

where [node,n1,n2,n3] are optional modifiers. This command instructs FEAP to compute the components of stress and strain in elements n1 to n2 at an increment of n3. If the modifier 'node' is given in a 2D problem, the stresses and strains are computed at nodal points instead of the selected stress/strain output points (usually the Gaussian points). In finite-element reliability analysis with CALREL-FEAP, the numbers of the elements whose stresses/strains are used in the limit-state functions are specified in the input file of CAL-REL. Therefore, there is no need to reenter the element numbers in the STRE command line. For this reason, the format of the command STRE has been shortened to:

STRE

The functions and formats of the other macro commands of FEAP are described in detail in Ref. 78.

6.4 Linkage of CALREL and FEAP

To link FEAP to CALREL, many modifications had to be made in both programs, especially in FEAP. Section 6.3 describes the changes made in the features of FEAP. The following is a list of the changes that have been made in the two codes to make the linkage possible.

1. The subroutines to allocate arrays in CALREL and FEAP have been modified to allow these two programs to share the same blank common block;
2. Several new common blocks have been created and several old common blocks have been modified to transfer data between CALREL and FEAP;

3. New file numbers have been assigned in FEAP to prevent data files from mixing up;
4. Several new arrays have been created in CALREL and FEAP to store the data needed in finite-element reliability analysis, e.g., the response gradient;
5. Several new routines have been created or modified in FEAP to incorporate the randomness in the structure and its environment, to compute the gradient, and to facilitate the user in formulating the user-defined subroutines UGFUN and UDGX;

Details of these modifications are described Appendix B.

6.5 Summary

A general-purpose computer code CALREL-FEAP has been developed to perform finite-element reliability analysis. In this code, the finite element program FEAP has been linked to the general-purpose reliability program CALREL to provide the required mechanical transformation. The connection between the two programs is through the user-defined subroutines UGFUN and UDGX. To use this program, the user should prepare the following:

1. An input file for CALREL;
2. An input file for FEAP;
3. An element routine in FEAP to compute element quantities, if not available in the existing library;
4. Three subroutines UGFUN, UDGX, and UDD to compute the limit-state functions, the gradients of the limit-state functions, and the user-provided distributions, if any.

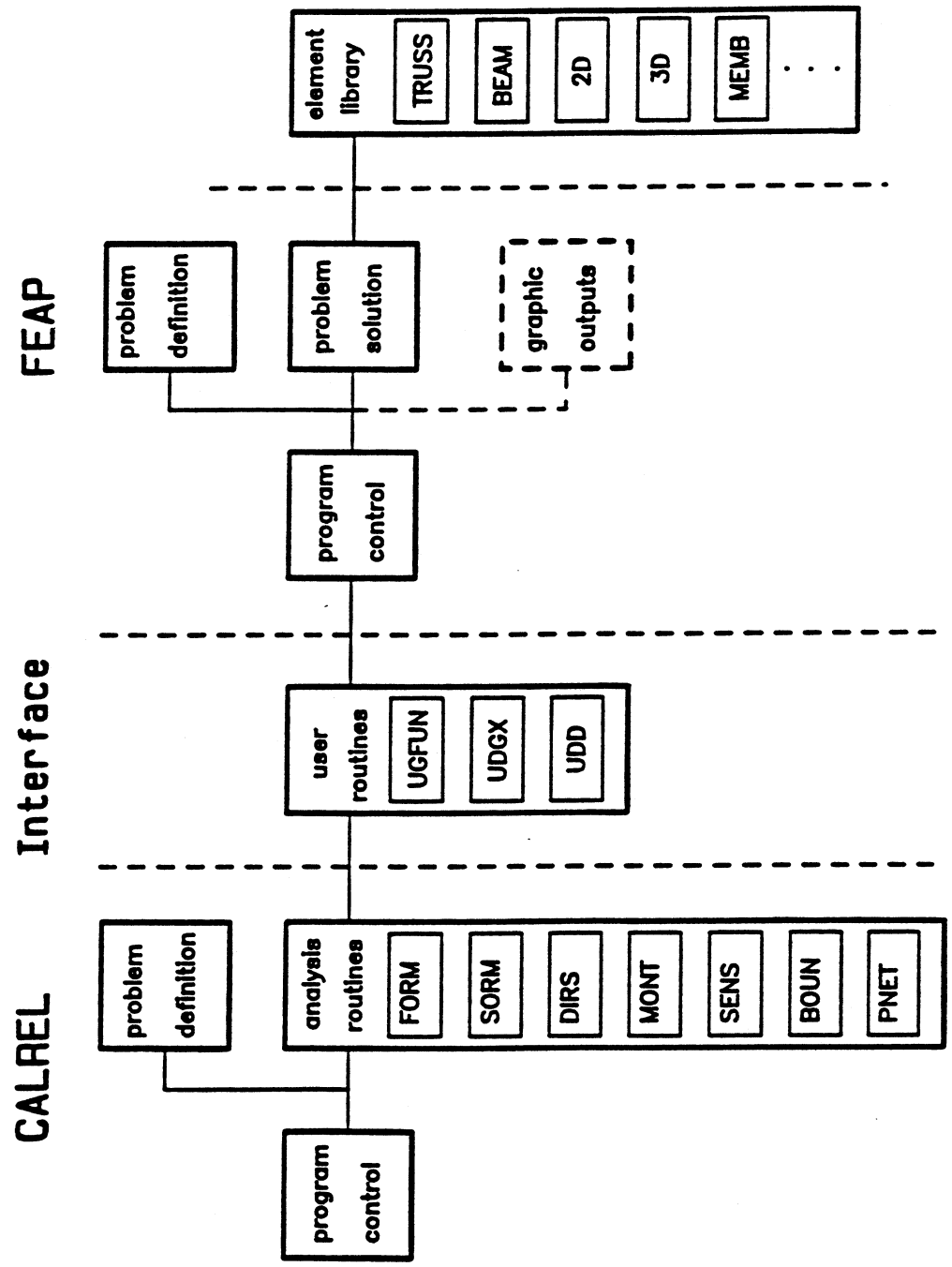


Figure 6.1 The Structure of CALREL-FEAP

CHAPTER 7

APPLICATION EXAMPLES

7.1 Introduction

Two application examples are presented here to illustrate the reliability methods described in the previous chapters. The first example examines the reliability of a built-up column. A displacement failure criterion is chosen to investigate the stability of the column. The second example examines the reliability of a plate with a random hole with two failure criteria. One criterion is in terms of the principal Cauchy stress at the stress concentration point, and the other is in terms of the maximum displacement along the plate edge. The union of the two failure events, representing system reliability, is also considered. These problems were solved using the finite-element reliability program CALREL-FEAP running on a MicroVax II/GPX workstation. The behaviors of the column and the plate are investigated using the results of FORM, SORM, and the first-order sensitivity analysis with respect to the distribution parameters and the deterministic parameters in the limit-state functions.

7.2 Example 7.1 -- Reliability of a Built-Up Column

The reliability of the two-dimensional, built-up column in Fig. 7.1 is studied. The column is composed of truss members with the strain energy function $E_{ss} D E_{ss}/2$, where D is the elastic modulus and $E_{ss} = (l^2 - L^2)/(2L^2)$ is the Green-Lagrangian strain measure, in which l and L are the current and original lengths of the members. Note that this formulation accounts for the effect of geometrical nonlinearity. The struts have a cross-sectional area of $1.59\,in^2$ and a random elastic modulus D_1, and the braces and battens have a cross-sectional area of $0.938\,in^2$ and a random elastic modulus D_2. D_1 and D_2 are assumed to be jointly lognormal with means $\mu_{D_1} = \mu_{D_2} = 30{,}000\ ksi$, coefficients of variation $\delta_{D_1} = \delta_{D_2} = 0.08$, and a correlation coefficient $\rho_{D_1 D_2} = 0.3$. The X_2-coordinates of the nodes are deterministic, but the X_1-coordinates are considered random to model column imperfection.

The X_1-coordinates are assumed to be independent normals with means $\mu_{X_{1i}} = 60$ or -60 *in* and standard deviations $\sigma_{X_{1i}} = 1.2\,in$. The buckling load for this column using the mean properties is determined to be 2,593 *kips*. The column is subjected to a lateral force F_1 at the midspan and an axial force F_2, which are independent random variables with log-normal distributions. F_1 has a mean $\mu_{F_1} = 20$ *kips* and a standard deviation $\sigma_{F_1} = 2$ *kips*, and F_2 has a variable mean μ_{F_2} and a coefficient of variation $\delta_{F_2} = 0.1$. Thus, 22 basic random variables are included in this problem. The column is considered failed if its horizontal displacement at the midspan exceeds the threshold u_0. Thus, the limit-state function is

$$g(\mathbf{v}) = u_0 - u_{10}(\mathbf{v}) \tag{7.1}$$

where $u_{10}(\mathbf{v})$ denotes the displacement at node 10 in the X_1 direction.

FORM and SORM were performed for $\mu_{F_2} = 500$ to 2,500 *kips* and $u_0 = 0$ to 30 *in*. The analytical expressions derived in Chapter 4 together with the adjoint method described in Chapter 5 were used to compute the gradient of the limit-state function. The matrices involved in the gradient computation are derived in Appendix A. Figure 7.2 shows plots of the first- and second-order failure probability estimates versus μ_{F_2} and u_0. For all the cases studied, FORM and SORM are in close agreement. This indicates that the limit-state surface around the design point is nearly flat, in spite of the nonlinearity of the structure. It is interesting to note in Fig. 7.2a that the failure probability curves for different allowable midspan displacements cluster together as the mean of the axial load increases, and they all approach probability one as μ_{F_2} approaches the buckling load of the mean column. This insensitivity of the failure probability to u_0 for large values of μ_{F_2} is an indication of instability of the column. The same phenomenon is observed in Fig. 7.2b, where the curves tend to become horizontal for increasing values of μ_{F_2}.

For each of the cases studied, the coordinates $\mathbf{v}^*$ and $\mathbf{s}^*$ of the design point define the most likely values of the basic variables and load effects in the failure set. (For non-

normal basic variables, these coordinates actually are not exactly at the most likely failure point, but the difference is usually negligible.) As an example, Table 7.1 lists the most likely values of the basic variables and Fig. 7.3 shows the corresponding undeformed and deformed configurations of the column for the failure event with $\mu_{F_2} = 1{,}500\,kips$ and $u_0 = 30\,in$. As one would expect, the most likely values of D_1 and D_2 are below their means, F_1 and F_2 are above their means, and the most likely undeformed configuration has a sine-shape imperfection on the compression side of the column.

As described in Section 2.5, a useful result in FORM is the sensitivities of the reliability index β with respect to distribution parameters (see Eq. 2.27). As an example, the sensitivities of the reliability index with respect to the means of the basic variables are examined here for 3 cases: (1) $\mu_{F_2} = 500\,kips$, $u_0 = 30\,in$, (2) $\mu_{F_2} = 1{,}500\,kips$, $u_0 = 30\,in$; and (3) $\mu_{F_2} = 1{,}500\,kips$, $u_0 = 5\,in$. The first two cases examine the reliability of the column against a large midspan deflection. In a sense, these cases represent failure of the column due to instability. The third case examines the reliability of the column against a relatively small midspan deflection, which may result with or without loss of stability. The sensitivities are scaled by the corresponding standard deviations of the basic variables to make them dimensionless. This also has the beneficial effect of making parameter variations equally likely in a statistical sense. These scaled measures can be directly compared to determine the relative importance of each basic variable. The results are shown in Fig. 7.4. In all cases the reliability index is strongly sensitive to the mean of the elastic modulus of the struts. This is because the softening of the structural stiffness basically comes from the P-δ effect of the struts, and the axial forces of the struts are proportional to D_1. Comparing Figs. 7.4a and 7.4b, it is seen that as the mean of F_2 increases by a factor of three, the sensitivity to the mean increases more than sevenfold. This is partly due to the increase in the standard deviation of F_2, since the coefficient of variation is kept fixed at $\delta_{F_2} = 0.1$, and partly because for the geometrically nonlinear column the lateral displacement is increasingly sensitive to larger values of the axial load. Comparing Figs. 7.4b

and 7.4c, the sensitivities to the column imperfection and the lateral forces are seen to decrease as the displacement threshold u_0 is increased. This is because a larger u_0 implies a state closer to the instability state, for which the magnitudes of the column imperfection and the lateral load do not matter.

As described in Section 2.5 and Eq. 2.28, FORM also provides the sensitivity of the reliability index with respect to any parameter in the limit-state function. In particular, observe that for the present example, the partial derivative $-\partial P_f/\partial u_0$ is identical to the PDF of the midspan deflection. For $\mu_{F_3} = 1500\,kips$, this PDF is computed based on FORM sensitivity analysis and is shown in Fig. 7.5.

7.3 Example 7.2 -- Reliability of a Stochastic Plate with a Random Hole

The reliability of the $32\times32\,in^2$ square plate shown in Fig. 7.6 which has a hole of random geometry is investigated. The plate is subjected to a uniformly distributed tensile load of random magnitude p at the two opposite edges. The magnitude p has a lognormal distribution with a mean of $100\,lb/in$ and standard deviation $20\,lb/in$. The coordinates of the hole boundary are assumed to be independent normals with mean values defining a circular hole of radius $2\,in$ and coefficients of variation equal to 0.04. The material of the plate is elastic with a quadratic strain energy function $\mathbf{EDE}/2$, where $\mathbf{E}$ is the Green-Lagrangian strain tensor, and $\mathbf{D}$ is the elasticity matrix defined in terms of Young's modulus E and Poisson's ratio ν. E and ν are modeled as independent, homogeneous random fields. The Young's modulus is modeled as a lognormal random field with a mean $\mu_E = 500\,psi$ and a coefficient of variation $\delta_E = 0.15$, and the Poisson's ratio is modeled as a uniform random field bounded between 0.2 and 0.4 (see section 5.3.1 and Ref. 14 for the definitions of these random fields). The autocorrelation coefficient functions of E and ν in the transformed normal space are assumed to have the isotropic forms

$$\rho_{Z_E Z_E}(\Delta X_1, \Delta X_2) = \exp\left[-\frac{\Delta X_1^2 + \Delta X_2^2}{(a_E L)^2}\right] \tag{7.2}$$

$$\rho_{Z_\nu Z_\nu}(\Delta X_1, \Delta X_2) = \exp\left[-\frac{\Delta X_1^2 + \Delta X_2^2}{(a_\nu L)^2}\right] \tag{7.3}$$

where $Z_E = \Phi^{-1}[F_E(E(\mathbf{X}))]$ and $Z_\nu = \Phi^{-1}[F_\nu(\nu(\mathbf{X}))]$ are the transformed random fields, $\Delta X_1^2 + \Delta X_2^2$ is the square of the distance between two points on the plate, $L = 32\,in$ is the side dimension of the plate, and a_E and a_ν are dimensionless measures of the correlation lengths of the two random fields.

Because the geometry and material properties of the plate are random, symmetry in general does not exist. However, for the sake of simplicity, symmetry properties are assumed and only a quarter of the plate is analyzed. The quarter plate is discretized into 144 finite elements and 30 random field elements for both the E and ν random fields, as shown in Fig. 7.7. After the discretizations, 85 random variables are considered, including 1 for the tensile load intensity, 24 for the coordinates of the nodes on the hole boundary, and 30 for each of the discretized random fields of E and ν. The referential formulation is used to compute the plate response including the effect of geometrical nonlinearity.

Two failure criteria are considered in this problem. The first criterion is the exceedance of the tensile principal Cauchy stress at element 133 above a threshold of 600 *psi*. The second is the exceedance of the horizontal displacement at node 13 above a threshold of 4 *in*. These criteria are formulated as limit-state functions

$$g_1(\tilde{T}^*(\mathbf{v})) = 600 - \tilde{T}^*(\mathbf{v}) \tag{7.4}$$

$$g_2(u_{13}(\mathbf{v})) = 4 - u_{13}(\mathbf{v}) \tag{7.5}$$

where $\tilde{T}^*(\mathbf{v})$ is the tensile principal Cauchy stress at the centroid of element 133, and $u_{13}(\mathbf{v})$ is the displacement of node 13 in the X_1 direction.

The problem was first analyzed by FORM and SORM with $a_E = 0.125$ and $a_\nu = 0.125$. In FORM, both the finite difference method and the analytical method described in Chapters 4 and 5 were used to compute the gradients of the limit-state functions. The essential matrices required for the analytical method are derived in Appendix A. The modified HL-RF method described in Chapter 3 with the initial point at the mean values of the basic variables was used to find the design point. In SORM, a point-fitted paraboloid together with Tvedt's double integral formula in Eq. 2.16 were used. The

number of iterations and the required CPU time for these solutions are listed in Table 7.2. It is clear that the analytical approach for computing the gradients provides a significant advantage. It is also clear that the SORM approximation requires considerable amount of additional computations beyond that required for FORM. Table 7.3 shows the estimates of the failure probability P_f and the generalized reliability index, defined here as $-\Phi^{-1}(P_f)$, based on FORM and SORM for each limit-state function. This table also lists the estimates of P_f and β for the union of the two failure events based on FORM system analysis [22] and directional simulation using second order approximation surfaces [37]. It is seen that the FORM and SORM estimates are in close agreement for both limit-state functions. Further investigations reveal that the curvatures of the fitted paraboloids at the design points are very small for both limit-state surfaces. This indicates that the two limit-state surfaces are nearly flat around their design points in spite of the nonlinearity of the mechanical and probability transformations. Hence, FORM provides good approximation for the present example. This method is used in the subsequent analysis of the plate.

Figures 7.8 and 7.9 show the contours of the local Young's modulus E^* and the local Poisson's ratio ν^* at the design point of the two limit-state surfaces. The shaded area in Fig. 7.8 indicates the region where the Young's modulus is above the mean value of E. Figure 7.10 shows the configuration of the hole at the design point of the stress limit state. The hole shape at the design point of the displacement limit state is not shown since it almost coincides with the mean shape.

Failure criteria in structural reliability evaluation are usually specified based on theoretical analysis and engineering judgement. For the present example, the two limit states in Eqs. 7.4 and 7.5 are selected based on our experience with linear elastic plates with circular holes, for which the location of stress concentration and maximum displacement under uniform tensile loads are well known. The coordinates of the design point obtained from FORM analysis provide an excellent means for examining the appropriateness of the selected limit states. For the present example, the contours of the tensile princi-

pal Cauchy stress and the undeformed and deformed shapes of the plate at the design point of the two limit-state surfaces are plotted in Figs. 7.11 and 7.12. These figures confirm that the maximum stress concentration indeed occurs in element 133 and that the maximum displacement indeed occurs at node 13 along the X_1 coordinate. These observations reassure the appropriateness of the selected limit-state functions. Had the stress contours indicated a different stress concentration point or had the deformed shapes indicated a different maximum displacement point, then one would have to conclude that the selected limit-state functions are not appropriate and that more critical failure conditions may exist. Clearly, such analysis would be useful for complex or nonlinear structures for which the behavior and the modes of failure cannot be easily prescribed.

Table 7.4 lists the scaled sensitivities of β with respect to the means and standard deviations of the load variable p, the coordinates of the nodes on the hole boundary, and the random fields E and ν. The sensitivities with respect to the random fields are obtained by summing up the sensitivities of the individual random field elements. In each case, the sensitivity is scaled by the standard deviation of the corresponding basic variable or random field. As mentioned earlier, these dimensionless sensitivity measures are directly comparable. It is seen in Table 7.4 that for the stress limit state, β is most sensitive to the mean and standard deviation of p. This is partly due to the large coefficient of variation of the load among the basic variables. However, a more fundamental reason is that while stresses are always directly related to loads, their dependence on material properties is indirect and only a result of statical indeterminacy. This relative insensitivity of stress limit-states to variabilities in material properties has been observed to an even greater extent for linear structures [15]. For the displacement limit state, highest sensitivities are to the mean of the Young's modulus random field and the standard deviation of the load. This is due to the strong dependence of the deformation of the plate to these two variables. This result is also consistent with previous results for linear structures [15].

It is clear from Table. 7.4 that variabilities in the geometry of the hole have no effect

on the reliability with respect to the displacement limit state. This is because the displacement at node 13 is a global response measure which is not affected by the detail of the hole geometry. On the other hand, the reliability with respect to the stress limit state is relatively sensitive to the geometry of the hole. Obviously, this is due to the localized nature of the stress limit state. For this limit state, the scaled sensitivities of β with respect to the means and standard deviations of the nodal coordinates around the hole are shown in Fig. 7.13. The vertical (horizontal) distance between each nodal point and the sensitivity curve indicates the sensitivity of β with respect to the mean or standard deviation of X_2 (X_1) coordinate of the node. The sensitivity measure is positive if the curve is drawn above (to the right of) the nodal point, and it is negative if the curve is drawn in the opposite side. It is seen that β is insensitive to the variations in the X_1 coordinates of the nodes, but it is sensitive to the variations in the X_2 coordinates of the nodes near the stress concentration point. Figure 7.13a shows that the reliability will decrease if the hole is wider in the X_2 direction near the stress concentration point and rapidly narrows away from that point. Figure 7.13b shows that increased uncertainty in the X_2 coordinates of the hole, particularly near the stress concentration point, will decrease the reliability. All these results are consistent with the theoretical results available for the linear elastic case.

Figure 7.14 shows the contour plots of the scaled sensitivities of β with respect to the mean of the local Young's modulus. In Figure 7.14a, the shaded band indicates the region where an increase in the mean of the local E will decrease the plate reliability for the stress limit state. This band is the main path through which the tensile load at the edge is transferred to element 133. If the band is stiffer than the remaining plate, more tensile load is distributed to this region, and element 133 takes more stress than it does when the plate is uniform. The stress contours of the design point in Fig. 7.11 and the sensitivity contours in Fig. 7.14a offer two different ways to improve the reliability of the plate with respect to the stress limit state. One is to mount a high-strength ring on the hole to keep the plate from cracking at the point of stress concentration, as is usually recommanded in classical mechanics. The other is to stiffen the region with high positive sensitivity

measures or soften the shaded band.

It is seen in Figure 7.14b that for the displacement limit state the sensitivity of β with respect to the mean of the local E is positive everywhere in the plate. Thus, the stiffening of any part of the plate will increase the reliability with respect to the displacement limit state. There are, of course, differences in the degree to which the stiffness of each element influences the displacement. It is clear from Fig. 7.14b that β will increase most if the mean of E in the region adjacent to node 13 is increased.

Figure 7.15 shows the contours of the scaled sensitivity of β with respect to the mean of the local Poisson's ratio. Unlike Young's modulus, little is known about how the variation of Poisson's ratio in a plate will influence the plate behavior, especially when geometrical nonlinearity is considered. The shaded areas in Figs. 7.15a and 7.15b indicate the regions where the sensitivity measures are negative. Thus, the reliability of the plate will increase if the Poisson's ratio in these regions increase.

Comparing Figs. 7.8 and 7.14, it is seen that the patterns of the contours of E^* and $\sigma_E \frac{\partial \beta}{\partial \mu_E}$ are similar for both limit states. This is because the plate is in the safe state at the mean point, and to shift the plate from the safe state to the limit state, the values of the discretized variables will either increase or decrease to make failure possible. The direction of change for a local E happens to be quite consistent with the sign of its corresponding $\sigma_E \frac{\partial \beta}{\partial \mu_E}$. This is because if an increase in the mean of a local E tends to decrease the plate reliability, for the local E to take a value higher than its mean tends to weaken the plate. For the same reason, the contours of ν^* and $\sigma_\nu \frac{\partial \beta}{\partial \mu_\nu}$ in Figs. 7.9 and 7.15 have similar patterns. The same phenomenon is observed when comparing the hole shape at the design point in Fig. 7.10 and the sensitivities of β with respect to the nodal coordinates of the hole in Fig 7.13 for the stress limit state.

Reliability sensitivity analyses are also performed with respect to the local standard deviation of the E and ν random fields. The contours of these sensitivity measures are

shown in Figs. 7.16 and 7.17. All these contour plots show the same trend: The sensitivity is negative everywhere, indicating less reliability for increasing uncertainty; also the plate reliability is most sensitive to the uncertainties in E and ν in the region where the stress or displacement is investigated.

As a final item of interest, the influences of the correlation length measures a_E and a_ν on the plate reliability are investigated. Figure 7.18a shows the variation of β with respect to a_E with $a_\nu = 0.125$, and Fig. 7.18b show the variation of β with respect ot a_ν with $a_E = 0.125$. As a_E increases from 0.1 to 0.5, β in Fig. 7.18a increases gradually from 2.64 to 2.70 for the stress limit state and drops from 2.45 to 2.13 for the displacement limit state. The comparatively low sensitivity of β to a_E for the stress limit state is due to the fact that the stress concentration is a localized phenomenon and is not overly sensitive to the spatial variations of E. As E becomes more uniform in the plate, i.e. as a_E grows, the plate becomes more reliable for the stress limit state and less reliable for the displacement limit state. The variation of the system reliability with respect to a_E is also shown in this figure. Since both failure criteria are included in the system reliability analysis, the corresponding index is always smaller than those of the individual limit states. Figure 7.18b shows that the reliability indices for the two limit states and the system remain nearly constant as the correlation length measure a_ν is varied. This implies that the plate reliability is insensitive to the spatial variability of Poisson's ratio.

This example illustrates that the finite-element reliability analysis not only provides estimates of the failure probabilities of structures, but also provides information which may help engineers gain insight into the behaviors of complex structures. The sensitivity measures are useful in identifying the relative significance of various parameters and sources of uncertainty affecting the reliability of the structure. They also provide indications for modifying a design in order to improve its reliability.

Table 7.1 Design Point of the Built-Up Column for $\mu_{F_2} = 1500 kips$ and $u_0 = 30 in$.

Basic Variables	Mean Value	Design Point
x_2	−60.00	−60.06
x_3	60.00	59.95
x_4	−60.00	−59.88
x_5	60.00	60.07
x_6	−60.00	−59.80
x_7	60.00	60.02
x_8	−60.00	−59.71
x_9	60.00	60.00
x_{10}	−60.00	−59.65
x_{11}	60.00	59.98
x_{12}	−60.00	−59.71
x_{13}	60.00	60.00
x_{14}	−60.00	−59.80
x_{15}	60.00	60.02
x_{16}	−60.00	−59.88
x_{17}	60.00	60.07
x_{18}	−60.00	−60.06
x_{19}	60.00	59.95
D_1	30000	24750
D_2	30000	27740
F_1	20.00	20.32
F_2	1500	2005

Note: Units are *in* for x_i, $kips/in^2$ for D_i, and *kips* for F_i.

Table 7.2 Required Number of Iterations and CPU Time of Example 7.2

Limit-State Function	FORM			SORM
	number of iterations	gradient computed by		
		analytical approach	finite difference	
g_1	8	21 min	2358 min	1186 min
g_2	4	10 min	1178 min	452 min

Table 7.3 FORM/SORM Solutions for Example 7.2

Criterion	FORM		SORM	
	β	P_{f1}	β	P_{f2}
stress limit-state	2.6371	4.181×10^{-3}	2.6570	3.942×10^{-3}
displacement limit-state	2.4191	7.780×10^{-3}	2.4178	7.808×10^{-3}
system reliability	2.341	9.615×10^{-3}	2.354	9.295×10^{-3}

Table 7.4 Sensitivity of β with Respect to Distribution Parameters

Variable or Field	Stress Limit-State		Displacement Limit-State	
	$\sigma \frac{\partial \beta}{\partial \mu}$	$\sigma \frac{\partial \beta}{\partial \sigma}$	$\sigma \frac{\partial \beta}{\partial \mu}$	$\sigma \frac{\partial \beta}{\partial \sigma}$
p	−0.5249	−1.9179	−0.5667	−1.9159
x_1	−0.0002	−0.0000	−0.0002	−0.0000
x_{14}	−0.0004	−0.0000	−0.0003	−0.0000
y_{14}	−0.0000	−0.0000	−0.0000	−0.0000
x_{27}	−0.0002	−0.0000	−0.0001	−0.0000
y_{27}	−0.0000	−0.0000	−0.0000	−0.0000
x_{40}	0.0004	−0.0000	−0.0000	−0.0000
y_{40}	0.0001	−0.0000	−0.0000	−0.0000
x_{53}	0.0015	−0.0000	−0.0001	−0.0000
y_{53}	0.0005	−0.0000	−0.0000	−0.0000
x_{66}	0.0029	−0.0000	−0.0003	−0.0000
y_{66}	0.0018	−0.0000	−0.0001	−0.0000
x_{79}	0.0042	−0.0000	−0.0003	−0.0000
y_{79}	0.0045	−0.0001	−0.0003	−0.0000
x_{92}	0.0054	−0.0001	−0.0004	−0.0000
y_{92}	0.0103	−0.0003	−0.0006	−0.0000
x_{105}	0.0061	−0.0001	−0.0003	−0.0000
y_{105}	0.0224	−0.0013	−0.0010	−0.0000
x_{118}	0.0058	−0.0001	−0.0002	−0.0000
y_{118}	0.0440	−0.0051	−0.0015	−0.0000
x_{131}	0.0040	−0.0000	−0.0001	−0.0000
y_{131}	0.0869	−0.0199	−0.0019	−0.0000
x_{144}	−0.0046	−0.0001	−0.0000	−0.0000
y_{144}	−0.1132	−0.0338	−0.0022	−0.0000
y_{157}	−0.2257	−0.1343	−0.0012	−0.0000
E	0.1540	−0.2630	0.7694	−0.3782
ν	−0.1100	−0.0614	−0.0014	−0.0021

Note: x_I and y_I denote the X_1 and X_2 coordinates of node I, respectively.

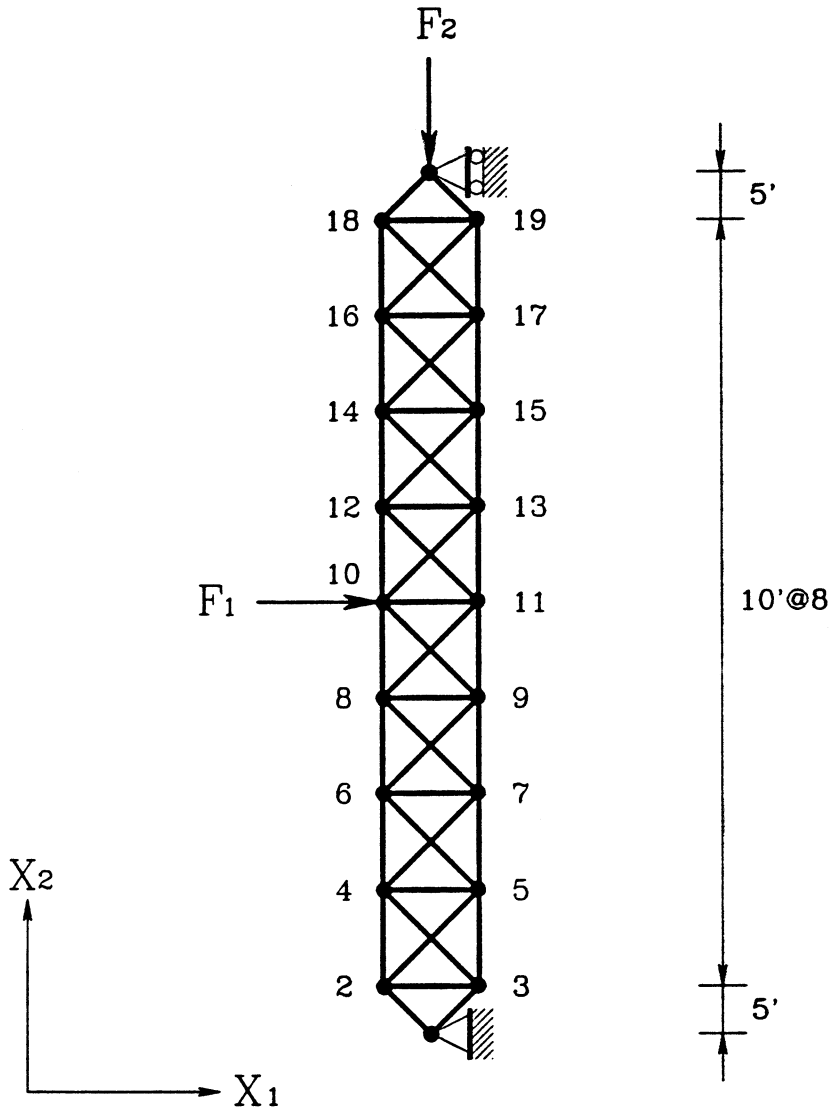
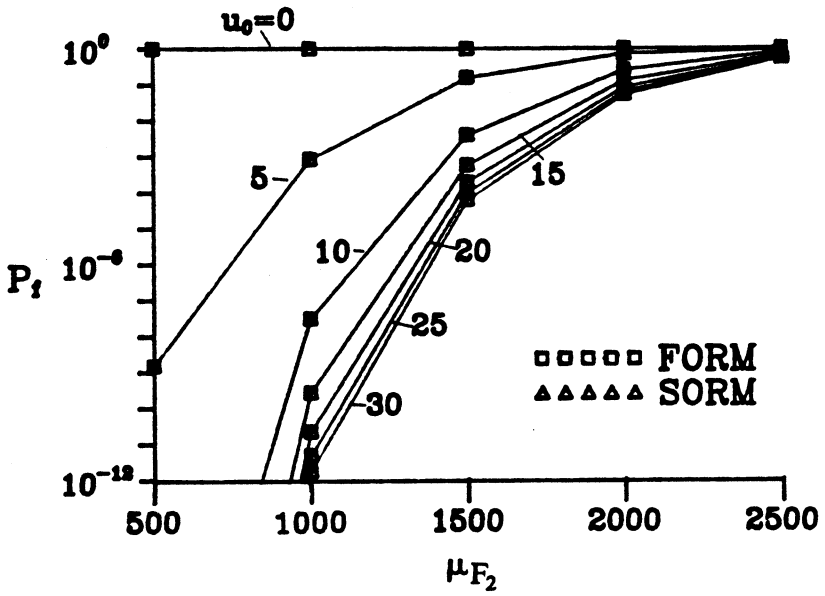
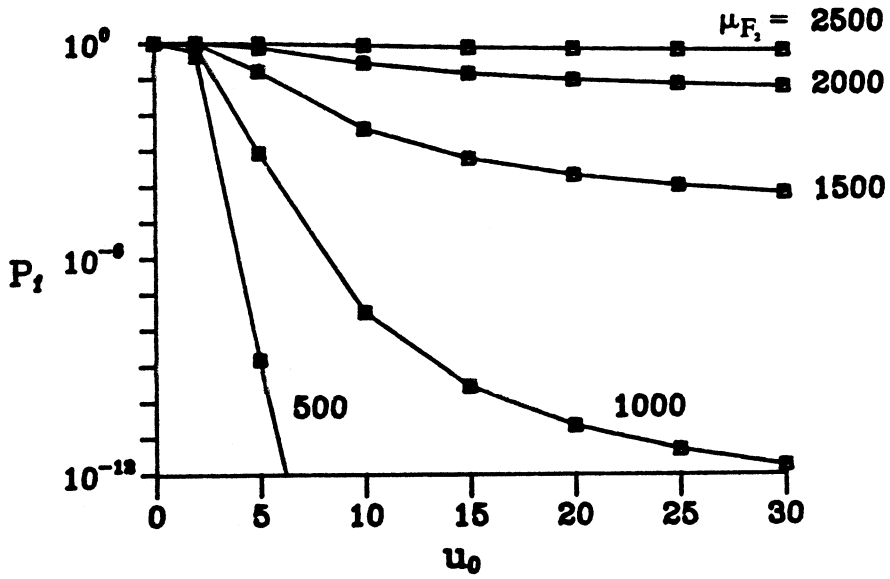


Figure 7.1 Example Built-Up Column



(a)



(b)

Figure 7.2 **FORM and SORM Estimates of Column Failure Probability: (a) Influence of μ_{F_2}; (b) Influence of u_0.**

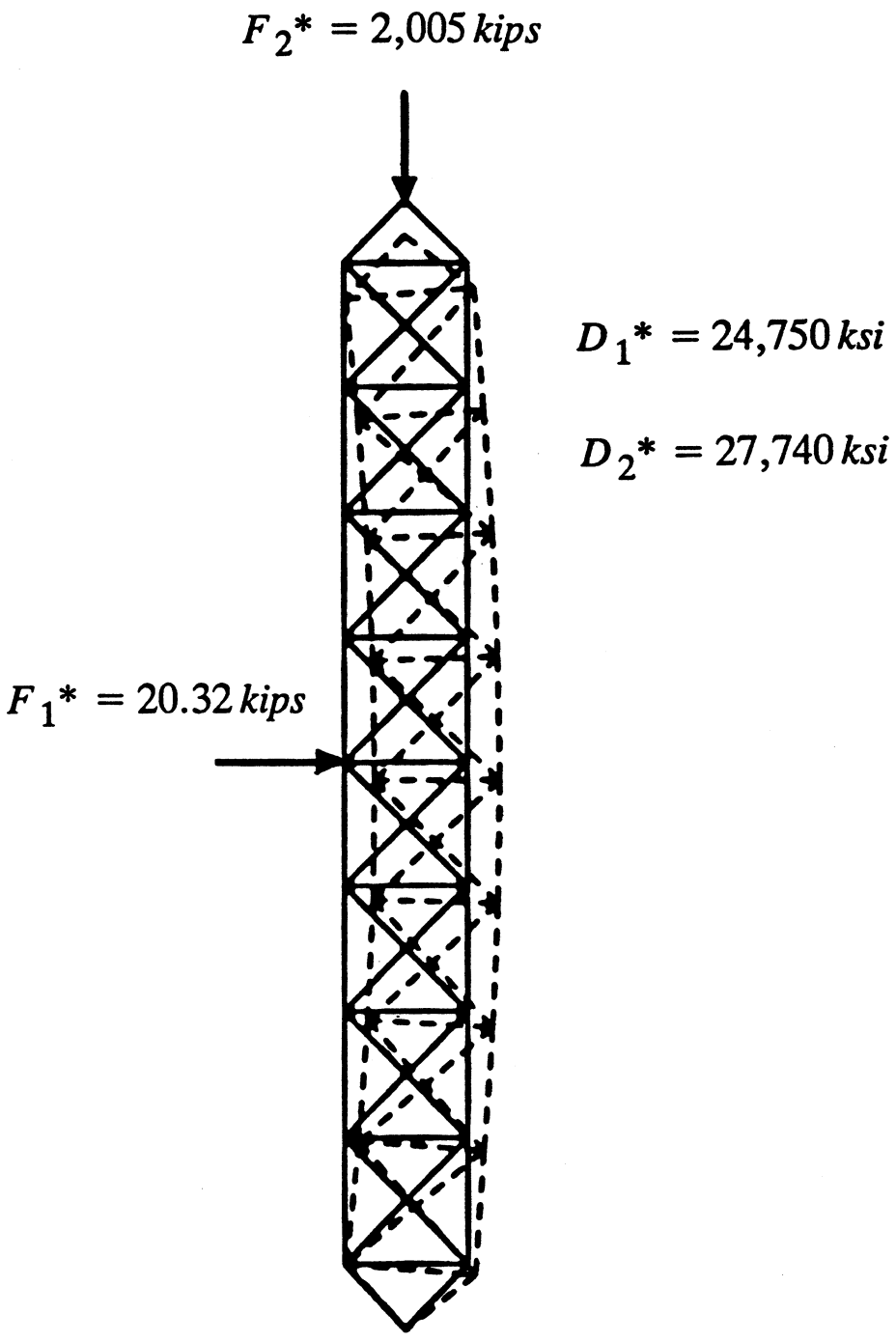


Figure 7.3 **Undeformed and Deformed Column Configurations at the Design Point for $\mu_{F_2} = 1{,}500 kips$ and $u_0 = 30 in$**

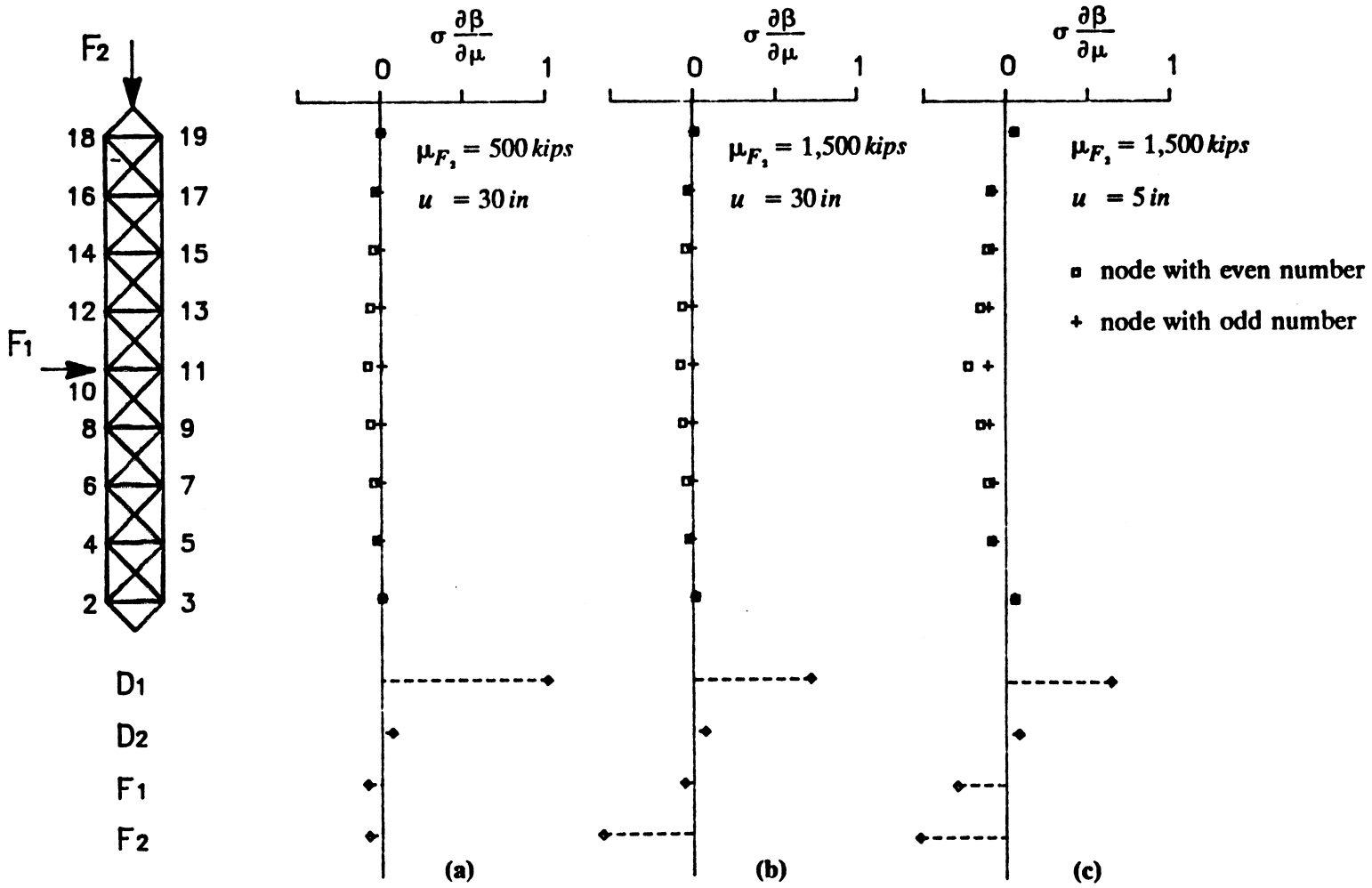


Figure 7.4 Sensitivity Measures $\sigma \frac{\partial \beta}{\partial \mu}$ for the Built-Up Column

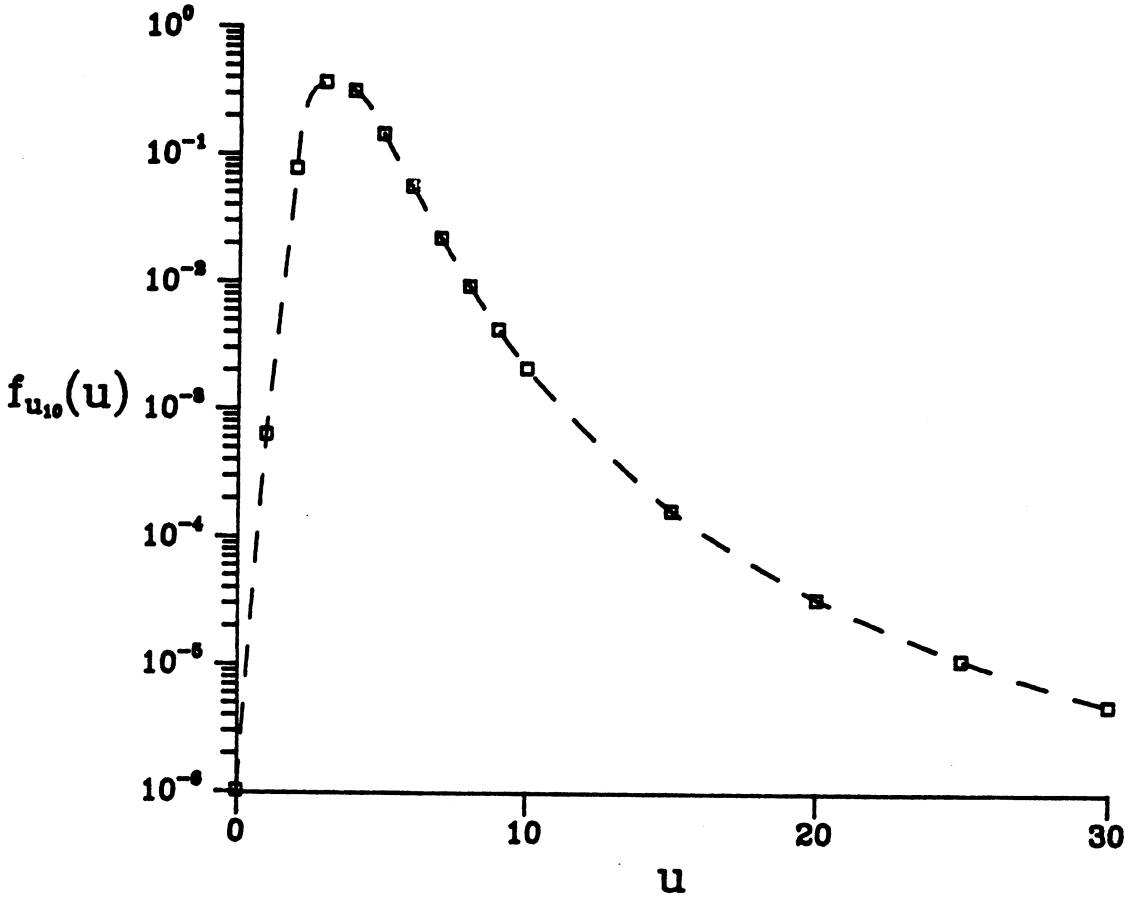


Figure 7.5 **PDF of Midspan Deflection for $\mu_{F_2} = 1500ksi$**

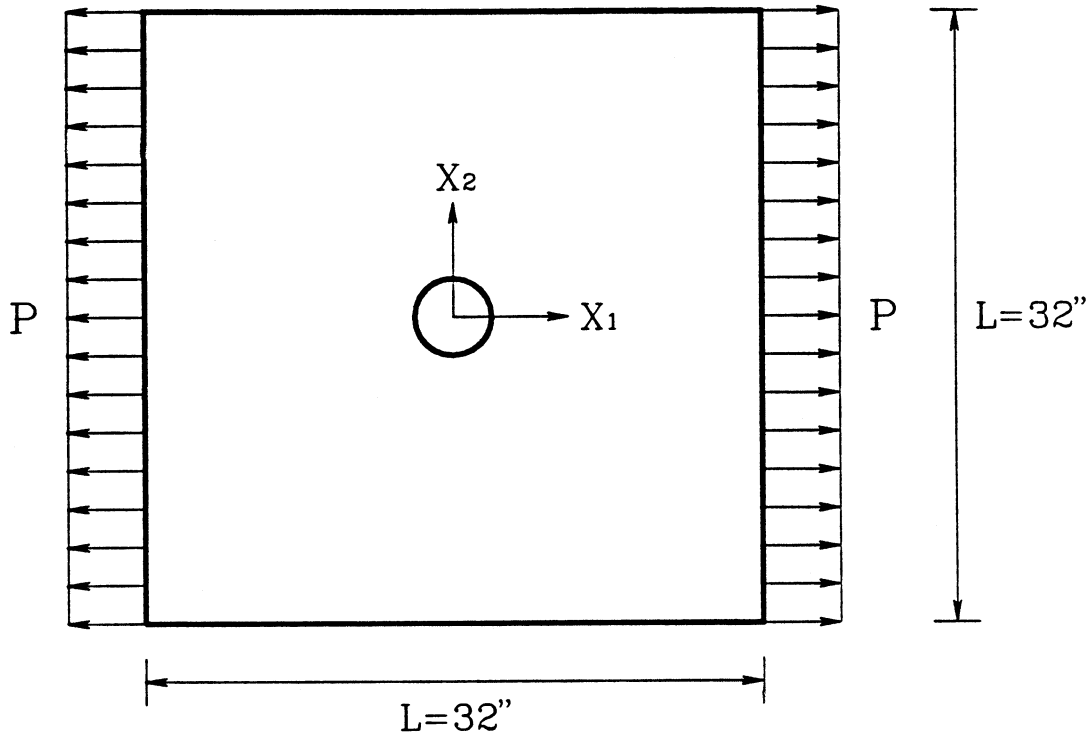
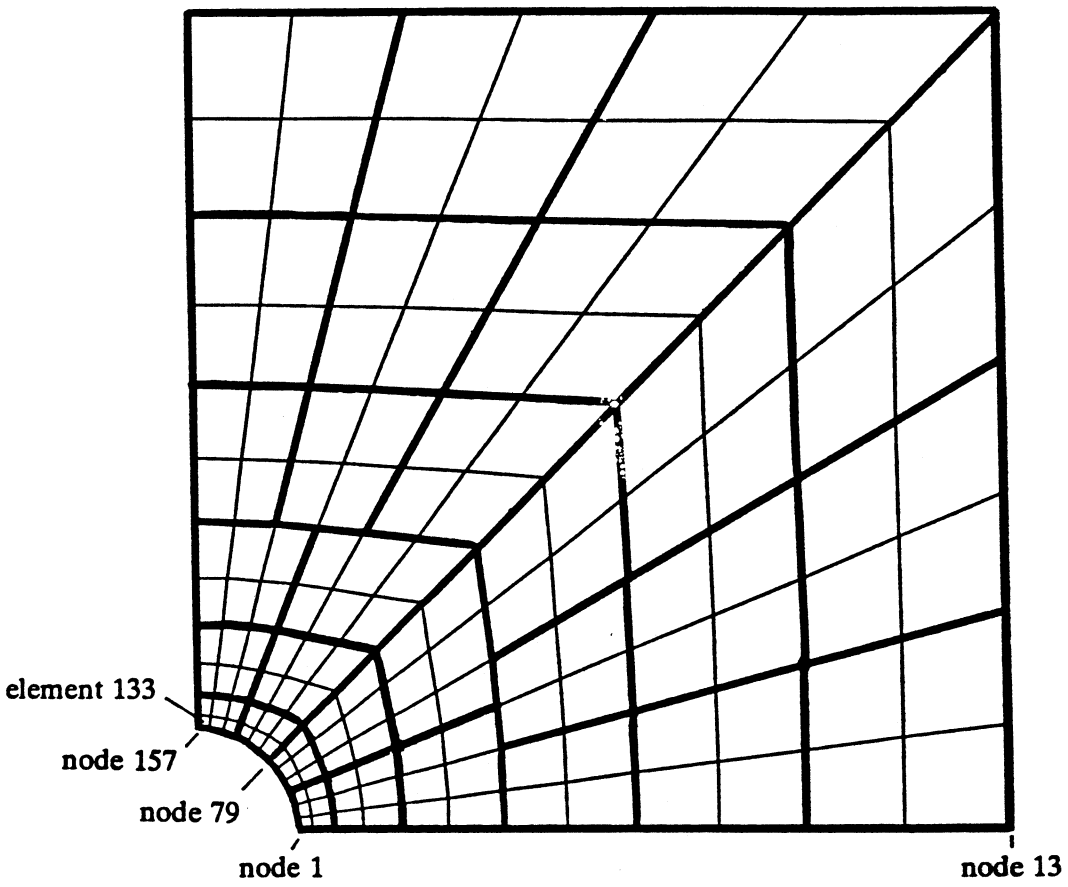
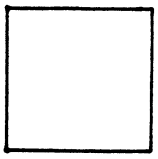
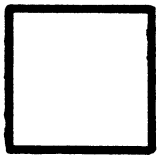


Figure 7.6 Stochastic Plate with a Random Hole



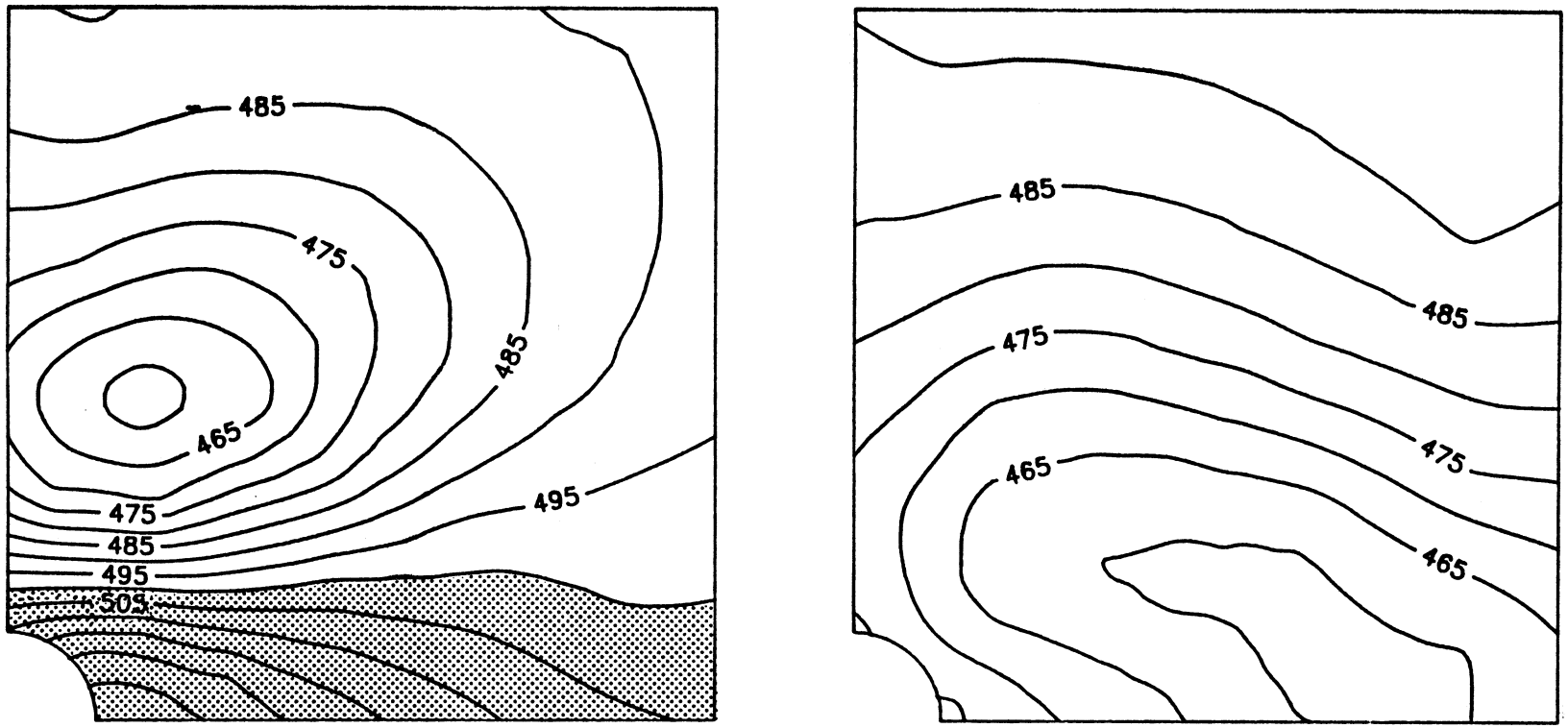


Finite Element Mesh



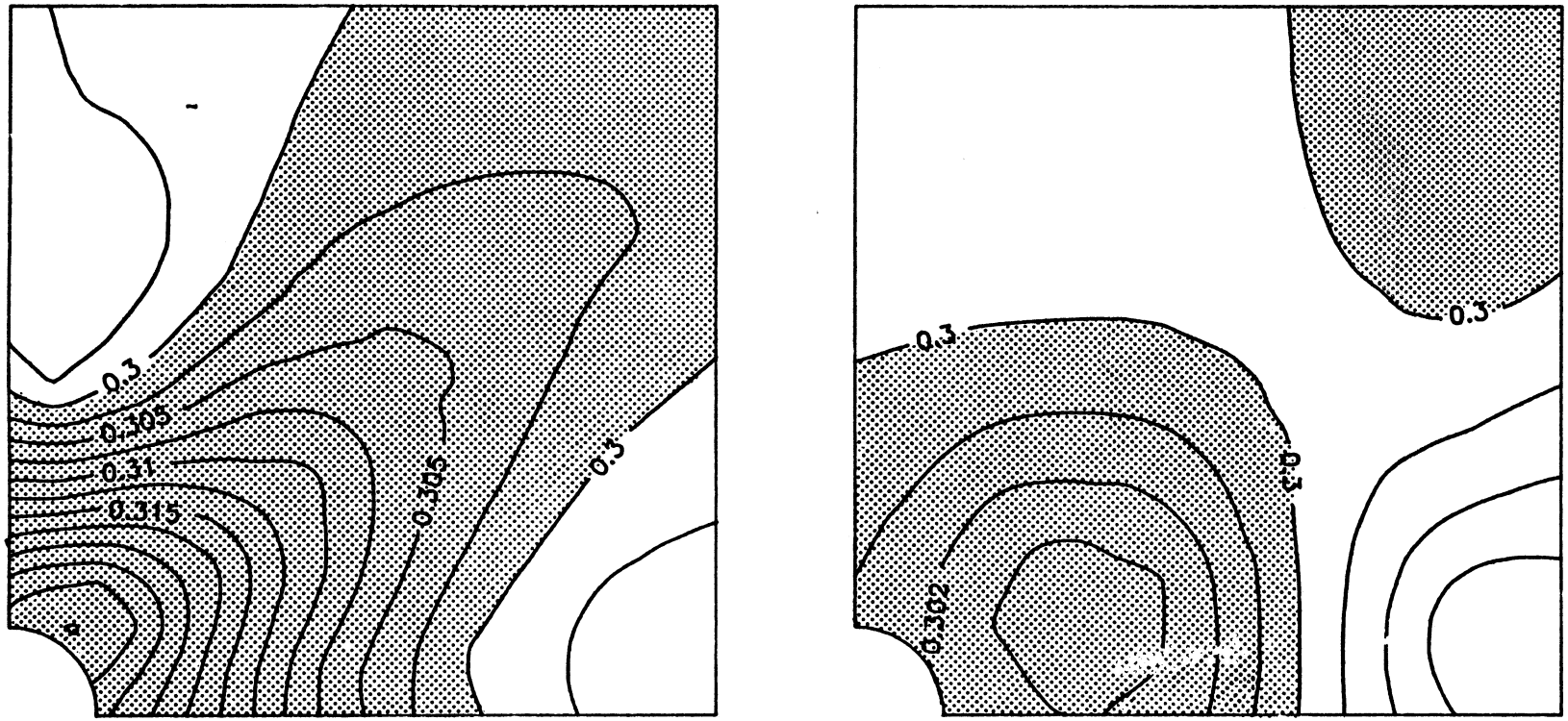
Random Field Mesh

Figure 7.7 Finite Element and Random Field Meshes



(a) (b)

Figure 7.8 **Contours of E^* for $a_E = a_v = 0.125$: (a) Stress Limit State; (b) Displacement Limit State.**



(a) (b)

Figure 7.9 **Contours of v^* for $a_E = a_v = 0.125$: (a) Stress Limit State; (b) Displacement Limit State.**

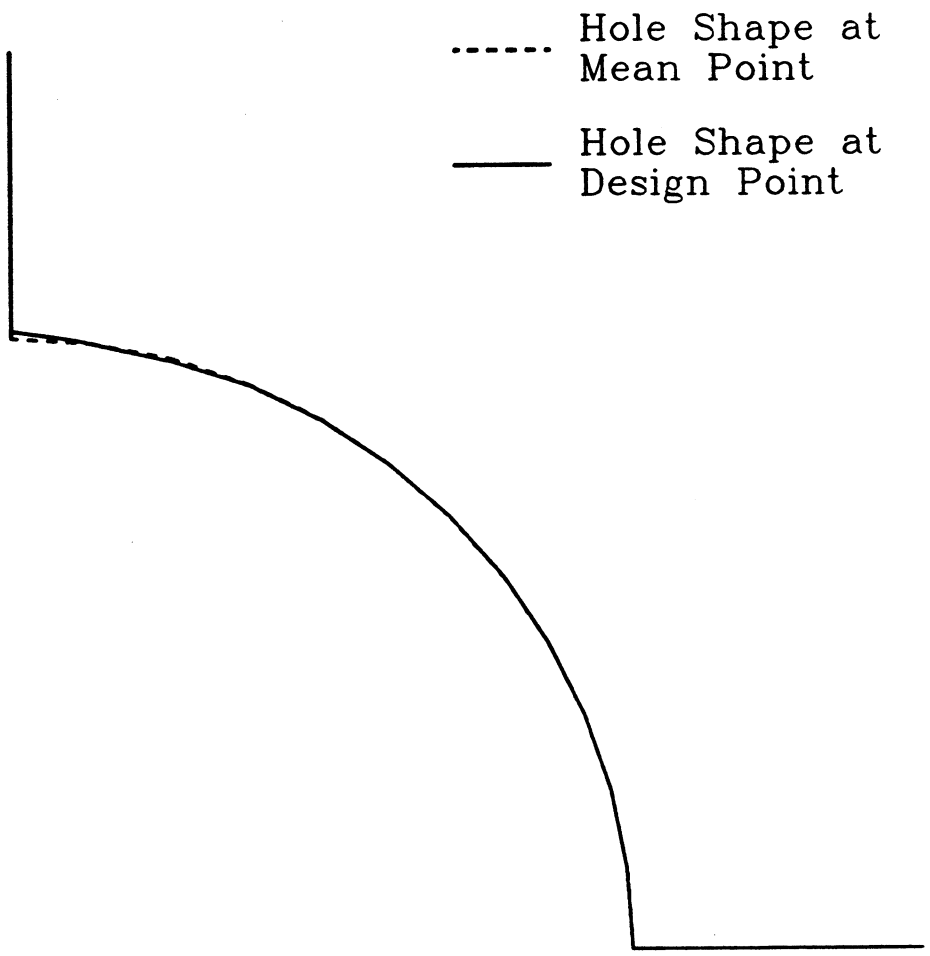
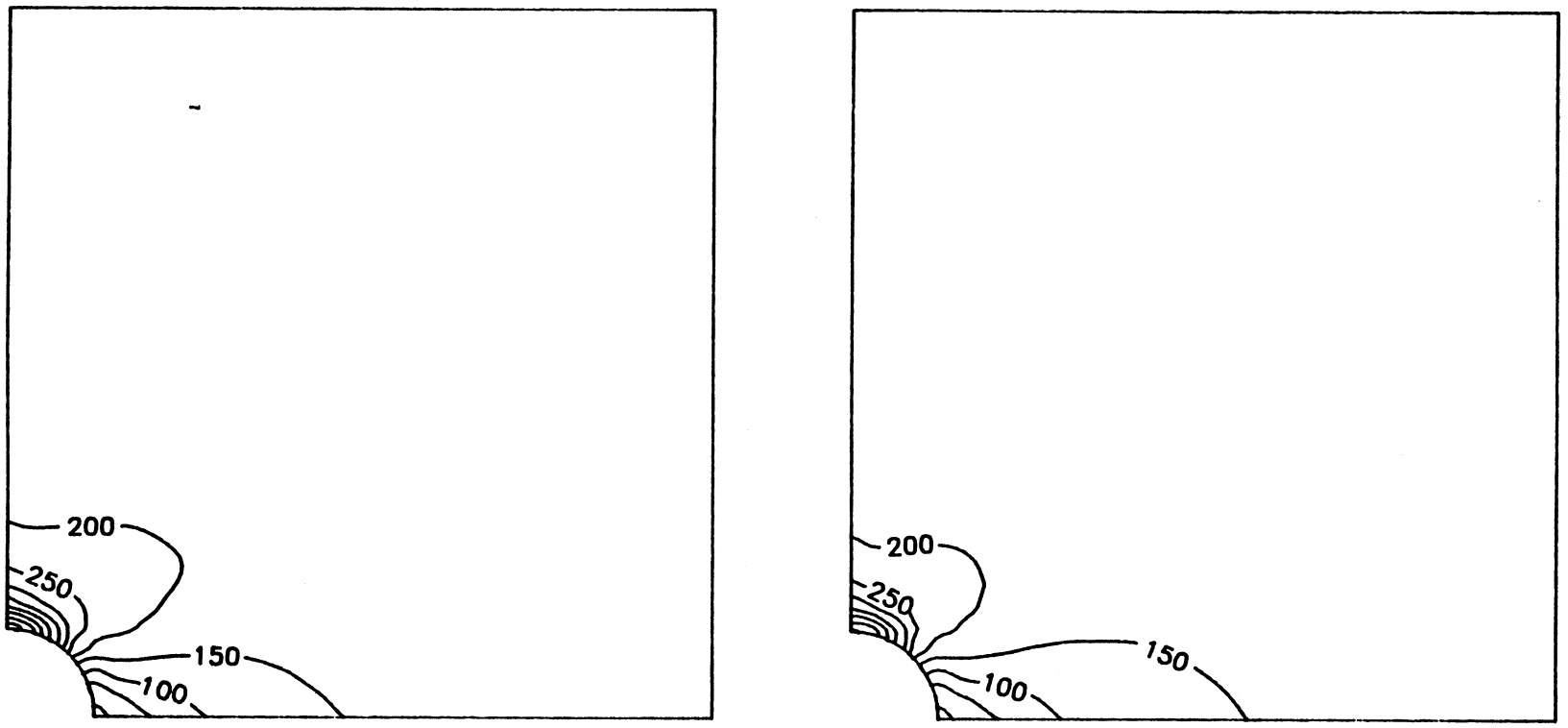


Figure 7.10 Hole Shape at the Design Point for the Stress Limit State



(a) (b)

Figure 7.11 Contours of Principal Tensile Stress at the Design Point for $a_E = a_v = 0.125$: (a) Stress Limit State; (b) Displacement Limit State.

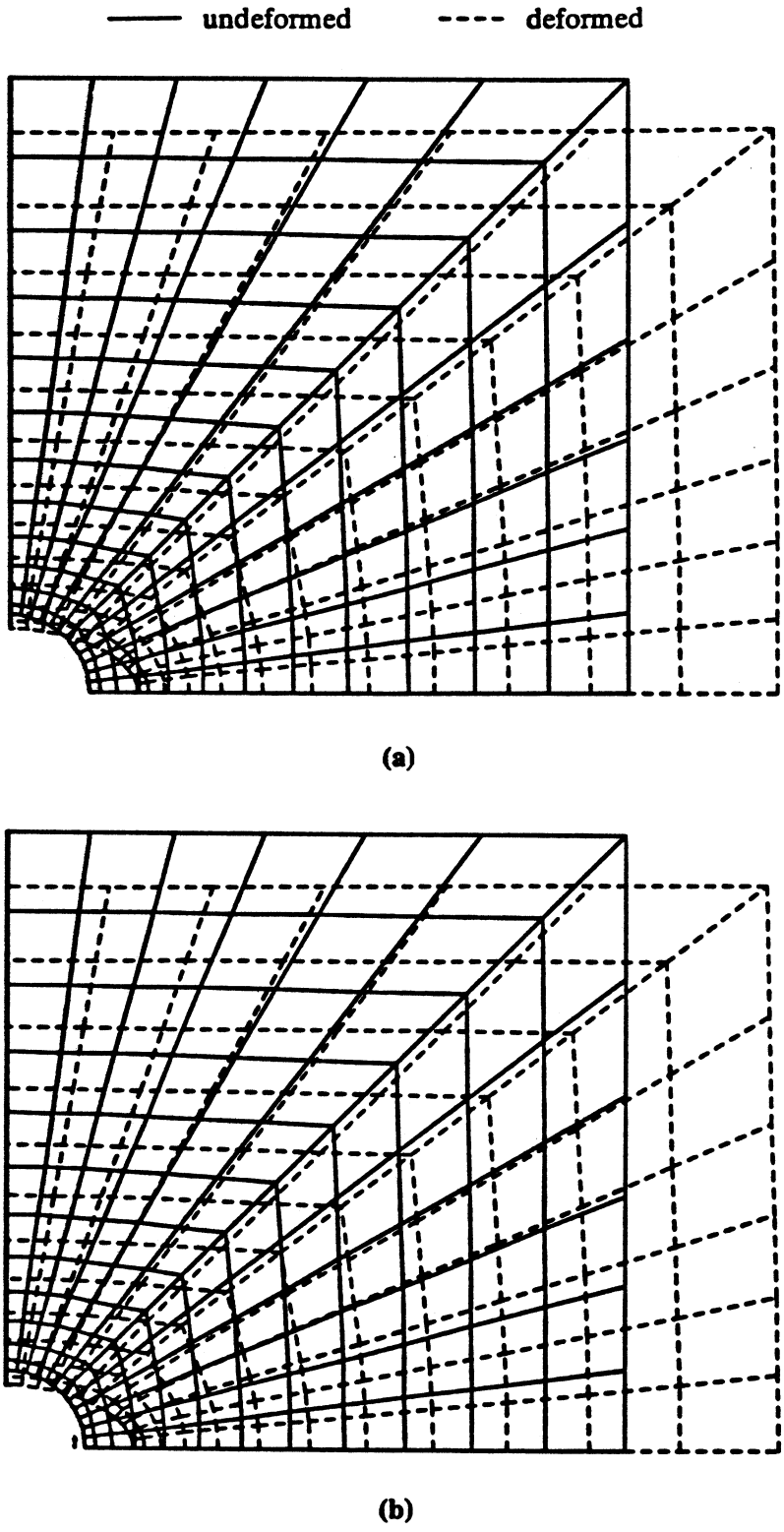
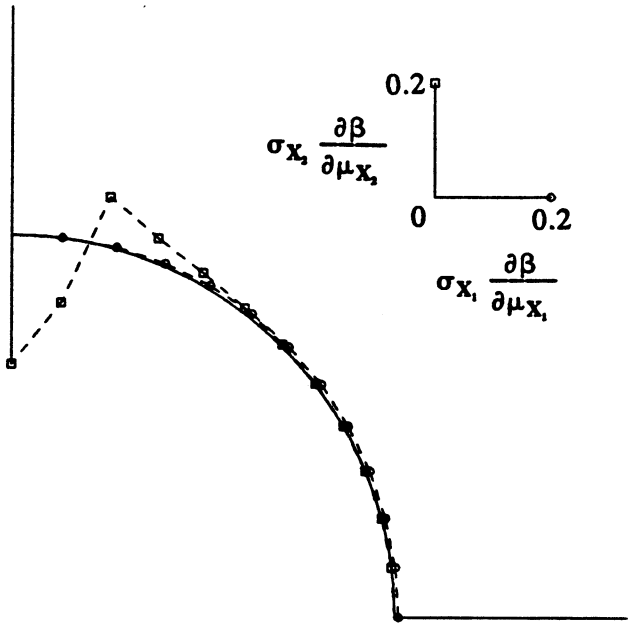
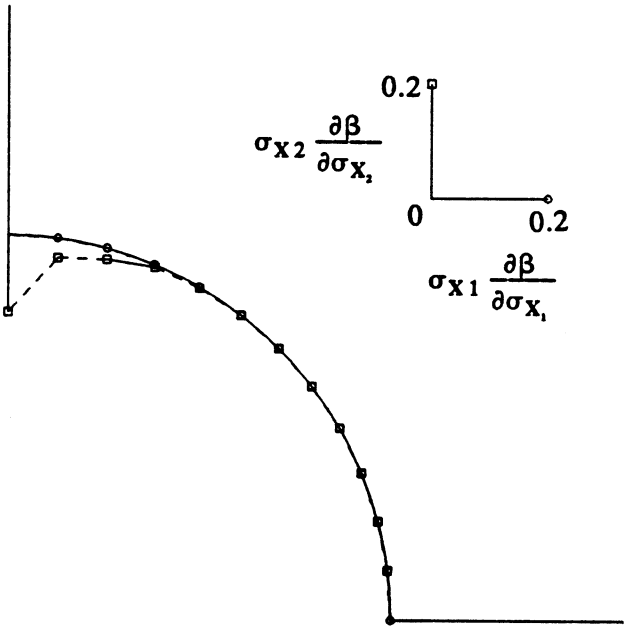


Figure 7.12 Undeformed and Deformed Configurations at the Design Point for $a_E = a_v = 0.125$: (a) Stress Limit State; (b) Displacement Limit State.

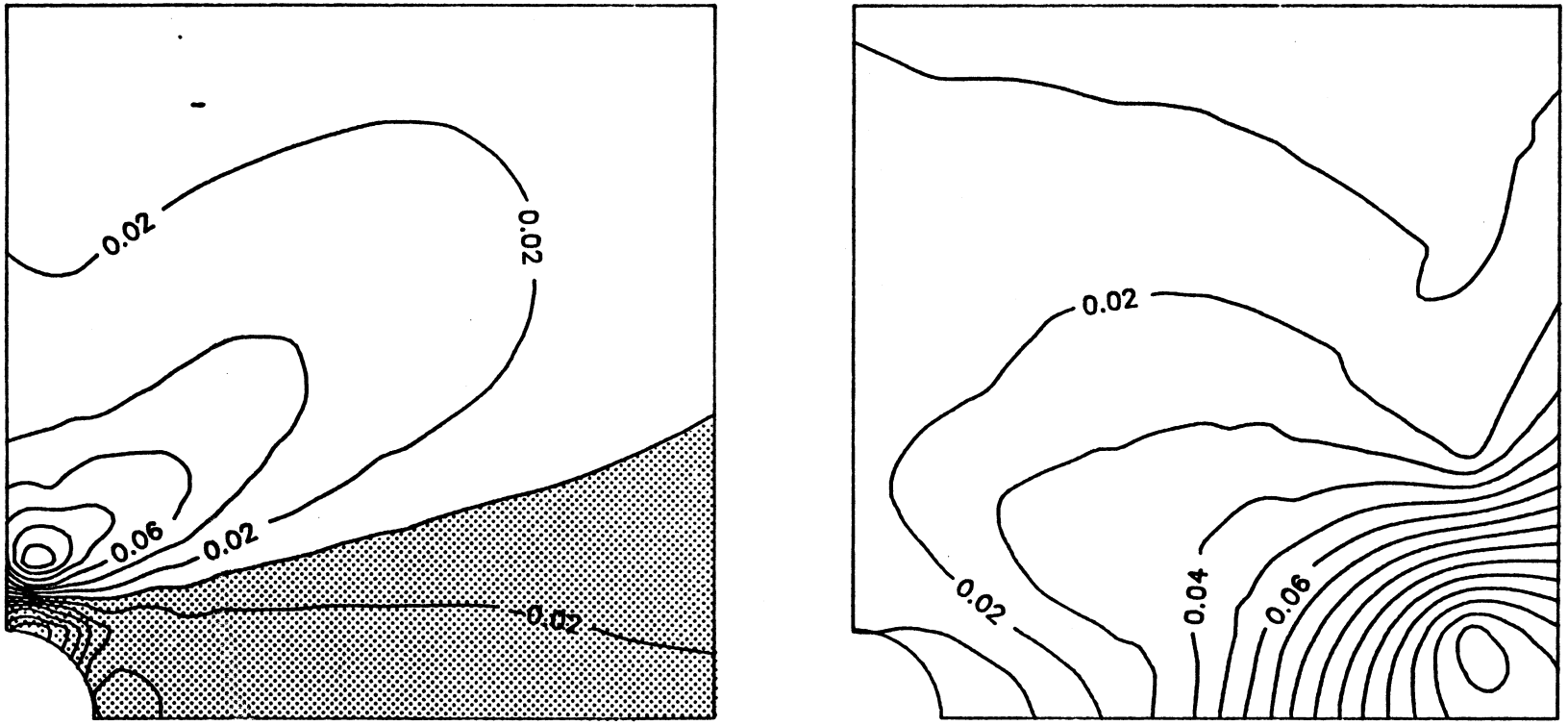


(a)



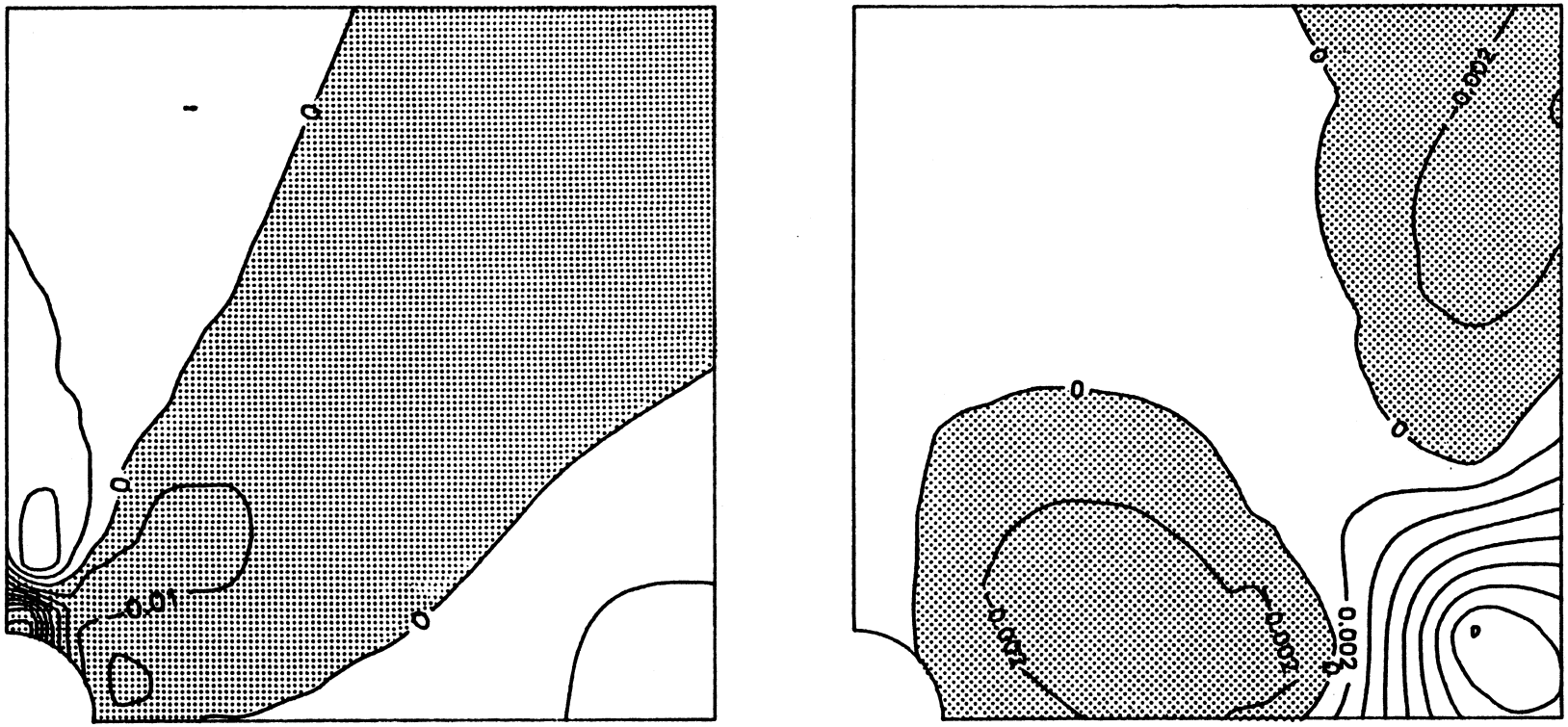
(b)

Figure 7.13 Reliability Sensitivity Measures for the Stress Limit State with $a_E = a_\nu = 0.125$: (a) Measure $\sigma_X \frac{\partial \beta}{\partial \mu_X}$; (b) Measure $\sigma_X \frac{\partial \beta}{\partial \sigma_X}$.



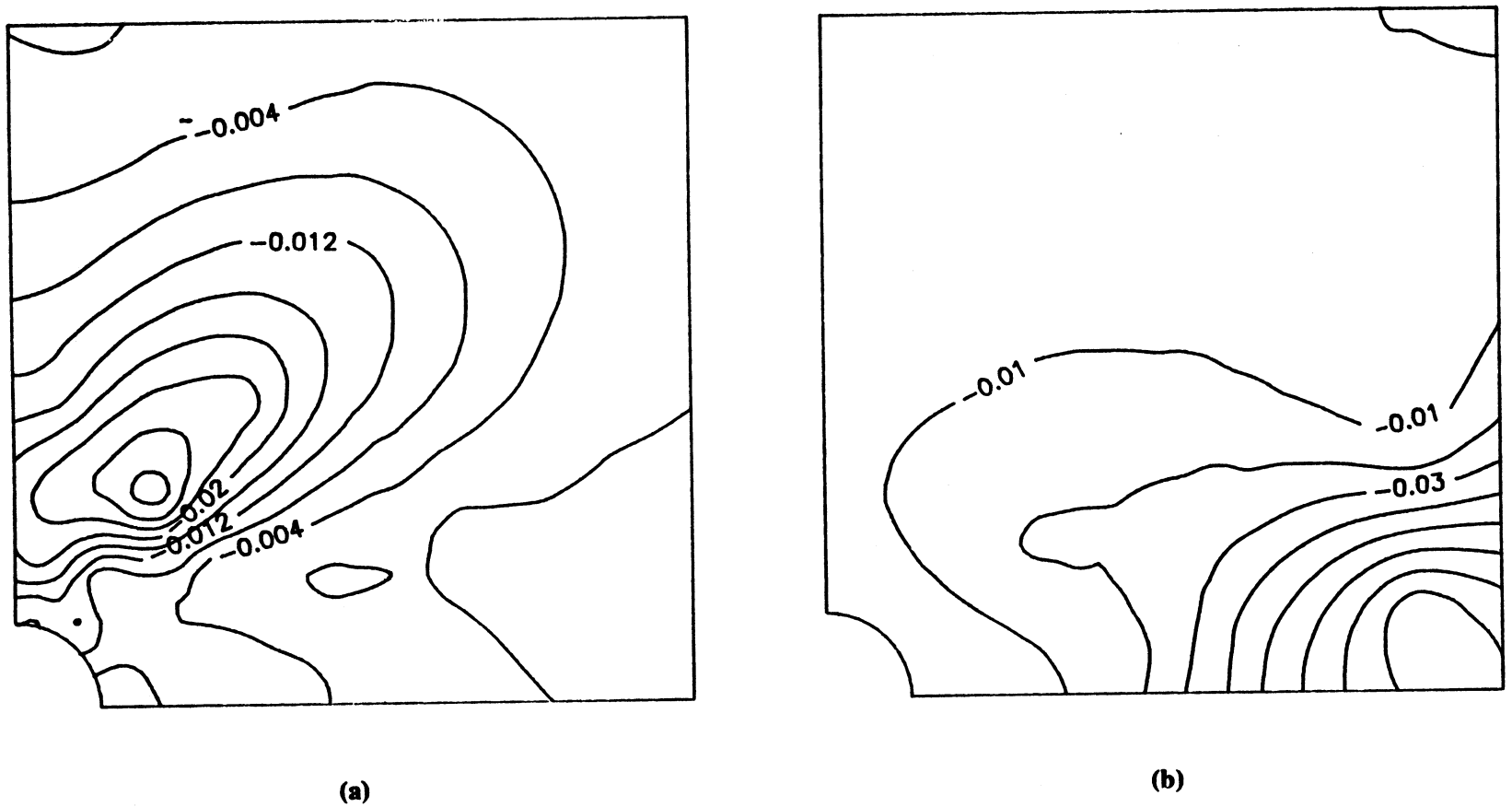
(a) (b)

Figure 7.14 **Contours of Local** $\sigma_E \frac{\partial \beta}{\partial \mu_E}$ **for** $a_E = a_v = 0.125$**: (a) Stress Limit State; (b) Displacement Limit State.**



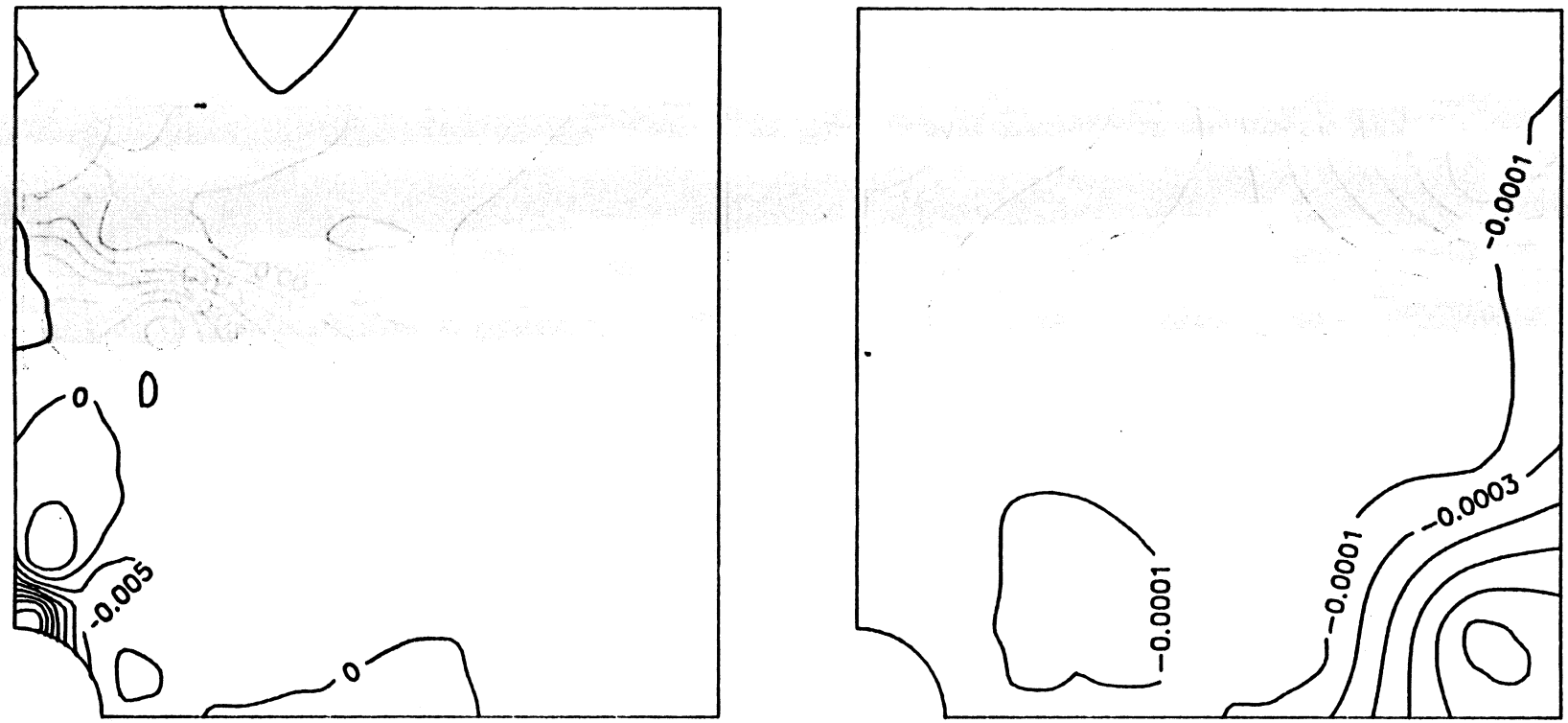
(a) (b)

Figure 7.15 **Contours of Local $\sigma_\nu \frac{\partial \beta}{\partial \mu_\nu}$ for $a_E = a_\nu = 0.125$: (a) Stress Limit State; (b) Displacement Limit State.**



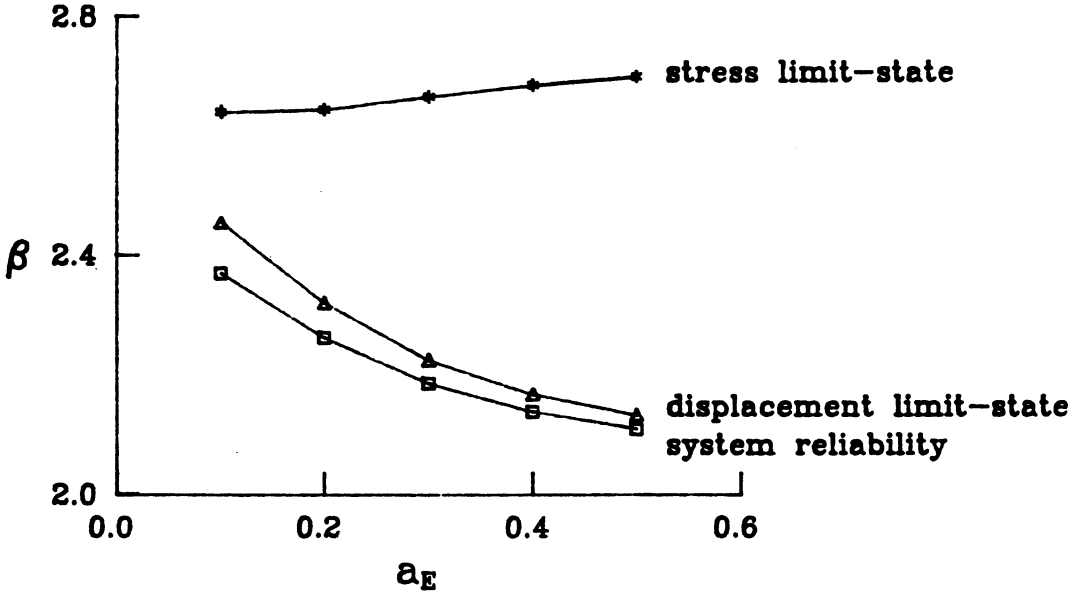
(a) (b)

Figure 7.16 **Contours of Local** $\sigma_E \frac{\partial \beta}{\partial \sigma_E}$ **for** $a_E = a_v = 0.125$**: (a) Stress Limit State; (b) Displacement Limit State.**

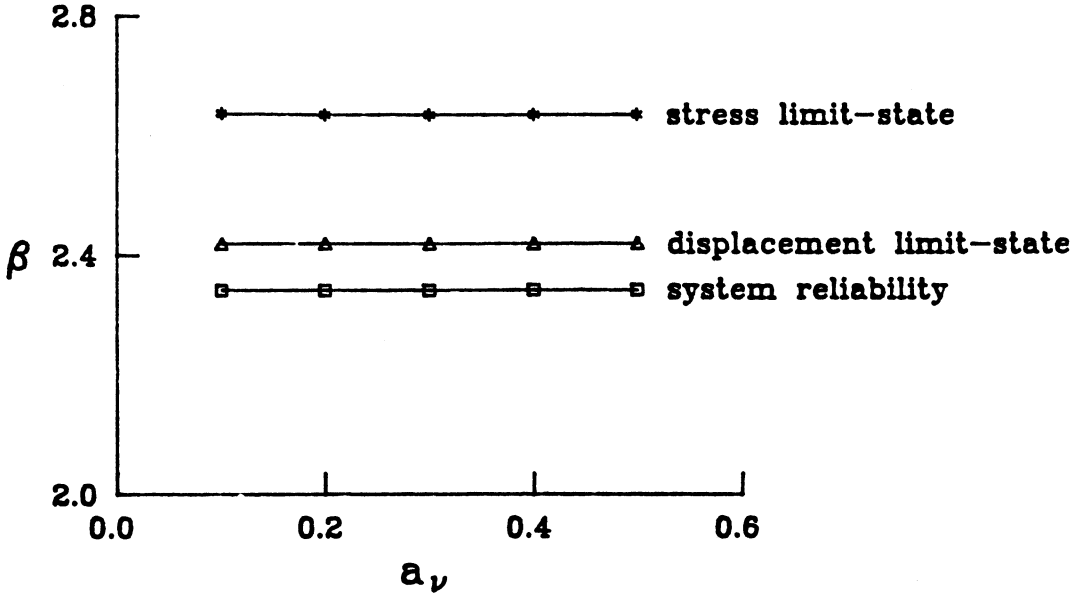


(a) (b)

Figure 7.17 **Contours of Local** $\sigma_v \frac{\partial \beta}{\partial \sigma_v}$ **for** $a_E = a_v = 0.125$**: (a) Stress Limit State; (b) Displacement Limit State.**



(a)



(b)

Figure 7.18 **Influence of Correlation Length on the Reliability Index:**

(a) Influence of a_E for $a_\nu = 0.125$; (b) Influence of a_ν for $a_E = 0.125$.

CHAPTER 8

CONCLUSIONS

8.1 Summary of Report

A general framework for reliability analysis of structures with geometrical nonlinearity is presented. The structure is either linear or nonlinear elastic, and is subjected to static loads. The material properties, geometry, and external loads of the structure are considered as random. These uncertain quantities are represented in terms of a vector of basic random variables. Failure criteria of the structure are expressed in terms of limit-state functions.

Two methods are used to assess the failure probabilities of the structure, namely, the first-order reliability method (FORM) and the second-order reliability method (SORM). In FORM, the limit-state surface in a transformed standard normal space is replaced by its tangent hyperplane at the design point, i.e., the point with the minimum distance from the origin. In SORM, the limit-state surface is fitted at the design point by a paraboloid obtained either by curvature fitting or point fitting.

In both methods, a constrained optimization algorithm is needed to search for the design point. Several optimization algorithms are compared for use in this application. The modified HL-RF method proposed in this study appears to be the most promising.

The structural response and its gradient, which are required in FORM and SORM, are computed by the finite element method. For nonlinear problems, it is important that the gradient is computed analytically since computation of the gradient by a finite difference scheme is not only time-consuming, but of questionable accuracy. In the present study, closed-form expressions for the gradient are derived for elastic structures with geometrical nonlinearity. The expressions are in terms of the tangent stiffness matrix, initial load stiffness matrix, and partial derivatives of the external and resisting force vectors. Similar to the stiffness matrix, the partial derivatives are formed first on the element level

and then assembled into global matrices. No iterations are required in the computation of the gradient in this analytical approach. The expressions presented are applicable to all types of elements and to both referential and spatial formulations. To further improve the efficiency of FORM and SORM, the adjoint method is used to compute the gradient of the limit-state function.

The uncertainties of the structure and its environment are often modeled by random fields. The Nataf model, which is applicable to non-Gaussian fields, is used in this study to model such random fields. Six methods to represent a random field in terms of random variables are compared for their suitability in the finite-element reliability analysis. In this study, the midpoint method is employed to discretize material-property random fields and the nodal-point method is employed to discretize geometry random fields because they are numerically stable and can be applied to non-Gaussian random fields.

In order to carry out the finite-element reliability method, a general-purpose reliability code, CALREL-FEAP, has been developed. The finite element routine, FEAP, has been implemented in the code to compute the structural response and its gradient. The CALREL-FEAP code not only performs FORM and SORM, but also has capabilities for Monte Carlo simulation, directional simulation, and series system reliability analysis. It has three attractive features. First, it incorporates all the time-saving techniques described in this report. Second, its structure is highly modulized. This makes the maintenance and extension of the code relatively easy. Third, it is flexible. The user can develop and implement in the code his/her own reliability method, or extend the finite element part of the code for reliability analysis of a structure with a special mechanical behavior.

The finite-element reliability method is illustrated with two examples: a built-up column and a stochastic plate with a random hole. The examples show that for these two cases, even though the basic random variables are non-normal and the structural behavior is highly nonlinear, the limit-state surfaces in the standard normal space are nearly flat, and FORM provides reasonable estimates of the failure probabilities. These examples also

show that the uncertainties in the material properties and structural geometry may have strong influence on the structural safety in spite of their small coefficients of variation. Sensitivities of failure probabilities with respect to the parameters in the limit-state functions and the distribution parameters of various random variables are examined in both cases. The sensitivity measures with respect to the parameters in the limit-state function can be used to construct the distribution of the structural response. The sensitivity measures with respect to the distribution parameters are shown to be useful in identifying the important sources of uncertainty and in developing means for design modification in order to improve the reliability.

8.2 Future Development

Although only the truss and plane stress elements with linear constitutive laws are included in the two examples of Chapter 7, the proposed finite-element reliability method is readily applicable to other types of geometrically nonlinear elements and constitutive laws. However, expressions for the stiffness matrix and the derivatives of the resisting and external forces with respect to the basic random variables need to be derived for each specific element type and constitutive law. This should be done in the future to make the element library of CALREL-FEAP more complete.

The finite-element reliability method and the CALREL-FEAP code can also be extended to other classes of structural problems. Such extensions would be particularly important for structures with inelastic materials and for structural dynamics problems. With the current version of CALREL-FEAP, such applications are possible (with proper formulation of the limit-state functions) if the response gradients are computed by a finite-difference scheme. However, for practical feasibility, it is necessary to develop and implement efficient algorithms for computing the response gradients. Such an extension would make CALREL-FEAP a powerful tool for structural reliability analysis in vast areas of application.

Other Areas that need further study and development include the modeling and representation of random fields, selection of mesh size, and various techniques to improve the efficiency of the gradient and the reliability computations.

Finally, it is important to note that the finite-element reliability method is in its infancy. As the field progresses and various applications are considered, new areas of study and development will emerge. It is hoped that the present study has taken an early but important step in this course of development.

APPENDIX A

DERIVATION OF ESSENTIAL MATRICES IN THE GRADIENT COMPUTATION

A.1 Truss Element of Example 7.1

Consider a truss element with a uniform cross-sectional area A, as shown in Fig. A.1. Let $\mathbf{U}_1$ and $\mathbf{U}_2$ denote the nodal displacements of the element, and $\mathbf{X}_1^d$ and $\mathbf{X}_2^d$ the original nodal coordinates. The Green-Lagrangian strain E_{ss} of the element is

$$E_{ss} = \frac{1}{2}\left(\frac{l^2}{L^2} - 1\right) = \frac{1}{L^2}\left(\Delta \mathbf{X}^{\mathrm{T}} \Delta \mathbf{U} + \frac{1}{2}\Delta \mathbf{U}^{\mathrm{T}} \Delta \mathbf{U}\right) \tag{A.1}$$

where $\Delta \mathbf{X} = \mathbf{X}_2^d - \mathbf{X}_1^d$, $\Delta \mathbf{U} = \mathbf{U}_2 - \mathbf{U}_1$, and $l = [(\Delta \mathbf{X} + \Delta \mathbf{U})^{\mathrm{T}}(\Delta \mathbf{X} + \Delta \mathbf{U})]^{1/2}$ and $L = [\Delta \mathbf{X}^{\mathrm{T}} \Delta \mathbf{X}]^{1/2}$ are the current and original lengths of the element, respectively. It follows that

$$dE_{ss} = \frac{1}{L^2}[-\Delta \mathbf{x}^{\mathrm{T}} \; \Delta \mathbf{x}^{\mathrm{T}}] \begin{bmatrix} d\mathbf{U}_1 \\ d\mathbf{U}_2 \end{bmatrix} \tag{A.2}$$

where $\Delta \mathbf{x} = \Delta \mathbf{X} + \Delta \mathbf{U}$. Hence, the strain-displacement matrix is

$$\bar{\mathbf{B}} = \frac{1}{L^2}[-\Delta \mathbf{x}^{\mathrm{T}} \; \Delta \mathbf{x}^{\mathrm{T}}] \tag{A.3}$$

The material of the truss elements is considered to be elastic with a strain energy function $E_{ss} D E_{ss}/2$. The second Piola-Kirchhoff stress is thus given by

$$T = D E_{ss} \tag{A.4}$$

Using Eqs. 4.19, A.3, and A.4, the element resisting force vector $\mathbf{R}_e$ is

$$\mathbf{R}_e = \int_0^L \frac{1}{L^2} \begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix} D E_{ss} A \, dL = \frac{D E_{ss} A}{L} \begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix} \tag{A.5}$$

The element tangent stiffness is obtained by taking the derivative of $\mathbf{R}_e$ with respect to the nodal displacements,

$$\mathbf{K}_{Te} = \frac{D A}{L^3} \begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix} [-\Delta \mathbf{x}^{\mathrm{T}} \; \Delta \mathbf{x}^{\mathrm{T}}] + \frac{D E_{ss} A}{L} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \tag{A.6}$$

Since only nodal loads with fixed directions are considered in Example 7.1, $\mathbf{P}$ simply

equals $[0,\ldots,0,F_x,0,\ldots,0,F_y]^T$.

Because the limit-state function in Example 7.1 is in terms of the nodal displacements, only four matrices need to be derived to compute the gradient. They are $\frac{\partial \mathbf{P}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}'}$ $\frac{\partial \mathbf{P}}{\partial \mathbf{v}_l}\Big|_{\mathbf{U}'}$ $\frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\Big|_{\mathbf{U}'}$ and $\frac{\partial \mathbf{R}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}'}$ By differentiation, these matrices are

$$\frac{\partial \mathbf{P}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}} = \mathbf{0} \tag{A.7}$$

$$\frac{\partial \mathbf{P}}{\partial F_x}\Big|_{\mathbf{U}} = [0,\ldots,0,1,0,\ldots,0]^T \tag{A.8}$$

$$\frac{\partial \mathbf{P}}{\partial F_y}\Big|_{\mathbf{U}} = [0,\ldots,0,1]^T \tag{A.9}$$

$$\frac{\partial \mathbf{R}_e}{\partial D}\Big|_{\mathbf{U}} = \frac{E_{ss}A}{L}\begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix} \tag{A.10}$$

$$\frac{\partial \mathbf{R}_e}{\partial \mathbf{X}_1^d}\Big|_{\mathbf{U}} = \frac{DE_{ss}A}{L}\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} + \frac{3DE_{ss}A}{L^3}\begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix}\Delta \mathbf{X}^T - \frac{DA}{L^3}\begin{bmatrix} -\Delta \mathbf{x} \\ \Delta \mathbf{x} \end{bmatrix}\Delta \mathbf{U}^T \tag{A.11}$$

$$\frac{\partial \mathbf{R}_e}{\partial \mathbf{X}_2^d}\Big|_{\mathbf{U}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{X}_1^d}\Big|_{\mathbf{U}} \tag{A.12}$$

The global matrices $\frac{\partial \mathbf{R}}{\partial D}\Big|_{\mathbf{U}}$ and $\frac{\partial \mathbf{R}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}}$ are obtained by assembling the corresponding element matrices.

A.2 Plane Stress Element of Example 7.2

In the conventional finite element method, an arbitrary-shaped, 4-node element Ω_e is often transformed into a master element $\hat{\Omega}_e$ through an isoparametric mapping before element matrices are computed (See Fig. A.2). Accordingly, the element coordinate system (X_1, X_2) is transformed into a natural coordinate system (ξ, η). Each point in $\hat{\Omega}_e$ is mapped onto Ω_e by the following equation:

$$\mathbf{X} = \sum_{\mathbf{I}=1}^{np} \mathbf{N}_\mathbf{I} \mathbf{X}_\mathbf{I}^d \tag{A.13}$$

where np is the number of nodal points of the element, $\mathbf{X}_I^d$ are the nodal coordinates, and N_I are the shape functions defined by

$$N_I = \frac{1}{4}(1+\xi_I\xi)(1+\eta_I\eta) \tag{A.14}$$

in which ξ_I and η_I are the coordinates of the nodal points in the natural coordinate system. These shape functions are also used in the displacement field interpolation, i.e.,

$$\mathbf{u} = \sum_{I=1}^{np} N_I \mathbf{U}_I \tag{A.15}$$

Since the material of the plate has a quadratic strain energy function, its constitutive law is

$$\mathbf{T} = \mathbf{D}\mathbf{E} \tag{A.16}$$

where $\mathbf{D}$ is the tangent elasticity matrix

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{A.17}$$

in which E is Young's modulus and ν is Poisson's ratio. From Eq. 4.14, the Green-Lagrangian strain $\mathbf{E}$ is

$$\mathbf{E} = \sum_{I=1}^{np} (\mathbf{B}_{0I} + \frac{1}{2}\mathbf{B}_{lI})\,\mathbf{U}_I \tag{A.18}$$

The linear and nonlinear strain-displacement matrices $\mathbf{B}_{0I}$ and $\mathbf{B}_{lI}$ for the element are obtained by specializing Eqs. 4.15 and 4.16 for the two dimensional case. They are

$$\mathbf{B}_{0I} = \begin{bmatrix} N_{I,X_1} & 0 \\ 0 & N_{I,X_2} \\ N_{I,X_2} & N_{I,X_1} \end{bmatrix} \tag{A.19}$$

$$\mathbf{B}_{lI} = \begin{bmatrix} \frac{\partial \mathbf{u}^T}{\partial X_1} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{u}^T}{\partial X_2} \\ \frac{\partial \mathbf{u}^T}{\partial X_2} & \frac{\partial \mathbf{u}^T}{\partial X_1} \end{bmatrix} \begin{bmatrix} N_{I,X_1}\mathbf{I} \\ N_{I,X_2}\mathbf{I} \end{bmatrix} = \mathbf{A}\,\mathbf{G}_I \tag{A.20}$$

where

$$\frac{\partial u}{\partial X_K} = \sum_{I=1}^{np} N_{I,X_s} U_I \tag{A.21}$$

and

$$\begin{bmatrix} N_{I,X_1} \\ N_{I,X_2} \end{bmatrix} = \frac{1}{J} \begin{bmatrix} X_{2,\eta} & -X_{2,\xi} \\ -X_{1,\eta} & X_{1,\xi} \end{bmatrix} \begin{bmatrix} N_{I,\xi} \\ N_{I,\eta} \end{bmatrix} \tag{A.22}$$

in which, using the approximation in Eq. A.13,

$$\begin{aligned} J &= X_{1,\xi} X_{2,\eta} - X_{1,\eta} X_{2,\xi} \\ &= \sum_{L=1}^{np} \sum_{K=1}^{np} (N_{L,\xi} N_{K,\eta} - N_{L,\eta} N_{K,\xi}) X_{L1}^d X_{K2}^d \end{aligned} \tag{A.23}$$

is the Jacobian of the coordinate transformation.

In Example 7.2, the plate is subjected to a uniformly distributed tensile load at the two opposite edges, and no body force is considered. It is further assumed that the tensile load remains proportional to the original size of the plate. Hence, $\mathbf{b}_0 = \mathbf{0}$, and $\bar{\mathbf{t}} = [p\ , 0]^T$ on the right edge. It is easily verified that for the nodes on the traction boundary

$$(\mathbf{P}_e)_I = \begin{bmatrix} \frac{pL}{2} \\ 0 \end{bmatrix} \tag{A.24}$$

where L is the original distance between the two nodes on the traction boundary.

The element resisting force at a node I is

$$(\mathbf{R}_e)_I = \int_{\Omega e} \bar{\mathbf{B}}_I^T \mathbf{D}\, \mathbf{E}\, d\Omega \tag{A.25}$$

Also, the IJ-th submatrix of the element tangent stiffness, denoted $(\mathbf{K}_{Te})_{IJ}$, is

$$(\mathbf{K}_{Te})_{IJ} = \frac{\partial (\mathbf{R}_e)_I}{\partial \mathbf{U}_J} = \int_{\Omega_e} \bar{\mathbf{B}}_I^T \mathbf{D}\, \bar{\mathbf{B}}_J\, d\Omega + \int_{\Omega_e} \mathbf{G}_I^T \mathbf{H}\, \mathbf{G}_J\, d\Omega \tag{A.26}$$

where

$$\mathbf{H} = \begin{bmatrix} S_{11}\mathbf{I} & S_{12}\mathbf{I} \\ S_{21}\mathbf{I} & S_{22}\mathbf{I} \end{bmatrix} \tag{A.27}$$

The two limit-state functions in Example 7.2 are

$$g_1(\tilde{T}^*(\mathbf{v})) = 600 - \tilde{T}^*(\mathbf{v}) \tag{A.28}$$

$$g_2(u_{13}(\mathbf{v})) = 4 - u_{13}(\mathbf{v}) \tag{A.29}$$

where $\tilde{T}^*(\mathbf{v})$ is the tensile principal Cauchy stress at the centroid of element 133, and $u_{13}(\mathbf{v})$ is the X_1 displacement of node 13. The gradients of these two limit-state functions are

$$\nabla g_1(\tilde{T}^*(\mathbf{v})) = -\nabla_{\mathbf{v}}\tilde{T}^*(\mathbf{v}) \tag{A.30}$$

$$\nabla g_2(u_{13}(\mathbf{v})) = -\nabla_{\mathbf{v}} u_{13}(\mathbf{v}) \tag{A.31}$$

Note that although the referential formulation is used in the finite element analysis, the first limit-state function is a function of the Cauchy stress tensor $\mathbf{T}^*$. Rearranging the terms in Eq. 4.4 and making use of Eq. A.16,

$$\mathbf{T}^* = \frac{1}{\det \mathbf{F}} \mathbf{F}\mathbf{D}\mathbf{E}\mathbf{F}^{\mathrm{T}} \tag{A.32}$$

where, by Eqs. 4.3 and A.15,

$$\mathbf{F} \approx \mathbf{I} + \sum_{\mathrm{I}=1}^{np} \mathbf{U}_{\mathrm{I}}[\mathrm{N}_{\mathrm{I},X_1}\ \mathrm{N}_{\mathrm{I},X_2}] \tag{A.33}$$

It is seen that the Cauchy stresses are not explicit functions of the Green-Lagrangian strains. Hence, rather than applying Eq. 5.1 directly, it is more convenient to expand Eq. A.30 as

$$\nabla g_1(\tilde{T}^*(\mathbf{v})) = -\frac{\partial \tilde{T}^*(\mathbf{v})}{\partial \mathbf{v}}\bigg|_{\mathbf{U}} - \frac{\partial \tilde{T}^*(\mathbf{v})}{\partial \mathbf{U}}\bigg|_{\mathbf{v}} \nabla_{\mathbf{v}}\mathbf{U} \tag{A.34}$$

The basic random variables considered include the tensile magnitude p, the discretized random fields E and ν, and the nodal coordinates of the hole boundary. Hence, six extra matrices need to be computed to obtain the gradient of the limit-state functions. They are $\frac{\partial \mathbf{P}}{\partial \mathbf{v}_l}\Big|_{\mathbf{U}}$, $\frac{\partial \mathbf{R}}{\partial \mathbf{v}_m}\Big|_{\mathbf{U}}$, $\frac{\partial \mathbf{T}^*}{\partial \mathbf{v}_m}\Big|_{\mathbf{U}}$, $\frac{\partial \mathbf{R}}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}}$, $\frac{\partial \mathbf{T}^*}{\partial \mathbf{X}^d}\Big|_{\mathbf{U}}$, and $\frac{\partial \mathbf{T}^*}{\partial \mathbf{U}}\Big|_{\mathbf{v}}$. The partial derivatives of $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{T}^*$ with respect to the load and material variables are

$$\frac{\partial (\mathbf{P}_e)_{\mathrm{I}}}{\partial p}\bigg|_{\mathbf{U}} = \begin{bmatrix} \frac{L}{2} \\ 0 \end{bmatrix} \quad \text{for nodes on the traction boundary} \tag{A.35}$$

$$\frac{\partial (\mathbf{R}_e)_I}{\partial E}\bigg|_U = \int_{\Omega_e} \bar{\mathbf{B}}_I^T \frac{\partial \mathbf{D}}{\partial E} \mathbf{E}\, d\Omega \tag{A.36}$$

$$\frac{\partial (\mathbf{R}_e)_I}{\partial \nu}\bigg|_U = \int_{\Omega_e} \bar{\mathbf{B}}_I^T \frac{\partial \mathbf{D}}{\partial \nu} \mathbf{E}\, d\Omega \tag{A.37}$$

$$\frac{\partial \mathbf{T}^*}{\partial E}\bigg|_U = \frac{1}{\det \mathbf{F}} \mathbf{F} \frac{\partial \mathbf{D}}{\partial E} \mathbf{E} \mathbf{F}^T \tag{A.38}$$

$$\frac{\partial \mathbf{T}^*}{\partial \nu}\bigg|_U = \frac{1}{\det \mathbf{F}} \mathbf{F} \frac{\partial \mathbf{D}}{\partial \nu} \mathbf{E} \mathbf{F}^T \tag{A.39}$$

where

$$\frac{\partial \mathbf{D}}{\partial E} = \frac{1}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{A.40}$$

$$\frac{\partial \mathbf{D}}{\partial \nu} = E \begin{bmatrix} \frac{2\nu}{(1-\nu^2)^2} & \frac{1+\nu^2}{(1-\nu^2)^2} & 0 \\ \frac{1+\nu^2}{(1-\nu^2)^2} & \frac{2\nu}{(1-\nu^2)^2} & 0 \\ 0 & 0 & -\frac{1}{2(1+\nu)^2} \end{bmatrix} \tag{A.41}$$

The partial derivatives of $\mathbf{R}$ and $\mathbf{T}^*$ with respect to the nodal coordinates are

$$\frac{\partial (\mathbf{R}_e)_I}{\partial X_j^d} = \int_{\Omega_e} \frac{\partial \bar{\mathbf{B}}_I^T}{\partial X_j^d} \mathbf{D}\mathbf{E} J\, d\Omega + \int_{\Omega_e} \bar{\mathbf{B}}_I^T \mathbf{D} \sum_{K=1}^{np} \left[\frac{\partial \mathbf{B}_{0K}}{\partial X_j^d} + \frac{1}{2} \frac{\partial \mathbf{B}_{lK}}{\partial X_j^d} \right] \mathbf{U}_K J\, d\Omega$$
$$+ \int_{\Omega_e} \bar{\mathbf{B}}_I^T \mathbf{D}\mathbf{E} \frac{\partial J}{\partial X_j^d} d\Omega \tag{A.42}$$

$$\frac{\partial \mathbf{T}^*}{\partial X^d}\bigg|_U = -\frac{1}{(\det \mathbf{F})^2} \frac{\partial \det \mathbf{F}}{\partial X^d}\bigg|_U \mathbf{F}\mathbf{D}\mathbf{E}\mathbf{F}^T + \frac{1}{\det \mathbf{F}} \frac{\partial \mathbf{F}}{\partial X^d}\bigg|_U \mathbf{D}\mathbf{E}\mathbf{F}^T$$
$$+ \frac{1}{\det \mathbf{F}} \mathbf{F}\mathbf{D} \frac{\partial \mathbf{E}}{\partial X^d}\bigg|_U \mathbf{F}^T + \frac{1}{\det \mathbf{F}} \mathbf{F}\mathbf{D}\mathbf{E} \frac{\partial \mathbf{F}^T}{\partial X^d}\bigg|_U \tag{A.43}$$

Note that in the above expressions, $\bar{\mathbf{B}}_I$, $\mathbf{B}_{0I}$, $\mathbf{B}_{lI}$, J, $\mathbf{F}$, and $\mathbf{E}$ are all functions of N_{I,X_L}. Thus, the derivatives of these terms with respect to X_j^d can be expressed in terms of the derivatives of N_{I,X_L}. Using Eqs. A.13 and A.22,

$$N_{I,x_1} = \frac{1}{J}\sum_{K=1}^{np}(N_{K,\eta}N_{I,\xi} - N_{K,\xi}N_{I,\eta})X^{\ell}_{K2} \tag{A.44}$$

$$N_{I,x_2} = \frac{1}{J}\sum_{K=1}^{np}(-N_{K,\eta}N_{I,\xi} + N_{K,\xi}N_{I,\eta})X^{\ell}_{K1} \tag{A.45}$$

Hence,

$$\frac{\partial N_{I,x_1}}{\partial X^{\ell}_{J1}} = -\frac{1}{J^2}\frac{\partial J}{\partial X^{\ell}_{J1}}\sum_{K=1}^{np}(N_{K,\eta}N_{I,\xi} - N_{K,\xi}N_{I,\eta})X^{\ell}_{K2} \tag{A.46}$$

$$\frac{\partial N_{I,x_1}}{\partial X^{\ell}_{J2}} = -\frac{1}{J^2}\frac{\partial J}{\partial X^{\ell}_{J2}}\sum_{K=1}^{np}(N_{K,\eta}N_{I,\xi} - N_{K,\xi}N_{I,\eta})X^{\ell}_{K2} + \frac{1}{J}(N_{J,\eta}N_{I,\xi} - N_{J,\xi}N_{I,\eta}) \tag{A.47}$$

$$\frac{\partial N_{I,x_2}}{\partial X^{\ell}_{J1}} = \frac{1}{J^2}\frac{\partial J}{\partial X^{\ell}_{J1}}\sum_{K=1}^{np}(N_{K,\eta}N_{I,\xi} - N_{K,\xi}N_{I,\eta})X^{\ell}_{K1} - \frac{1}{J}(N_{J,\eta}N_{I,\xi} - N_{J,\xi}N_{I,\eta}) \tag{A.48}$$

$$\frac{\partial N_{I,x_2}}{\partial X^{\ell}_{J2}} = \frac{1}{J^2}\frac{\partial J}{\partial X^{\ell}_{J2}}\sum_{K=1}^{np}(N_{K,\eta}N_{I,\xi} - N_{K,\xi}N_{I,\eta})X^{\ell}_{K1} \tag{A.49}$$

where

$$\frac{\partial J}{\partial X^{\ell}_{J1}} = \sum_{L=1}^{np}(N_{L,\eta}N_{J,\xi} - N_{J,\eta}N_{L,\xi})X^{\ell}_{L2} \tag{A.50}$$

$$\frac{\partial J}{\partial X^{\ell}_{J2}} = \sum_{L=1}^{np}(N_{J,\eta}N_{L,\xi} - N_{L,\eta}N_{J,\xi})X^{\ell}_{L1} \tag{A.51}$$

Finally, the partial derivative of $\mathbf{T}^*$ with respect to the nodal displacements is

$$\left.\frac{\partial \mathbf{T}^*}{\partial \mathbf{U}}\right|_{\mathbf{v}} = -\frac{1}{(\det \mathbf{F})^2}\left.\frac{\partial \det \mathbf{F}}{\partial \mathbf{U}}\right|_{\mathbf{x}'} \mathbf{F}\mathbf{D}\mathbf{E}\mathbf{F}^{\mathrm{T}} + \frac{1}{\det \mathbf{F}}\left.\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right|_{\mathbf{x}'} \mathbf{D}\mathbf{E}\mathbf{F}^{\mathrm{T}}$$
$$+ \frac{1}{\det \mathbf{F}}\mathbf{F}\mathbf{D}\left.\frac{\partial \mathbf{E}}{\partial \mathbf{U}}\right|_{\mathbf{x}'} \mathbf{F}^{\mathrm{T}} + \frac{1}{\det \mathbf{F}}\mathbf{F}\mathbf{D}\mathbf{E}\left.\frac{\partial \mathbf{F}^{\mathrm{T}}}{\partial \mathbf{U}}\right|_{\mathbf{x}'} \tag{A.52}$$

where

$$\left.\frac{\partial \mathbf{F}}{\partial U_{I1}}\right|_{\mathbf{x}'} = \begin{bmatrix} N_{I,x_1} & N_{I,x_2} \\ 0 & 0 \end{bmatrix} \tag{A.53}$$

$$\left.\frac{\partial \mathbf{F}}{\partial U_{I2}}\right|_{\mathbf{x}'} = \begin{bmatrix} 0 & 0 \\ N_{I,x_1} & N_{I,x_2} \end{bmatrix} \tag{A.54}$$

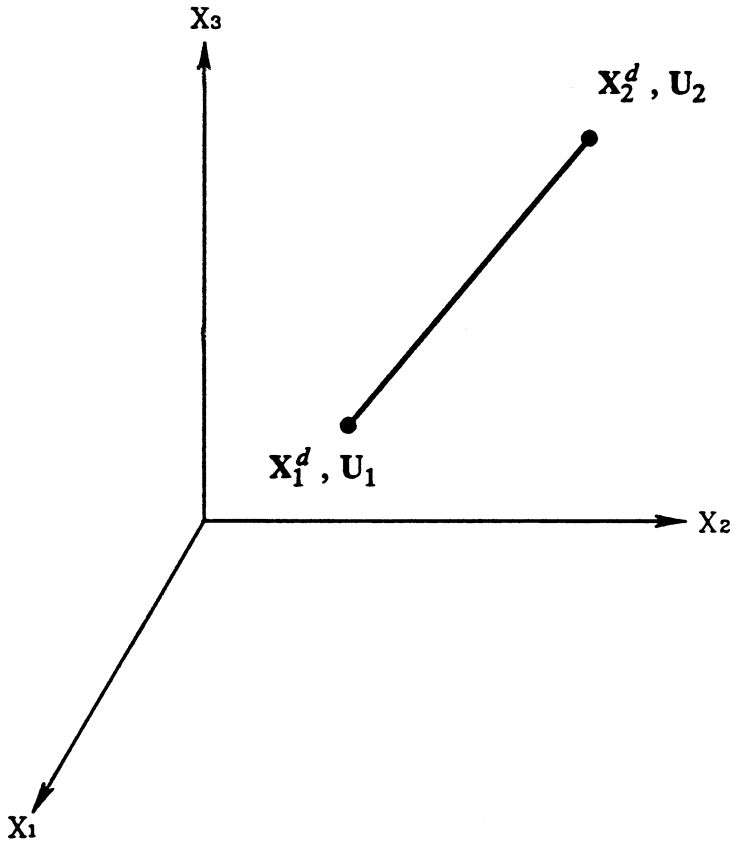


Figure A.1 Truss Element

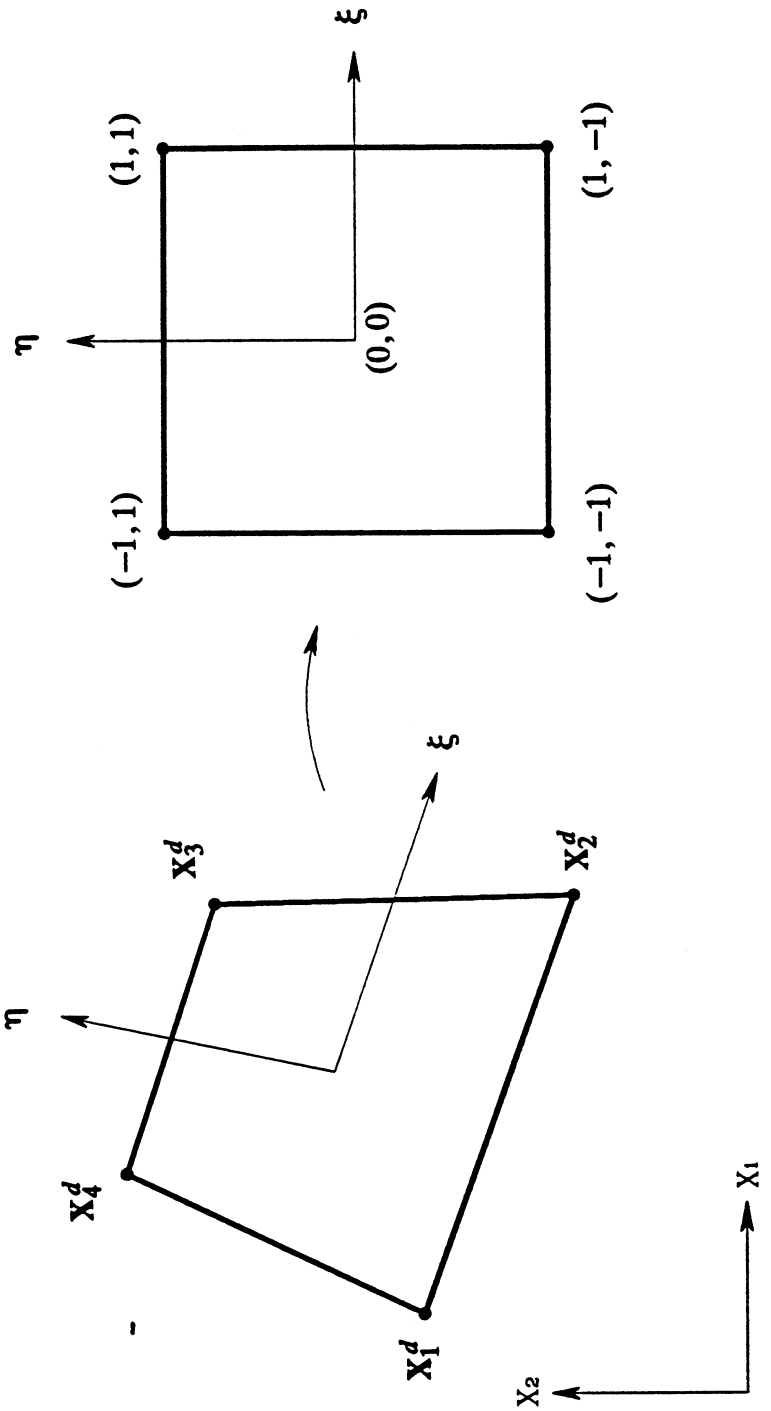


Figure A.2 Coordinate Transformation for 4-node Element

APPENDIX B

PROGRAM CALREL-FEAP

B.1 Linkage of CALREL and FEAP

This section describes the changes made in CALREL and FEAP to make the linkage of these two codes possible.

B.1.1 Memory Allocation

As independent programs, CALREL and FEAP have their own dynamic array allocations. In CALREL, a large integer array is declared in the blank common block, and a package developed by E. Wilson for use in the finite element code CAL-80 is employed to allocate memory. This package includes subroutines which allocate new arrays, locate existing arrays, and delete existing arrays. The initial position of the unused portion of the blank common block, *next*, is updated whenever the package is invoked and stored in a common block /dbsy/. FEAP also declares an integer array in the blank common block, and it uses a subroutine PSETM to allocate new arrays. Subroutine PSETM keeps a record of the length, *kmax*, of the used portion of the array in the common block /plong/. It would be a waste of memory space if CALREL and the subprogram FEAP used two independent allocation arrays. In order for the subprogram FEAP to share the same allocation array as CALREL, both the common blocks /dbsy/ and /plong/ have been passed to the main routine of FEAP. The variables *next* and *kmax* are updated in that routine so that both CALREL and FEAP are informed of the currently available space. In this way, the overlapping of array allocations by the two codes is avoided, while the allocation routine of each code remains essentially untouched. This point is important since it reduces the changes to be made in each code and, consequently, the possibility of coding errors. The conceptual diagram of the memory allocation of CALREL-FEAP is shown in Fig. B.1.

B.1.2 Common Blocks and Arguments

In order to transfer control parameters between CALREL and FEAP, some modifications on the common blocks have been made. Five new common blocks have been created:

```
common /crel/  ifeap,ifeapo,ifres
common /jjcf/  nmnc,ngnc,ntdc,nmnf,ngnf,ntdf,nrvc,nrvf,nrvm,nrvb
common /mdat3/ n16,n17,n18,n19,n20,n21,n22,n23,n24,n25,n26,n27,n28
common /mdat4/ nes0,les0,nsts,ndeb,ndeu,nesc,nele,ndrb
common /grad/  flgf,flgr
```

Common block /crel/ stores the current limit-state function number, *ifeap*, the limit-state function number in the previous call to FEAP, *ifeapo*, and the flag *ifres*, which indicates whether this is a restart run of FEAP or not. Common blocks /jjcf/, /mdat3/, and /mdat4/ contain the addresses of the new arrays created in FEAP. Common block /grad/ contains two logical flags, *flgf* and *flgr*. These flags control whether the structural response and its gradient are to be computed by FEAP or not. Their values are assigned in CALREL according to the kind of analysis that CALREL is performing. For example, during the search for the design point, the structural response needs to be computed at each iteration step, while the gradient may not be needed, depending on the optimization scheme used. On the other hand, when performing sensitivity analysis with respect to the deterministic parameters in the limit-state function, CALREL keeps the response fixed and requires computing the gradient only.

Two existing common blocks in CALREL have been passed to FEAP:

```
common /flag/ icl,igr,igf
common /prob/ ngf,ngp,nrx,ntp,nry,ntf
```

The first block contains the control parameters of CALREL, and the second block stores the size of the reliability problem.

Finally, two existing common blocks in FEAP have been extended:

```
common /fdata/ fl(11),pfr,folw
common /mdata/ nn,n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,m1
```

The new term *folw* in /fdata/ is a flag indicating whether the external loads are follower forces; the term *m1* in /mdata/ is the address of the first array allocated in the blank common by FEAP.

In addition to the parameters transferred through common blocks, the basic random variables and the current limit-state function number are passed from the user-defined routines UGFUN and UDGX to FEAP as arguments in the call statements.

B.1.3 File Numbers

Both CALREL and FEAP open several files to store problem data. Unfortunately, some of the file numbers used in these two programs were identical. To prevent the data files from mixing up, new file numbers have been assigned in subroutines FILNAM, PHIST, PMACR, PMACR3, PMAN, READER, and TAPER in FEAP.

B.1.4 New Arrays in CALREL and FEAP

Two new arrays have been created in CALREL. These arrays store the numbers of the elements whose stresses or strains are used in the limit-state functions and the starting addresses of these element numbers with respect to each limit-state function.

In FEAP, several new arrays have been created to save the matrices required in the computation of the gradient. Other arrays have been created as pointer arrays for the external loads, material properties, and structural geometry variables which are random and must be updated every time FEAP is called. These pointers indicate whether a load parameter, material property, or nodal coordinate is random, and, if it is, they assign the number of the associated random variable so that the current value of the variable can be determined for the running of FEAP (see Fig. B.1).

B.1.5 New and Modified Subroutines in FEAP

Several subroutines in FEAP have been modified and several new routines have been added. Basically, these subroutines can be classified into three groups. The first group of subroutines have been created or modified to incorporate the randomness in the structure and its environment. For instance, every time FEAP is called, a subroutine is executed to assign the current values of the random variables to the load parameters, material properties, and structural geometry. Another subroutine then generates new nodal coordinates using the new values of the geometry parameters. The second group of subroutines have been created or modified for the computation of the gradient. Subroutines that form the matrices $\frac{\partial \mathbf{R}}{\partial \mathbf{v}}\Big|_{\mathbf{U}}$, $\frac{\partial \mathbf{X}^d}{\partial \mathbf{v}_g}$, etc., belong to this group. The third group consists of new subroutines that are designed for the convenience of the user in formulating the user-defined subroutines UGFUN and UDGX. Two of these are called by UGFUN to extract displacements and stresses/stains from FEAP in order to compute the limit-state functions. Other routines are called in UDGX to extract the quantities used in the computation of the response gradient. These subroutines and their functions are listed in Table B.1. In addition to the three groups of subroutines mentioned above, a few facility subroutines have been written to handle frequently performed operations.

B.1.6 User-Defined Subroutines

As mentioned in Chapter 6, there are three subroutines in CALREL, namely, UGFUN, UDGX, and UDD, and one element routine in FEAP that should be developed by the user. The formats of UGFUN, UDGX, and UDD remains the same as in the original version of CALREL. However, subroutines UGFUN and UDGX, which serve as an interface between CALREL and FEAP, must contain certain statements to extract quantities computed by FEAP. The element routine of FEAP should be modified. Arrays associated with random material properties, nodal coordinates, and boundary tractions must be transferred to this routine. In addition, the routine should contain extra steps to form the

essential matrices if the response gradient is to be computed by analytical expressions. Details of these user-defined routines are described in the subsequent sections.

B.2 Input File of CALREL

To run **CALREL**, the user should provide an input file containing a list of macro commands that execute the required operations and the data that describe the problem or specify solution schemes.

B.2.1 Macro Commands of CALREL

The formats and functions of the macro command are listed below. Note that only the first four characters are required to identify a macro command. A command may be followed by a list of modifiers. All modifiers inside brackets have default values and are optional.

CALRel NRX=*nrx* [NGF=*ngf*] [NIG=*nig*] [NDP=*ntp*] [NMC=*nmc*] [NTL=*ntl*]

CALRel defines the size of a problem, where

ngf : number of limit-state functions, default = 1.

nig : number of independent groups of basic random variables, default = 1.

nrx : total number of basic random variables.

ntp : number of deterministic parameters in the limit-state functions, default=0.

nmc : number of minimum cut sets, default = *ngf*.

ntl : total number of components in all minimum cut sets, default = *nmc*.

This command should be the first line of the input file.

RESTart

RESTart restarts a previous CALREL run. All problem data are read from the files saved in the previous run. The user should skip command CALRel if RESTart is issued. This command should be the first line of the input file.

DATA

DATA reads the input data of a problem. The user should provide the problem data following this command. The format of the input data is described in the next section. This command can be skipped in a restart analysis.

END

END stops data input mode and evokes macro command mode.

FORM [INI=*ini*] [IST=*ist*] [IGF=*igf*] [NPR=*npr*]

FORM performs first-order reliability analyses on individual limit-state functions, where

ini : initialization flag, default = 0.

ini = −1, start from the design point of the previous run;

ini = 0, start from the mean point;

ini = 1, start from the point specified by the user.

ist : restart flag, default = 0.

ist = 0, analyze a new problem;

ist = 1, continue an unconverged problem.

igf : limit-state function to be analyzed, default = 1.

igf applies to component analysis only.

npr : print interval, default = 0.

npr < 0, print no first-order results;

npr = 0, print the final results of FORM;

npr > 0, print the results after every *npr* steps.

BOUNd [IBT=*ibt*]

BOUNd calculates the first-order failure probability bounds for series systems, where

ibt : type of bounds required, default = 0.

ibt = 1, unimodal bounds;

ibt = 2, relaxed bimodal bounds;

ibt = 3, bimodal bounds;

ibt = 0, all of the above.

Command FORM must have been executed before this command.

PNET [RHO=*rho*]

PNET estimates the first-order failure probability by the PNET method, where

rho : the threshold used in PNET method, default = 0.6.

Command FORM must have been executed before this command.

SENSitivity [ISC=*isc*] [ISV=*isv*] [IGF=*igf*]

SENSitivity performs first-order sensitivity analysis with respect ot the parameters defining the probability distributions and with respect to the deterministic parameters defining the

limit-state functions, where

isc : type of sensitivity analysis required, default = 1.

isc = 1, sensitivity analysis at component level;

isc = 2, sensitivity analysis for a series system.

isv : type of parameters for sensitivity analysis, default = 0.

isv = 1, sensitivity analysis with respect to distribution parameters;

isv = 2, sensitivity analysis with respect to deterministic parameters;

isv = 0, both the above.

igf : the same as in FORM.

Command FORM must have been executed before this command, and so must command BOUNd if *isc* = 2.

SORM [ISO=*iso*] [ITG=*itg*] [IGF=*igf*]

SORM performs second-order reliability analysis on individual limit-state functions, where

iso : type of second-order analysis to be performed, default = 1.

iso = 1, point-fitting method;

iso = 2, curvature-fitting method;

iso = 3, both the above methods.

itg : type of integration scheme used; default = 2 for the point-fitting method and default = 1 for the curvature-fitting method.

itg = 1, improved Breitung formula;

itg = 2, improved Breitung formula and Tvedt's double integral.

igf : the same as in FORM.

Command FORM must have been executed before this command.

DIRS [IFS=*ifs*] [IST=*ist*] [NSM=*nsm*] [NPR=*npr*] [COV=*cov*] [STP=*stp*]

DIRS performs directional simulation, where

ifs : type of surface used in simulation, default = 2 if SORM has been executed, otherwise default = 1.

ifs = 0, exact limit-state surface;

ifs = 1, first-order approximation surface;

ifs = 2, second-order approximation surface.

ist : restart flag, default = 0.

ist = 0, perform a new directional simulation;

ist = 1, continue a previous simulation.

nsm : total number of trials, default = 10000.

npr : print interval for simulation results, default = *nsm*/50.

cov : threshold for coefficient of variation of probability estimate, default = 0.05.

stp : seed for random number generation; should lie in the range [1,2147483647]. If not specified, *stp* is generated by the program.

Simulation stops at *nsm* trials or after the sample coefficient of variation falls below *cov*, whichever arrives first. If *ifs* = 1 (2), command FORM (SORM) must have been executed.

MONT [IST=*ist*] [NSM=*nsm*] [NPR=*npr*] [COV=*cov*] [STP=*stp*]

MONT performs the Monte Carlo simulation, where *ist*, *nsm*, *npr*, *cov*, and *stp* are the same as defined in DIRS.

EXIT

EXIT stops program execution.

B.2.2 Input Data of CALREL

The input data of **CALREL** is composed of several data sections. Each section starts with a title describing the type of data to follow. All input data are in free format. Data items are separated by spaces, commas, or tabs.

TITLe
n
text

TITLe reads the description of the problem, where

n : the number of text lines that follow.

text : *n* lines of text describing the problem.

FLAG
icl,igr

FLAG reads analysis flags, where

icl : problem type, default = 1.

icl = 1, component reliability;

icl = 2, series system reliability;

icl = 3, general system reliability.

igr : flag for gradient computation, default = 0.

igr = 0, gradient computed by finite difference;

igr = 1, gradient provided by the user in subroutine UDGX.

OPTImization
iop,ni1,ni2,tol,op1,op2,op3

OPTImization reads optimization parameters, where

iop : type of optimization scheme used, default = 2.

iop = 1, the HL-RF method;

iop = 2, the modified HL-RF method;

iop = 3, the gradient projection method;

iop = 4, the sequential quadratic programming method.

ni1 : maximum number of iteration cycles, default = 100, maximum = 100. [1]

ni2 : maximum steps in line search, default = 4.

tol : convergence tolerance, default = 0.001, minimum = 0.001. [1]

op1 : step size reduction factor in line search, default = 1.0 for *iop* = 1, default = 0.5 for *iop* = 2 or 3.

op2 : optimization parameter.

iop = 2, parameter c in the merit function, default = 10;

iop = 3, convergence tolerance for line search, default = *tol*.

op3 : optimization parameter.

iop = 3, Maximum step size in line search, default = 4.

STATistics[2]
igt,nge,ngm
xn,nv,id,p_1*,*p_2*,*p_3*,*p_4*,*x_0
ro
...

STATistics reads statistical data of the basic random variables. The variables are divided into *nig* statistically independent groups. For each group, the user should provide:

igt : group type.

igt = 1, all variables in the group are statistically independent;

igt = 2, variables described by their marginal distributions and correlation matrix;

igt = 3, variables described by conditional and/or marginal distributions.

nge : number of variables in the group.

ngm : number of variables in a type 3 group which are described by their marginal distributions.

For each of the nge variables, provide: [3]

xn : variable name. [4]

nv : variable number.

id : *ids* = |*id*| = distribution type.

ids = 0, deterministic;

ids = 1, normal;

ids = 2, lognormal;

ids = 3, gamma;

ids = 4, shifted exponential;

ids = 5, shifted Rayleigh;

ids = 6, uniform;

ids = 7, beta;

ids = 11, type-I largest value;

ids = 12, type-I smallest value;

ids = 13, type-II largest value;

ids = 14, type-III smallest value;

ids > 50, user-defined distributions.

p_1 : distribution parameter 1. [5]

id > 0, p_1 : mean value;

id < 0, p_1 : as defined in Table B.2.

p_2 : distribution parameter 2. [5]

id > 0, p_2 : standard deviation;

id < 0, p_2 : as defined in Table B.2.

p_3 : distribution parameter 3. [5]

p_4 : distribution parameter 4. [5]

x_0 : initial value of the variable, needed when *ini* = 1.

If *igt* = 2, or *igt* = 3 with *ngm* > 0, input the lower triangle of the correlation matrix for variables described by their marginal distributions; read after the distribution data of the variables. Otherwise, skip *ro* and continue to input the distribution data for the next group.

ro : lower triangle of the correlation matrix excluding the diagonals. Read it row-wise and in a triangular form.

Repeat the above input data for all groups.

PARAmeter
tp

PARAmeter reads the deterministic parameters defining the limit-state functions, where

tp : *ntp* deterministic parameters.

LIMIt
nesc
ig,nelm
elm
...

LIMIt reads the numbers of the elements whose components of stress or strain are used in the limit-state functions, where

nesc : total number of stress and strain components in an element.

ig: limit-state function number.

nelm : total number of elements whose stresses/strains are used in this limit-state function.

elm : element numbers of the above *nelm* elements.

ig, *nelm*, and *elm* should be repeated for all limit-state functions which are expressed in terms of the element stresses/strains. This data section must be ended with a blank line.

CUTSets[6]
mc

CUTSets reads minimum cut sets, where

mc : components of minimum cut sets. Use zero to indicate the end of each cut set. Use a negative sign to indicate the complement of a limit-state function. E.g., for minimum cut sets (1,2,4) and (-2,5,7), *mc* = 1 2 4 0 -2 5 7 0.

Notes:

(1) To override the restrictions $ni1 < 100$ and $tol > 0.001$, input negative *nil* and *tol* values. Their absolute values will be used in the analysis regardless of the limits.

(2) In a normal run without a restart option, all data sections can be skipped except STAT which must be input at least once.

(3) If a type 3 group contains variables described by marginal distributions, i.e. $ngm > 0$, then such variables must be defined before the variables described by conditional distributions.

(4) The first four characters of the line is treated as the variable name. The user should provide *nv*, *ids*, p_1, p_2, p_3, p_4, and x_0 from the fifth character on. No comma should

be inserted between *xn* and *nv*.

(5) For variable groups with *igt* = 3, entries for parameters p_1, p_2, p_3, and p_4 can be real or integer numbers. If real, then the parameter is assigned the entered value. If an integer ip_k is entered for parameter p_k, then p_k is set equal to the ip_k-th basic random variable in the group. The ip_k-th variable must have been defined before. This allows assignment of distributions whose parameters are themselves random variables.

(6) If CUTS is skipped in a system problem, then the system is treated as a series system.

(7) Only the first four characters are required to identify a data section type.

(8) The order of data sections is arbitrary.

B.2.3 An Example Input File of CALREL

```
CALR ngf=1 nig=3 nrx=6 ntp=1
DATA
TITLE
1
example input file
FLAG
1 0
OPTI
2 90 4 0.001 0 0 0
STAT
1 1
r    1,1,100.,20.,0.,0.,100.
2 2
v1   2,2,1000.,150.,0.,0.,810.
m1   3,-6,50.,100.,0.,0.,75.
```

```
0.5
3 3 2
v2    4,2,1000.,150.,0.,0.,810.
m2    5,-6,50.,100.,0.,0.,75.
q     6,51,4,5,10.,0.,810.
0.5
PARA
2.0
END
FORM ini=1 npr=1
SENS
SORM iso=1 itg=1
DATA
PARA
3.0
END
FORM ini=-1 npr=-1
SORM iso=1 itg=1
EXIT
```

B.3 Input File of FEAP

The input file of the subprogram FEAP consists of two parts. The first part is composed of data sections which define the control information and the mesh of the problem. The second part contains a list of macro commands defining the solution algorithm. Details of this input file is available in Ref. 78 and Section 6.3.

B.4 User-Defined Subroutines in CALREL

The formats and arguments of the three user-defined subroutines in CALREL are:

```
      subroutine ugfun(g,v,tp,ig)
      implicit double precision (a-h,o-z)
      common /prob/ngf,nig,nrv,ntp,nry,ntf
      dimension v(nrv),tp(ntp)
      go to (10,20,...) ig
10    g = ...
      return
20    g = ...
      return
      end

      subroutine udgx(v,dgv,tp,ig)
      implicit double precision (a-h,o-z)
      common /prob/ngf,nig,nrv,ntp,nry,ntf
      dimension v(nrv),dgv(nrv),tp(ntp)
      go to (10,20,...) ig
10    dgv(1) = ...
      ...
      return
20    dgv(1) = ...
      ...
      return
      end

      subroutine udd(v,par,sg,ids,cdf,pdf,bnd,ib)
      implicit double precision (a-h,o-z)
      common /prob/ngf,nig,nrv,ntp,nry,ntf
```

```
      dimension v(nrv),par(4),bnd(2)
      go to (10,20,...) ids-50
10    cdf = ...
      pdf = ...
      sg = ...
      ib  = ...
      bnd(1) = ...
      bnd(2) = ...
      return
20    ...
      return
      end
```

The definitions of the arguments of the above subroutine are as follows:

Argument	Definition
g	the value of the limit-state function
v	a vector of basic random variables
tp	deterministic parameters in the limit-state functions
ig	the number of the limit-state function whose value/gradient is computed
dgv	gradient of the limit-state function in the original space
par	parameters defining the probability distribution of the variable
sg	standard deviation of the variable
ids	distribution number of the variable
cdf	cumulative probability of the variable
pdf	probability density of the variable

bnd	lower/upper bounds of the distribution
ib	code of distribution bounds
	ib = 0, the distribution has no bounds
	ib = 1, the distribution has a lower bound
	ib = 2, the distribution has a upper bound
	ib = 3, the distribution has lower and upper bounds

The argument *sg* is used in computing derivatives and only an approximate estimate is needed. All real arguments of these subroutines must be in double precision.

In CALREL-FEAP, subroutine UGFUN is essentially composed of two sections: The first section checks if FEAP is to be executed, and the other computes the limit-state functions for a given set of values of the basic variables. A simple example of UGFUN is as follows:

```
      subroutine ugfun(g,v,tp,ig)
      implicit real*8 (a-h,o-z)
      logical flgf,flgr
      common /prob/ngf,nig,nrv,ntp,nry,ntf
      common /grad/ flgf,flgr
      dimension v(nrv),tp(ntp)
c---- check if FEAP is to be executed or not
      if(flgf) call feap(v,ig)
c---- switch to ig-th limit-state function
      go to (10,20), ig
10    g = tp(1) - getstr(1,133)
      return
20    g = tp(2) - getdis(1,13)
      return
```

```
end
```

In this example, the first failure criterion is expressed as the exceedance of the first component of the stress/strain vector of element 133 above the threshold tp(1), and the second criterion is expressed as the exceedance of the first component of the displacements at node 13 above the threshold tp(2). The double precision functions getstr(i,j) and getdis(i,j) respectively extract the i-th stress/strain component of element j and the i-th displacement component of node j from the results obtained in FEAP. The order of the stress/strain components is arbitrary and is specified by the user in the element routine. For a truss element, for example, one can define the first component as the axial force and the second component as the axial deformation, or vice versa.

For geometrically nonlinear, elastic structures, the gradient of the limit-state function is computed by the adjoint method to reduce the amount of computation. It was shown in Chapter 4 that the partial derivatives of $\boldsymbol{\epsilon}$, $\boldsymbol{\sigma}$, $\mathbf{R}$, and $\mathbf{P}$ are dependent on the element type. These are systematically computed in the element routine of FEAP. On the other hand, the partial derivatives of $g(.)$ are directly related to the particular limit-state function used and must be computed by expressions provided by the user. Therefore, in subroutine UDGX, the user should provide expressions for the partial derivatives of $g(.)$ and call appropriate subroutines to extract the essential matrices from FEAP. An example of UDGX for the limit-state functions shown in the above example of UGFUN is as follows (see Table B.1 and the steps listed in section 5.2 of Chapter 5 for clarity):

```
subroutine udgx(dgv,v,tp,ig)
implicit real*8 (a-h,o-z)
common mtot,np,m(1)
common /prob/ ngf,ngp,nrv,ntp,nry,ntf
common /cdata/ numnp,numel,numat,nen,neq,iprgx
common /mdata/ nn,n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,m1
dimension v(nrv),dgv(nrv),tp(ntp)
```

```
      save /prob/,/cdata/,/mdata/
      data nlam,ndsu,ndsv/0,0,0/
c---- allocate an neq×1 array for λ
c---- neq : number of degrees of freedom of the structure
      if(nlam.eq.0) call define('lamd',nlam,neq,1)
c---- switch to ig-th limit-state function
      go to (10,20), ig
c---- set ntot = total number of stress/strain components used in g(.)
10    ntot = 1
c---- allocate an ntot×neq array ∂σ/∂ε |v ∂ε/∂U |v
      if(ndsu.eq.0) call define('dsdu',ndsu,ntot,neq)
c---- allocate an ntot×nrv array for ∂σ/∂v |U + ∂σ/∂ε ∂ε/∂v |U
      if(ndsv.eq.0) call define('dsdv',ndsv,ntot,nrv)
c1 -- compute ∂g/∂σ ∂σ/∂ε |v ∂ε/∂U |v
c1.1 compute ∂g/∂σ
      dgds = -1.
c1.2 extract ∂σ/∂ε |v ∂ε/∂U |v from FEAP and store at m(ndsu)
      call dsdu(1,133,m(ndsu))
c1.3 compute the product ∂g/∂σ ∂σ/∂ε |v ∂ε/∂U |v and store at m(nlam)
      call mult(dgds,m(ndsu),m(nlam),1,1,neq)
c2 -- compute λ from λ^T K = ∂g/∂σ ∂σ/∂ε |v ∂ε/∂U |v
      call solvl(m(nlam))
c3 -- extract ∂P/∂v |U - ∂R/∂v |U from FEAP and compute λ^T [ ∂P/∂v |U - ∂R/∂v |U ]
```

call tldrv(dgv,m(nlam))

c4 -- compute $\frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial v}\Big|_v \frac{\partial \epsilon}{\partial v}\Big|_U$

c4.1 extract $\frac{\partial \sigma}{\partial v}\Big|_v \frac{\partial \epsilon}{\partial v}\Big|_U$ from FEAP

call dsdv(nste,nelm,m(ndsv))

c4.2 compute $\frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial v}\Big|_v \frac{\partial \epsilon}{\partial v}\Big|_U$

call mult(dgds,m(ndsv),m(ndsv),1,ntot,nrv)

c5 -- compute $\nabla_v g = \frac{\partial g}{\partial \sigma}\frac{\partial \sigma}{\partial v}\Big|_v \frac{\partial \epsilon}{\partial v}\Big|_U + \lambda^T \left[\frac{\partial P}{\partial v}\Big|_U - \frac{\partial R}{\partial v}\Big|_U\right]$

call addm(dgv,m(ndsv),dgv,1,nrv)

return

c1 -- compute $\frac{\partial g}{\partial U}$

20 call dgu(m(nlam),m(n7),2,neq)

c2 -- compute λ from $\lambda^T K = \frac{\partial g}{\partial U}$

call solvl(m(nlam))

c3 -- extract $\frac{\partial P}{\partial v}\Big|_U - \frac{\partial R}{\partial v}\Big|_U$ from FEAP and

c3 -- compute $\nabla_v g = \lambda^T \left[\frac{\partial P}{\partial v}\Big|_U - \frac{\partial R}{\partial v}\Big|_U\right]$

call tldrv(dgv,m(nlam))

return

end

subroutine dgu(dgdu,id,ndf,neq)

c dgdu : $\partial g/\partial U$

c id : equation pointer

c ndf : number of degrees of freedom at each node

c neq : number of degrees of freedom of the structure

```
      implicit real*8 (a-h,o-z)
      dimension dgdu(neq),id(ndf,1)
      do 10 i=1,neq
10    dgdu(i)=0.
      n=id(1,13)
      dgdu(n)=-1.
      return
      end
```

In this example, subroutine DGU computes $\frac{\partial g}{\partial U}$, *dgdu* is a vector storing $\frac{\partial g}{\partial U}$, and *dgds* is a vector storing $\frac{\partial g}{\partial \sigma}$. These vectors and the parameter *ntot* should be specified by the user. Note that in general subroutines DSDU and DSDV are executed *ntot* times in UDGX. The facility subroutines called in UDGX and their functions are listed Table B.1.

Subroutine UDD, which is used to define probability distributions not available in the CALREL library, does not change in finite-element reliability analysis. As a simple example, consider a basic random variable whose PDF is defined as

$$f_S(s) = \begin{cases} (s-8)/4 & 8 < s \leq 10 \\ (12-s)/4 & 10 < s \leq 12 \\ 0 & \textit{elsewhere} \end{cases} \tag{B.1}$$

Since this distribution is not in the library, it must be defined in the subroutine UDD.

```
      subroutine udd(v,par,sg,ids,cdf,pdf,bnd,ib)
      implicit double precision (a-h,o-z)
      common /prob/ngf,nig,nrv,ntp,nry,ntf
      dimension v(nrv),par(4),bnd(2)
c.... par(1)=8.0, par(2)=10.0, par(3)=12.0.
      s = v(2)
      if(s.le.par(1)) then
```

```
  cdf = 0.d0
  pdf = 0.d0
else if(s.le.par(2)) then
  cdf = s*s/8.d0 - 2.d0*s + 8.d0
  pdf = ( s - 8.d0 ) / 4.d0
else if(s.le.par(3)) then
  cdf = 3.d0*s - s*s/8.d0 - 17.d0
  pdf = ( 12.d0 - s ) / 4.d0
else
  cdf = 1.d0
  pdf = 0.d0
endif
sg = 0.8165
bnd(1) = par(1)
bnd(2) = par(3)
ib = 3
return
end
```

B.5 Element Routine in FEAP

In finite-element reliability analysis, in general the element routine for a particular element may have to include the following steps:

1. Define the element parameters; e.g., the constitutive parameters, the density, etc. (isw=1);
2. Check features for each element (isw=2)
3. Compute the tangent stiffness matrix and/or the residual force (isw=3);
4. Output the element stresses/strains (isw=4);

5. Compute the mass matrix (isw=5);
6. Compute the residual forces (isw=6);
7. Compute the boundary loads (isw=7);
8. Output the nodal stresses/strains (isw=8);
9. Compute the damping matrix (isw=9);
10. Calculate error indicators (isw=10);
11. Compute the gradient of the surface loads with respect to the load parameters (isw=11);
12. Compute the gradient of the surface loads with respect to the nodal coordinates (isw=12);
13. Compute the gradient of the resisting forces with respect to the material parameters (isw=13);
14. Compute the gradient of the resisting forces with respect to the nodal coordinates (isw=14);
15. Compute the gradient of the stresses/strains with respect to the material parameters (isw=15);
16. Compute the gradient of the stresses/strains with respect to the nodal coordinates (isw=16);
17. Compute the gradient of the stresses/strains with respect to the nodal displacements (isw=17).

In CALREL-FEAP, the execution of the above steps is controlled by the argument *isw* passed to the element routine. Not all the seventeen steps are necessary in every element routine. The steps to be included in each particular case depend on the type of problem at hand and the method of solution used. For instance, steps 5 and 9 are not necessary in a static problem. If the element routine is available in the element library of FEAP, usually steps 1 to 10 already exist. These steps are adequate if the response gradient is to be

computed by a finite difference scheme. Steps 11 to 17 are required only if the response gradient is to be computed using analytical expressions such as those in Chapter 4. These gradients are not available in FEAP and must be provided by the user. In the current versions of CALREL-FEAP, the gradients for geometrically nonlinear structures are available for elastic truss and 2D elements.

As an example, the element routine for the geometrically nonlinear truss element in Example 7.1 is listed below. Note that many steps in the element routine are omitted because they are not used in the analysis. For instance, the steps to compute the gradients of strains and stresses are not included because the limit-state function is in terms of displacements.

```
      subroutine elmt01(d,u,x,ix,t,s,p,ndf,ndm,nst,isw,
     *                  brv,jm,pjm,nimp,sts,m)
      implicit double precision (a-h,o-z)
c.... any dimensional truss element routine
c.... finite deformation
c.... linear elastic
      character*4 o,head
      logical fl,pfr,folw
      common /bdata/  o,head(20)
      common /cdata/  numnp,numel,nummat,nen,neq,ipr
      common /fdata/  fl(11),pfr,folw
      common /mdat4/  nes0,les0,nsts,ndeb,ndeu,nesc,nele,ndrb,ndxc
      common /eldata/ dm,n,ma,mct,iel,nel
      common /iofile/ ifile,jfile
      save /bdata/,/cdata/,/fdata/,/mdat4/,/eldata/,/iofile/
      dimension d(1),u(ndf,1),ix(1),x(ndm,1),t(1),s(nst,nst),p(1)
      dimension brv(1),jm(19),pjm(18),nimp(2,1),sts(1),m(1)
```

```
      dimension db(3),dx(3),xx(3),dx0(3)
      if(isw.eq.-1) go to 110
      go to (1,2,3,4,5,4,7,2,9,2,11,12,4,14,15,16),isw
c.... input material properties
1     call readln(pjm,jm,'rd',2,brv,nimp,1,nnrv)
110   d(1) = pjm(1)
      d(2) = pjm(2)
      jm(19) = 1
      if(isw.gt.0) write(jfile,2000) d(1),d(2)
      d(4) = d(1)*d(2)
c.... set initial values for non-printed coordinates
      do 120 i = 1,3
        xx(i) = 0.0
120   continue
2     return
c.... compute element stiffness matrix
3     xl = 0.0
      xl0 = 0.0
      dd = d(4)
      do 31 i = 1,ndm
        dx0(i) = x(i,2) - x(i,1)
        dx(i) = dx0(i) + u(i,2) - u(i,1)
        db(i) = dd * dx(i)
        xl = xl + dx(i)**2
        xl0 = xl0 + dx0(i)**2
31    continue
      xlc = xl0*dsqrt(xl0)
      do 32 i = 1,ndm
```

```
      dx(i) = dx(i) / xlc
32    continue
      ess = xl - xl0
      pp = dd*ess/2.d0/xlc
      i1 = 0
      do 37 ii = 1,2
       j1 = i1
       do 36 jj = ii,2
        do 34 i = 1,ndm
        do 33 j = 1,ndm
         s(i+i1,j+j1) = db(i)*dx(j)
33      continue
         s(i+i1,i+j1) = s(i+i1,i+j1) + pp
34      continue
        j1 = j1 + ndf
        pp = -pp
        do 35 j = 1,ndm
         dx(j) = -dx(j)
35      continue
36     continue
       i1 = i1 + ndf
37    continue
c.... construct symmetric part
      do 38 i = 1,ndm
      do 38 j = 1,ndm
       s(i+ndf,j) = s(j,i+ndf)
38    continue
c.... form a residual
```

```
      pp = pp*xlc
      do 39 i = 1,ndf
       p(i) =  pp*dx(i)
       p(i+ndf) = -pp*dx(i)
39    continue
      return
c.... isw=4 : output stress and strain in element
c.... isw=6 : compute residual force
c.... isw=13: compute ∂(residual force)/∂(material properties)
4     xl = 0.0
      xl0=0.0
      do 41 i = 1,ndm
       dx0(i) = x(i,2) - x(i,1)
       dx(i) = dx0(i) + u(i,2) - u(i,1)
       xl = xl + dx(i)**2
       xl0 = xl0 + dx0(i)**2
       xx(i) = (x(i,2) + x(i,1))/2.
41    continue
      ess = (xl - xl0)/2.d0/xl0
      sig = d(4)*ess
      xl0 = dsqrt(xl0)
      if (isw.eq.4) then
c.... print the stress/strain values for the element
       sts(1)=sig
       sts(2)=ess
       mct = mct - 1
       if(mct.le.0) then
        write(jfile,2001) o,head
```

```
        mct = 50
       endif
       write(jfile,2002) n,dm,ma,xx,sig,ess
c.... compute right hand side
      else if(isw.eq.6) then
       sig = sig / xl0
       do 43 i = 1,ndf
        p(i    ) = dx(i)*sig
        p(i+ndf) = -p(i)
43     continue
      else
       if(jm(1).eq.0) return
       sig = d(2)*ess/xl0
       do 44 i = 1,ndf
        p(i    ) = -dx(i)*sig
        p(i+ndf) = -p(i)
44     continue
      endif
      return
c.... compute element mass matrices
5     return
c.... compute surface tractions
7     return
c.... form consistent damping matrix
9     return
c.... error indicator calculation
10    return
c.... compute ∂(surface load)/∂(force brv)
```

```
11    return
c.... compute ∂(surface load)/∂(nodal coordinates)
12    return
c.... compute ∂(resisting force)/∂(nodal coordinates)
14    call drdx01(d,x,u,ndf,ndm,nst,p)
      return
c.... compute ∂(stress,strain)/∂(material properties)
15    return
c.... compute ∂(stress,strain)/∂(nodal coordinates)
16    return
c.... compute ∂(stress,strain)/∂(displacements)
17    return
c.... formats
1000 format(8f10.0)
2000 format(5x,'truss element'/
     1      5x,'          modulus = ',e13.4,/,
     2      5x,'          area    = ',e13.4,//)
2001 format(a1,20a4//
     +       9x,'truss element'//
     +       3x,'element  material',6x,'1-coord',6x,'2-coord',
     +       6x,'3-coord',6x,' force',9x,'strain')
2002 format(i10,2i5,3f13.4,2e16.5)
      end

      subroutine drdx01(d,x,u,ndf,ndm,nst,drx)
      implicit double precision (a-h,o-z)
c.... compute ∂R/∂X,  R: internal force, X: coordinates
      dimension d(1),x(ndm,1),u(ndf,1),drx(nst,1),dx(3),dx0(3),du(3)
```

```
c.... compute current configuration for follower force
      xl = 0.0
      xl0 = 0.0
      do 20 i=1,ndm
      dx0(i) = x(i,2) - x(i,1)
      du(i) = u(i,2) - u(i,1)
      dx(i) = dx0(i) + du(i)
      xl = xl + dx(i)**2
      xl0 = xl0 + dx0(i)**2
20    continue
      ess = (xl - xl0)/2.d0/xl0
      sig = d(4)*ess
      xl0 = dsqrt(xl0)
      xlc = xl0**3
      t1 = -3.d0*sig/xlc
      t2 = d(4)/xlc
      t3 = sig/xl0
      do 40 i = 1,ndf
      do 30 j = 1,ndm
30      drx(i,j) = t1*dx(i)*dx0(j) + t2*dx(i)*du(j)
40      drx(i,i) = drx(i,i)+t3
      do 50 i = 1,ndf
      do 50 j = 1,ndm
        drx(i,j+ndm) = -drx(i,j)
        drx(i+ndf,j) = -drx(i,j)
        drx(i+ndf,j+ndm) = drx(i,j)
50    continue
      return
```

```
end
```

Table B.1 Facility Subroutines in CALREL-FEAP

Subroutine	Function
feap(v,ig)	executes subprogram FEAP
getdis(i,j)	extracts the i-th displacement component of node j from FEAP
getstr(i,j)	extracts the i-th stress/strain component of element j from FEAP
define(ar,na,nr,nc)	allocates a real $nr \times nc$ matrix named ar and return the address na
addm(**a**,**b**,**c**,nr,nc)	computes the $nr \times nc$ matrix $\mathbf{c} = \mathbf{a} + \mathbf{b}$
mult(**a**,**b**,**c**,na,nb,nc)	computes $\mathbf{c} = \mathbf{a}\,\mathbf{b}$, where **a** is an $na \times nb$ matrix and **b** is an $nb \times nc$ matrix
solvl(**a**)	solves for λ from $\lambda^T \mathbf{K} = \mathbf{a}$ and returns λ in array **a**
tldrv(**a**,λ)	computes $\mathbf{a} = \lambda^T \left[\frac{\partial \mathbf{P}}{\partial \mathbf{v}} \Big\|_{\mathbf{U}} - \frac{\partial \mathbf{R}}{\partial \mathbf{v}} \Big\|_{\mathbf{U}} \right]$
dsdu(i,j,**a**)	extracts $\frac{\partial \sigma}{\partial \epsilon} \Big\|_{\mathbf{v}} \frac{\partial \epsilon}{\partial \mathbf{U}} \Big\|_{\mathbf{v}}$ for the i-th stress/strain component of element j from FEAP and stores in **a**
dsdv(i,j,**a**)	extracts $\frac{\partial \sigma}{\partial \mathbf{v}} \Big\|_{\mathbf{v}} \frac{\partial \epsilon}{\partial \mathbf{v}} \Big\|_{\mathbf{U}}$ for the i-th stress/strain component of element j from FEAP and stores in **a**

Table B.2 CALREL Probability Distribution Library

Name	Type	PDF, $f_v(v)$	CDF, $F_v(v)$	Parameters				Note
				P_1	P_2	P_3	P_4	
Normal	1	$\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{v-\mu}{\sigma}\right)^2\right]$	$\Phi\left[\frac{v-\mu}{\sigma}\right]$	μ	$0<\sigma$			1
Lognormal	2	$\frac{1}{\sqrt{2\pi}\zeta v}\exp\left[-\frac{1}{2}\left(\frac{\ln v-\lambda}{\zeta}\right)^2\right]$, $0<v$	$\Phi\left[\frac{\ln v-\lambda}{\zeta}\right]$	λ	$0<\zeta$			1
Gamma	3	$\frac{\lambda(\lambda v)^{k-1}}{\Gamma(k)}\exp(-\lambda v)$, $0\le v$	$\frac{\Gamma(k,\lambda v)}{\Gamma(k)}$	$0<\lambda$	$0<k$			2,3
Shifted Exponential	4	$\lambda\exp[-\lambda(v-v_0)]$, $v_0\le v$	$1-\exp(-\lambda(v-v_0))$	$0<\lambda$	v_0			
Shifted Rayleigh	5	$\frac{(v-v_0)}{\alpha^2}\exp\left[-\frac{1}{2}\left(\frac{v-v_0}{\alpha}\right)^2\right]$, $v_0\le v$	$1-\exp\left[-\frac{1}{2}\left[\frac{v-v_0}{\alpha}\right]^2\right]$	α	v_0			
Uniform	6	$\frac{1}{b-a}$, $a\le v\le b$	$\frac{v-a}{b-a}$	a	b			
Beta	7	$\frac{(v-a)^{q-1}(b-v)^{r-1}}{B(q,r)(b-a)^{q+r-1}}$, $a\le v\le b$		$0<q$	$0<r$	a	b	4
Type I Largest Value	11	$\alpha_u\exp[-\alpha_u(v-u_u)-\exp(-\alpha_u(v-u_u))]$	$\exp[-\exp(-\alpha_u(v-u_u))]$	u_u	$0<\alpha_u$			
Type I Smallest Value	12	$\alpha_1\exp[\alpha_1(v-u_1)-\exp(\alpha_1(v-u_1))]$	$1-\exp[-\exp(\alpha_1(v-u_1))]$	u_1	$0<\alpha_1$			
Type II Largest Value	13	$\frac{k}{u_u}\left(\frac{u_u}{v}\right)^{k+1}\exp\left[-\left(\frac{u_u}{v}\right)^k\right]$, $0<v$	$\exp\left[-\left(\frac{u_u}{v}\right)^k\right]$	u_u	$0<k$			5
Type III Smallest Value	14	$\frac{k}{u_1-\epsilon}\left(\frac{v-\epsilon}{u_1-\epsilon}\right)^{k-1}\exp\left[-\left(\frac{v-\epsilon}{u_1-\epsilon}\right)^k\right]$, $\epsilon\le v$	$1-\exp\left[-\left(\frac{v-\epsilon}{u_1-\epsilon}\right)^k\right]$	u_1	$0<k$			6

Notes

1. $\Phi(v) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{v} e^{-\frac{u^2}{2}} du$ is the standard normal cumulative probability.

2. $\Gamma(k) = \int_{0}^{\infty} e^{-u} u^{k-1} du$ is the gamma function. For integer k, $\Gamma(k) = (k-1)!$.

3. $\Gamma(k,v) = \int_{0}^{v} e^{-u} u^{k-1} du$ is the incomplete gamma function. Note that $\Gamma(k) = \Gamma(k,\infty)$.

   The incomplete gamma function is available in most mathematical software libraries.

4. $B(q,r) = \frac{\Gamma(q)\Gamma(r)}{\Gamma(q+r)}$ is the beta function.

5. This distribution is also known as the *Gumbel* distribution.

6. This distribution is also known as the *Weibull* distribution.

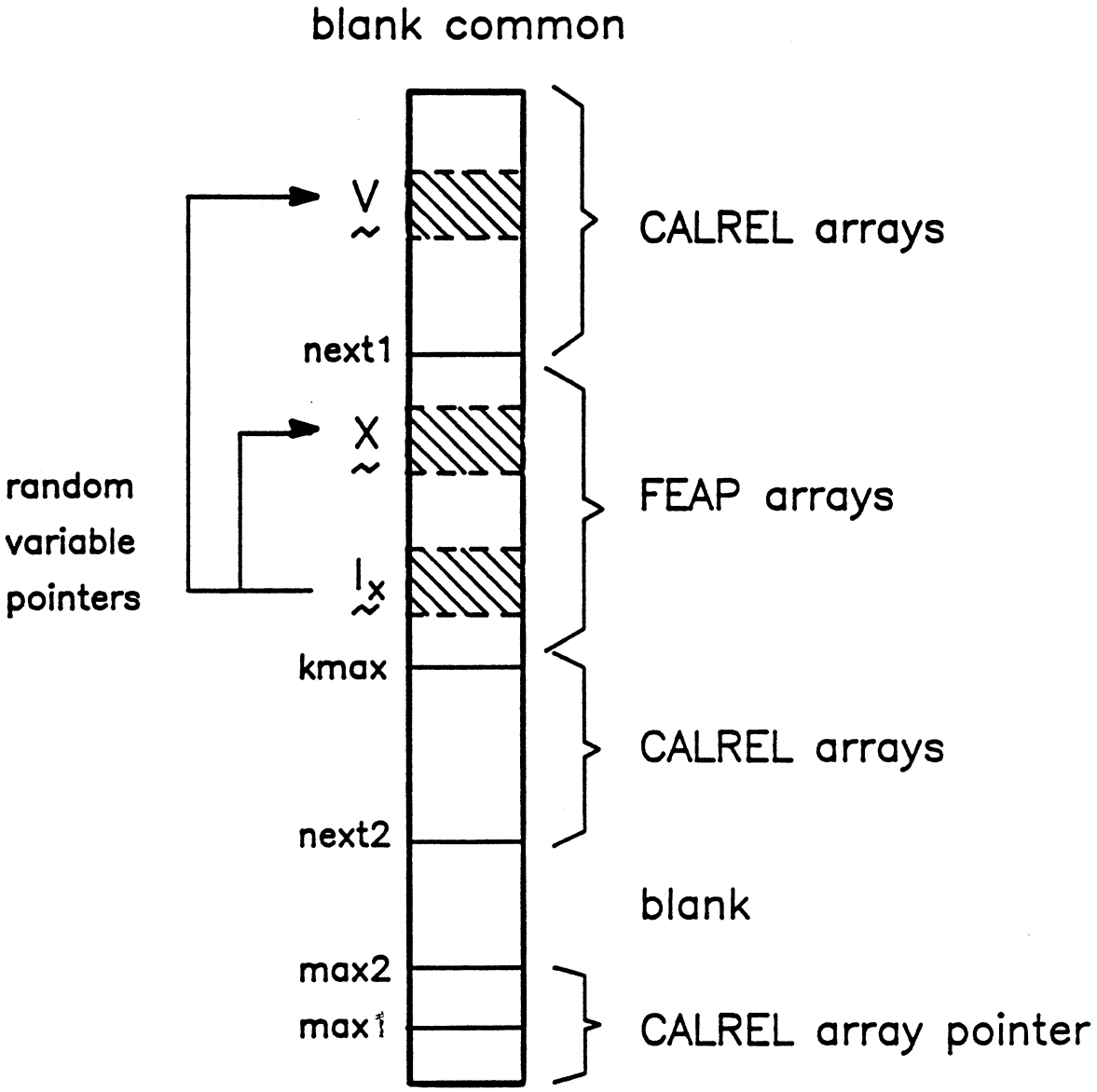


Figure B.1 CALREL-FEAP Memory Allocation

References

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

2. R. Araya, "Seismic Hazard Analysis: Improved Models, Uncertainties and Sensitivities," thesis presented to the Department of Civil Engineering University of California, Berkeley, CA in partial fulfillment of the requirements for the degree of Doctor of Philosophy, March 1988..

3. T. Arnbjerg-Nielsen and P. Bjerager, "Finite Element Reliability Method with Improved Efficiency by Sensitivity Analysis," in *Proceedings*, Joint ASME/SES Applied Mechanics and Engineering Sciences Conference, vol. ADM-93, pp. 15-26, Berkeley, CA, June 1988.

4. J. S. Arora and E. J. Haug, "Methods of Design Sensitivity Analysis in Structural Optimization," *AIAA Journal*, vol. 17, no. 9, pp. 970-974, 1979.

5. C. J. Astill, S. B. Nosseir, and M. Shinozuka, "Impact Loading on Structures with Random Properties," *Journal of Structural Mechanics*, vol. 1, no. 1, pp. 63-77, 1972.

6. G. B. Baecher and T. S. Ingra, "Stochastic FEM in Settlement Predictions," *Journal of Geotechnical Engineering*, vol. 107, no. 4, pp. 449-463, ASCE, April 1981.

7. K. Breitung, "Asymptotic Approximations for Multinormal Integrals," *Journal of Engineering Mechanics*, vol. 110, no. 3, pp. 357-366, ASCE, March 1984.

8. J. D. Collins and W. T. Thomson, "The Eigenvalue Problem for Structural Systems with Statistical Properties," *AIAA Journal*, vol. 7, no. 4, pp. 642-648, 1969.

9. K. Dems and Z. Mróz, "Variational Approach by Means of Adjoint Systems to Structural Optimization and Sensitivity Analysis - I, Variation of Material Parameters within Fixed Domain," *International Journal of Solids and Structures*, vol. 19, no. 8, pp. 677-692, 1983.

10. K. Dems and Z. Mróz, "Variational Approach by Means of Adjoint Systems to Structural Optimization and Sensitivity Analysis - II, Structure Shape Variation," *International Journal of Solids and Structures*, vol. 20, no. 6, pp. 527-552, 1984.

11. B. A. Dendrou and E. N. Houstis, "An Inference-Finite Element Model for Field Applications," *Applied Mathematical Modeling*, vol. 1, IPC Science and Technology Publications, Guildford, England, 1978.

12. B. A. Dendrou and E. N. Houstis, "Uncertainty Finite Element Dynamic Analysis," Report CSD-TR 271, Department of Civil Engineering, Purdue University, West Lafayette, Indiana, July 1978.

13. A. Der Kiureghian, "Finite Element Methods in Structural Safety Studies," in *Proceedings*, ASCE-STD Symposium on Structural Safety Studies, ASCE Convention, pp. 40-52, Denver, Colorado, May 1985.

14. A. Der Kiureghian, "Multivariate Distribution Models for Structural Reliability," in *Transactions*, the 9th International Conference on Structural Mechanics in Reactor Technology, vol. M, pp. 373-379, Lausanne, Switzerland, Aug. 1987.

15. A. Der Kiureghian and B.-J. Ke, "The Stochastic Finite Element Method in Structural Reliability," *Probabilistic Engineering Mechanics*, vol. 3, no. 2, pp. 83-91, 1988.

16. 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, vol. I, pp. 395-404, Kobe, Japan, May 1985.

17. A. Der Kiureghian, H-Z. Lin, and S-J. Hwang, "Second-Order Reliability Approximations," *Journal of Engineering Mechanics*, vol. 113, no. 8, pp. 1208-1225, ASCE, Aug. 1987.

18. A. Der Kiureghian and P-L. Liu, "Structural Reliability Under Incomplete Probability Information," *Journal of Engineering Mechanics*, vol. 112, no. 1, pp. 721-740, ASCE, Jan. 1986.

19. A. Der Kiureghian and R. L. Taylor, "Numerical Methods in Structural Reliability," in *Proceedings*, Fourth International Conference on Applications of Statistics and Probability in Soil and Structural Engineering, pp. 769-784, Florence, Italy, June 1983.

20. J. B. Dias and J. C. Nagtegaal, "Efficient Algorithms for Use in Probabilistic Finite Element Analysis," *Advances in Aerospace Structural Analysis*, no. AD-09, pp. 37-50, ASME, 1985.

21. J. B. Dias and S. Nakazawa, "An Approach to Probabilistic Finite Element Analysis Using a Mixed-Iterative Formulation," in *Proceedings*, Joint ASME/SES Applied Mechanics and Engineering Sciences Conference, vol. ADM-93, pp. 75-86, Berkeley, CA, June 1988.

22. O. Ditlevsen, "Narrow Reliability Bounds for Structural Systems," *Journal of Structural Mechanics*, vol. 7, no. 4, pp. 453-472, Dec. 1979.

23. B. Fiessler, H-J. Neumann, and R. Rackwitz, "Quadratic Limit States in Structural Reliability," *Journal of Engineering Mechanics*, vol. 105, no. 4, pp. 661-676, ASCE, Aug. 1979.

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

25. M. Grigoriu, "Crossings of Non-Gaussian Translation Processes," *Journal of Engineering Mechanics*, vol. 110, no. 4, pp. 610-620, ASCE, April 1984.

26. F. Guers and R. Rackwitz, "Time-Variant Reliability of Structural Systems Subject to Fatigue," in *Reliability and Risk Analysis in Civil Engineering, Proceedings of ICASP5*, ed. N. C. Lind, vol. 1, pp. 495-505, Institute for Risk Research, University of Waterloo, Ontario, Canada, 1987.

27. K. Handa and K. Anderson, "Application of Finite Element Methods in the Statistical Analysis of Structures," in *Proceedings*, Third International Conference on Structural Safety and Reliability, pp. 409-417, Trondheim, Norway, June 1981.

28. A. M. Hasofer and N. C. Lind, "Exact and Invariant Second-Moment Code Format," *Journal of Engineering Mechanics*, vol. 100, no. 1, pp. 111-121, ASCE, Feb. 1974.

29. T. Hisada and S. Nakagiri, "A Note on Stochastic Finite Element Method (Part 3) -- An Extension of the Methodology to Nonlinear Problems," *Seisan-Kenkyu*, vol. 32, no. 12, pp. 14-17, Institute of Industrial Science, University of Tokyo, Tokyo, Japan, Dec. 1980.

30. T. Hisada and S. Nakagiri, "Stochastic Finite Element Method Developed for Structural Safety and Reliability," in *Proceedings*, 3rd International Conference on Structural Safety and Reliability, pp. 395-408, Trondheim, Norway, June 1981.

31. T. Hisada and S. Nakagiri, "Role of the Stochastic Finite Element Method in Structural Safety and Reliability," in *Proceedings*, 4th International Conference on Structural Safety and Reliability, pp. 385-394, Kobe, Japan, May 1985.

32. T. Hisada, S. Nakagiri, and T. Nagasaki, "Stochastic Finite Element Analysis of Uncertain Intrinsic Stresses Caused by Structural Misfits," in *Transactions*, Seventh International Conference on Structural Mechanics in Reactor Technology, vol. M, pp. 199-206, Chicago, Illinois, Aug. 1983.

33. M. Hohenbichler and R. Rackwitz, "Improvement of Second-Order Reliability Estimates by Importance Sampling," to appear in *Journal of Engineering Mechanics*, ASCE.

34. M. Hohenbichler and R. Rackwitz, "Non-Normal Dependent Vectors in Structural Safety," *Journal of Engineering Mechanics*, vol. 107, no. 6, pp. 1227-1238, ASCE, Dec. 1981.

35. T. Igusa and A. Der Kiureghian, "Response of Uncertain Systems to Stochastic Excitation," *Journal of Engineering Mechanics*, vol. 114, no. 5, pp. 812-832, ASCE, May 1988.

36. M. Lawrence, "Basis Random Variables in Finite Element Analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 1849-1863, 1987.

37. H-Z. Lin and A. Der Kiureghian, "Second-Order System Reliability Using Directional Simulation," in *Reliability and Risk Analysis in Civil Engineering, Proceedings of ICASP5*, ed. N. C. Lind, vol. 2, pp. 930-937, Institute for Risk Research, University of Waterloo, Ontario, Canada, 1987.

38. P-L. Liu and A. Der Kiureghian, "Optimization Algorithms for Structural Reliability Analysis," Report No. UCB/SESM-86/09, Department of Civil Engineering, Division of Structural Engineering and Structural Mechanics, University of California, Berkeley, CA, Sept. 1986.

39. P-L. Liu and A. Der Kiureghian, "Reliability of Geometrically Nonlinear Structures," in *Proceedings*, ASCE-EMD/GTD/STD Joint Specialty Conference on Probabilistic Methods, pp. 164-168, Blacksburg, VA, May 1988.

40. P-L. Liu and A. Der Kiureghian, "Optimization Algorithms for Structural Reliability," in *Proceedings*, Joint ASME/SES Applied Mechanics and Engineering Sciences Conference, pp. 185-196, Berkeley, CA, June 1988.

41. W. K. Liu, T. Belytschko, and A. Mani, "Probabilistic Finite Elements for Nonlinear Structural Dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 56, pp. 61-88, 1986.

42. W. K. Liu, T. Belytschko, and A. Mani, "Random Field Finite Elements," *International Journal for Numerical Methods in Engineering*, vol. 23, pp. 1831-1845, 1986.

43. W. K. Liu, G. Besterfield, M. Lawrence, and T. Belytschko, "Kuhn-Tucker Optimization Based Reliability Analysis for Probabilistic Finite Elements," in *Proceedings*, Joint ASME/SES Applied Mechanics and Engineering Sciences Conference, vol. ADM-93, pp. 135-150, Berkeley, CA, June 1988.

44. M. Loeve, *Probability Theory, 4th edition*, Springer-Verlag, New York, 1977.

45. F. A. Lootsma, "Comparative Performance Evaluation, Experimental Design, and Generation of Test Problems in Nonlinear Optimization," in *Computational Mathematical Programming*, NATO ASI series, Series F, Computer and System Sciences, vol. 15, pp. 249-260, Springer-Verlag, Berlin, 1984.

46. D. G. Luenberger, *Linear and Nonlinear Programming, 2nd edition,* Addison-Wesley, Reading, Massachusetts, 1984.

47. H. O. Madsen, "Fast Integration for Time Variant Reliability," A/S Veritas Research Report, Hovik, Norway, 1986.

48. H. O. Madsen, "Omission Sensitivity Factors," *Structural Safety*, vol. 5, pp. 335-345, 1988.

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

50. L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium,* Prentice-Hall, Englewood Cliffs, N.J., 1969.

51. S. Nakagiri and T. Hisada, "A Note on Stochastic Finite Element Method (Part 6) -- An Application in Problems of Uncertain Elastic Foundation," *Seisan-Kenkyu*, vol. 35, no. 1, pp. 20-23, Institute of Industrial Science, University of Tokyo, Tokyo, Japan, Jan. 1983.

52. A. Nataf, "Determination des Distribution dont les Marges Sont Donees," *Comptes Rendus de l'Academie des Sciences*, vol. 225, pp. 42-43, Paris, 1962.

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

54. G. Righetti and K. Harrop-Williams, "Finite Element Analysis of Random Soil Media," *Journal of Geotechnical Engineering*, vol. 114, no. 1, pp. 59-75, ASCE, Jan. 1988.

55. J. Rosen, "The Gradient Projection method for Nonlinear Programming, Part II.

Nonlinear Constraints," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, pp. 514-532, 1961.

56. M. Rosenblatt, "Remarks on a Multivariate Transformation," *Annals of Mathematical Statistics*, vol. 23, pp. 470-472, 1952.

57. Y. S. Ryu, M. Haririan, C. C. Wu, and J. S. Arora, "Structural Design Sensitivity Analysis of Nonlinear Response," *Computers & Structures*, vol. 21, no. 1/2, pp. 245-255, June 1985.

58. K. Schittkowski, "On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function," *Mathematische Operationsforschung und Statistik, Ser. Optimization*, vol. 14, pp. 197-216, 1983.

59. I. E. Segal, "Fiducial Distribution of Several Parameters with Applications to a Normal System," in *Proceedings of Cambridge Philosophical Society*, vol. 34, pp. 41-47, 1938.

60. M. Shinozuka, "Basic Analysis of Structural Safety," *Journal of Structural Engineering*, vol. 109, no. 3, pp. 721-740, ASCE, March 1983.

61. M. Shinozuka and G. Dasgupta, "Stochastic Finite Element Methods in Dynamics," in *Proceedings*, 3rd ASCE EMD Specialty Conference on Dynamic Response of Structures, pp. 44-54, University of California at Los Angeles, Los Angeles, CA, 1986.

62. M. Shinozuka and E. Lenoe, "A probabilistic Model for Spatial Distribution of Material Properties," *Engineering Fracture Mechanics*, vol. 8, pp. 217-227, 1976.

63. P. D. Spanos and R. Ghanem, "Stochastic Finite Element Expansion for Random Media," Report NCEER-88-0005, March 1988.

64. G. Strang, *Linear Algebra and Its Applications*, Academic Press, New York, 1980.

65. S. Tani, S. Nakagiri, and T. Higashino, "Assessment of the Reliability Indices of CFRP Laminated Plate," in *Proceedings*, Joint ASME/SES Applied Mechanics and Engineering Sciences Conference, vol. ADM-93, pp. 27-36, Berkeley, CA, June 1988.

66. J. Todd, *Survey of Numerical Analysis*, McGraw-Hill, New York, N.Y., 1962.

67. L. Tvedt, "Two Second-Order Approximations to the Failure Probability," A/S Veritas Research Report, Hovik, Norway, 1984.

68. L. Tvedt, "On the Probability Content of a Parabolic Failure Set in a Space of Independent Standard Normally Distributed Random Variables," A/S Veritas Research Report, Hovik, Norway, 1985.

69. L. Tvedt, "Second Order Probability by an Exact Integral," A/S Veritas Research Report, Hovik, Norway, 1988.

70. E. H. Vanmarcke, "Probabilistic Modeling of Soil Profiles," *Journal of Geotechnical Engineering*, vol. 103, no. 11, pp. 1227-1246, ASCE, Nov. 1977.

71. E. H. Vanmarcke, *Random Fields: Analysis and Synthesis*, MIT Press, Cambridge, Mass., 1983.

72. E. H. Vanmarcke and M. Grigoriu, "Stochastic Finite Element Analysis of Simple Beams," *Journal of Engineering Mechanics*, vol. 109, no. 5, pp. 1203-1214, ASCE, Oct. 1983.

73. Y. K. Wen and H.-C. Chen, "On Fast Integration for Time Variant Structural Reliability," *Probabilistic Engineering Mechanics*, vol. 2, no. 3, pp. 156-162, Sept. 1987.

74. C. C. Wu and J. S. Arora, "Design Sensitivity Analysis and Optimization of Nonlinear Structural Response Using Incremental Procedure," *AIAA Journal*, vol. 25, no. 8, pp. 1118-1125, 1987.

75. F. Yamazaki and M. Shinozuka, "Digital Generation of Non-Gaussian Stochastic Fields," *Journal of Engineering Mechanics*, vol. 114, no. 7, ASCE, July 1988.

76. F. Yamazaki, M. Shinozuka, and G. Dasgupta, "Neumann Expansion for Stochastic Finite Element Analysis," *Journal of Engineering Mechanics*, vol. 114, no. 8, pp. 1335-1354, ASCE, Aug. 1988.

77. W. I. Zangwill, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Engle-

wood Cliffs, N.J., 1969.

78. O. C. Zienkiewicz, *The Finite Element Method,* McGraw-Hill, New York, N.Y., 1977.