Title

Safety Index by First-Order Second-Moment Reliability Method

Authors
Kitagawa, Makoto
Der, Kiureghian

Publication Date
1980-12-01

SAFETY INDEX BY FIRST-ORDER SECOND-MOMENT RELIABILITY METHOD

by

Makoto Kitagawa
Graduate Student, Department of Civil Engineering
University of California, Berkeley

and

Armen Der Kiureghian
Assistant Professor of Civil Engineering
University of California, Berkeley

A report on research sponsored by
the National Science Foundation

Report No. UCB/SESM-80/09
Division of Structural Engineering
and Structural Mechanics
Department of Civil Engineering
University of California, Berkeley

December 1980

NISEE/COMPUTER APPLICATIONS
DAVIS HALL
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720
(415) 642-5113

TABLE OF CONTENTS

		Page
1.	Introduction	1
2.	The Mean-Value First-Order Second-Moment Method(MVFOSM)	3
	2.1 Formulation of the method	3
	2.2 Lack of invariance	4
	2.3 Probability information	5
3.	The Advanced First-Order Second-Moment Method(FOSM)	6
	3.1 Formulation of the method	6
	3.2 Correlated random variables	7
	3.3 Inclusion of the distribution information	8
4.	Computer Program	10
5.	Comparison between MVFOSM and FOSM Methods	12
	5.1 Effect of non-linear performance function	12
	5.2 Effect of non-normal variables	12
	5.3 Effect of variable correlation	14
6.	Summary and Conclusion	14
7.	Acknowledgement	15
8.	References	16
	Tables	18
	Figures	25
	Appendix I User's Guide for Program FOSM	55
	Appendix II Computer Program FOSM	61
	Appendix III Example Run of Computer Program FOSM	69

1. Introduction

Any structure must be designed so that it withstands dead loads, live loads and any unusual disturbances, to which it might be subjected during its life time. At the same time, however, a structure design must be done within certain economic and functional constraints. On the face of uncertainties in loads and structural properties, these goals can only be accomplished by establishing a reasonable level of safety for the structure.

Since Freudenthal proposed his classical reliability theory (1), the application of probabilitistic concepts in structural design has been a major concern among researchers and engineers. Freudenthal proposed the probability of failure as a safety measure instead of the conventional safety factor.

However, the classical theory of reliability is appropriate only when our information on structural resistances, loads, etc., is comlete and accurate mathematical analysis can be performed. In practice, the information required for such analysis may not be completely available. One also has to recognize the existence of uncertainties arising from both inherent randomness and statistical error in actual design applications (2). Furthermore, direct calculation of failure probability requires numerical integration, which is not appropriate in engineering practice and in code-making.

To overcome those shortcomings of the classical theory, the first-order second-moment reliability theory was developed as a practical alternative (3). This theory stands on the basic recognition that the state of incomplete knowledge and information is unavoidable. It is characterized by its way of representing the uncertainty in structural design variables. Namely, only the first and second moments, i.e. means and variances, of design variables are used in analysis.

Furthermore, the performance function of the structure, which describes the criterion for its safety, is linearized through a first-order Taylor expansion. For these reasons, the method has come to be known as the first-order second-moment (FOSM) reliability method.

A useful measure of safety, which may serve as an alternative to the probability of failure, is obtained in this method and is known as the safety index. Using this index, the explicit use of failure probability in design equation is avoided. In the original development of the FOSM method, the performance function was linearized at mean values of design variables. For this reason, this method is known as the mean-value first-order second-moment (MVFOSM) method.

Although the MVFOSM method gives exact solutions when the performance function is linear and the design variables are normal, further study of the method revealed serious problems. Firstly, the information of non-normal variables, even if available, cannot be included in this theory in a logical manner. Secondly, for non-linear performance function, the linear approximation at the mean point results in lack of invariance relative to the safety index. That is to say, the safety index in this method depends on the particular formulation of the performance function, letting the MVFOSM method lack invariance.

It has been shown that the lack of invariance relative to the formulation of the performance function, and the inability to include distribution information can be overcome, while keeping the simple algebra of the first-order second-moment theory, if the linearization is performed at a point on the "failure surface" (4). This method is called herein as the advanced first-order second-moment method (FOSM).

The authors have developed a computer program for evaluating the safety index for an arbitrary performance function by the FOSM method. This development is presented in this report. Using this program, the safety indices based on MVFOSM and FOSM methods were computed and compared for a number of cases of interest. The comparison was done in terms of the non-linearity of the performance function, non-normality of the design variables, and the cross-correlation between design variables. Through these comparisons, the safety index obtained from the MVFOSM method was found to be inappropriate in some cases.

Information obtained through analyses, such as in this report, is useful as a tool in establishing reliability-based design formats in the future.

2. The Mean-Value First-Order Second-Moment Method (MVFOSM)

2.1 Formulation of the method

Neglecting time effects, the safety criterion of a particular structure can be written in general in terms of a performance function.

$$Z = g\,(X_1 , X_2 , \cdots\cdots, X_n) \quad \text{------------} \quad (2.1)$$

where X_i's represent structural variables which might influence the structural resistances and/or the external load actions. In the second-moment approach, the uncertainty of the variables, X_i's , is expressed only through their means (first moment) μ_i's, and their standard deviations σ_i's (square root of the second central moment).

The performance function is usually formulated such that $Z > 0$ denotes survival and $Z \leq 0$ denotes failure of the structure. Then, the failure surface is described in the n-dimensional space of X_i's as

$$g\,(x_1 , x_2 , \cdots\cdots, x_n) = 0 \quad \text{------------} \quad (2.2)$$

The first two terms of the Taylor expansion of Eq.(2.1) at the point $\{ x_1^* , x_2^* , \cdots, x_n^* \}$ yields

$$Z = g\,(x_1^* , x_2^* , \cdots\cdots, x_n^*) + \sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_* (X_i - x_i^*) \quad \text{---} \quad (2.3)$$

Eq.(2.3) is an approximation of Eq.(2.1). If the performance function is linear, Eq.(2.3) is exact. The MVFOSM method selects $\{ x_1^* , x_2^* , \cdots\ x_n^* \}$ to equal the mean $\{ \mu_1 , \mu_2 , \cdots, \mu_n \}$.

This yields the mean of Z

$$\mu_Z \doteqdot g\,(\mu_1 , \mu_2 , \cdots\cdots, \mu_n) \quad \text{------------} \quad (2.4)$$

and the standard deviation of Z

$$\sigma_Z \doteqdot \left[\sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_*^2 \sigma_i^2 + \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \rho_{ij} \left(\frac{\partial g}{\partial X_i}\right)_* \left(\frac{\partial g}{\partial X_j}\right)_* \sigma_i \sigma_j \right]^{\frac{1}{2}} \quad \text{---} \quad (2.5)$$

in which ρ_{ij} is the correlation coefficient between X_i and X_j , and the derivatives are evaluated at the mean values. In terms of these quantities, the safety index is defined as

$$\beta = \mu_Z / \sigma_Z \qquad (2.6)$$

Observe that β can be interpreted as the distance from the mean of g(•) to the origin in terms of the standard deviation of g(•), as shown in Fig.2.1. As the safety index is computed through only the first two moments of the design variables, it is a useful and convenient safety measure from the practical standpoint.

2.2 Lack of invariance

Eqs. (2.4) and (2.5) show that the calculated mean and variance of Z will depend on how the performance function is defined. This means that the calculated safety index is influenced by how the performance function is formulated, although the same failure surface may be involved. That is to say, the safety index based on the MVFOSM method lacks invariance relative to the formulation of the performance function.

To illustrate this lack of invariance, the following example is presented. Assume that only two variables are included : R, structural resistance, and S, load effect. The failure event is defined by ($R \leq S$). The following formulations of the performance function are consistent with this failure criterion:

$$Z_1 = R - S \qquad (2.7)$$

$$Z_2 = R / S - 1 \qquad (2.8)$$

$$Z_3 = \ln R - \ln S \qquad (2.9)$$

Define the first two moments of R and S by (μ_R , σ_R^2) and (μ_S , σ_S^2), respectively. Assuming no correlation between R and S, following the procedure dicussed above, one obtains the corresponding safety indices as

$$\beta_1 = \frac{r - 1}{\sqrt{r^2 \delta_R^2 + \delta_S^2}} \qquad (2.10)$$

$$\beta_2 = \frac{r - 1}{r} \frac{1}{\sqrt{\delta_R^2 + \delta_S^2}} \qquad (2.11)$$

$$\beta_3 = \frac{\ln r}{\sqrt{\delta_R^2 + \delta_S^2}} \qquad (2.12)$$

in which $r = \mu_R / \mu_S$, $\delta_R = \sigma_R / \mu_R$, $\delta_S = \sigma_S / \mu_S$

Table 2.1 and Figs. 2.2 and 2.3 show results obtained for the safety indices from Eqs.(2.10)--(2.12). Although Eqs.(2.7)--(2.9) constitute the same failure surface, the results show significant discrepancy between the computed safety indices for the three functions. As Eq.(2.7) is a linear expression, the safety index based on this formulation is exact. On the other hand, the discrepancy in results based on Eqs.(2.8) and (2.9) from the result of Eq.(2.7) is caused by the linearization of the respective performance functions. This discrepancy arises because of expansion of $g(\cdot)$ about the mean values. To overcome this shortcoming of the MVFOSM method, it is necessary to modify the way the performance function is linearized.

2.3 Probability information

To relate the safety index to failure probability, information on the probability distribution of Z is required. However, the distribution of Z remains unknown, since no distribution assumption was made with regard to X_i's.

If the function $g(\cdot)$ is linear or is well approximated by a linear approximation, the central limit theorem of probability theory suggests that the distribution of Z may approach the normal distribution for large n. Following this principle, one may estimate the failure probability as

$$P_f = \Phi(-\beta) \qquad (2.13)$$

where Φ is the standard normal cumulative distribution. Even when the normal distribution cannot be justified, the above may be regarded as a consistent measure of the failure probability.

Eq.(2.13) gives the correct solution of the failure probability for the case of normal variable and linear performance function. In practice, however, one may know that some variables are non-normally distributed. The MVFOSM is unable to incorporate such information in the analysis in a rational manner.

3. The Advanced First-Order Second-Moment Method (FOSM)

3.1 Formulation of the method

The lack of invariance of β to the choice of performance function arises because of linearization of the performance function about the mean point. It has been recognized that the linear expansion of the performance function should take place not about the mean point but about a point on the failure surface ($g(\cdot)=0$). That point lies in the upper tails of load distributions and in the lower tails of resistance distributions. The linearization point on the surface is chosen such that the resulting β is a minimum.

To do so, it is required to determine the precise point $\{x_1^*, x_2^*, ----, x_n^*\}$ on the failure surface (i.e. such that $g(x_1^*, x_2^*, ----x_n^*)=0$) where the linearization is to be made. Therefore, the safety checking can be considered to be finding the point on the failure surface representing the failure criterion, and measuring the distance from the mean to the linearization point.

The difference of the concept of searching the safety index in the MVFOSM method and FOSM method is schematically shown in Fig.3.1. This figure shows that the tangent plane taken at the mean deviates significantly from that taken at the failure point. As the location of point " A " in Fig.3.1 depends on the formulation of the performance function, the safety index changes with a change in the formulation of this function. In the FOSM method, however, once the failure point " B " is fixed, an invariant tangent plane is obtained, even if different formulations of the performance function are considered.

If the design variables are uncorrelated, using Eqs.(2.3)-(2.5), the mean and variance of Z are

$$\mu_Z = \sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_* (\mu_i - x_i^*) \qquad (3.1)$$

where $g(x_1^*, x_2^*, ----, x_n^*) = 0$ is used, and

$$\sigma_Z = \left[\sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_*^2 \sigma_i^2\right]^{\frac{1}{2}} = \sum_{i=1}^{n} \alpha_i \left(\frac{\partial g}{\partial X_i}\right)_* \sigma_i \qquad (3.2)$$

respectively, in which

$$\alpha_i \equiv \left(\frac{\partial g}{\partial X_i}\right)_* \sigma_i \Big/ \sqrt{\sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_*^2 \sigma_i^2} \qquad (3.3)$$

All derivatives in the above equations are evaluated at the point $\{x_1^*, x_2^*, ----, x_n^*\}$.

As the safety index β is defined by μ_Z / σ_Z, using Eqs.(3.1) and (3.2),

$$\sum_{i=1}^{n} \left(\frac{\partial g}{\partial X_i}\right)_* (\mu_i - x_i^* - \alpha_i \beta \sigma_i) = 0 \qquad (3.4)$$

Thus, the linearization point $\{x_1^*, x_2^*, ----, x_n^*\}$ and the safety index β are found by solving the system of equations

$$\begin{cases} x_i^* = \mu_i - \alpha_i \beta \sigma_i, & i = 1, 2, ----, n \\ g(x_1^*, x_2^*, ----, x_n^*) = 0 \end{cases} \qquad (3.5)$$

For non-linear performance functions, Eq.(3.5) can only be solved through iteration to obtain the minimum value of β. The iteration procedure is discussed in the next chapter.

3.2 Correlated random variables

In the preceding section, the design variables were assumed to be uncorrelated. When these variables are correlated, the problem can be solved by transforming into a set of uncorrelated variables as follows :

Let $\{\mu_X\} = \begin{Bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{Bmatrix}$ and $[V_X] = \begin{bmatrix} \sigma_1^2 & \rho_{ij}\sigma_i\sigma_j \\ & \ddots \\ & & \sigma_n^2 \end{bmatrix} = E\left[(\{X\} - \{\mu_X\})(\{X\} - \{\mu_X\})^T\right]$

denote the mean vector and the covariance matrix of correlated variables,

X_i ($i=1,2,----,n$), where ρ_{ij}= correlation coefficient between X_i and X_j. For positive-definite $[V_X]$, an orthogonal transformation, $\{W\}=[Z]^T\{X\}$, is possible, which transforms $[V_X]$ into a diagonal matrix of eigen-values.

$$[V_W]=[Z]^T\ [V_X][Z] \qquad (3.6)$$

The set of variables $\{W\}$, with mean, $\{\mu_W\}=[Z]^T\{\mu_X\}$, and diagonal variance matrix, $[V_W]$, are uncorrelated. Thus, the method described above can be applied to these variables to compute the safety index as before.

In this case, it is necessary to compute the transformed performance function and its derivatives. Using the inverse transformation,

$$\{X\}=([Z]^T)^{-1}\{W\} \qquad (3.7)$$

the performance function $g_X(X_1, X_2, ------, X_n)$ in the X-space is formulated as $g_W(W_1, W_2, ----, W_n)$ in the W-space. The derivatives of $g_W(\cdot)$, i.e. $\partial g_W/\partial W_i$ ($i=1,2,-----,n$), are computed as a linear combination of $\partial g_X/\partial X_i$ ($i=1,2,-----n$) as follows:

$$\left(\frac{\partial g_W}{\partial W_j}\right)_*=\sum_{i=1}^{n}\left(\frac{\partial g_X}{\partial X_i}\right)_*\left(\frac{\partial X_i}{\partial W_j}\right) \qquad (3.8)$$

where $\dfrac{\partial X_i}{\partial W_j}$=ij component of the matrix $([Z]^T)^{-1}$.

It should be noted herein that the eigen matrix Z satisfies

$$([Z]^T)^{-1}=[Z] \qquad (3.9)$$

when the computed eigen vectors are normalized such that each has a unit Euclidean length.

3.3 Inclusion of distribution information (5)

In the preceding discussion, nothing was said about the probability information contained in the safety index, or the inclusion of the distribution information into the FOSM method. In some cases, however, the information on the failure probability will be required. Or, in some cases, we may know the type of probability distribution for some variables.

As dicussed in 2.3, relating the safety index to the failure probability through Eq.(2.13) is a reasonable idea when the variables are normally distributed or when their distributions are not specified. On the other hand, when some variables are known to be non-normally distributed, it is necessary to obtain " equivalent normal " distributions for the non-normal variables in order to retain the same solution process.

The fitting of normal distributions to non-normal distributions is done at the linearization point, $X_i = x_i^*$. When the CDF, $F_{X_i}(X)$, and its PDF, $f_{X_i}(X)$, of X_i are known, the fitting into a normal CDF and PDF is done by setting,

$$F_{X_i}(x_i^*) = \phi\left(\frac{x_i^* - \mu_i'}{\sigma_i'}\right) \quad (3.10)$$

$$f_{X_i}(x_i^*) = \frac{1}{\sigma_i'}\varphi(\phi^{-1}(F_{X_i}(x_i^*))) \quad (3.11)$$

where μ_i' and σ_i' are the equivalent mean and standard deviation, and $\varphi(\cdot)$ and $\phi(\cdot)$ are the PDF and CDF of standard normal variable. Solving for μ_i' and σ_i', one obtains :

$$\mu_i' = x_i^* - \frac{\varphi\phi^{-1}(F_{X_i}(x_i^*)))\,\phi^{-1}(F_{X_i}(x_i^*))}{f_{X_i}(x_i^*)} \quad (3.12)$$

$$\sigma_i' = \frac{\varphi(\phi^{-1}(F_{X_i}(x_i^*)))}{f(x_i^*)} \quad (3.13)$$

In the numerical calculation, this fitting is done for each non-normal variable in each iteration searching for the minimum β and the corresponding failure point x_i^*. When the variables are correlated, the same procedure can be applied; i.e. Eqs. (3.12) and (3.13) are used in the original space (X-space). In this case, the same correlation coefficient ρ_{ij} is assumed during the iteration, yielding the i-j component of the covariance matrix $\rho_{ij}\sigma_i'\sigma_j'$. Since the covariance matrix changes at each iteration, a new eigen matrix has to be recomputed in each iteration to make an orthogonal transformation.

4. Computer Program

The computer program " FOSM " was developed to compute the safety index according to the advanced FOSM method for any arbitrary performance function. The program uses an iteration method to compute the minimum value of β and the corresponding failure point. The iteration starts by assuming initial values of α_i's and β. The initial value of β is computed by the MVFOSM method. And the first values of α_i's are computed at the mean point. The flow chart for the algorithm of the program is described below and in Fig.4.1.

(1) The user is required to provide the followings (①).*

1) Performance function ——— $Z = g_X(X_1, X_2, \cdots, X_n)$
2) Derivative functions ——— $(\partial g_X / \partial X_i)$ $i = 1, 2, \cdots, n$
3) Mean value ———— $\{\mu_X\} = \{\mu_1, \mu_2, \cdots, \mu_n\}$
4) Covariance matrix ———— $[V_X] = \begin{bmatrix} \sigma_1^2 & & \rho_{ij}\sigma_i\sigma_j \\ & \sigma_2^2 & \\ & & \ddots & \sigma_n^2 \end{bmatrix}$
5) Type of distribution for each variable X_i

(2) For correlated variables, orthogonal transformation is performed through the eigen matrix $[Z]$ (② — ③)

1) Find eigen matrix $[Z]$, such that $[V_W] = [Z]^T [V_X][Z]$ is a diagonal matrix with eigen values.
2) Transform mean $\{\mu_X\}$ to $\{\mu_W\}$: $\{\mu_W\} = [Z]^T \{\mu_X\}$

(3) Initial value of α_i's are computed (④).

$$\alpha_i = \frac{(g_W / W_i)_*}{\sqrt{\sum_{j=1}^{n} (g_W / W_j)_*^2 \, \sigma_{W_j}^2}} \qquad (i = 1, 2, \cdots, n)$$

computed at $\{W\} = \{\mu_W\}$ (Eq.(3.8))

(4) Initial value for β is computed by MVFOSM method (⑤)

Use Eqs. (2.4) — (2.6).

---

* The number in the circle means the block number in Fig.4.1.

(5) Modify the value of β , such that the point $\{w_i^*\} = \{\mu_{W_i}\} - \{\alpha_i \beta \sigma_{W_i}\}$ is on the failure surface. This is done by solving

$$g_W(\mu_{W1} - \alpha_1 \beta \sigma_{W1} , \text{-------------} , \mu_{Wn} - \alpha_n \beta \sigma_{Wn}) = 0$$

This equation is solved by Newton's method. Using the computed value of β , the failure point on the surface is computed (⑥ -- --⑧).

(6) Using the distribution information of each variable, fitting to normal distribution is done (⑨ -- ⑪).

1) When the variable has a normal distribution, or when the distribution is not specified, the mean μ_i and the standard deviation σ_i remains unchanged.

2) When the variable is non-normally distributed, the mean and standard deviation are modified using Eqs. (3.12) and (3.13).

3) After fitting is completed, new mean vector, $\{\mu'_X\}$, and covariance matrix, $[V'_X]$, are computed. In constructing $[V'_X]$, the correlation coefficients ρ_{ij} are assumed to be unchanged.

(7) Transform the mean vector $\{\mu'_X\}$, and the covariance matrix, $[V'_X]$, into an uncorrelated set of variables through a new eigen matrix $[Z']$. The procedure is the same as that in step (2). (⑫)

(8) New failure point, x_i^*, and cosine directors, α_i's, are computed using the latest means and standard deviations (⑬ -- ⑭).

(9) As done in step (5), modification of β must be done so that the linearization point is on the failure surface (⑮ -- ⑯)

(10) The convergence of iteration is checked by comparing the latest β and α_i's with those obtained in the preceding step (⑰ -- ⑱).

(11) If β and/or α_i's have not converged, go back to step (6) and repeat calculation. If they are converged, print results (⑰ -- -⑲).

5. Comparison between MVFOSM and FOSM Methods

5.1 Effect of non-linear performance function

In section 2.2, the " lack of invariance " in the MVFOSM method, which is caused by the non-linearity of the performance function, was discussed. The FOSM method, on the other hand, is invariant relative to the formulation of the performance function. To illustrate this, safety indices for the three formulations of the performance function in Eqs. (2.7)--(2.9) are computed and are shown in Table 5.1 and Figs. 5.1--5.2.

Observe that results for all three formulations are the same, except for small deviation, less than 0.4 %, which are due to the convergence tolerance assumed in the iteration process. Note that formulation 1 (Eq.(2.7)), which is linear, requires no iteration. Therefore, the results corresponding to this formulation are exact.

5.2 Effect of non-normal variables

To see the effect of non-normal variables, the safety index was computed for the performance function $Z = R - S$, where R and S were assumed to have the following distributions.

Case	R	S	Index
1	Normal	Normal	N/N
2	Log-normal	Log-normal	LN/LN
3	Gamma	Gamma	GAM/GAM
4	Gamma	Normal	GAM/N
5	Extreme-III	Extreme-I	EX3/EX1
6	Extreme-III	Extreme-II	EX3/EX2

Computations were carried out for all the above cases and the mean and coefficient of variation values given in the following table.

Mean Value	μ_R	40, 60
	μ_S	20
Coefficient of Variation	$\delta_R = \sigma_R / \mu_R$	0.1, 0.15, 0.2, 0.25, 0.3
	$\delta_S = \sigma_S / \mu_S$	0.1, 0.3

To obtain the accuracies of the MVFOSM and FOSM methods, the exact failure probability for each case was also computed using

$$P_f = \int_{-\infty}^{\infty} F_R(s) f_S(s) ds \quad \text{------------------} \quad (5.1)$$

where $F_R(\cdot)$ = CDF of variable R, and $f_S(\cdot)$ = PDF of variable S. The numerical integration required in Eq.(5.1) was carried out using the IMSL routine called " DCADRE ". Then, the safety index, β, was computed from

$$\beta = \phi^{-1}(1 - P_f) \quad \text{------------------------} \quad (5.2)$$

Results for the above computations are shown in Tables 5.2 and 5.3 for β and Tables 5.4 and 5.5 for P_f. These results are also shown plotted in Figs. 5.3--5.12 for β and Figs. 5.13--5.22 for P_f. From these results, one may make the following remarks :

(1) The effect of non-normal distribution is significant when $r = \mu_R/\mu_S$ is large or when δ_R and δ_S are small. This is because the probability of failure in such cases becomes more sensitive to the tails of the distributions of R and S. On the other hand, this effect becomes comparatively small for small r or large δ_R and δ_S.

(2) Results based on the FOSM method closely agrees with those obtained by numerical integration of the exact expression. Thus, by transformation into equivalent normal distribution through Eqs. (3.12)--(3.13), accurate results can be obtained.

(3) When variables have extreme-value distributions, e.g. cases EX3/EX1 or EX3/EX2, the safety index seems to be smaller than that which is based on normal distributions. Thus, the assumption of normal distribution gives unconservative results.

(4) The negligence of distribution information may result in significant misjudgement, when a high safety index is required. Therefore, the inclusion of such information, when available, is essential in safety analysis.

5.3 Effect of variable correlation

To examine the effect of correlation between design variables, the safety index for the performance function $Z = R - S$ was computed assuming $\mu_R = 40$, $\mu_S = 20$, $\sigma_R = 4$, $\sigma_S = 2$ and correlation coefficient $\rho_{RS} = -0.5, -0.4, -0.3, -0.2, -0.1, 0.0, 0.1, 0.2, 0.3, 0.4, 0.5$. No non-normal distributions were assumed for R and S. The results of the computation are shown in Table 5.6.

Observe that the safety index varies significantly with the correlation coefficient. Thus, inclusion of information on correlation between variables is also essential in structural safety analysis.

6. Summary and Conclusion

This study can be summarized and concluded as follows:

(1) The MVFOSM method gives the correct solution of the safety index and the failure probability in the case of linear performance function with normal variables. But, this method has the following shortcomings.

   1) Significant error is introduced by linearization of the non-linear performance function about the mean point. This results in lack of invariance of the computed safety index relative to the formulation of the performance function.

   2) Even if information on non-normal distribution of some variables is available, such information cannot be rationally included in the analysis.

(2) These shortcomings are overcome by the FOSM method, while keeping the simple algebra of the second-moment approach.

   1) The lack of invariance in the MVFOSM method can be overcome by doing the linear expansion of the performance function about a point located on the failure surface instead of about the mean point

   2) The information on non-normal variables can be included in the analysis, by fitting the non-normal distribution to normal distribution at the failure point.

(3) When the variables are correlated, the safety index can be computed by transforming the correlated variables into a set of uncorrelated variables. This calculation is possible for a performance function with non-normal variables, if correlation coefficients are assumed to be unchanged.

(4) The effect of the non-linearity of the performance function cannot be neglected in the MVFOSM approach. FOSM method is essential for a non-linear performance function.

(5) The inclusion of the information on non-normal variables, when available, is essential for accurate safety evaluation, especially when P_f is small or when β is large.

(6) The effect of correlation between design variables cannot be neglected.

(7) The basic idea of the computer program "FOSM" is to assume initial values of β and α_i's and to carry out iteration until these values are converged. The initial value of β is computed by the MVFOSM method, and the first values of α_i's are computed at the mean point. These initial assumptions seem to be fine to obtain the correct solution when the failure surface is smooth. In practice, most structural problems have smooth failure surfaces. Therefore, this program should be useful for most structural reliability problems.

7. Acknowledgement

This work was supported by the U. S. National Science Foundation under Grant No. ENG-7905906. This support is gratefully acknowledged.

8. References

(1) Freudenthal, A.M., Garrelts, J.M., and Shinozuka, M., "The Analysis of Structural Safety," Journal of the Structural Division, ASCE, Vol. 92, No. ST 1, Proc. Paper 4682, February, 1966, pp267--325

(2) Ang, A. H-S. and Amin, M., "Safety Factors and Probability in Structural Design," Journal of the Structural Division, ASCE, Vol. 95, No. ST 7, Proc. Paper 6667, July, 1969, pp1389--1405

(3) Ang, A. H-S. and Cornell, C.A., "Reliability Bases of Structural Safety and Design," Journal of the Structural Division, ASCE, Vol. 100, No. ST 9, Proc. Paper 10777, September, 1974, pp1755--1769

(4) Hasofer, A.M. and Lind N.C., "Exact and Invariant Second-Moment Code Format," Journal of the Engineering Mechanics Division, ASCE, Vol. 100, No. EM 1, Proc. Paper 10376, February, 1974, pp111--121

(5) Rackwitz, R., "Practical Probabilitistic Approach to Design," Technical University of Munich, Institut für Bauingenieurwesen III, May, 1976

List of Tables and Figures

Table 2.1 Lack of invariance (MVFOSM method)

Table 5.1 Effect of nonlinearity of performance function

Table 5.1 --- Table 5.5

Effect of non-normal variables

Table 5.6 Effect of variable correlation

Figure 2.1 Definition of safety index

Figure 2.2 --- Figure 2.3

Lack of invariance (MVFOSM method)

Figure 3.1 Conceptual comparison of MVFOSM and FOSM methods

Figure 3.2 Fitting to normal distribution

Figure 4.1 Flow chart of computer program "FOSM"

Figure 5.1 --- Figure 5.2

Effect of nonlinear performance function (FOSM method)

Figure 5.3 --- Figure 5.12

Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

Figure 5.13 --- Figure 5.22

Comparisons of probability of failure, P_f , based on MVFOSM and FOSM methods with exact solutions

TAB. 2.1 LACK OF INVARIANCE (MVFOSM METHOD)

$\gamma = \mu_R/\mu_S$	2						3					
$\delta_S = \sigma_S/\mu_S$	0.1			0.3			0.1			0.3		
Performance Function / $\delta_R = \sigma_R/\mu_R$	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
$Z = R - S$	4.472	2.425	1.644	2.774	2.000	1.491	6.325	3.288	2.209	4.714	2.981	2.108
$Z = R/S - 1$	3.536	2.236	1.581	1.582	1.387	1.179	4.714	2.981	2.108	2.108	1.849	1.571
$Z = \ln R/S$	4.901	3.100	2.192	2.192	1.922	1.634	7.768	4.913	3.474	3.474	3.047	2.589

TAB.5.1 EFFECT OF NONLINEARITY OF PERFORMANCE FUNCTION

$\gamma = \mu_R/\mu_S$	2 $\left(\begin{matrix}\mu_R=40\\ \mu_S=20\end{matrix}\right)$						3 $\left(\begin{matrix}\mu_R=60\\ \mu_S=20\end{matrix}\right)$					
$\delta_S = \sigma_S/\mu_S$	0.1			0.3			0.1			0.3		
Performance Function / $\delta_R = \sigma_R/\mu_R$	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
$Z=R-S$	4.472	2.425	1.644	2.774	2.000	1.491	6.325	3.288	2.209	4.714	2.981	2.108
$Z=R/S-1$	4.487	2.433	1.649	2.786	2.007	1.496	6.347	3.297	2.215	4.730	2.994	2.116
$Z=\ln R/S$	4.472	2.425	1.644	2.786	2.000	1.491	6.325	3.288	2.209	4.714	2.981	2.108

TAB. 5.2 EFFECT OF NON–NORMAL VARIABLES (β for $r=2$)

Mean Value		Standard Deviation		MVFOSM method	Advanced FOSM method					
μ_R	μ_S	σ_R	σ_S		N/N	LN/LN	GAM/GAM	GAM/N	EX3/EX1	EX3/EX2
40	20	4	2	4.472	4.472 (4.472)	4.914 (4.914)	4.847 (4.848)	4.981 (4.929)	3.487 (3.405)	3.486 (3.926)
		6		3.162	3.162 (3.162)	3.828 (3.828)	3.624 (3.624)	3.647 (3.657)	2.748 (2.693)	2.764 (2.680)
		8		2.425	2.425 (2.425)	3.060 (3.060)	2.832 (2.832)	2.832 (2.841)	2.260 (2.219)	2.273 (2.215)
		10		1.961	1.961 (1.961)	2.514 (2.514)	2.294 (2.294)	2.288 (2.297)	1.912 (1.878)	1.923 (1.877)
		12		1.644	1.644 (1.644)	2.113 (2.113)	1.908 (1.908)	1.902 (1.909)	1.646 (1.617)	1.655 (1.617)
		4	6	2.774	2.774 (2.774)	2.359 (2.359)	2.478 (2.479)	2.798 (2.811)	2.325 (2.254)	2.208 (2.148)
		6		2.357	2.357 (2.357)	2.202 (2.202)	2.257 (2.259)	2.419 (2.459)	2.152 (2.059)	2.135 (2.019)
		8		2.000	2.000 (2.000)	2.024 (2.024)	2.023 (2.025)	2.089 (2.114)	1.927 (1.844)	1.998 (1.843)
		10		1.715	1.715 (1.715)	1.842 (1.842)	1.800 (1.802)	1.813 (1.842)	1.714 (1.649)	1.807 (1.657)
		12		1.491	1.491 (1.491)	1.670 (1.670)	1.598 (1.600)	1.583 (1.614)	1.524 (1.463)	1.611 (1.481)

Note: Exact values obtained through numerical integration of Eq.(5.1) are shown in parenthesis.

TAB.5.3 EFFECT OF NON-NORMAL VARIABLES (β for $r=3$)

Mean Value		Standard Deviation		MVFOSM method	Advanced FOSM method					
μ_R	μ_S	σ_R	σ_S		N/N	LN/LN	GAM/GAM	GAM/N	EX3/EX1	EX3/EX2
60	20	6	2	6.325	6.325 (6.325)	7.788 (7.788)	7.734 (7.553)	7.732 (7.694)	4.629 (4.572)	4.621 (4.472)
		9		4.339	4.339 (4.339)	6.088 (6.088)	5.564 (5.411)	5.615 (5.450)	3.676 (3.626)	3.683 (3.613)
		12		3.288	3.288 (3.288)	4.888 (4.888)	4.321 (4.321)	4.330 (4.338)	3.063 (3.017)	3.059 (3.014)
		15		2.643	2.643 (2.643)	4.040 (4.040)	3.498 (3.498)	3.496 (3.504)	2.620 (2.586)	2.621 (2.585)
		18		2.209	2.209 (2.209)	3.420 (3.420)	2.917 (2.917)	2.913 (2.919)	2.279 (2.257)	2.287 (2.257)
		6	6	4.714	4.714 (4.714)	3.666 (3.666)	4.080 (4.081)	4.944 (5.021)	3.467 (3.390)	2.937 (2.884)
		9		3.698	3.698 (3.698)	3.433 (3.433)	3.647 (3.649)	4.076 (4.093)	3.105 (3.029)	2.865 (2.763)
		12		2.981	2.981 (2.981)	3.169 (3.169)	3.217 (3.220)	3.420 (3.443)	2.737 (2.673)	2.725 (2.563)
		15		2.476	2.476 (2.476)	2.901 (2.901)	2.831 (2.834)	2.919 (2.944)	2.422 (2.368)	2.486 (2.329)
		18		2.108	2.108 (2.108)	2.646 (2.646)	2.496 (2.501)	2.527 (2.554)	2.156 (2.110)	2.227 (2.100)

Note: Exact values obtained through numerical integration of Eq.(5.1) are shown in parenthesis.

TAB.5.4 EFFECT OF NON-NORMAL VARIABLES (P_f for $r=2$)

Mean Value μ_R	Mean Value μ_S	Standard Deviation σ_R	Standard Deviation σ_S	MVFOSM method	Advanced FOSM method N/N	LN/LN	GAM/GAM	GAM/N	EX3/EX1	EX3/EX1
40	20	4	2	3.87×10^{-6}	3.87×10^{-6} (3.87×10^{-6})	4.59×10^{-7} (4.59×10^{-7})	6.27×10^{-7} (6.25×10^{-7})	3.16×10^{-7} (4.13×10^{-7})	2.44×10^{-4} (3.31×10^{-4})	2.45×10^{-4} (4.41×10^{-4})
		6		7.83×10^{-4}	7.83×10^{-4} (7.83×10^{-4})	6.45×10^{-5} (6.45×10^{-5})	1.45×10^{-4} (1.45×10^{-4})	1.33×10^{-4} (1.27×10^{-4})	3.00×10^{-3} (3.54×10^{-3})	2.86×10^{-3} (3.68×10^{-3})
		8		7.65×10^{-3}	7.65×10^{-3} (7.65×10^{-3})	1.11×10^{-3} (1.11×10^{-3})	2.32×10^{-3} (2.31×10^{-3})	2.32×10^{-3} (2.25×10^{-3})	1.19×10^{-2} (1.33×10^{-2})	1.15×10^{-2} (1.34×10^{-2})
		10		2.49×10^{-2}	2.49×10^{-2} (2.49×10^{-2})	5.97×10^{-3} (5.97×10^{-3})	1.09×10^{-2} (1.07×10^{-2})	1.11×10^{-2} (1.08×10^{-2})	2.79×10^{-2} (3.02×10^{-2})	2.72×10^{-2} (3.02×10^{-2})
		12		5.01×10^{-2}	5.01×10^{-2} (5.01×10^{-2})	1.73×10^{-2} (1.73×10^{-2})	2.82×10^{-2} (2.82×10^{-2})	2.86×10^{-2} (2.81×10^{-2})	4.98×10^{-2} (5.29×10^{-2})	4.90×10^{-2} (5.29×10^{-2})
		4	6	2.77×10^{-3}	2.77×10^{-3} (2.77×10^{-3})	9.17×10^{-3} (9.17×10^{-3})	6.61×10^{-3} (6.59×10^{-3})	2.57×10^{-3} (2.47×10^{-3})	1.00×10^{-2} (1.21×10^{-2})	1.36×10^{-2} (1.58×10^{-2})
		6		9.21×10^{-3}	9.21×10^{-3} (9.21×10^{-3})	1.38×10^{-2} (1.38×10^{-2})	1.20×10^{-2} (1.20×10^{-2})	7.77×10^{-3} (7.36×10^{-3})	1.57×10^{-2} (1.98×10^{-2})	1.64×10^{-2} (2.17×10^{-2})
		8		2.28×10^{-2}	2.28×10^{-2} (2.28×10^{-2})	2.15×10^{-2} (2.15×10^{-2})	2.15×10^{-2} (2.14×10^{-2})	1.83×10^{-2} (1.73×10^{-2})	2.70×10^{-2} (3.26×10^{-2})	2.29×10^{-2} (3.26×10^{-2})
		10		4.32×10^{-2}	4.32×10^{-2} (4.32×10^{-2})	3.27×10^{-2} (3.27×10^{-2})	3.60×10^{-2} (3.58×10^{-2})	3.49×10^{-2} (3.28×10^{-2})	4.33×10^{-2} (5.02×10^{-2})	3.54×10^{-2} (4.88×10^{-2})
		12		6.80×10^{-2}	6.80×10^{-2} (6.80×10^{-2})	4.75×10^{-2} (4.75×10^{-2})	5.50×10^{-2} (5.48×10^{-2})	5.67×10^{-2} (5.33×10^{-2})	6.38×10^{-2} (7.17×10^{-2})	5.36×10^{-2} (6.93×10^{-2})

Note: Exact values obtained through numerical integration of Eq.(5.1) are shown in parenthesis.

TAB.5.5 EFFECT OF NON-NORMAL VARIABLES (P_f for $\gamma = 3$)

Mean Value		Standard Deviation		MVFOSM method	Advanced FOSM method					
μ_R	μ_S	σ_R	σ_S		N/N	LN/LN	GAM/GAM	GAM/N	EX3/EX1	EX3/EX2
60	20	6	2	1.27×10^{-10}	1.27×10^{-10} (1.27×10^{-10})	4.07×10^{-19} (4.07×10^{-19})	7.11×10^{-15} (1.88×10^{-14})	7.11×10^{-15} (1.54×10^{-15})	1.84×10^{-6} (2.41×10^{-6})	1.91×10^{-6} (3.87×10^{-6})
		9		7.17×10^{-6}	7.17×10^{-6} (7.17×10^{-6})	5.71×10^{-10} (5.71×10^{-10})	1.32×10^{-8} (3.13×10^{-8})	9.85×10^{-9} (2.52×10^{-8})	1.19×10^{-4} (1.44×10^{-4})	1.15×10^{-4} (1.51×10^{-4})
		12		5.05×10^{-4}	5.05×10^{-4} (5.05×10^{-4})	5.08×10^{-7} (5.08×10^{-7})	7.75×10^{-6} (7.75×10^{-6})	7.46×10^{-6} (7.18×10^{-6})	1.09×10^{-3} (1.28×10^{-3})	1.11×10^{-3} (1.29×10^{-3})
		15		4.11×10^{-3}	4.11×10^{-3} (4.11×10^{-3})	2.67×10^{-5} (2.67×10^{-5})	2.34×10^{-4} (2.34×10^{-4})	2.36×10^{-4} (2.29×10^{-4})	4.40×10^{-3} (4.85×10^{-3})	4.38×10^{-3} (4.86×10^{-3})
		18		1.36×10^{-2}	1.36×10^{-2} (1.36×10^{-2})	3.13×10^{-4} (3.13×10^{-4})	1.77×10^{-3} (1.77×10^{-3})	1.79×10^{-3} (1.75×10^{-3})	1.13×10^{-2} (1.20×10^{-2})	1.11×10^{-2} (1.20×10^{-2})
		6	6	1.21×10^{-6}	1.21×10^{-6} (1.21×10^{-6})	1.23×10^{-4} (1.23×10^{-4})	2.25×10^{-5} (2.24×10^{-5})	3.83×10^{-7} (2.57×10^{-7})	2.63×10^{-4} (3.50×10^{-4})	1.66×10^{-3} (1.96×10^{-3})
		9		1.09×10^{-4}	1.09×10^{-4} (1.09×10^{-4})	2.98×10^{-4} (2.98×10^{-4})	1.33×10^{-4} (1.32×10^{-4})	2.29×10^{-5} (2.13×10^{-5})	9.52×10^{-4} (1.23×10^{-3})	2.09×10^{-3} (2.86×10^{-3})
		12		1.43×10^{-3}	1.43×10^{-3} (1.43×10^{-3})	7.66×10^{-4} (7.66×10^{-4})	6.47×10^{-4} (6.41×10^{-4})	3.13×10^{-4} (2.88×10^{-4})	3.10×10^{-3} (3.76×10^{-3})	3.21×10^{-3} (5.18×10^{-3})
		15		6.64×10^{-3}	6.64×10^{-3} (6.64×10^{-3})	1.86×10^{-3} (1.86×10^{-3})	2.32×10^{-3} (2.30×10^{-3})	1.76×10^{-3} (1.62×10^{-3})	7.73×10^{-3} (8.94×10^{-3})	6.46×10^{-3} (9.93×10^{-3})
		18		1.75×10^{-2}	1.75×10^{-2} (1.75×10^{-2})	4.07×10^{-3} (4.07×10^{-3})	6.27×10^{-3} (6.20×10^{-3})	5.75×10^{-3} (5.32×10^{-3})	1.55×10^{-2} (1.74×10^{-2})	1.30×10^{-2} (1.79×10^{-2})

Note: Exact values obtained through numerical integration of Eq.(5.1) are shown in parenthesis.

TAB. 5.6 EFFECT OF VARIABLE CORRELATION

ρ_{RS}	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5
β	3.780	3.892	4.016	4.152	4.324	4.472	4.663	4.880	5.130	5.423	5.774
P_F	7.85×10^{-5}	4.96×10^{-5}	2.96×10^{-5}	1.65×10^{-5}	7.66×10^{-6}	3.87×10^{-6}	1.56×10^{-6}	5.32×10^{-7}	1.45×10^{-7}	2.93×10^{-8}	3.38×10^{-9}

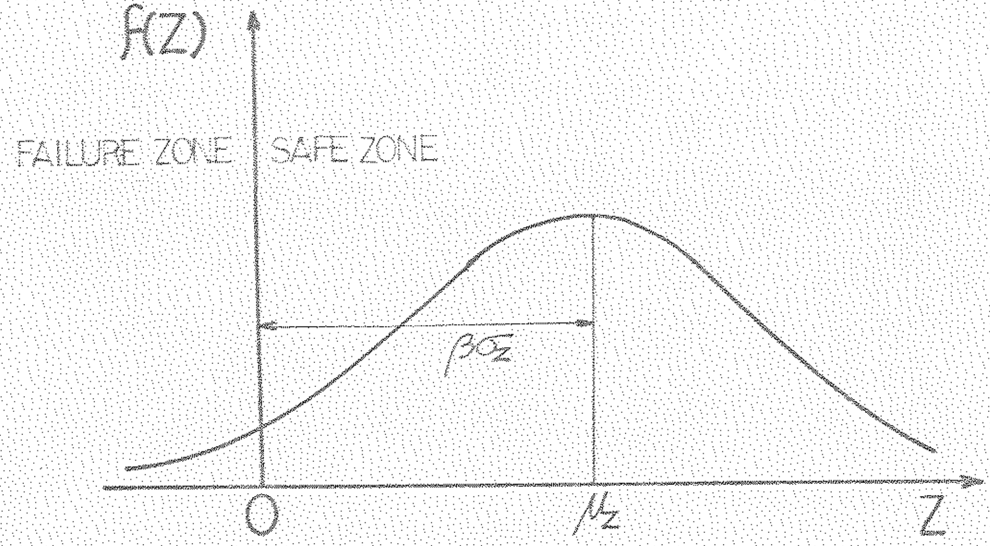


FIG.2.1 DEFINITION OF SAFETY INDEX

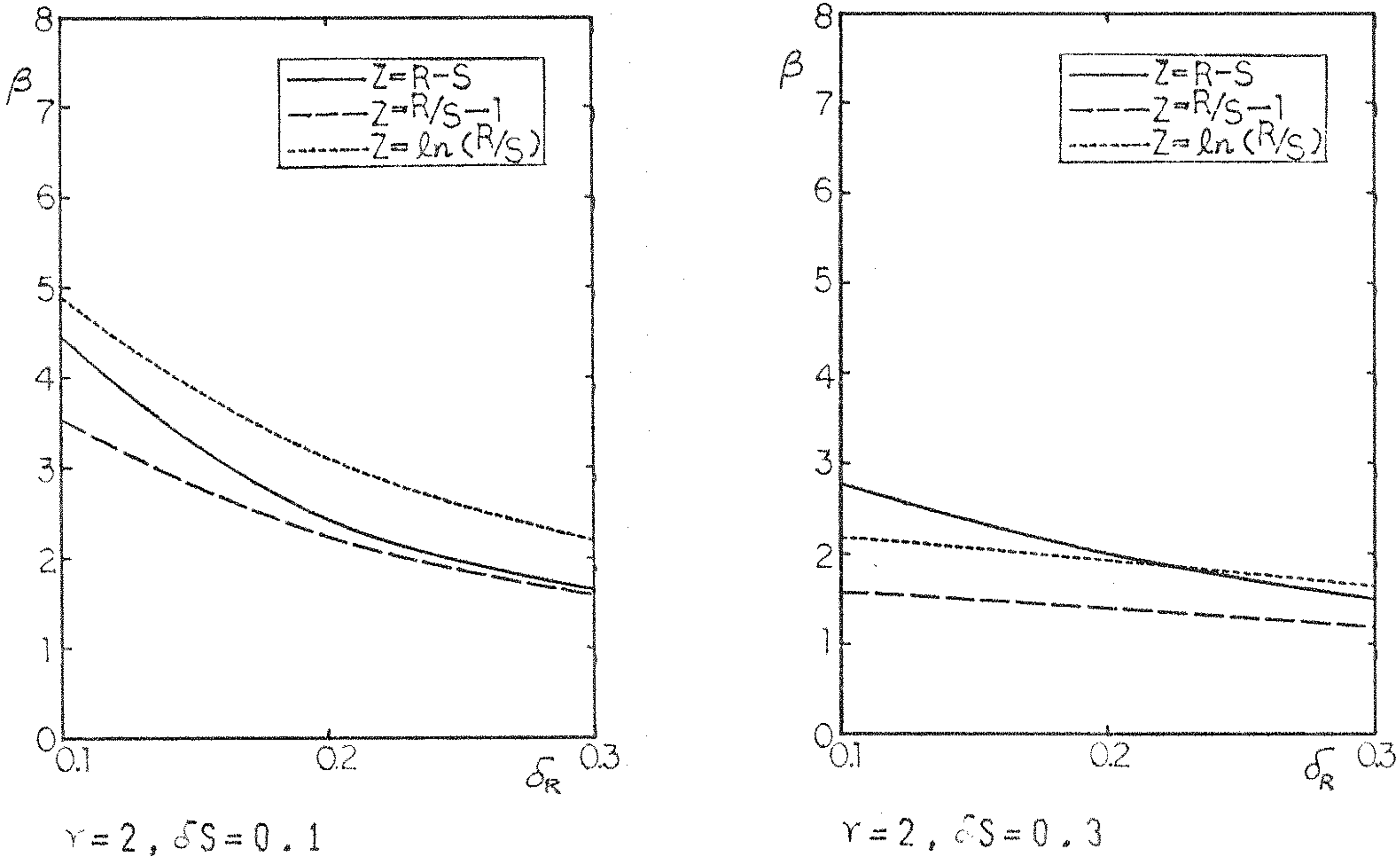
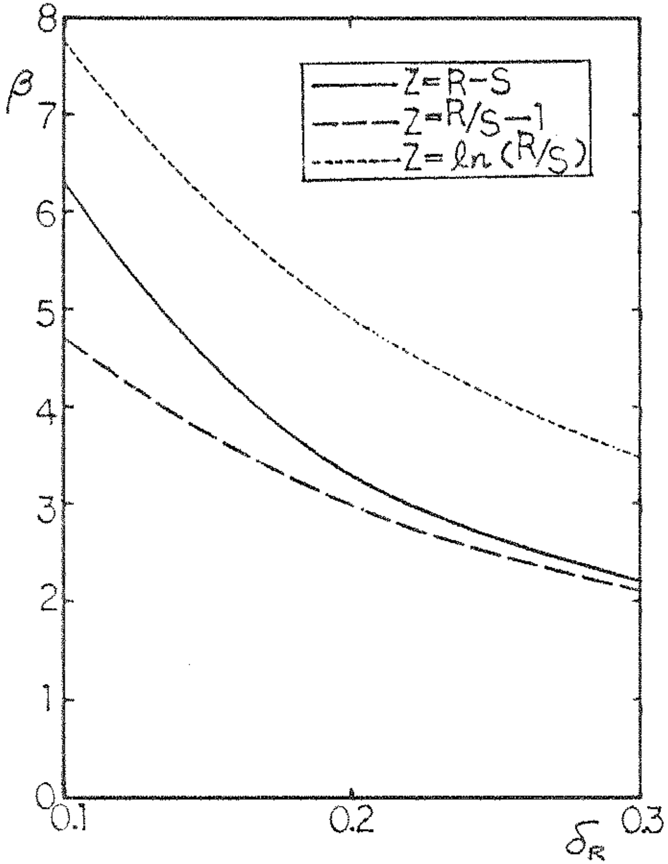
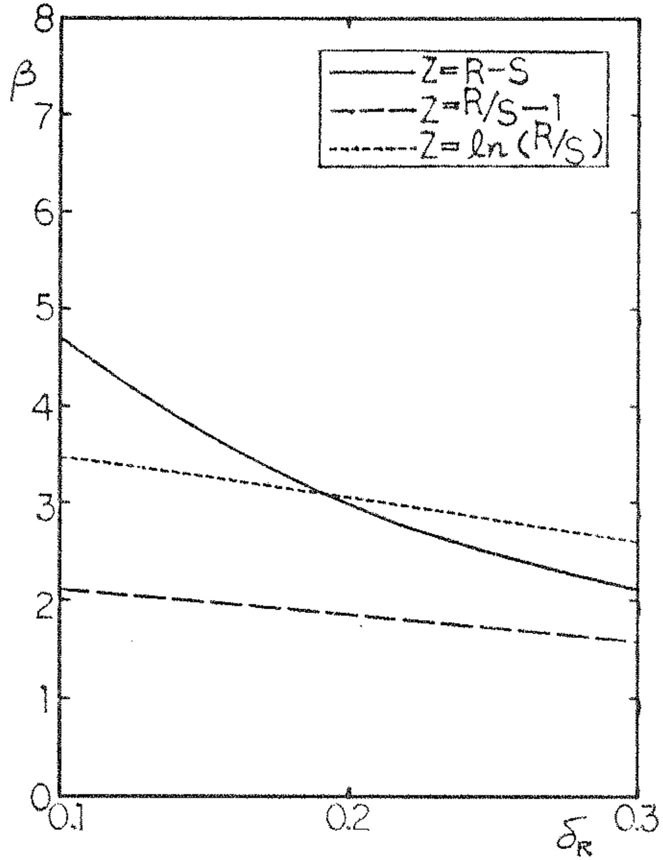


FIG.2.2 LACK OF INVARIANCE ($\gamma = 2$)
(MVFOSM METHOD)



γ = 3 , δS = 0 . 1



γ = 3 , δS = 0 . 3

FIG. 2.3 LACK OF INVARIANCE (γ =3)
(MVFOSM METHOD)

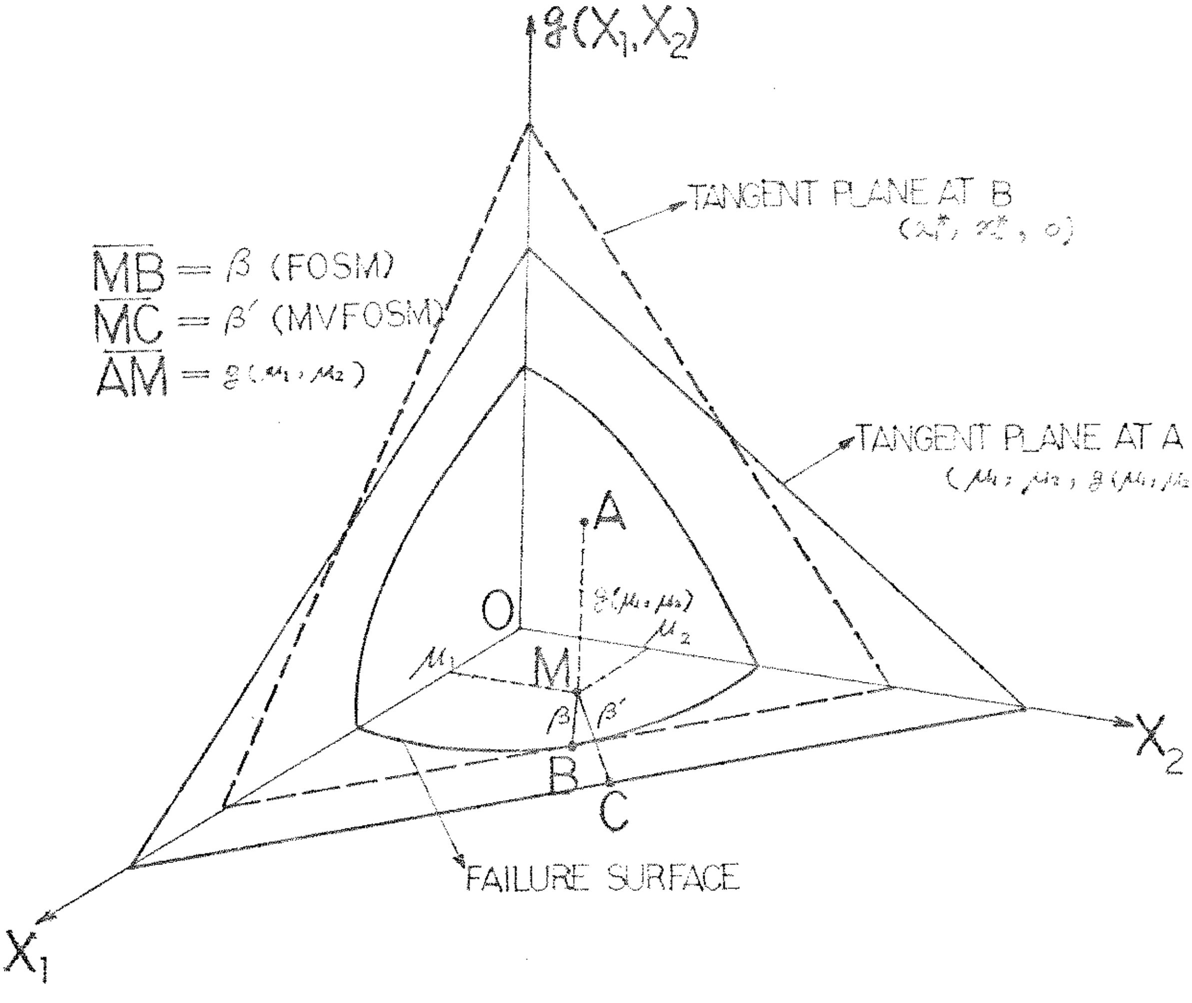


FIG. 3.1 CONCEPTUAL COMPARISON OF MVFOSM AND FOSM METHODS

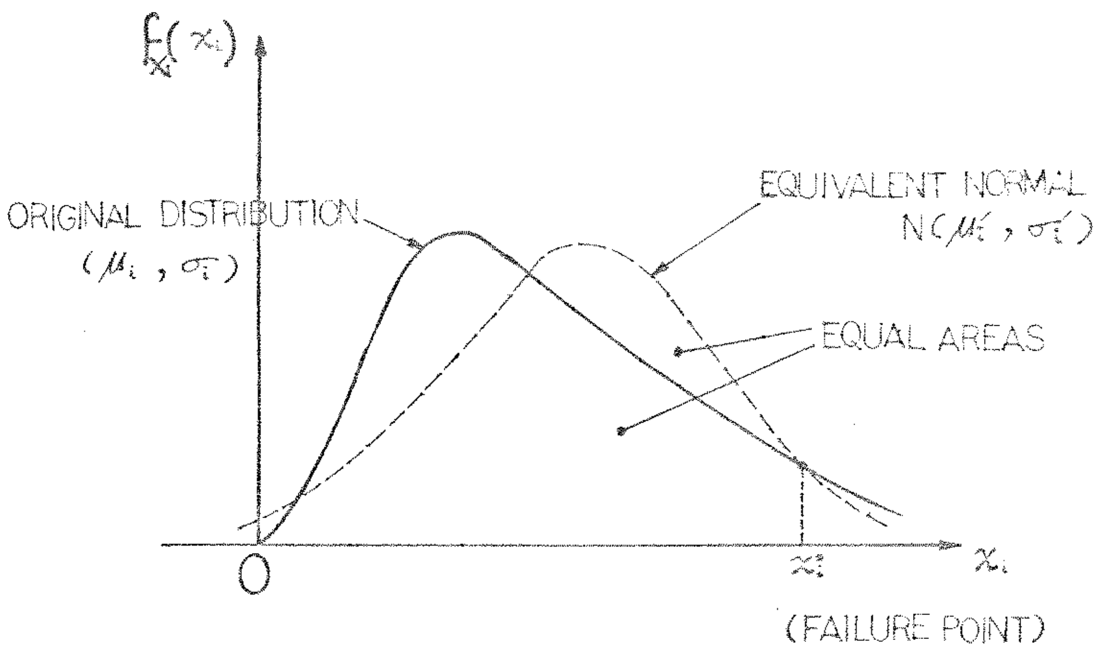


FIG.3.2 FITTING TO NORMAL DISTRIBUTION

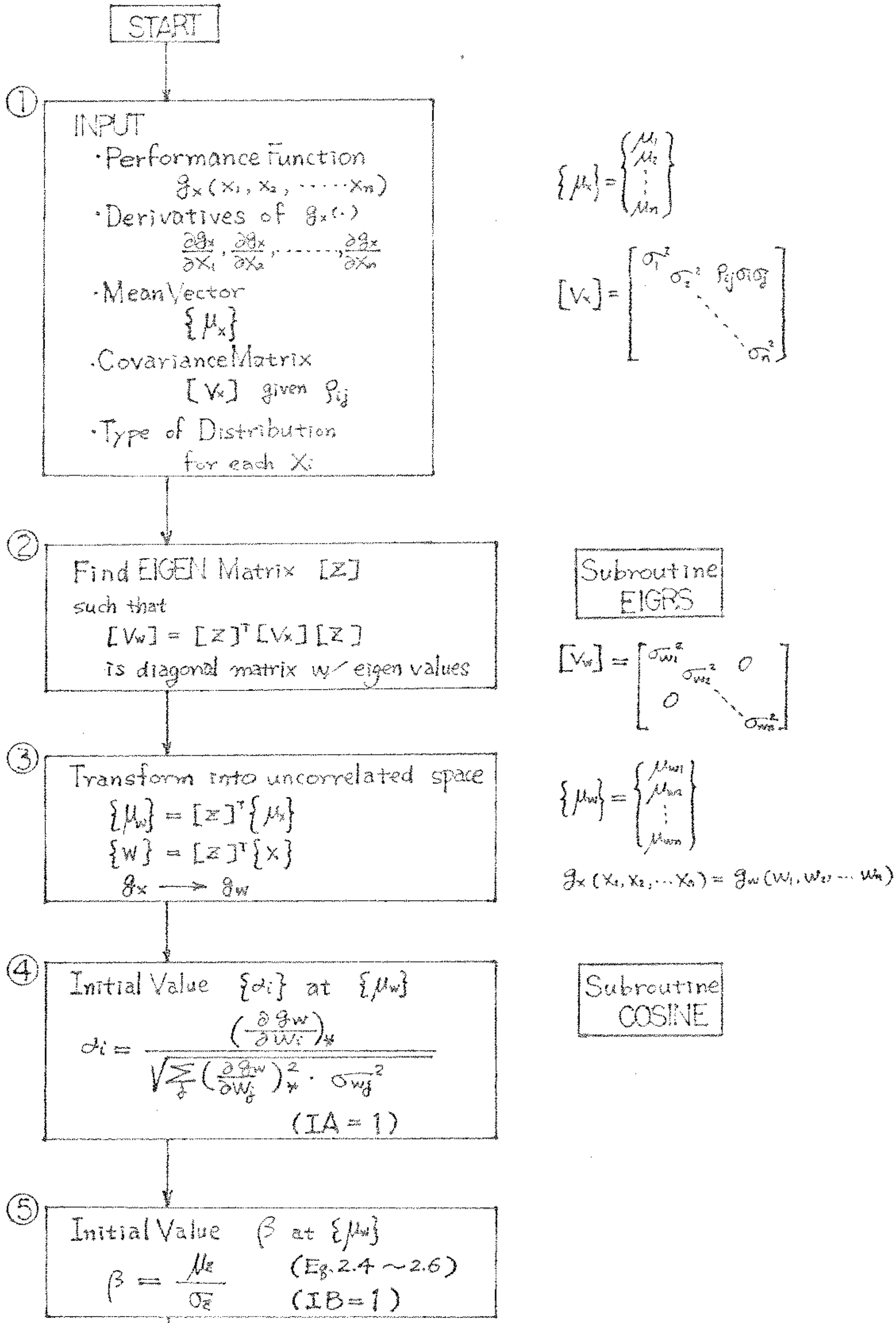


FIG. 4.1

FLOW CHART OF

COMPUTER PROGRAM "FOSM"

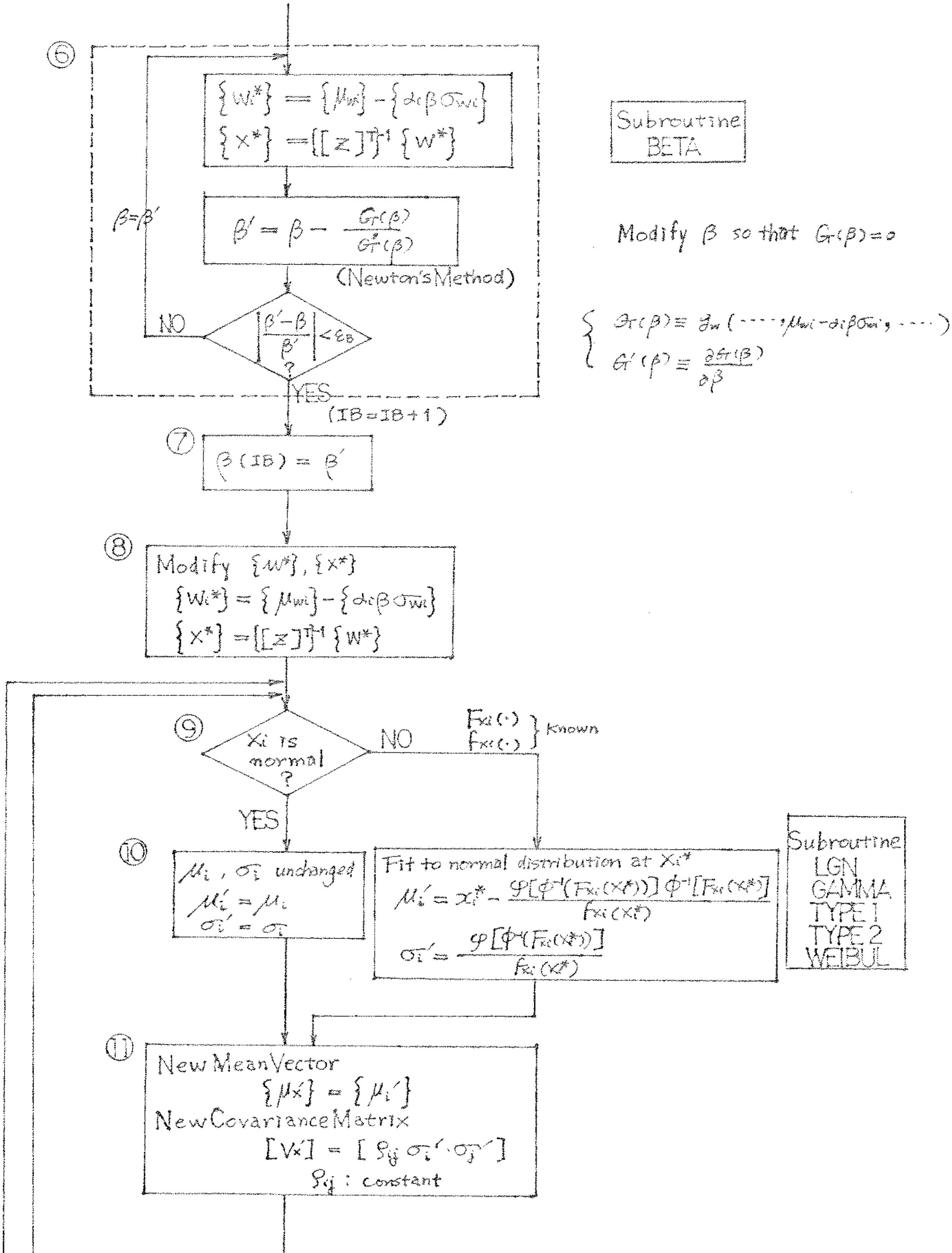


FIG. 4.1 (CONTINUED)

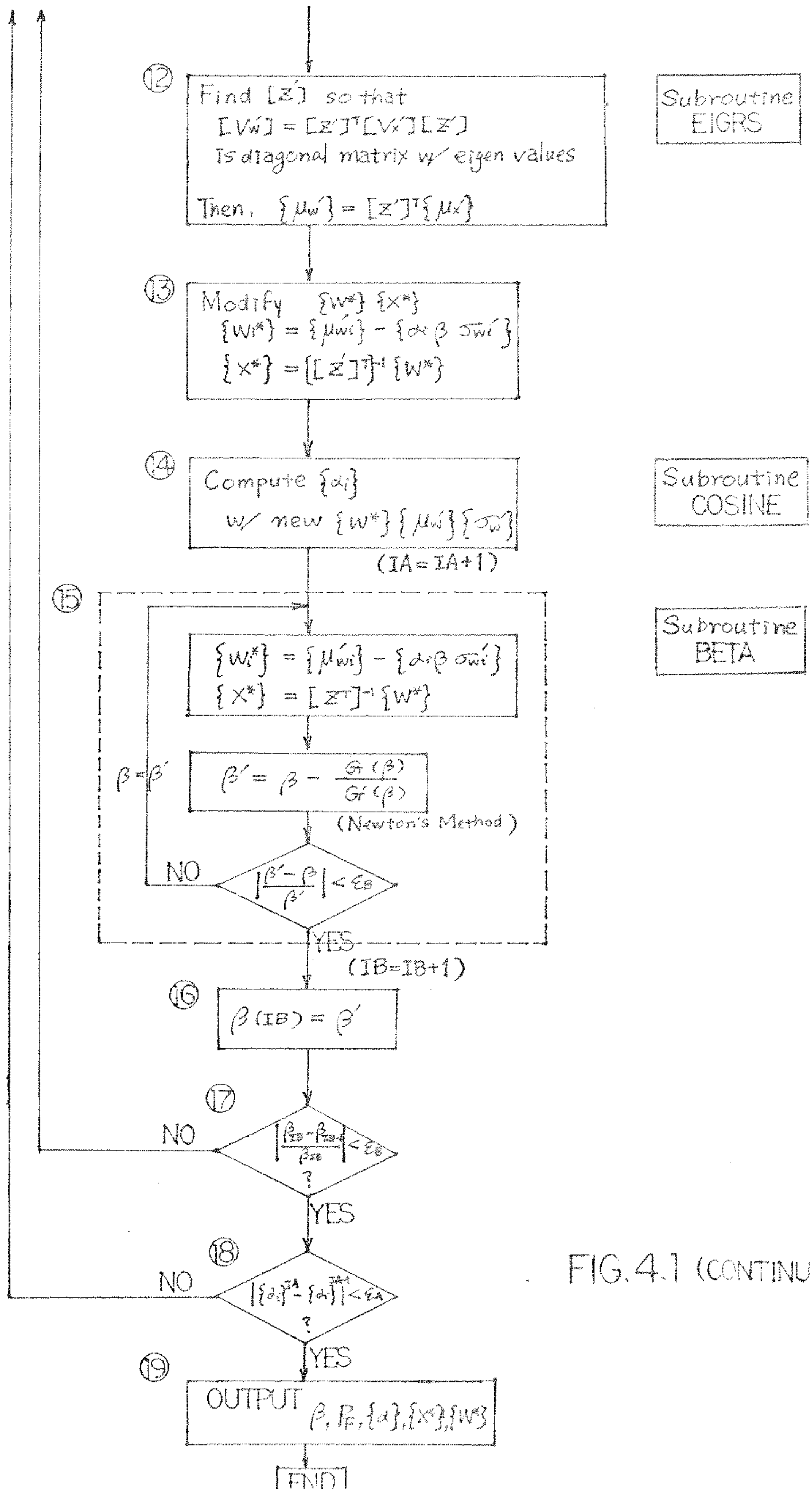
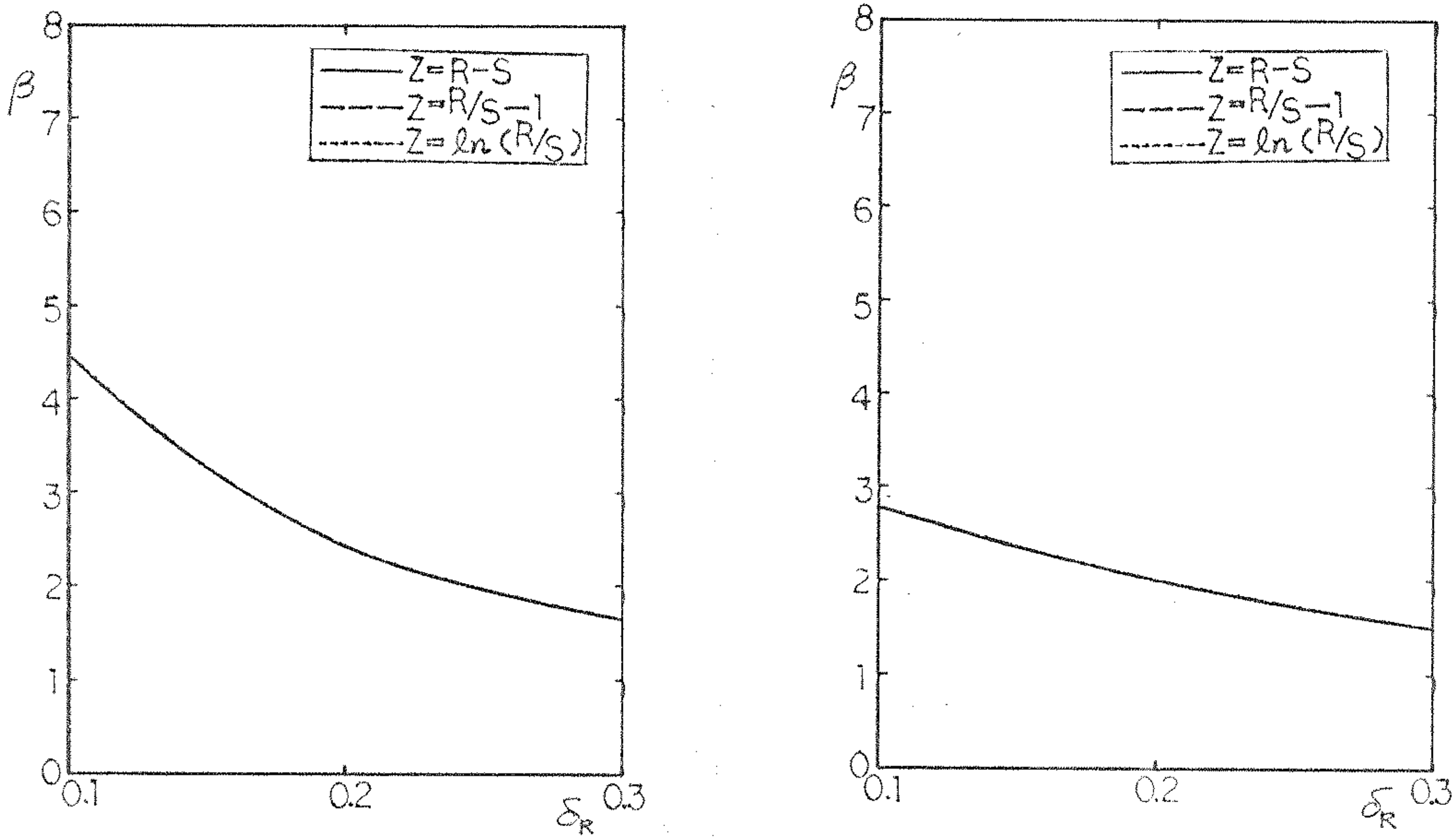
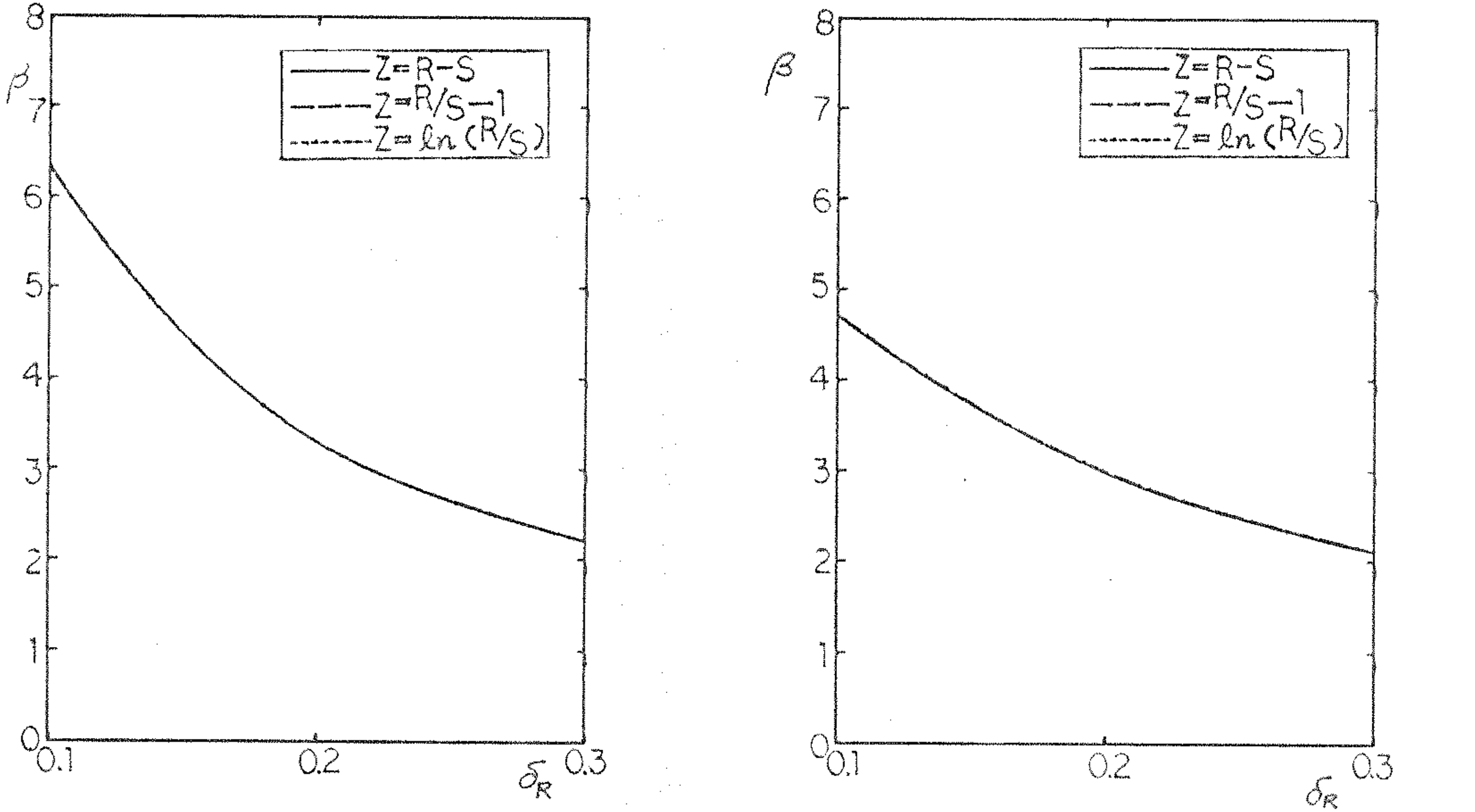


FIG. 4.1 (CONTINUED)



$\gamma = 2$, $\delta S = 0.1$ $\gamma = 2$, $\delta S = 0.3$

FIG.5.1 EFFECT OF NONLINEAR PERFORMANCE FUNCTION ($\gamma = 2$)
(FOSM METHOD)



$\gamma = 3$, $\delta S = 0.1$ $\qquad$ $\gamma = 3$, $\delta S = 0.3$

FIG. 5.2 EFFECT OF NONLINEAR PERFORMANCE FUNCTION ($\gamma = 3$)
(FOSM METHOD)

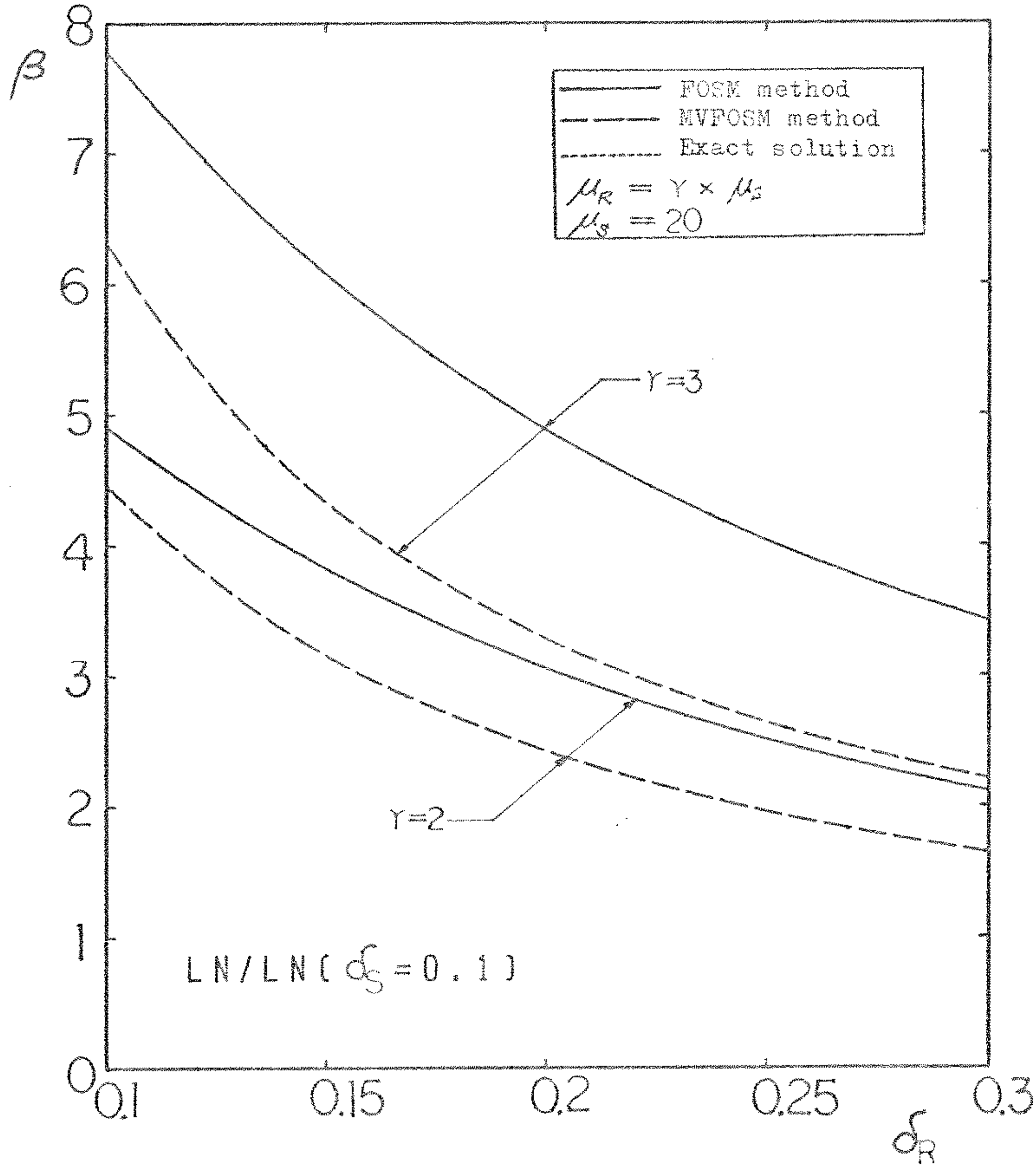


Fig. 5.3 Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

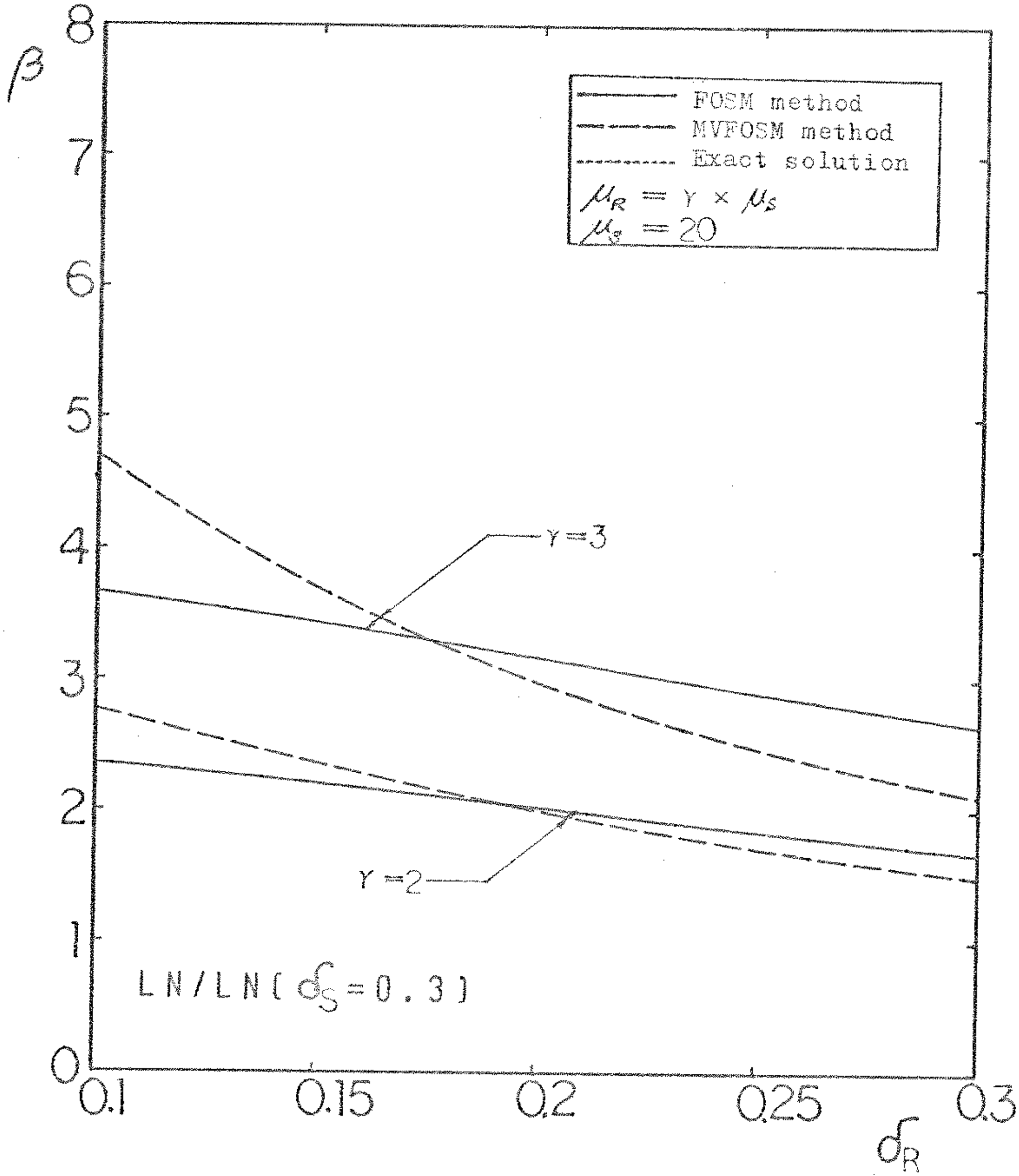


Fig. 5.4 Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

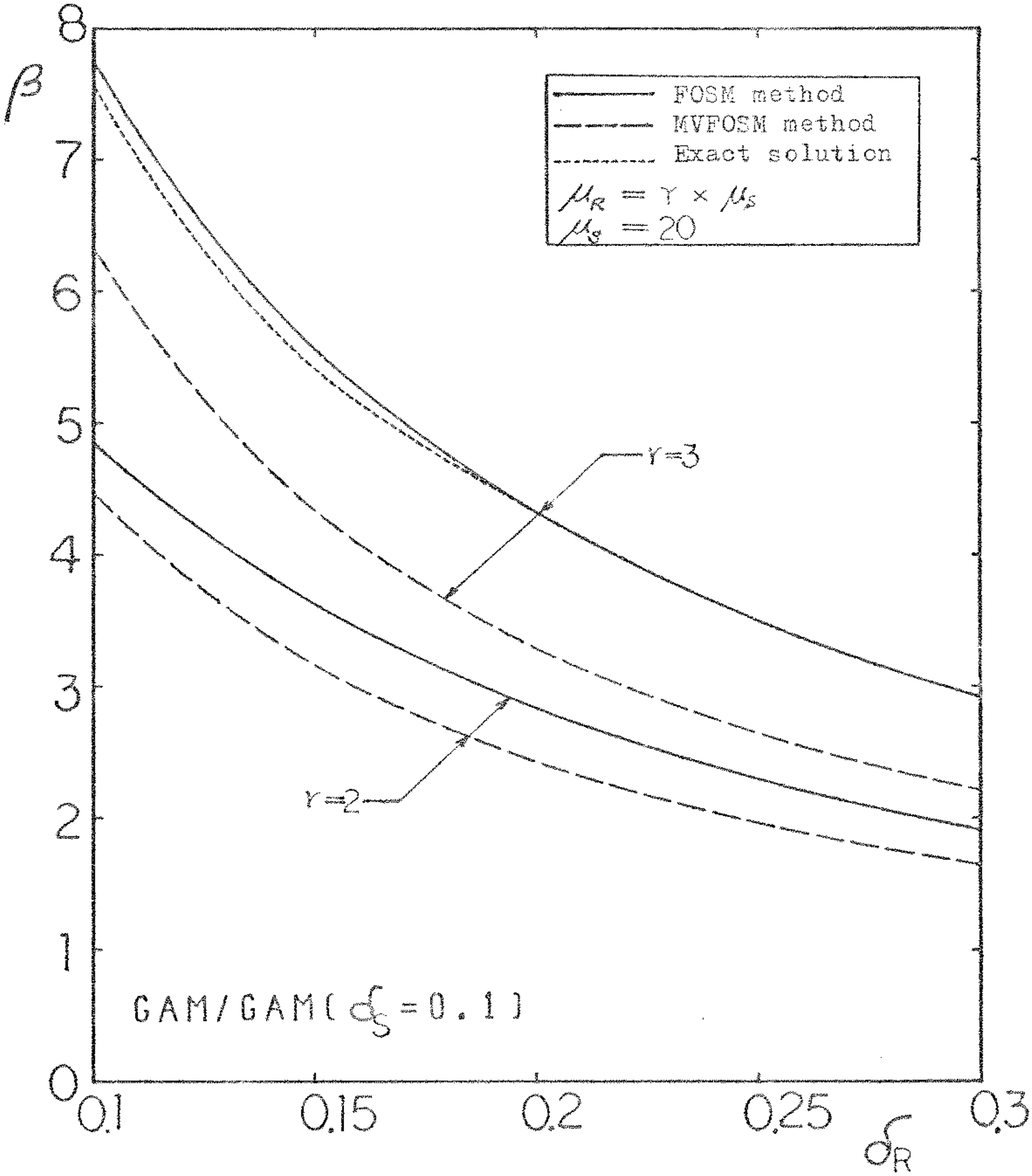


Fig. 5.5 Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

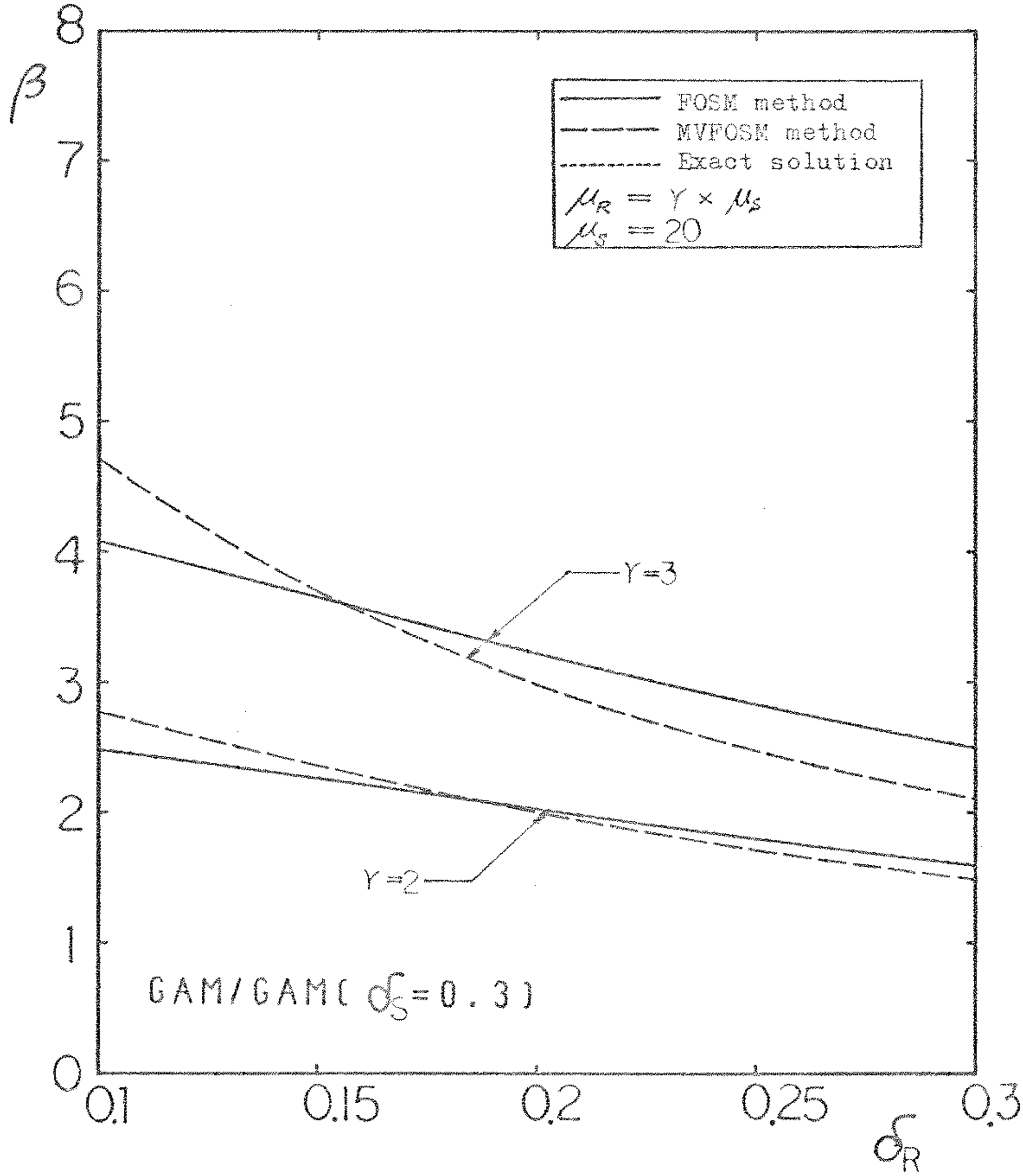


Fig 5.6 Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

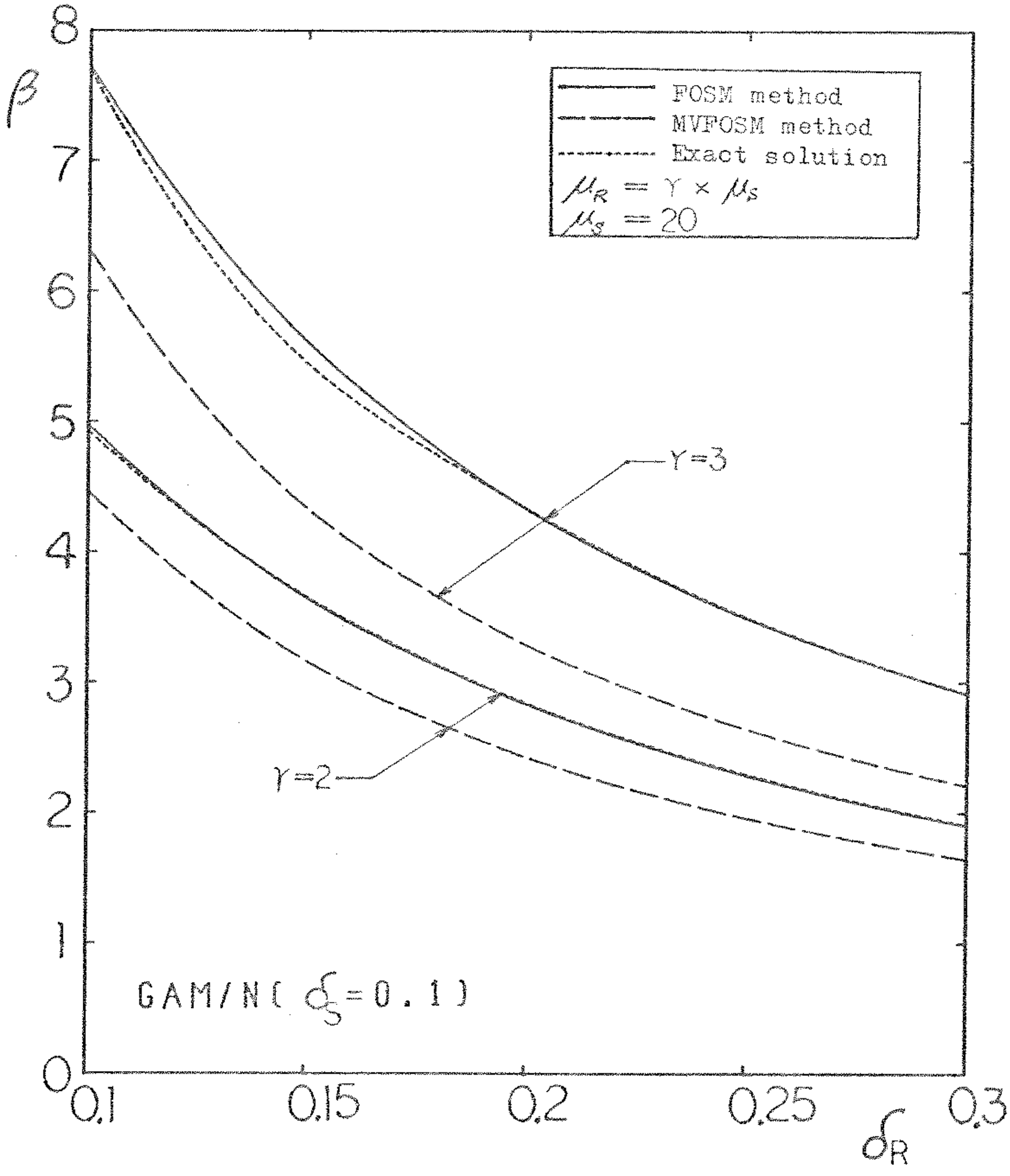


Fig. 5.7 Coparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

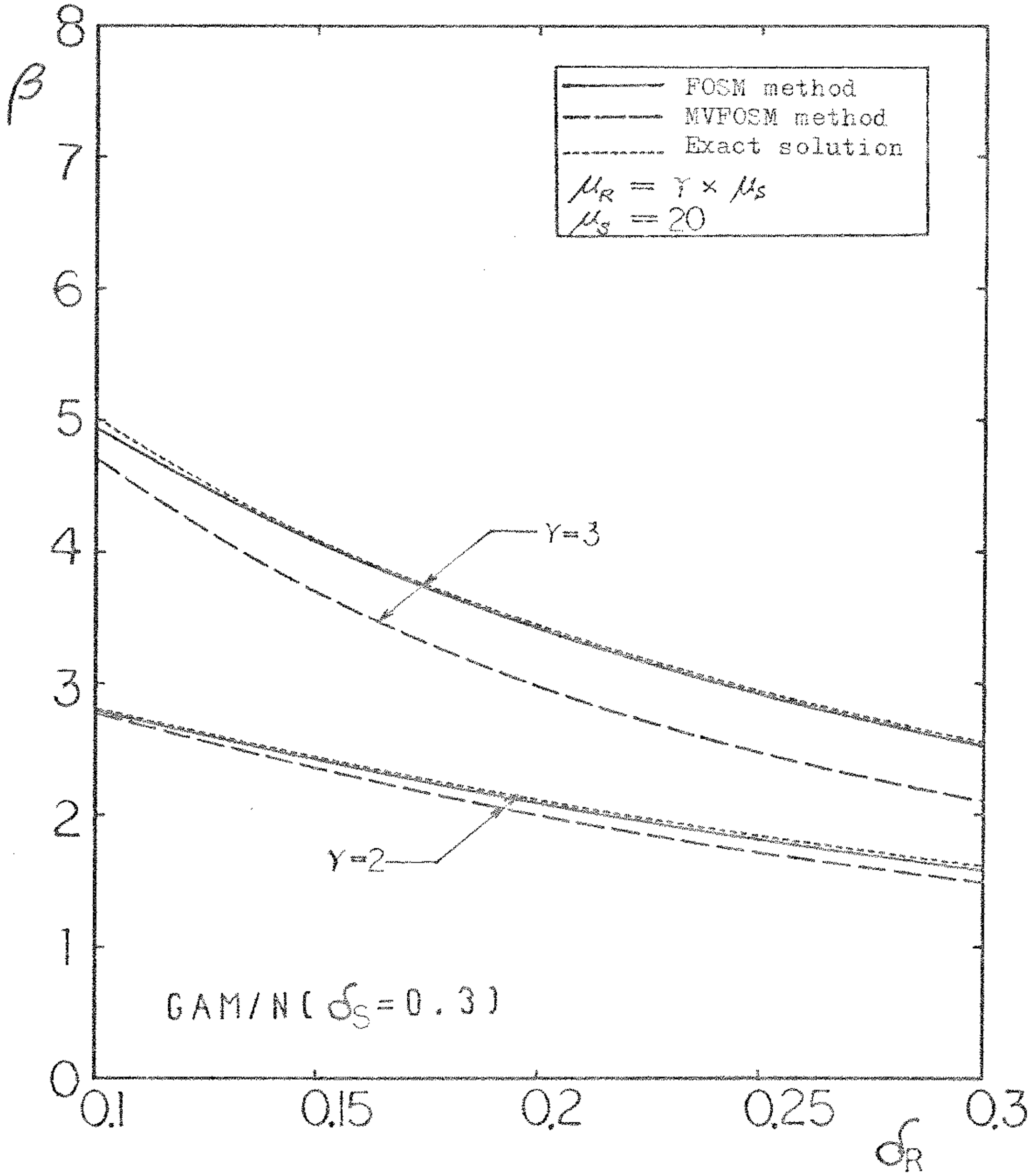


Fig. 5.8 Comparisons of safety indices, β, based on MVFOSM and FOSM mehtods with exact solutions

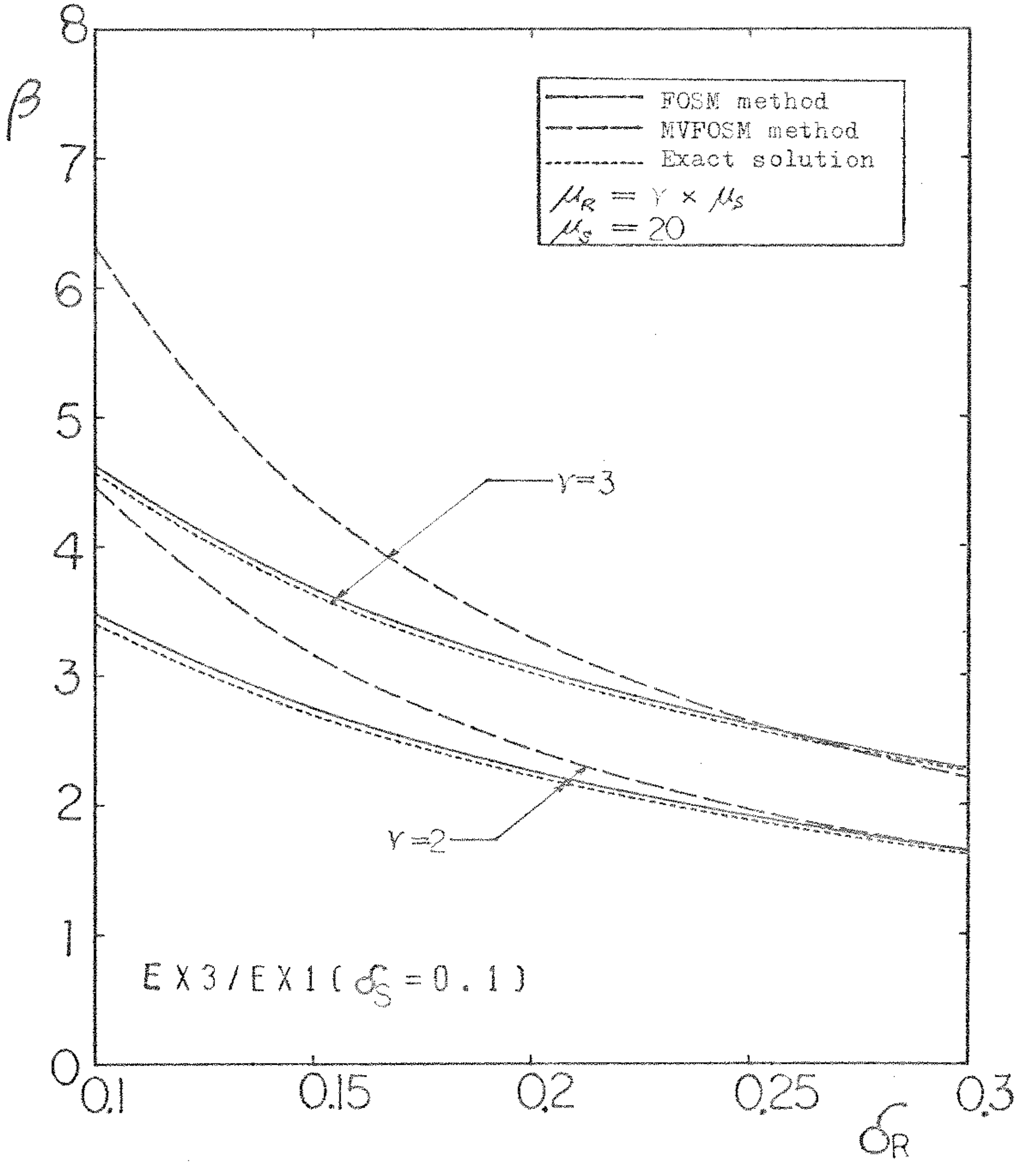


Fig. 5.9 Comparisons of safety indices, β, based on MVFOSM and FOSM methods with exact solutions

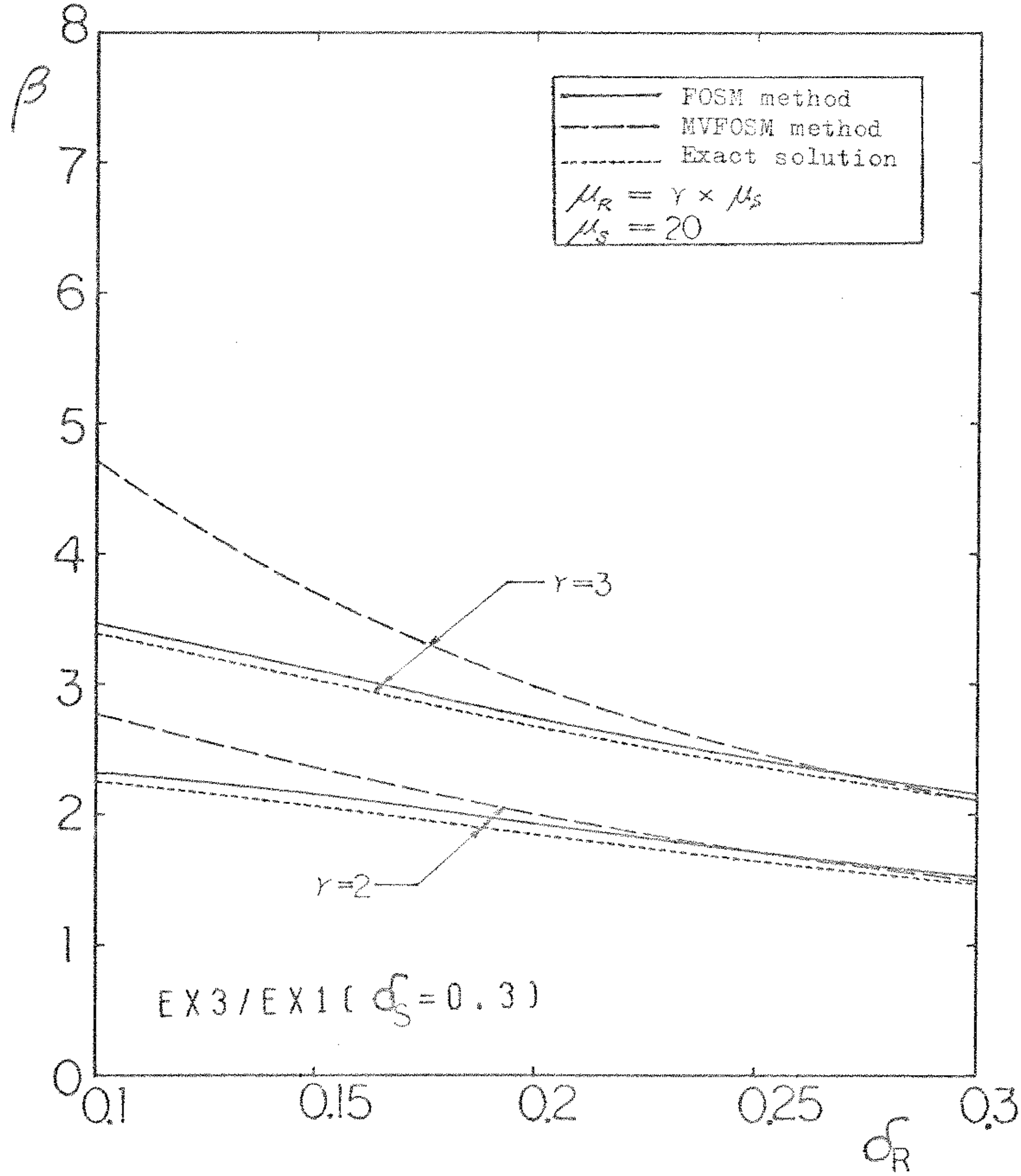


Fig. 5.10 Comparisons of safety indices, β, based on MVFOSM and FOSM methods with exact solutions

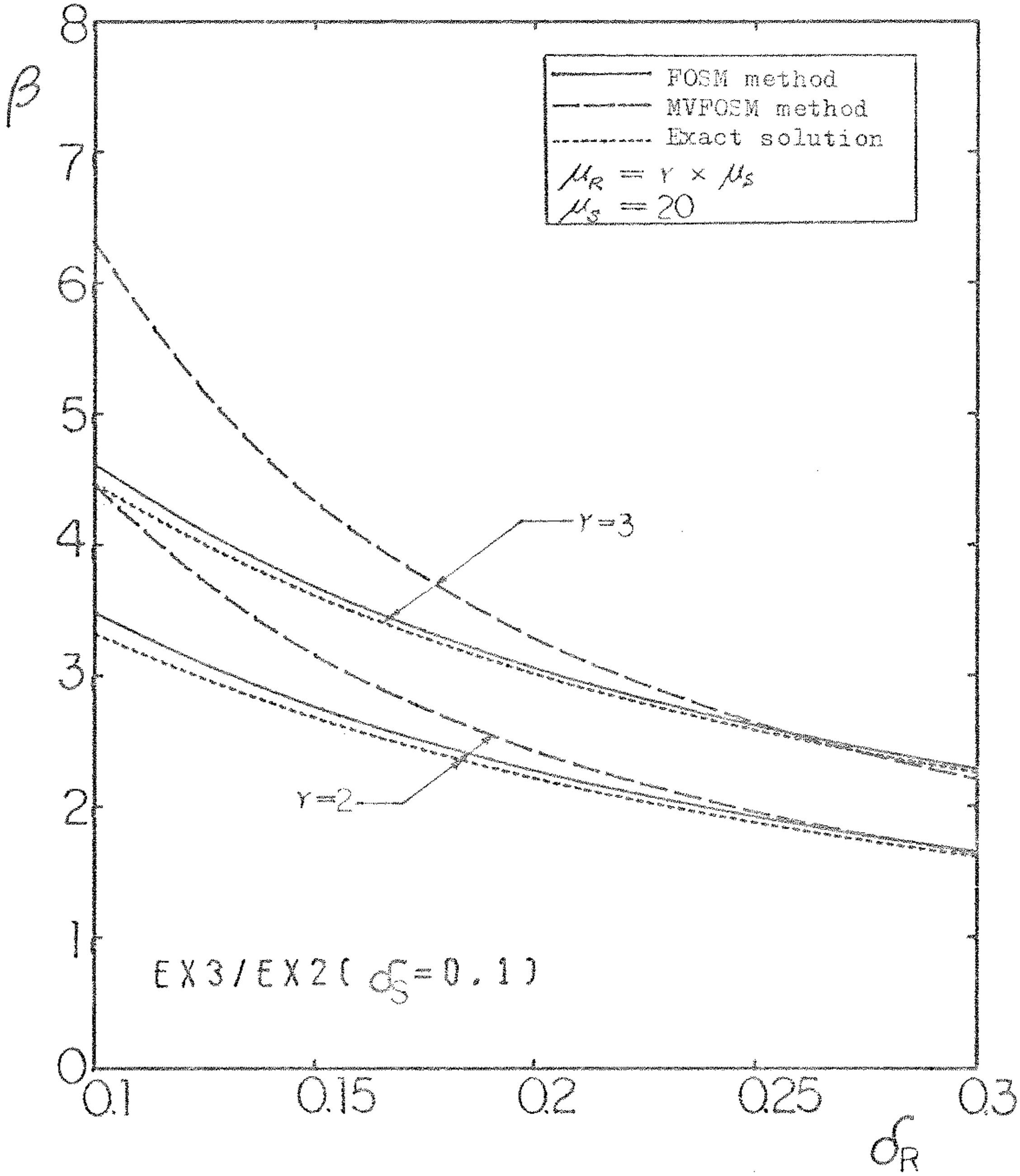


Fig. 5.11 Comparisons of safety indices, β, based on MVFOSM and FOSM mehtods with exact solutions

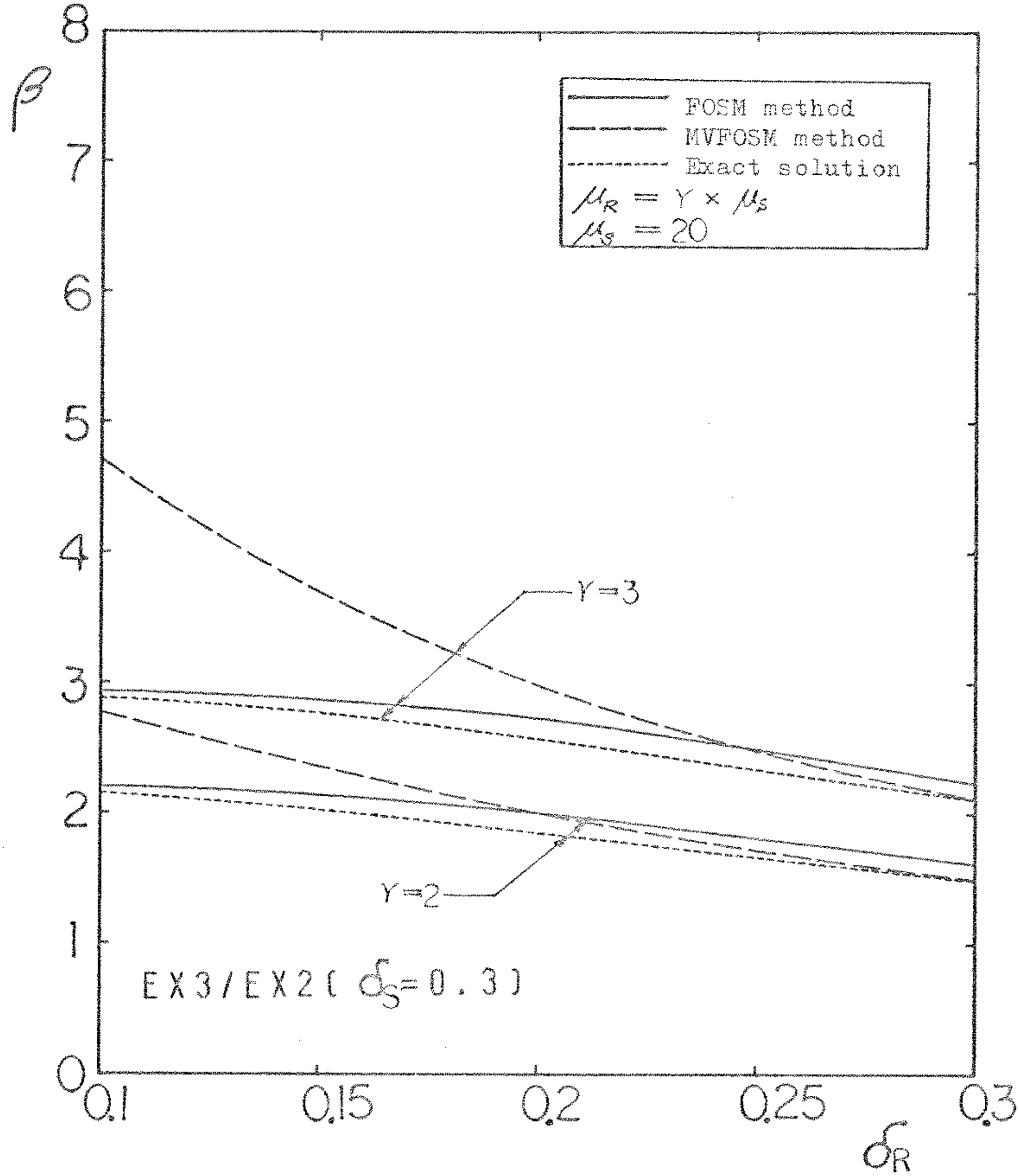


Fig. 5.12 Comparisons of safety indices, β , based on MVFOSM and FOSM methods with exact solutions

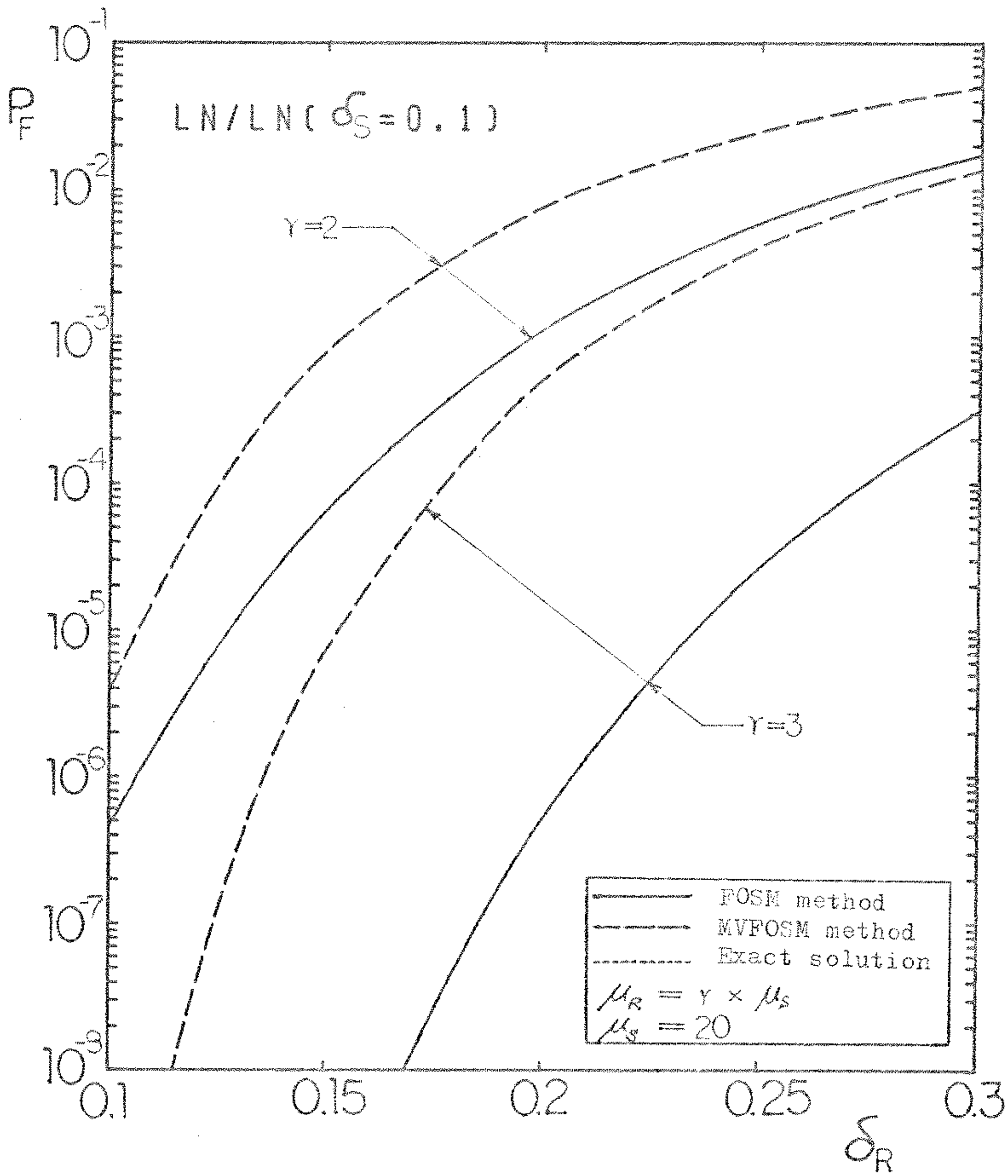


Fig. 5.13 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

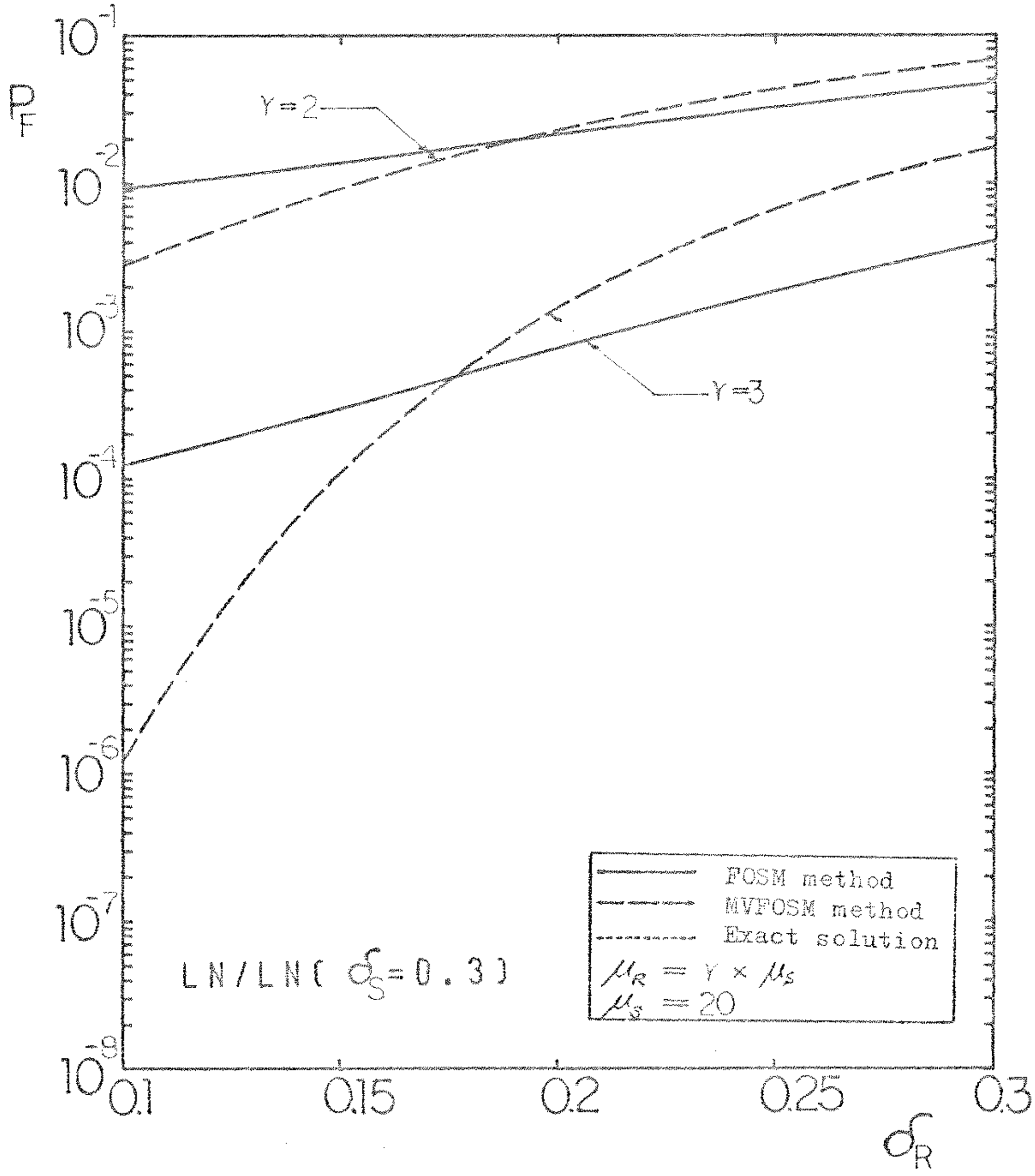


Fig. 5.14 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM mehtods with exact solutions

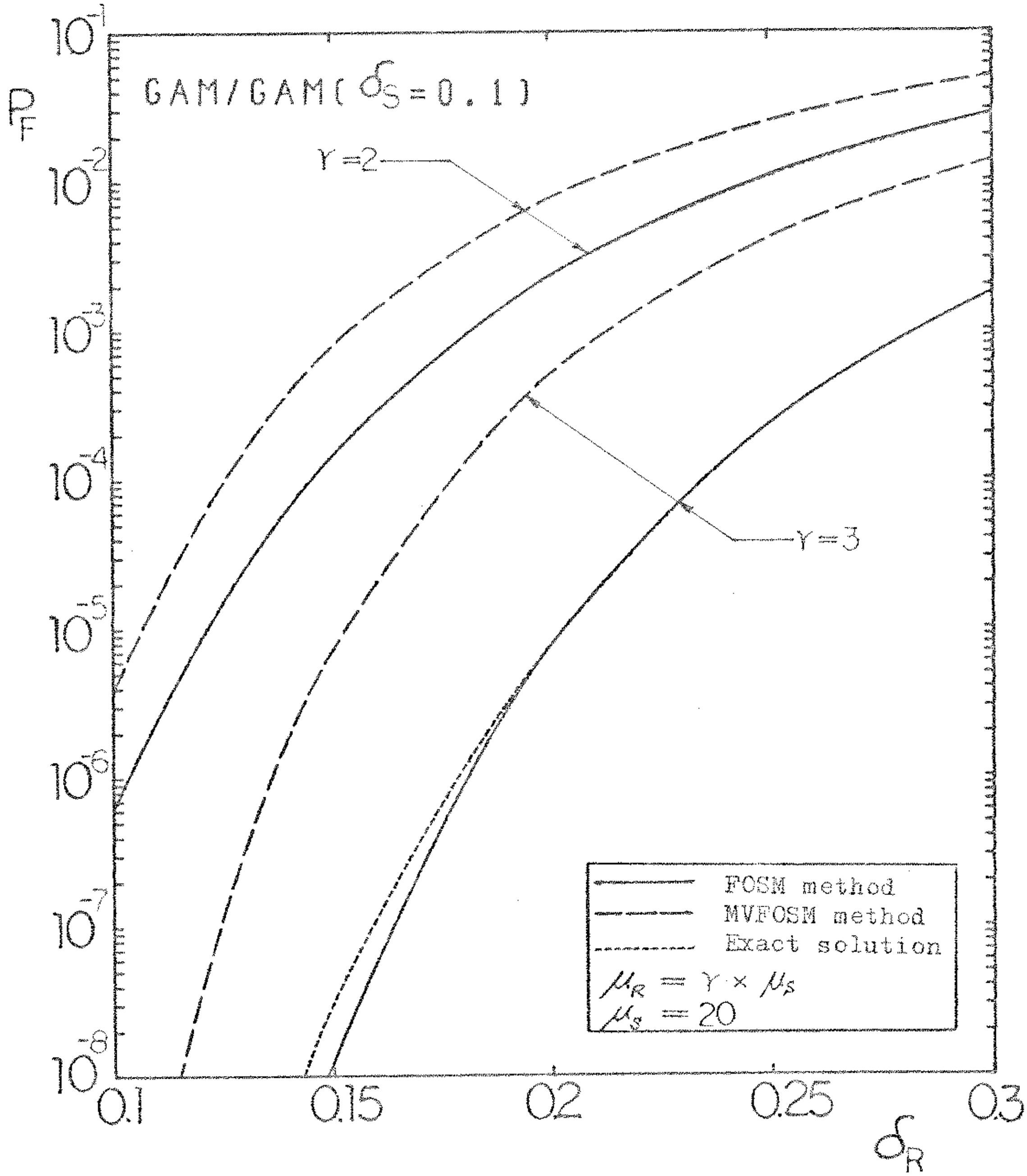


Fig. 5.15 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

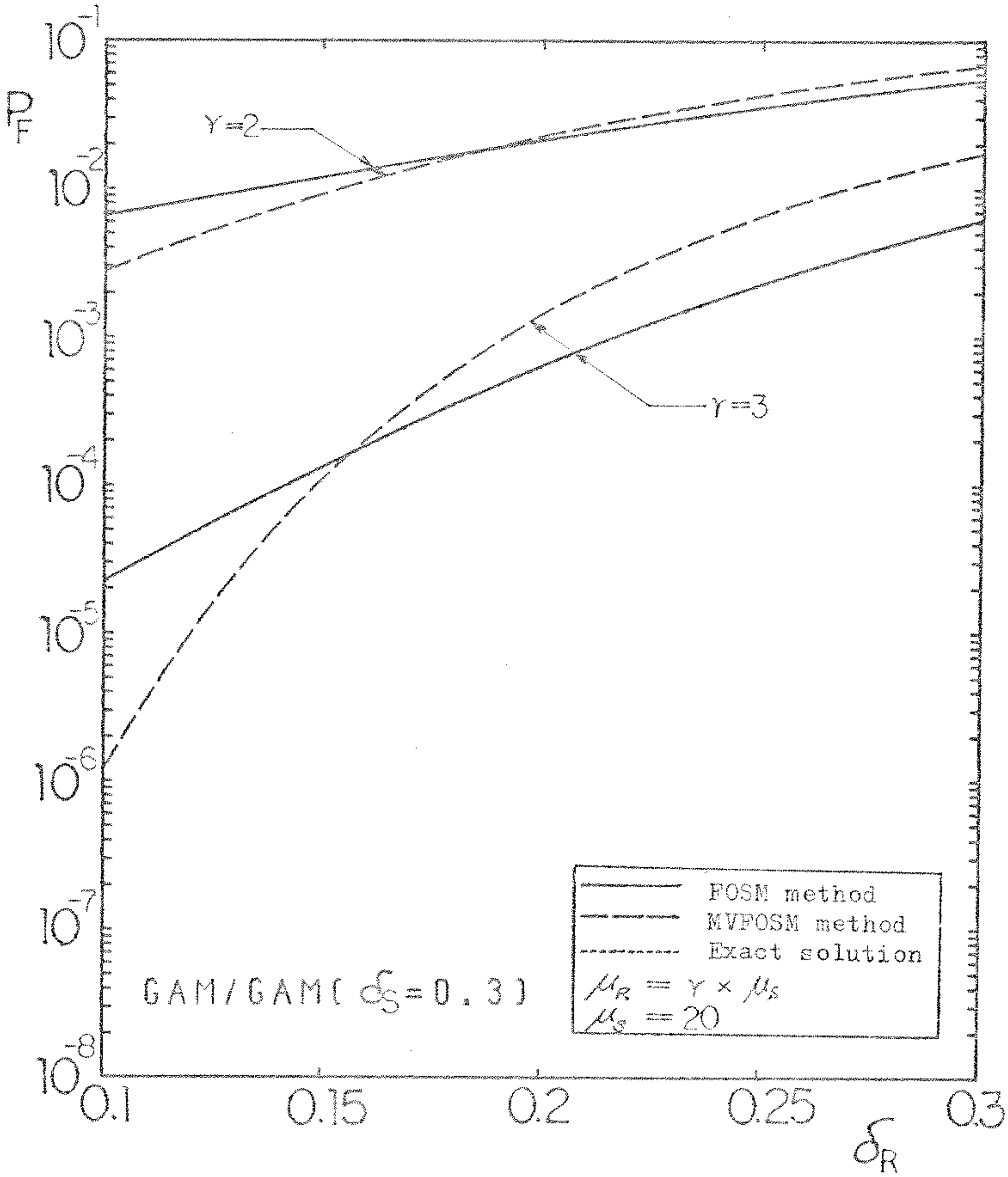


Fig. 5.16 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

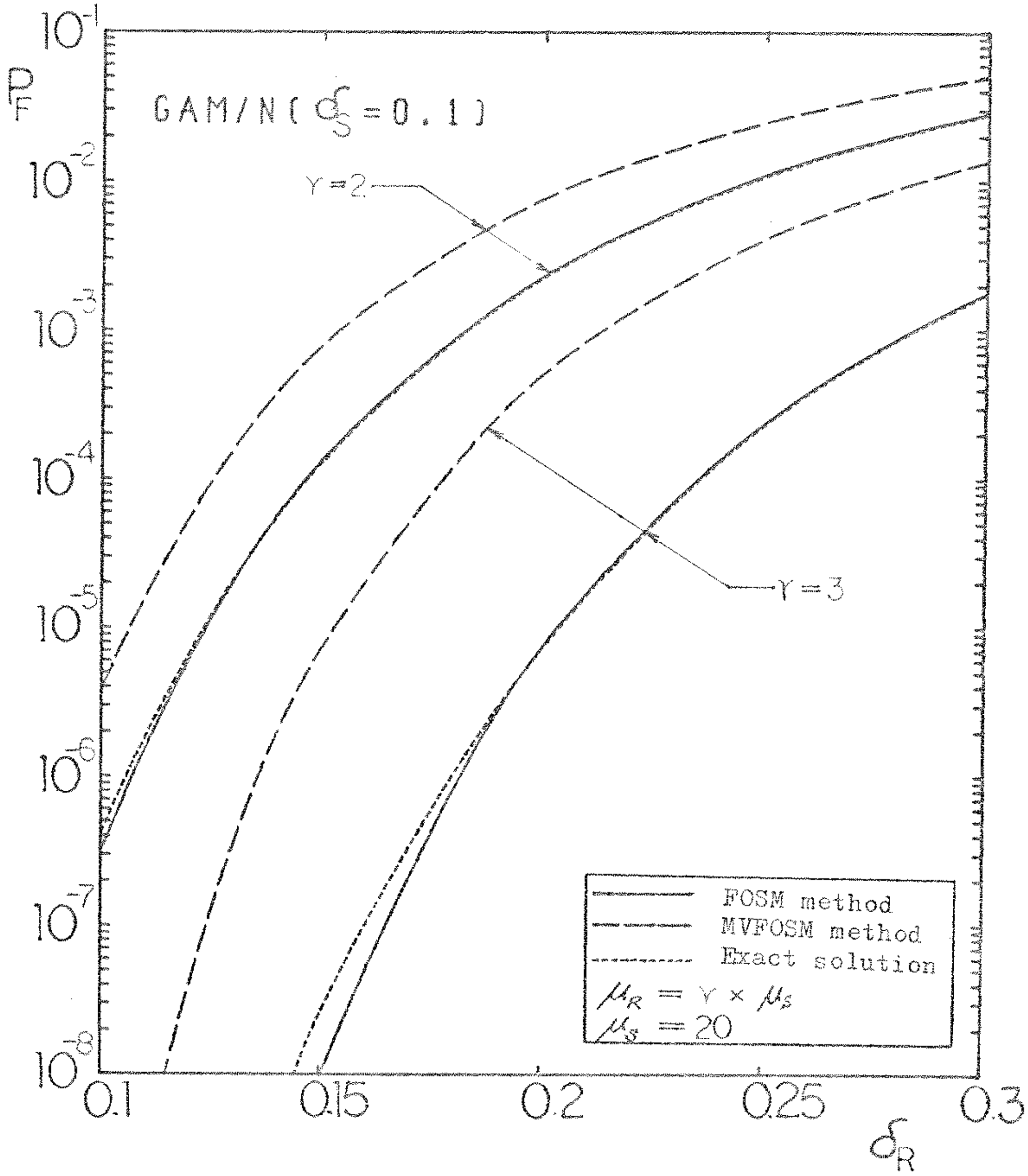


Fig. 5.17 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

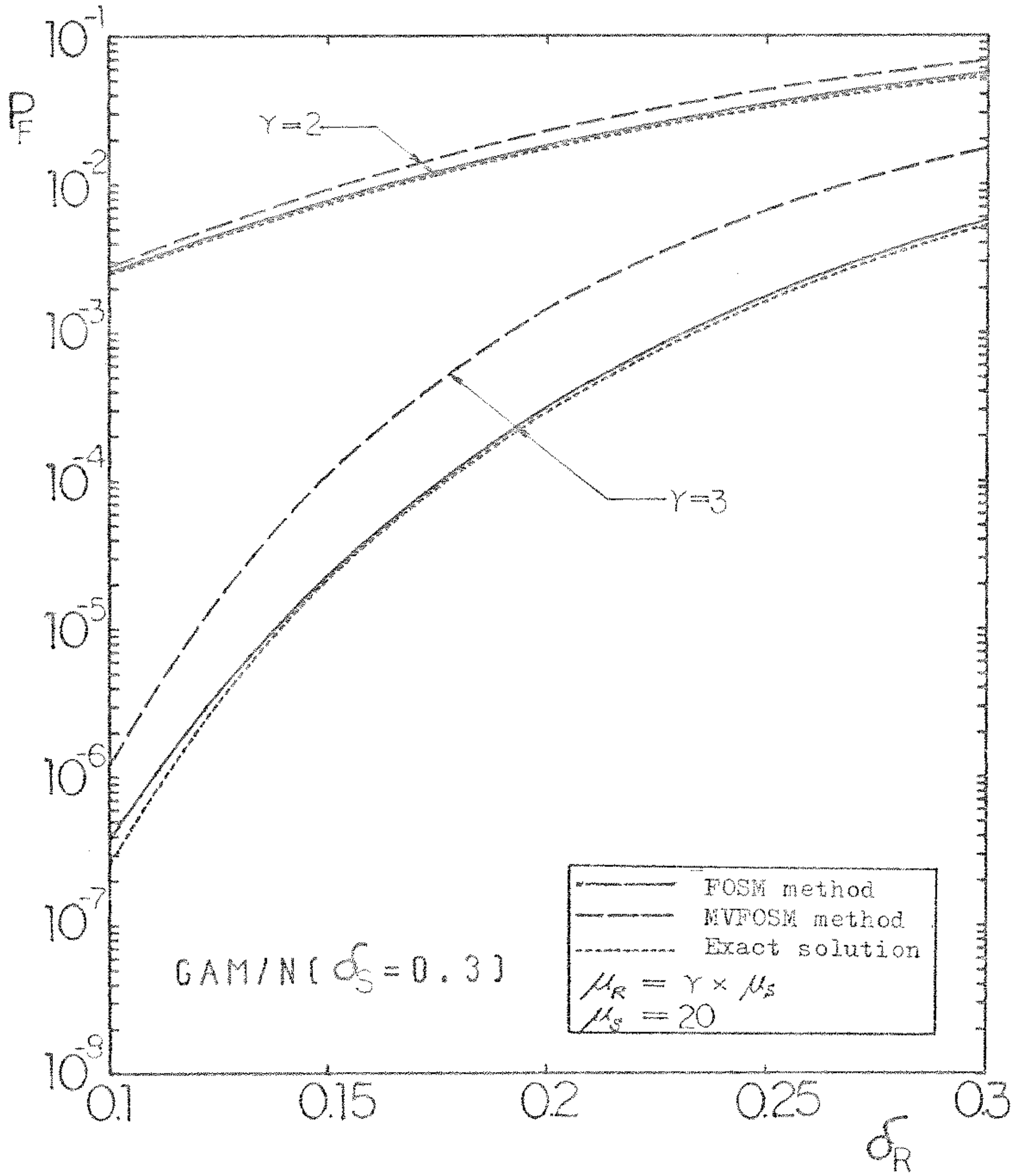


Fig. 5.18 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

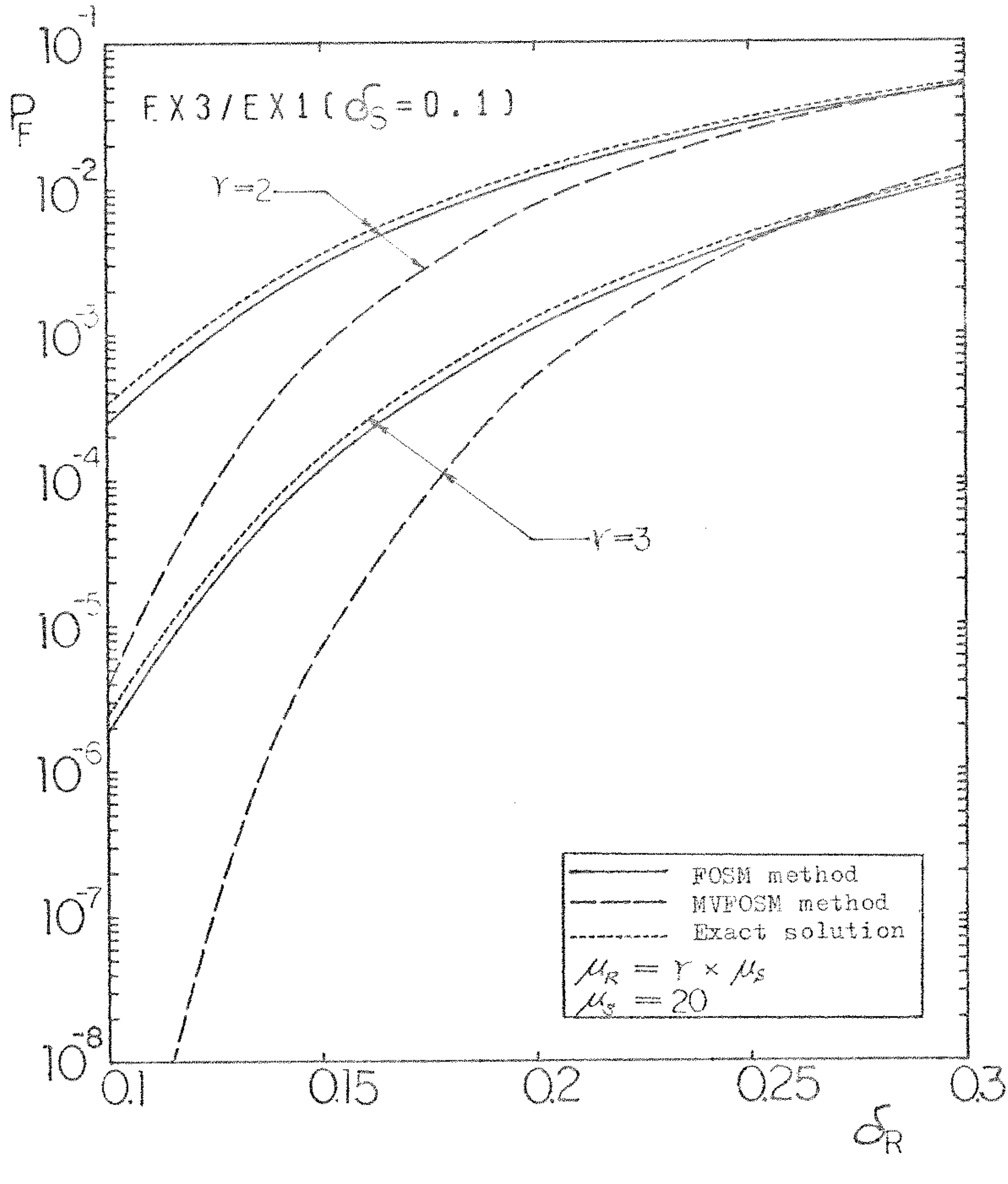


Fig. 5.19 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

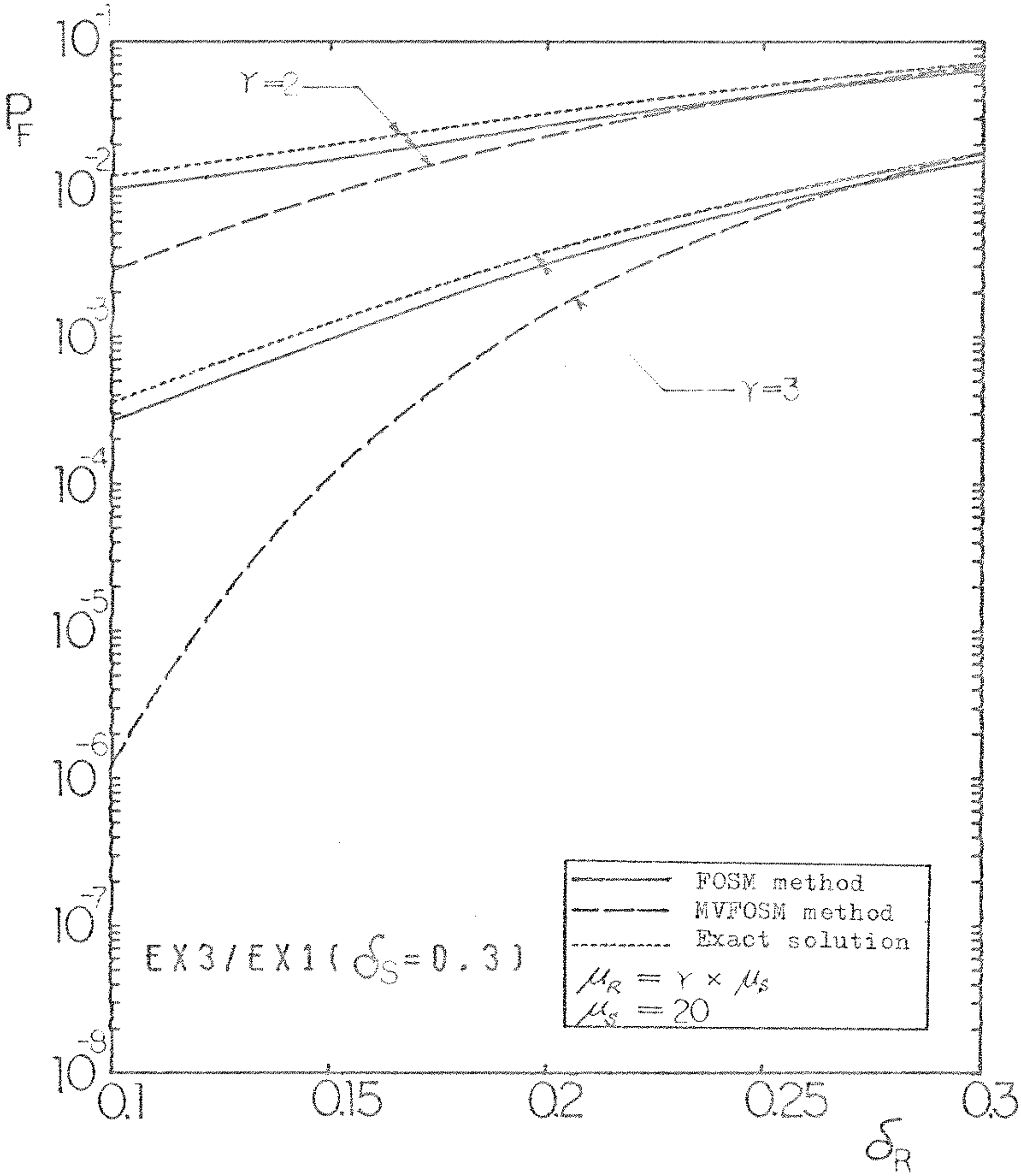


Fig. 5.20 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

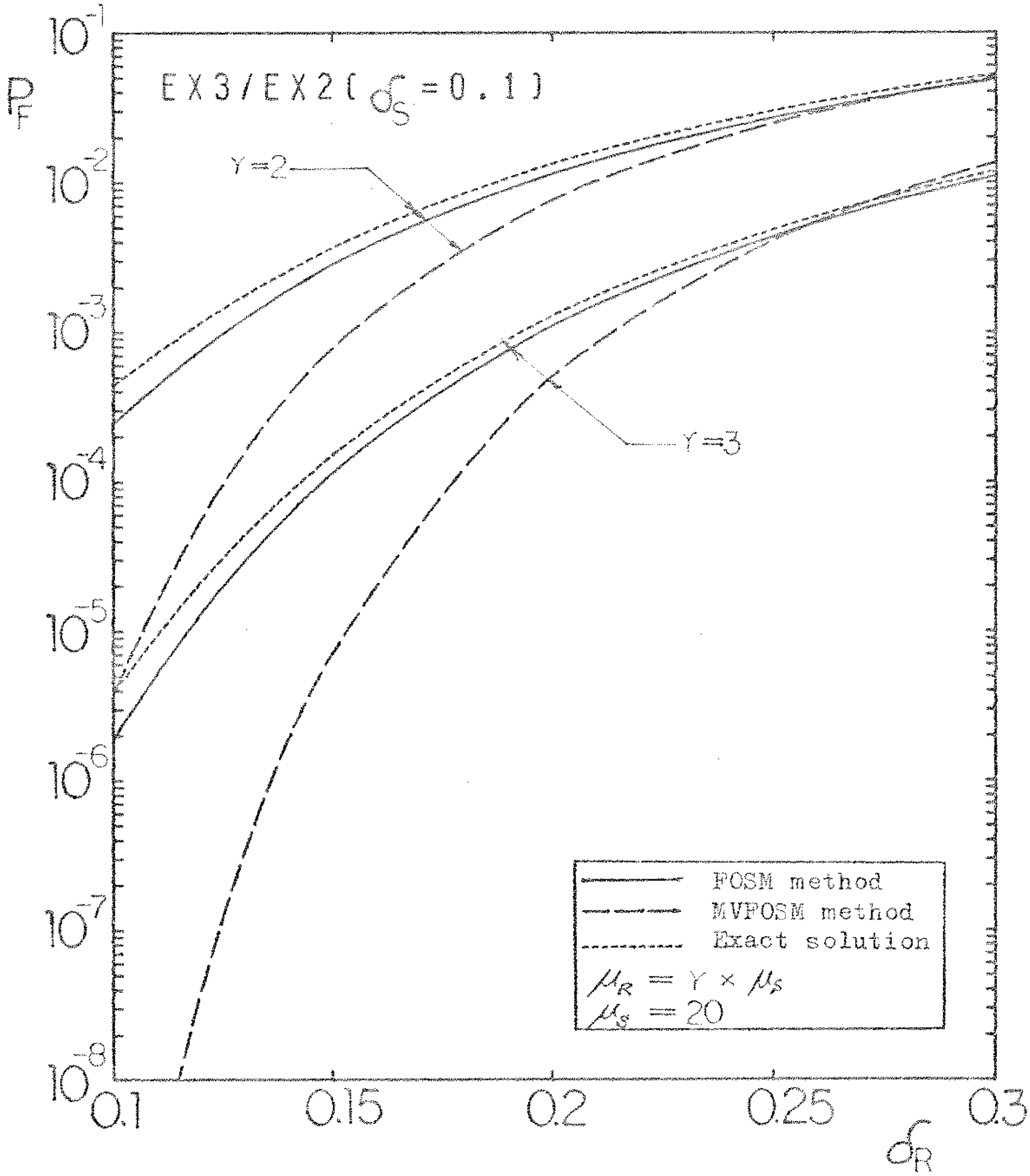


Fig. 5.21 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

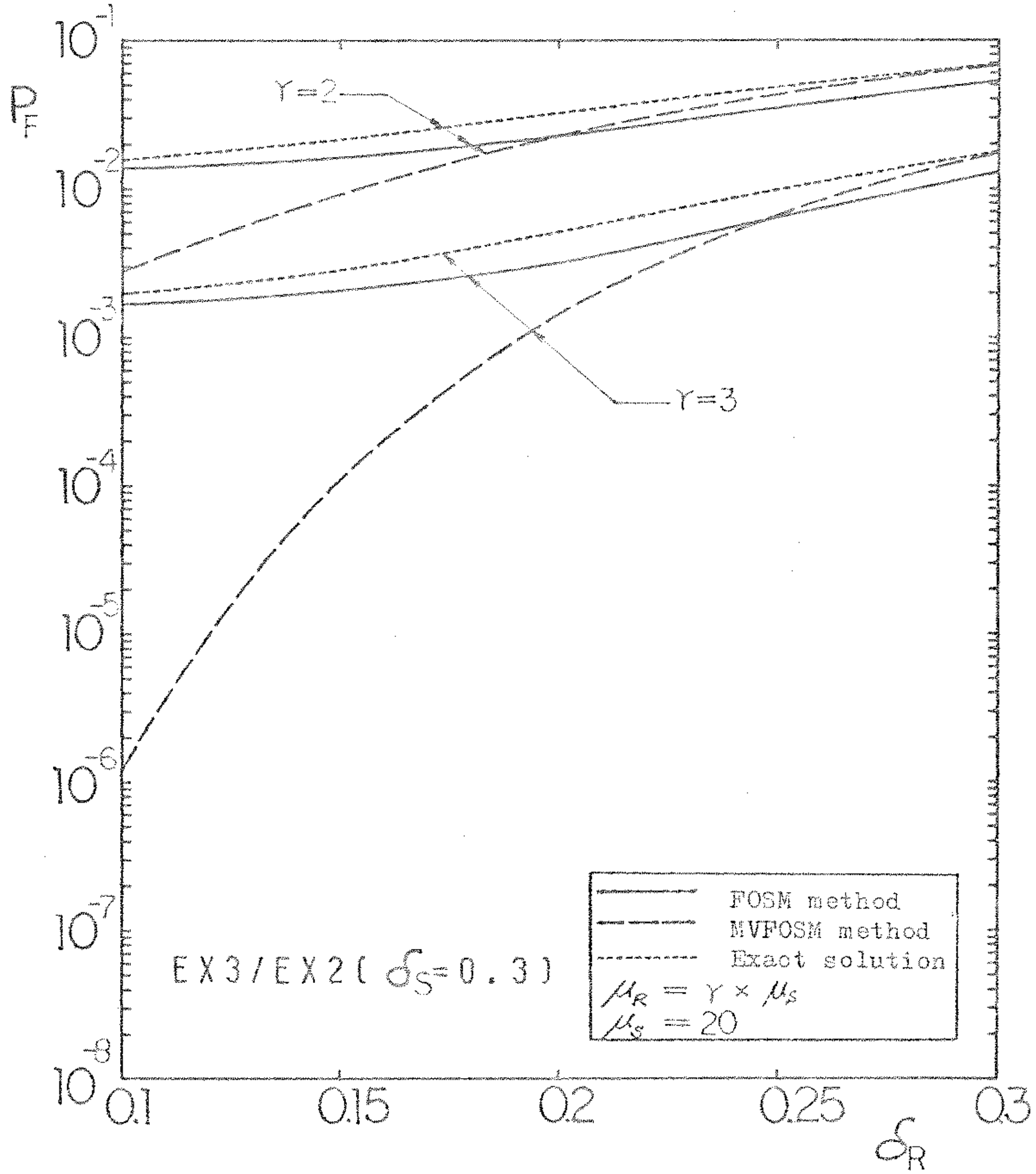


Fig. 5.22 Comparisons of probability of failure, P_f, based on MVFOSM and FOSM methods with exact solutions

Appendix I User's Guide for Program "FOSM"

1. Program organization

Program "FOSM" consists of a main program and the following subprograms. The flow chart of the program is shown in Fig. 4.1.

(1) EIGRS

The function of "EIGRS" is to find an eigen matrix $[Z]$, which transforms the covariance matrix $[V_X]$ into the diagonal matrix of eigen values $[V_W]$ through Eq.(3.6). To use this program, the covariance matrix $[V_X]$, which is real and symmetric, is transformed into a vector $\{V_{XA}\}$ by a subprogram VCVTFS. Both programs belong to IMSL routines.

(2) DG

The subprogram "DG" computes the numerical value of $\partial g_X/\partial X_i$ $(i=1,\text{------},n)$ for a certain value of $\{X\}=(x_1,\text{-----},x_n)$. This subprogram is provided by the user.

(3) DIGW

The subprogram "DIGW" computes the numerical value of $\partial g_W/\partial W_j$ $(j=1,\text{-----},n)$ as a linear combination of $\partial g_X/\partial X_i$ $(i=1,\text{-----},n)$ through Eq.(3.8).

(4) COSINE

The function of "COSINE" is to compute $\{\alpha_i\}$ defined by Eq.(3.3). $\{\alpha_i\}$ in the reduced space (W-space), which is defined by the following equation, is computed by using "COSINE" with "DG" and "DIGW".

$$\alpha_i=(\partial g_W/\partial W_i)_* \, \sigma_{Wi}\Big/\sqrt{\sum_{i=1}^{n}(\partial g_W/\partial W_i)_*^2\,\sigma_{Wi}^2}$$

(5) BETA

The function of "BETA" is to solve the following equation in terms of

$$G(\beta)=g_W(\mu_{W1}-\alpha_1\beta\sigma_{W1},\text{----------},\mu_{Wn}-\alpha_n\beta\sigma_{Wn})=0$$

so that the linearization point is on the failure surface. The equation is solved using Newton's method

$$\beta_{i+1}=\beta_i-G(\beta_i)/G'(\beta_i)$$

(6) LGN, GAM, TYPE1, TYPE2, WEIBL, EXPO

These subprograms aim to fit the non-normal distributions to "equivalent normal distributions" through Eqs.(3.12) and (3.13). The subprograms correspond to the following distributions.

LGN --------- Log normal dis.

GAM ------- Gamma dis.

TYPE1 ----- Type I Extreme dis.

TYPE2 ----- Type II Extreme dis.

WEIBL ----- Weibull dis. (Type III Extreme dis.)

EXPO ------- Exponential dis.

(7) WTOX

The function of "WTOX" is to transform variable $\{W\}$ into the original space(X-space) by multiplying the eigen matrix $[Z]$.

(8) GX(X)

This is a function declaration of the performance function $g_X(x_1,\text{-----},x_n)$. This program must be provided by the user.

(9) MDNOR

The function of "MDNOR" is to compute the failure probability from the safety index through Eq.(2.13).It belongs to IMSL routines.

2. Definition of parameters

NX	: Number of variables $(n \leqq 10)$
GX(X)	: Performance function $g_X(X_1,\ X_2,\text{-----},X_n)$
X(10)	: Variables of performance funtion $\{X\}=(X_1,\ X_2,\text{------},\ X_n)$
W(10)	: Variables transformed into uncorrelated space $\{W\}=(W_1,\ W_2,\text{------},\ W_n)$
NDIS(10)	: Type of distribution for each variable X_i 1 = Normal / Unspecified, 2 = Log-Normal, 3 = Gamma 4 = Type-1, 5=Type-2, 6 = Exponential, 7 = Weibull
Z(10,10)	: Eigen matrix to transform variables from correlated space (X-space) to uncorrelated space (W-space).
DGX(10)	: Partial derivatives of $g_X(X_1,\ X_2,\text{----},\ X_n)$ $(\partial g_X/\partial X_1,\partial g_X/\partial X_2,\text{----},\ \partial g_X/\partial X_n)$
DGW(10)	: Partial derivatives of $g_W(W_1,\ W_2,\text{-----},\ W_n)$ $(\partial g_W/\partial W_1,\partial g_W/\partial W_2,\text{----},\ \partial g_W/\partial W_n)$
EX(10)	: Mean vector of $\{X\}$ $\{\mu_X\}=(\mu_{X_1},\mu_{X_2},\text{-----},\mu_{X_n})$
EW(10)	: Mean vector of $\{W\}$ $\{\mu_W\}=(\mu_{W_1},\mu_{W_2},\text{-----},\mu_{W_n})$
VX(10,10)	: Covariance matrix of $\{X\}$ $[V_X]=\begin{bmatrix}\sigma_1^2 & \ddots & \\ \rho_{ij}\sigma_i\sigma_j & & \sigma_n^2\end{bmatrix}$

RO(10,10) : Correlation coefficients matrix $[\rho]=\begin{bmatrix}\rho_{11} & & \rho_{ij} \\ & \ddots & \\ & & \rho_{nn}\end{bmatrix}$

VW(10,10) : Transformed variance matrix $[V_W]=\begin{bmatrix}\sigma_{W1}^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{Wn}^2\end{bmatrix}$

SGW(10) : Standard deviation vector of $\{W\}$

$\{\sigma_W\}=(\sigma_{W1}, \sigma_{W2}, ------, \sigma_{Wn})$

AEX(10) : Modified mean vector after fitting to normal distributions

$\{\mu_X'\}=(\mu_1', \mu_2', ------, \mu_n')$

ASGX(10) : Modified standard deviation vector after fitting to normal distributions

$\{\sigma_X'\}=(\sigma_1', \sigma_2', -----, \sigma_n')$

AVX(10,10) : Modified covariance matrix after fitting to normal distributions

$[V_X']=\begin{bmatrix}\sigma_1'^2 & & \rho_{ij}\sigma_i'\sigma_j' \\ & \ddots & \\ & & \sigma_n'^2\end{bmatrix}$

AEW(10) : Mean vector $\{\mu_W'\}$, obtained by transformation of $\{\mu_X'\}$

ASGW(10) : Variance matrix $[V_W']$, obtained by transformation of $[V_X']$

IA : Index showing iteration number of calculation of $\{\alpha_i\}$

IB : Index showing iteration number of calculation of β

NIA : Allowable max. number of IA ($NIA \leqq 30$)

NIB : Allowable max. number of IB ($NIB \leqq 30$)

A(10,30) : A(i,IA) means α_i at the iteration stage of IA

B(30) : B(IB) means β at the iteration stage of IB

EPA : Convergence tolerance for α_i. Check is done by

$|A(i,IA) - A(i,IA-1)| < EPA \quad (i = 1, ---, n)$

EPB : Convergence tolerance for β. Check is done by

$|(B(IB) - B(IB-1))| / B(IB) < EPB$

PF : Probability of failure

NCASE : Total number of cases of analysis

3. Programs provided by a user

As described in 1. a user of this program is required to provide two subprograms as follows.

(1) The performance function is defined through a FUNCTION STATEMENT. The user is required to program the followings.

```
FUNCTION GX(X)
DIMENSION X(1)
GX= ---------
RETURN
END
```

Example: For the performance function, $g_X(X_1, X_2, X_3) = X_1^2 - X_2X_3$, the program includes the following:

```
GX=X(1)*X(1) - X(2)*X(3)
```

(2) Derivatives of the performance function are defined in the subroutine DG as follows:

```
SUBROUTINE DG(X,DGX)
DIMENSION X(1),DGX(1)
DGX(1)= ---------
DGX(2)= ---------
          :
DGX(·)= ---------
RETURN
END
```

The user is required to program the parts indicated by "---------".

Example: For the performance function described above, it is programmed as follows :

```
DGX(1)=2.0*X(1)
DGX(2)=-X(3)
DGX(3)=-X(2)
```

4. Data input to FOSM

(1) Master control card (2I5)

Columns	Variable	Entry
1---5	NX	Total number of variables ($NX \leq 10$)
6---10	NCASE	Number of cases of analysis

(2) Convergence control card (2F10.5 , 2I5)

Columns	Variable	Entry
1---10	EPA	Convergence tolerance for the computation of $\{\alpha_i\}$
11---20	EPB	Convergence tolerance for the computation of β
21---25	NIA	Maximum number of iteration for the computation of $\{\alpha_i\}$ ($NIA \leqq 30$)
26---30	NIB	Maximum number of iteration for the computation of β ($NIB \leqq 30$)

(3) Variable data card

Card 1 Heading (7A10)

Columns	Variable	Entry
1---70	TIT1--TIT7	Enter the heading information for use in labeling the output for each case

(Note) Begin each new set of data with a new heading card.

Card 2 Distributions and mean values (I5, F10.3)

Columns	Variable	Entry
1---5	NDIS(10)	Type of distribution of each variable
6---15	EX(10)	Mean value of each variable

(Note 1) This card is prepared for each variable, and put in order following the sequence of variables. Therefore, total number of card 2 is NX.

(Note 2) NDIS : 1---Normal/Unspecified

2---Lognormal

3---Gamma

4---Extreme (Type I)

5---Extreme (Type II)

6---Exponential

7---Weibull (Extreme Type III)

Card 3 Covariances (7F10.3)

Columns	Variable	Entry
1---10	VX(1,1)	Enter the first row of the covariance matrix
11---20	VX(1,2)	at first, then move to the second row, and
21---30	VX(1,3)	etc. When the first card is full, continue
etc.		to the second card, and etc.

(Note) When the input for card 3 is over, a user can go to the next case. Input another set of variable data from card 1.

4. Output of FOSM

Example of the output is shown in Appendix III.

Appendix II Computer Program "FOSM"

```
               PROGRAM FOSM(INPUT,TAPE5=INPUT,OUTPUT,TAPE6=OUTPUT)
000003         DIMENSION EX(10),VX(10,10),XM(50),NDIS(10),D(10),U(65),EW(10),
              1VW(10,10),Z(10,10),SGW(10),DG(10),DGW(10),A(10,30),B(50),
              2W(10),X(10),AEW(10),ASGW(10),SEX(10),ASG(10),FO(10,10),RVX(10,10)
C***********************************************************************INPUT
000003         READ(5,100) NX,NCASE
000013     100 FORMAT(2I5)
000013         WRITE(6,200) NX,NCASE
000023     200 FORMAT(1H1,10X,10HDATA INPUT/1H0,24H(NUMBER OF VARIABLES)=,I5
              1/1H0,22H(CASE NUMBER OF CASES)=,I5)
000023         READ(5,105) EPA,EPB,NIA,NIB
000037     105 FORMAT(2F10.5,2I5)
000037         WRITE(6,205) EPA,EPB,NIA,NIB
000053     205 FORMAT(1H0,21HCONVERGENCE CONDITION/1H0,4HEPA=,F10.5,5X,4HEPB=,
              1F10.5/1H0,4HNIA=,I5,5X,4HNIB=,I5)
000053         ICASE=1
000054     110 READ(5,111) TIT1,TIT2,TIT3,TIT4,TIT5,TIT6,TIT7
000076     111 FORMAT(7A10)
000076         WRITE(6,112) TIT1,TIT2,TIT3,TIT4,TIT5,TIT6,TIT7
000120     112 FORMAT(1H1,7A10)
000120         WRITE(6,150)
000124     150 FORMAT(1H0,4X,1HI,1X,4HNDIS,5X,2HEX)
000124         DO 10 I=1,NX
000126         READ(5,101) NDIS(I),EX(I)
000135     101 FORMAT(I5,F10.5)
000135         WRITE(6,201) I,NDIS(I),EX(I)
000147     201 FORMAT(1H0,2I5,F10.5)
000147      10 CONTINUE
000152         WRITE(6,160)
```

```
000135      160 FORMAT(1H0,20HCDIS=1 NORMAL OR UNSPECIFIED/1H0,
               116HCDIS=2 LOGNORMAL/1H0,12HCDIS=3 GAMMA/1H0,
               212HCDIS=4 TYPE1/1H0,12HCDIS=5 TYPE2/1H0,
               318HCDIS=6 EXPONENTIAL/1H0,14HCDIS=7 WEIBULL)
000155          READ(5,102) ((VX(I,J),J=1,NX),I=1,NX)
000174      102 FORMAT(7F10.3)
000174          DO 11 I=1,NX
000176          DO 11 J=1,NX
000177       11 RO(I,J)=VX(I,J)/SQRT(VX(I,I)*VX(J,J))
000216          WRITE(6,302) ((I,J,VX(I,J),J=1,NX),I=1,NX)
000240      302 FORMAT(1H0,17HCOVARIANCE MATRIX/1H0,5(3HVX(,I2,1H,,I2,2H)=,F10.3,
               15X))
         C*********************************************************FIND EIGEN MATRIX
000240          IX=10
000241          IJOB=2
000242          CALL VCVTFS(VX,NX,IX,VXA)
000245          CALL EIGRS(VXA,NX,IJOB,W,Z,IX,WK,IER)
000255          WRITE(6,205) ((I,J,Z(I,J),J=1,NX),I=1,NX)
000300      205 FORMAT(1H0,12HEIGEN MATRIX/1H0,5(3HZ(,I2,1H,,I2,2H)=,F10.3,5X))
         C*****************************************************TRANSFORM INTO W-SPACE
000300          DO 14 I=1,NX
000302          DO 14 J=1,NX
000303          VW(I,J)=0.0
000306          DO 14 L=1,NX
000310          DO 14 K=1,NX
FOSM     FORTRAN COMPILATION           RUN 2.300-75274       03 DEC 80 10:57:50  PAGE NO. 2

000311       14 VW(I,J)=VW(I,J)+Z(K,I)*VX(K,L)*Z(L,J)
000337          WRITE(6,207) ((I,J,VW(I,J),J=1,NX),I=1,NX)
000361      207 FORMAT(1H0,29HTRANSFORMED COVARIANCE MATRIX/1H0,
               15(3HVW(,I2,1H,,I2,2H)=,F10.3,5X))
000361          DO 15 I=1,NX
000363       15 SGW(I)=SQRT(VW(I,I))
000373          DO 16 I=1,NX
000374          EW(I)=0.0
000375          DO 16 J=1,NX
000377       16 EW(I)=EW(I)+Z(J,I)*EX(J)
000412          WRITE(6,208) (I,EW(I),I=1,NX)
000426      208 FORMAT(1H0,22HTRANSFORMED MEAN VALUE/1H0,
               15(3HEW(,I2,2H)=,F10.3,5X))
         C*******************************************************INITIAL ESTIMATE
000426          IA=1
000427          CALL DG(EX,DGX)
000431          CALL DIGW(NX,DGX,DGW,Z)
000434          CALL COSINE(NX,IA,DGW,SGW,A)
000440          EG=GX(EX)
000443          VGW=0.0
000444          DO 17 I=1,NX
000445          DO 17 J=1,NX
000446       17 VGW=VGW+RO(I,J)*DGW(I)*DGW(J)*SGW(I)*SGW(J)
000466          BT0=EG/SQRT(VGW)
000471          IF(BT0.LE.0.0) GO TO 81
000472          WRITE(6,507) BT0
000500      507 FORMAT(1H0,13HINITIAL BETA=,F10.3)
         C************************************************************MODIFY BETA
000500          CALL BETA(NX,IA,BT0,BT,W,X,SGW,EW,Z,A)
000512          IB=1
000513          B(IB)=BT
000515          DO 18 I=1,NX
000517       18 W(I)=EW(I)-A(I,IA)*B(IB)*SGW(I)
000533          CALL WTOX(NX,Z,W,X)
         C**********************************************FIT TO NORMAL DISTRIBUTION
000536       30 DO 40 I=1,NX
```

```
000540          EXX=EX(I)
000542          SIG=SQRT(VX(I,I))
000544          XX=X(I)
000550          IF(NDIS(I).EQ.1) GO TO 41
000552          IF(NDIS(I).EQ.2) GO TO 42
000554          IF(NDIS(I).EQ.3) GO TO 43
000556          IF(NDIS(I).EQ.4) GO TO 44
000560          IF(NDIS(I).EQ.5) GO TO 45
000562          IF(NDIS(I).EQ.6) GO TO 46
000564          IF(NDIS(I).EQ.7) GO TO 47
000566       41 AEX(I)=EXX
000570          ASGX(I)=SIG
000572          GO TO 40
000572       42 CALL LGN(EXX,SIG,XX,EE,SGM)
000576          AEX(I)=EE
000600          ASGX(I)=SGM
000602          GO TO 40
000602       43 CALL GAM(EXX,SIG,XX,EE,SGM)
FUSN     FORTRAN COMPILATION              RUN 2.3Q0-75274        02 DEC 80 10:57:50  PAGE NO. 

000606          AEX(I)=EE
000610          ASGX(I)=SGM
000612          GO TO 40
000612       44 CALL TYPE1(EXX,SIG,XX,EE,SGM)
000616          AEX(I)=EE
000620          ASGX(I)=SGM
000622          GO TO 40
000622       45 CALL TYPE2(EXX,SIG,XX,EE,SGM)
000626          AEX(I)=EE
000630          ASGX(I)=SGM
000632          GO TO 40
000632       46 CALL EXPO (EXX,SIG,XX,EE,SGM)
000636          AEX()=EE
000640          ASGX(I)=SGM
000642          GO TO 40
000642       47 CALL WEIBL(EXX,SIG,XX,EE,SGM)
000646          AEX(I)=EE
000650          ASGX(I)=SGM
000652       40 CONTINUE
C*******************************************************NEW MEAN,COVARIANCE,EIGEN MATRIX
000655          DO 50 I=1,NX
000656          DO 50 J=1,NX
000657       50 AVX(I,J)=RO(I,J)*ASGX(I)*ASGX(J)
000674          CALL VCVTFS(AVX,NX,IX,VXA)
000677          CALL EIGRS(VXA,NX,IJOB,D,Z,IX,WK,IER)
000707          DO 51 I=1,NX
000711          DO 51 J=1,NX
000712          VW(I,J)=0.0
000715          DO 51 L=1,NX
000717          DO 51 K=1,NX
000720       51 VW(I,J)=VW(I,J)+Z(K,I)*AVX(K,L)*Z(L,J)
000740          DO 52 I=1,NX
000747       52 ASGW(I)=SQRT(VW(I,I))
000757          DO 53 I=1,NX
000760          AEW(I)=0.0
000761          DO 53 J=1,NX
000763       53 AEW(I)=AEW(I)+Z(J,I)*AEX(J)
000776          DO 55 I=1,NX
001000       55 U(I)=AEW(I)-A(I,IA)*B(IB)*ASGW(I)
001014          CALL WTOX(NX,Z,U,X)
C************************************************************MODIFY ALPHA BETA
001017          IA=IA+1
001021          IF(IA.EQ.NIA) GO TO 60
```

```
001023          CALL DG(X,DGX)
001025          CALL DIGN(NX,DGX,DGX,Z)
001030          CALL COSINE(NX,IA,DGW,ASGW,A)
001034          BT0=B(IB)
001036          CALL BETA(NX,IA,BT0,BT,W,X,ASGW,ADW,Z,A)
001050          IB=IB+1
001052          B(IB)=BT
001054          DO 56 I=1,NX
001055       56 W(I)=AFN(I)-A(I,IA)*B(IB)*ASGW(I)
001071          CALL WTOX(NX,Z,W,X)
      C[illegible]CONVERGENCE TEST
FOSM      FORTRAN COMPILATION            RUN 2.300-75274        03 DEC 80 10:57:50  PAGE NO.

001074          IB1=IB-1
001076          BX=B(IB1)
001100          IF(ABS((BT-BX)/BT).GE.EPB) GO TO 30
001104          IA1=IA-1
001106          DO 62 I=1,NX
001110       62 IF(ABS(A(I,IA)-A(I,IA1)).GE.EPA) GO TO 30
      C[illegible]OUTPUT
001123          WRITE(6,300)
001126      300 FORMAT(1H1,6HOUTPUT/1H0,31HCOORDINATE AND COSINE DIRECTION/1H0,
              18HVARIABLE,5X,4HX(I),11X,4HW(I),11X,4HA(I))
001126          DO 70 I=1,NX
001130       70 WRITE(6,302) I,X(I),W(I),A(I,IA)
001150      302 FORMAT(1H0,I5,5X,F10.3,5X,F10.3,5X,F10.3)
001150          CALL NORMAL(BT,P)
001152          PF=1.0-P
001154          WRITE(6,301) BT,PF
001164      301 FORMAT(1H0,17HSAFETY INDEX(BT)=,F10.3/1H0,
              127HPROBABILITY OF FAILURE(PF)=,E10.3)
001164          GO TO 90
001165       80 WRITE(6,107)
001171      107 FORMAT(1H0,34HFAILED TO CONVERGE IN THIS PROGRAM)
001171          GO TO 90
001172       81 WRITE(6,108)
001176      108 FORMAT(1H0,38HSTOP CALCULATION(UNSTABLE STRUCTURE))
001176       90 ICASE=ICASE+1
001200          IF(ICASE.LE.NCASE) GO TO 110
001202          STOP
001204          END
```

```
              SUBROUTINE DG[illegible]
[illegible]   DIMENSION [illegible]
[illegible]   [illegible]
[illegible]   DG[illegible]
[illegible]   DO [illegible]
000012        RETURN
[illegible]   END
```

```
              SUBROUTINE DIGW(NX,DGX,DGW,Z)
000007        DIMENSION DGX(1),DGW(1),Z(10,1)
000007        DO 10 J=1,NX
000010        DGW(J)=0.0
000012        DO 10 I=1,NX
000013     10 DGW(J)=DGW(J)+Z(I,J)*DGX(I)
000027        RETURN
000027        END
```

```
              SUBROUTINE COSINE(NX,IA,DGW,SGW,A)
000010        DIMENSION DGW(1),SGW(1),A(10,1)
000010        SG=0.0
000011        DO 10 I=1,NX
000012     10 SG=SG+DGW(I)*DGW(I)*SGW(I)*SGW(I)
000021        SG=SQRT(SG)
000023        DO 20 I=1,NX
000027     20 A(I,IA)=DGW(I)*SGW(I)/SG
000045        RETURN
000046        END
```

```
              SUBROUTINE WTOX(NX,Z,W,X)
000007        DIMENSION W(1),X(1),Z(10,1)
000007        DO 10 I=1,NX
000010        X(I)=0.0
000012        DO 10 J=1,NX
000013     10 X(I)=X(I)+Z(I,J)*W(J)
000027        RETURN
000027        END
```

```
      [illegible]
      [illegible]
     1(10,10),A(10,1)
      IG=0
    8 IG=IG+1
      IF(IG.EQ.30) GO TO 20
      DO 9 I=1,NX
    9 W(I)=HEW(I)+A(I,IA)*BT0+ASGW(I)
      CALL WTOX(NX,Z,W,X)
      CALL DG(X,DGX)
      CALL DIGW(A,DGX,DGW,Z)
      DGST=0.0
      DO 10 I=1,NX
   10 DGST=DGST+DGW(I)*(A(I,IA)+ASGW(I))
      GX=G(X)
   11 IF(DGST.EQ.0.0) GO TO 12
      BT1=BT0-GX/DGST
      GO TO 15
   12 DGST=DGST-0.05*DGW(I)*A(I,IA)*ASGW(I)
      GO TO 11
   13 DO 14 I=1,NX
   14 W(I)=HEW(I)+A(I,IA)*BT1+ASGW(I)
      CALL WTOX(NX,Z,W,X)
      G=G(X)
      IF(GX*G.LT.0.0) GO TO 15
      BT=BT1
      GO TO 16
   15 BT=(BT1+BT0)/2.0
   16 IF(ABS((BT-BT0)/BT).LT.0.01) GO TO 20
      BT0=BT
      GO TO 8
   20 RETURN
      END
```

```
      SUBROUTINE LGN(EXI,SIG,XX,EE,SGM)
      XI=SQRT(ALOG(1.0+(SIG/EXI)**2))
      XL=ALOG(EXI)-0.5*XI*XI
      CX=(ALOG(XX)-XL)/XI
      CALL MDNOR(CX,P)
      CALL MDNRIS(P,Y,IER)
      FAI=F(Y)
      PD=F(CX)/XI/XX
      SGM=FAI/PD
      EE=XX-FAI*Y/PD
      RETURN
      END
```

```
      SUBROUTINE GAM(EXI,SIG,XX,EE,SGM)
      XK=EXI*EXI/SIG/SIG
      XK1=XK-1.0
      XLAM=XK/EXI
      GM=GAMMA(XK)
      X=XLAM*XX
      CALL MDGAM(X,XK,PROB,IER)
      CALL MDNRIS(PROB,Y,IER)
      FAI=F(Y)
      PD=XLAM*(X**XK1)*EXP(-X)/GM
      SGM=FAI/PD
      EE=XX-FAI*Y/PD
      RETURN
      END
```

```
                        SUBROUTINE [illegible]
000010                  [illegible]
000011                  [illegible]
000014                  [illegible]
000032                  CALL MDNRIS(P,Y,IER)
000034                  FAI=F(Y)
000036                  PD=F*(ALP*EXP(-ALP*(XX-U)))
000053                  SGM=FAI/PD
000055                  EE=XX-FAI*Y/PD
000060                  RETURN
000061                  END
```

```
                        SUBROUTINE [illegible](EXX,SIG,XX,EE,SGM)
000010                  COV=SIG/EXX
000011                  XK=2.0
000012               10 X1=1.0+1.0/XK
000014                  X2=1.0+2.0/XK
000017                  GM1=GAMMA(X1)
000021                  GM2=GAMMA(X2)
000023                  VY=SQRT(GM2/GM1/GM1-1.0)
000031                  IF(COV-VY) 20,50,30
000037               20 XK=XK+1.0
000041                  GO TO 10
000042               30 XK=XK-0.1
000044                  X1=1.0+1.0/XK
000046                  X2=1.0+2.0/XK
000051                  GM1=GAMMA(X1)
000053                  GM2=GAMMA(X2)
000055                  VY=SQRT(GM2/GM1/GM1-1.0)
000063                  IF(COV-VY) 40,50,30
000071               40 XK=XK+0.01
000073                  X1=1.0+1.0/XK
000075                  X2=1.0+2.0/XK
000100                  GM1=GAMMA(X1)
000102                  GM2=GAMMA(X2)
000104                  VY=SQRT(GM2/GM1/GM1-1.0)
000112                  IF(COV-VY) 40,50,50
000120               50 U=EXX/GM1
000121                  XK1=XK+1.0
000123                  P=EXP(-(U/XX)**XK)
000136                  CALL MDNRIS(P,Y,IER)
000140                  FAI=F(Y)
000142                  PD=(XK/U)*((U/XX)**XK1)*P
000154                  SGM=FAI/PD
000156                  EE=XX-FAI*Y/PD
000161                  RETURN
000162                  END
```

```
                        SUBROUTINE EXPO(EXX,SIG,XX,EE,SGM)
000010                  XLA=1.0/EXX
000011                  P=1.0-EXP(-XLA*XX)
000021                  CALL MDNRIS(P,Y,IER)
000034                  FAI=F(Y)
000036                  PD=XLA*EXP(-XLA*XX)
000041                  SGM=FAI/PD
000043                  EE=XX-FAI*Y/PD
000046                  RETURN
000047                  END
```

```
      SUBROUTINE WEIB([illegible],SIG,[illegible],IER)
      COV=S[illegible]
      [illegible]=1.0
   10 X1=1.0+1.0/XK
      X2=1.0+2.0/XK
      GM1=GAMMA(X1)
      GM2=GAMMA(X2)
      VY=SQRT(GM2/(GM1*GM1)-1.0)
      IF(COV-VY) 20,50,30
   20 XK=XK+1.0
      GO TO 10
   30 XK=XK-0.1
      X1=1.0+1.0/XK
      X2=1.0+2.0/XK
      GM1=GAMMA(X1)
      GM2=GAMMA(X2)
      VY=SQRT(GM2/(GM1*GM1)-1.0)
      IF(COV-VY) 40,50,30
   40 XK=XK+0.01
      X1=1.0+1.0/XK
      X2=1.0+2.0/XK
      GM1=GAMMA(X1)
      GM2=GAMMA(X2)
      VY=SQRT(GM2/(GM1*GM1)-1.0)
      IF(COV-VY) 40,50,30
   50 U=([illegible]/GM1)
      XKI=XK-1.0
      P=1.0-EXP(-(X0/U)**XK)
      CALL MDNRIS(P,Y,IER)
      FAI=F(Y)
      PD=(XK/U)*((X0/U)**XKI)*(1.0-P)
      SGM=FAI/PD
      XE=X0-FAI*Y/PD
      RETURN
      END

      FUNCTION F(X)
      F=(EXP(-0.5*X*X))/2.5066
      RETURN
      END

      FUNCTION GX(X)
      DIMENSION X(1)
      GX=X(1)*X(2)-X(3)
      RETURN
      END
```

Appendix III Example Run of Computer Program FOSM

As an example, the calculation of the safety index for the following cases was carried out.

(1) Performance function $g_X(X_1, X_2, X_3) = X_1 \cdot X_2 - X_3$

(2) Computation

1) Case-1 :

$$\{\mu_X\} = (40, 50, 1000)$$

$$[V_X] = \begin{bmatrix} 25 & 10 & 0 \\ 10 & 25 & 0 \\ 0 & 0 & 40000 \end{bmatrix}$$

Distributions of variables are not specified.

2) Case-2 :

$$\{\mu_X\} = (40, 50, 1000)$$

$$[V_X] = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 40000 \end{bmatrix}$$

X_1 ---- Log-Normal

X_2 ---- Normal

X_3 ---- Type I

3) Case-3 :

$$\{\mu_X\} = (40, 50, 1000)$$

$$[V_X] = \begin{bmatrix} 25 & 10 & 0 \\ 10 & 25 & 0 \\ 0 & 0 & 40000 \end{bmatrix}$$

X_1 ---- Log-Normal

X_2 ---- Normal

X_3 ---- Type I

The results of this computation are as follows:

Subprograms provided by a user

(1) FUNCTION GX(X)

(2) SUBROUTINE(X,DGX)

Input Format

(3) Master Control Card

(4) Convergence Control Card

Output Format

FORTRAN CODING FORM

Program "FOSM" Input Format (1)
Coded By
Checked By

Identification (73–80)

Date
Page 1 of 2

C FOR COMMENT — STATEMENT NUMBER (1–5), Cont. (6), FORTRAN STATEMENT (7–72)

```
      FUNCTION GX(X)
      DIMENSION X(1)
      GX=X(1)*X(2)-X(3)
      RETURN
      END

      SUBROUTINE DG(X,DGX)
      DIMENSION X(1),DGX(1)
      DGX(1)=X(2)
      DGX(2)=X(1)
      DGX(3)=-1.0
      RETURN
      END


    3     3
      
      0.01      0.01      30      30
```

① Performance function ← $g_x(X) = x_1 x_2 - x_3$

② Derivatives of performance function
← $\partial g_x / \partial x_1 = x_2$
← $\partial g_x / \partial x_2 = x_1$
← $\partial g_x / \partial x_3 = -1$

③ ← Master Control Card (NX, NCASE)

④ ← Convergence Control Card (EPA, EPB, N1A, N1B)

FORTRAN CODING FORM

Program "FOSM" Input Format (2)
Coded By
Checked By
Identification 73–80
Date
Page 2 of 2

C FOR COMMENT ⑤ Variable data cards

Statement Number	Cont.	FORTRAN Statement	Notes
CASE1		(CØRRELATED VARIABLES/DISTRIBUTIONS ARE NØT SPECIFIED)	← Heading (Card 1)
1		40.0	← Distribution and Mean (Card 2)
1		50.0	
1		1,000.0	
		25.0 10.0 0.0 10.0 25.0 0.0 0.0	Covariance (Card 3)
		0.0 40,000.0	
CASE2		(UNCØRRELATED VARIABLES/DISTRIBUTIØNS ARE SPECIFIED)	← Heading (Card 1)
2		40.0	← Distribution and Mean (Card 2)
1		50.0	
4		1,000.0	
		25.0 0.0 0.0 0.0 25.0 0.0 0.0	Covariance (Card 3)
		0.0 40,000.0	
CASE3		(CØRRELATED VARIABLES/DISTRIBUTIØNS ARE SPECIFIED)	← Heading (Card 1)
2		40.0	← Distribution and Mean (Card 2)
1		50.0	
4		1,000.0	
		25.0 10.0 0.0 10.0 25.0 0.0 0.0	Covariance (Card 3)
		0.0 40,000.0	

" FOSM " Output Format

```
DATA INPUT
NX(NUMBER OF VARIABLES)=     3
NCASE(NUMBER OF CASES)=     3
CONVERGENCE CONDITION
EPG=      .00001      EPB=      .01000
NIA=   30       NIB=   [illegible]
```

```
ASE1(CORRELATED VARIABLES /DISTRIBUTIONS ARE NOT SPECIFIED)

   I  NDIS      EX

   1    1     40.000

   2    1     50.000

   3    1   1000.000

NDIS=1 NORMAL OR UNSPECIFIED

NDIS=2 LOGNORMAL

NDIS=3 GAMMA

NDIS=4 TYPE1

NDIS=5 TYPE2

NDIS=6 EXPONENTIAL

NDIS=7 WEIBULL

COVARIANCE MATRIX

VX( 1, 1)=    25.000    VX( 1, 2)=    10.000    VX( 1, 3)=     0.        VX( 2, 1)=    10.000    VX( 2, 2)=    25.
X( 2, 3)=     0.        VX( 3, 1)=     0.        VX( 3, 2)=     0.        VX( 3, 3)= 40000.000    VX(

EIGEN MATRIX

Z( 1, 1)=    -.707      Z( 1, 2)=    -.707      Z( 1, 3)=     0.        Z( 2, 1)=     .707      Z( 2, 2)=    -.707
( 2, 3)=     0.        Z( 3, 1)=     0.        Z( 3, 2)=     0.        Z( 3, 3)=     1.000     Z(

TRANSFORMED COVARIANCE MATRIX

VW( 1, 1)=    15.000    VW( 1, 2)=    -.000     VW( 1, 3)=     0.        VW( 2, 1)=    -.000     VW( 2, 2)=    35.
W( 2, 3)=     0.        VW( 3, 1)=     0.        VW( 3, 2)=     0.        VW( 3, 3)= 40000.000    VW(

TRANSFORMED MEAN VALUE

EW( 1)=     7.071    EW( 2)=   -63.640    EW( 3)=  1000.000    EW(

INITIAL BETA=     2.290
OUTPUT

COORDINATE AND COSINE DIRECTION

VARIABLE      X(I)          W(I)          A(I)

   1         30.592         7.895        -.084

   2         41.757       -51.158        -.832

   3       1278.584      1278.584        -.540

SAFETY INDEX(BT)=     2.537
```

```
ASE3(UNCORRELATED VARIABLES/DISTRIBUTIONS ARE SPECIFIED)

   I NDIS      EX
   1    2    40.000
   2    1    50.000
   3    4  1000.000

DIS=1 NORMAL OR UNSPECIFIED
DIS=2 LOGNORMAL
DIS=3 GAMMA
DIS=4 TYPE1
DIS=5 TYPE2
DIS=6 EXPONENTIAL
DIS=7 WEIBULL

OVARIANCE MATRIX
X( 1, 1)=    25.000    VX( 1, 2)=     0.        VX( 1, 3)=     0.        VX( 2, 1)=     0.        VX( 2, 2)=    25.000
( 2, 3)=     0.        VX( 3, 1)=     0.        VX( 3, 2)=     0.        VX( 3, 3)= 40000.000    VX(

IGEN MATRIX
( 1, 1)=     1.000    Z( 1, 2)=     0.        Z( 1, 3)=     0.        Z( 2, 1)=     0.        Z( 2, 2)=     1.000
 2, 3)=     0.        Z( 3, 1)=     0.        Z( 3, 2)=     0.        Z( 3, 3)=     1.000    Z(

RANSFORMED COVARIANCE MATRIX
W( 1, 1)=    25.000    VW( 1, 2)=     0.        VW( 1, 3)=     0.        VW( 2, 1)=     0.        VW( 2, 2)=    25.000
( 2, 3)=     0.        VW( 3, 1)=     0.        VW( 3, 2)=     0.        VW( 3, 3)= 40000.000    VW(

RANSFORMED MEAN VALUE
W( 1)=    40.000    EW( 2)=    50.000    EW( 3)=  1000.000    EW(

NITIAL BETA=     2.649

OUTPUT

COORDINATE AND COSINE DIRECTION
VARIABLE      X(I)          W(I)          A(I)
    1        34.732        34.732
    2        45.226        45.226         .364
    3      1575.350      1575.350        -.837

SAFETY INDEX(BT)=     2.626

PROBABILITY OF FAILURE(PF)=  .432E-02
```

CASE3:UNRELATED VARIABLES /DISTRIBUTIONS ARE SPECIFIED)

I	NDIS	EX
1	2	40.000
2	1	50.000
3	4	1000.000

NDIS=1 NORMAL OR UNSPECIFIED

NDIS=2 LOGNORMAL

NDIS=3 GAMMA

NDIS=4 TYPE1

NDIS=5 TYPE2

NDIS=6 EXPONENTIAL

NDIS=7 WEIBULL

COVARIANCE MATRIX

```
VX( 1, 1)=   25.000   VX( 1, 2)=   10.000   VX( 1, 3)=   0.        VX( 2, 1)=   10.000   VX( 2, 2)=   25.000
VX( 2, 3)=   0.       VX( 3, 1)=   0.       VX( 3, 2)=   0.        VX( 3, 3)= 40000.000   VX(
```

EIGEN MATRIX

```
Z( 1, 1)=   -.707   Z( 1, 2)=   -.707   Z( 1, 3)=   0.    Z( 2, 1)=   .707   Z( 2, 2)=   -.707
Z( 2, 3)=   0.      Z( 3, 1)=   0.      Z( 3, 2)=   0.    Z( 3, 3)=   1.000   Z(
```

TRANSFORMED COVARIANCE MATRIX

```
VW( 1, 1)=   15.000   VW( 1, 2)=   -.000   VW( 1, 3)=   0.   VW( 2, 1)=   -.000   VW( 2, 2)=   35.000
VW( 2, 3)=   0.       VW( 3, 1)=   0.      VW( 3, 2)=   0.   VW( 3, 3)= 40000.000   VW(
```

TRANSFORMED MEAN VALUE

```
EW( 1)=   7.071   EW( 2)=   -63.640   EW( 3)=   1000.000   EW(
```

INITIAL BETA=   2.290

OUTPUT

COORDINATE AND COSINE DIRECTION

VARIABLE	X(I)	W(I)	A(I)
1	33.673	-4.510	-.136
2	43.627	54.926	.611
3	1472.731	1472.731	-.780

SAFETY INDEX(BT)=   2.477

PROBABILITY OF FAILURE(PF)=   .662E-02