

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

Estudo comparativo de métodos computacionais de avaliação de
confiabilidade estrutural

por

Carla Simone de Albuquerque

Dissertação para obtenção do título de
Mestre em Modelagem Computacional

Rio Grande, agosto, 2021

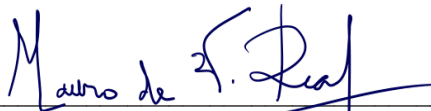
Carla Simone de Albuquerque

“Estudo comparativo de métodos computacionais de avaliação de confiabilidade estrutural”

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal do Rio Grande - FURG, como requisito parcial para obtenção do Grau de Mestre. Área de concentração: Modelagem Computacional.

Aprovado em: 23/08/2021

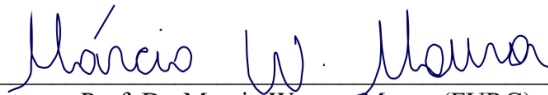
BANCA EXAMINADORA



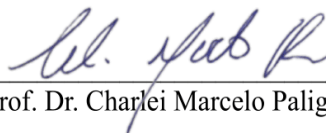
Prof. Dr. Mauro de Vasconcellos Real (Orientador – FURG)



Prof. Dr. Liércio André Isoldi (FURG)



Prof. Dr. Marcio Wrague Moura (FURG)



Prof. Dr. Charlei Marcelo Paliga (UFPel)

Rio Grande - RS
2021

Ficha Catalográfica

A345e Albuquerque, Carla Simone de.

Estudo comparativo de métodos computacionais de avaliação de confiabilidade estrutural / Carla Simone de Albuquerque. – 2021. 132 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Modelagem Computacional, Rio Grande/RS, 2021.

Orientador: Dr. Mauro de Vasconcellos Real.

1. Confiabilidade Estrutural 2. Probabilidade de Falha 3. Índice de Confiabilidade 4. Monte Carlo 5. FOSM 6. FORM 7. SORM I. Real, Mauro de Vasconcellos II. Título.

CDU 004

Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

Dedico este trabalho aos meus filhos (Jardel e Josias Neto), meus pais (Fátima e Josias), meu esposo (Tadeu), minhas irmãs (Telma, Adriana, Ana Cristina e Patrícia) e a minha avó Maria (in memoriam).

AGRADECIMENTOS

A Deus, fonte que me abasteceu de força, fé e foco para nunca desistir.

Aos meus familiares, pelo apoio, incentivo, admiração e amor, que foram as armas dessa etapa de minha vida, em especial os meus pais (Fátima e Josias) e as minhas irmãs (Telma, Adriana, Ana Cristina e Patrícia).

Aos meus filhos (Jardel e Josias Neto), por suportarem minha ausência durante o curso e por serem minha fonte de determinação.

Ao meu esposo (Tadeu), pelo companheirismo incondicional, presente por todo este percurso de minha vida, alimentando-me a alma com muito amor, carinho, força e fé para trilhar esse caminho.

Ao Professor Dr. Mauro de Vasconcellos Real, pela orientação, disponibilidade, colaboração, ensinamentos, paciência e por me fazer descobrir uma área tão fascinante da Estatística/Engenharia.

Aos membros da banca examinadora (Liércio, Márcio e Charlei), pela inestimável colaboração para o aperfeiçoamento do conteúdo desta Dissertação.

Aos meus cunhados (Antonio José e João), por todas as vezes que ajudaram meus filhos na minha ausência.

Aos amigos que conquistei no curso, pelos momentos de alegria, conhecimento compartilhado e descontração dentro e fora da universidade durante o período do Mestrado.

Aos mestres, que das mais diversas maneiras, me motivaram a escolher esse caminho: Ricardo e Nilton (Orientadores da Engenharia Civil e Matemática).

À Universidade Federal do Rio Grande (FURG) e ao PPGMC, por me possibilitar condições necessárias para a conclusão deste curso.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro que permitiu a realização deste trabalho.

“Grandes realizações não são feitas por impulso, mas por uma soma de pequenas realizações.”

(Vicent Van Gogh)

RESUMO

A segurança é um requisito essencial de um sistema estrutural. A confiabilidade é uma ferramenta adicional que está em processo de crescimento e importância na engenharia, por permitir quantificar incertezas nas variáveis de projeto e, assim, auxiliar na tomada de decisões mais adequadas em relação à segurança de uma estrutura. O presente trabalho tem por objetivo comparar e analisar métodos de confiabilidade estrutural, aplicados a vários exemplos de funções de estado limite. Esses métodos são ferramentas importantes para essa análise, pois identificam e quantificam incertezas em variáveis aleatórias, permitindo avaliar a probabilidade de falha da estrutura. Os métodos de confiabilidade estrutural foram programados e simulados na linguagem *Python*. Após a programação foi realizado um estudo comparativo da utilização dos métodos estudados (Simulação de Monte Carlo e métodos de transformação), levando em consideração os resultados do índice de confiabilidade e probabilidade de falha. O desempenho desses métodos foi verificado e analisado através de dezenove exemplos de funções de estado limite lineares, não lineares, implícitas e explícitas. Os resultados indicam que a simulação de Monte Carlo força bruta (MCFB) e com amostragem por importância (MCAI) se mostraram bastante eficientes para os exemplos estudados neste trabalho, com valores iguais ou bem próximos aos valores de referência da literatura. O método de primeira ordem e segundo momento (FOSM) apresentou limitações em alguns exemplos, quando as variáveis aleatórias básicas não são normais e a função estado limite é não linear. O método de confiabilidade de primeira ordem (FORM) trabalha com uma linearização da superfície de falha, dessa forma, para funções com alta não linearidade não funciona bem. O método de confiabilidade de segunda ordem (SORM) melhorou os resultados do FORM, incluindo informações adicionais sobre a curvatura da função estado limite. Por fim, os códigos programados e simulados na linguagem *Python* se mostraram bem eficientes para todos os métodos, pois os resultados foram bastante rápidos e precisos para as análises realizadas.

Palavras-chaves: Confiabilidade estrutural, Probabilidade de falha, Índice de confiabilidade, Monte Carlo, FOSM, FORM, SORM.

ABSTRACT

Safety is an essential requirement of a structural system. Reliability is an additional tool in process of growing and importance in engineering, as it allows us to quantify uncertainties in the design variables and, thus assisting in making more suitable decisions regarding the safety of a structure. The present work aims to compare and analyze structural reliability methods, applied to various examples of limit state functions. These methods are important tools for this analysis, because they identify and quantify uncertainties in random variables, allowing the evaluation of the probability of failure of the structure. Structural reliability methods were programmed and simulated in the Python language. After programming, a comparative study of the use of the studied methods (Monte Carlo simulation and transformation methods) was carried out, taking into account the results of the reliability index and failure probability. The performance of these methods was verified and analyzed through nineteen examples of linear, non-linear, implicit and explicit limit state functions. The results indicate that the simulation of Monte de Carlo brute force (MCFB) and with importance sampling (MCAI) proved to be quite efficient for the examples studied in this work, with values equal to or very close to the reference values of literature. The First Order and Second Moment Method (FOSM) presented limitations in some examples, when the basic random variables are not normal and the limit state function is nonlinear. The first order reliability method (FORM) works with a linearization of the failure surface, so for functions with high non-linearity it does not work well. The second order reliability method (SORM) has improved the FORM results by including additional information about the curvature of the limit state function. Finally, the codes programmed and simulated in Python language proved to be very efficient for all methods, as the results were very fast and accurate for the analysis performed.

Keywords: Structural reliability, Probability of failure, Reliability index, Monte Carlo, FOSM, FORM, SORM.

ÍNDICE

1. INTRODUÇÃO.....	19
1.1. Justificativa.....	20
1.2.1. Objetivo geral.....	20
1.2.2. Objetivos específicos	21
1.3. Metodologia.....	21
1.4. Organização do trabalho.....	22
2. REVISÃO BIBLIOGRÁFICA	23
2.1. Estado da Arte	23
2.1.1. Artigos em periódicos	23
2.1.2. Dissertações e Teses.....	27
2.2. Relevância da pesquisa.....	30
3. CONCEITOS DE PROBABILIDADE E ESTATÍSTICA	31
3.1. Conceitos estatísticos.....	31
3.1.1. Média, variância e desvio padrão.....	31
3.2. Conceitos de probabilidade	32
3.2.1. Variáveis aleatórias	33
3.2.2. Distribuição de probabilidade	34
4. CONFIABILIDADE ESTRUTURAL	39
4.1. Estados limites.....	39
4.2. Função estado limite.....	41
5. MÉTODOS DE CONFIABILIDADE.....	45
5.1. Simulação de Monte Carlo	45
5.1.1. Monte Carlo força bruta.....	47
5.1.2. Monte Carlo com amostragem por importância.....	49
5.2. Métodos de transformação.....	52
5.2.1. FOSM.....	53
5.2.2. FORM	55
FORM HLRF	63
FORM iHLRF	66
5.2.3. SORM	68
6. RESULTADOS	73
6.1. Verificação dos métodos	73

6.1.1. Problema 1: Função estado limite não linear com variáveis normais.....	74
6.1.2. Problema 2: Função estado limite não linear com variáveis mistas.....	75
6.1.3. Problema 3: Função estado limite altamente não linear com variáveis normais	79
6.1.4. Problema 4: Função estado limite altamente não linear com variáveis mistas	81
6.1.5. Problema 5: Função estado limite implícita.....	82
6.2. Estudo comparativo	85
7. CONCLUSÕES	108
8. REFERÊNCIAS BIBLIOGRÁFICAS	110
APÊNDICE 1 – MCFB implementado na linguagem Python	115
APÊNDICE 2 – MCAI implementado na linguagem Python	117
APÊNDICE 3 – FOSM implementado na linguagem Python.....	120
APÊNDICE 4 – FORM HLRF implementado na linguagem Python.....	122
APÊNDICE 5 – FORM iHLRF implementado na linguagem Python.....	124
APÊNDICE 6 – SORM implementado na linguagem Python	128

LISTA DE FIGURAS

Figura 3.1 – Função Densidade de Probabilidade (a) e Função de Distribuição Acumulada (b)	33
Figura 3.2 – Distribuição Normal.....	35
Figura 3.3 – Distribuição Lognormal.....	36
Figura 3.4 – Distribuição Gumbel.....	37
Figura 4.1 – Representação genérica para a função estado limite.....	41
Figura 4.2 – Função estado limite para um sistema com duas variáveis.....	42
Figura 4.3 – Representação do índice de confiabilidade	44
Figura 5.1 – Métodos de confiabilidade estrutural	45
Figura 5.2 – Geração de números aleatórios com distribuição prescrita.....	46
Figura 5.3 – Fluxograma do algoritmo do método Monte Carlo força bruta	49
Figura 5.4 – Método de Monte Carlo com Amostragem por Importância.....	50
Figura 5.5 – Fluxograma do algoritmo do método Monte Carlo com amostragem por importância.....	51
Figura 5.6 – Aproximação de primeira e segunda ordem (FORM e SORM)	52
Figura 5.7 – Transformação do espaço normal para o normal equivalente.....	56
Figura 5.8 – Transformações das variáveis X_1 e X_2 em variáveis normais padrão independentes	61
Figura 5.9 – Representação do método FORM com o algoritmo HLRF	65
Figura 5.10 – Rotação das coordenadas	68
Figura 5.11 – Algoritmo 5.4. Fluxograma do método SORM.....	68
Figura 6.1 – Histograma da função estado limite do problema 1.....	74
Figura 6.2 – Resultados dos índices de confiabilidade da viga de concreto armado	78
Figura 6.3 – Comparativo do índice de confiabilidade para a taxa de armadura do aço ($\rho = 0,15\%$).....	78
Figura 6.4 – Comparativo do índice de confiabilidade para a taxa de armadura do aço ($\rho = 0,95\%$).....	79
Figura 6.5 – Esquema estrutural da treliça plana.....	82
Figura 6.6 – Histograma da função estado limite da treliça plana	83
Figura 6.7 – Histograma comparativo da função estado limite dos exemplos 1 e 2.	87
Figura 6.8 – Limite de falha e os domínios de falha e segurança, para o exemplo 3.....	89
Figura 6.9 – Limite de falha e os domínios de falha e segurança, para o exemplo 4.....	90

Figura 6.10 – Vista esquemática da treliça de telhado	100
Figura 6.11 – Estrutura de treliça de 10 barras.....	102
Figura 6.12 – Viga continua de três vãos	104
Figura 6.13 - Comparativo do índice de confiabilidade dos exemplos analisados.....	106

LISTA DE TABELAS

Tabela 5.1 – Número de simulações para diferentes probabilidade de falha	49
Tabela 6.1 – Médias e desvios padrão das variáveis aleatórias do Problema 1	74
Tabela 6.2 – Comparativo dos resultados para o Problema 1	75
Tabela 6.3 – Viga de concreto armado	77
Tabela 6.4 – Modelo probabilístico das variáveis aleatórias	77
Tabela 6.5 – Médias e desvios padrão das variáveis aleatórias do Problema 3	80
Tabela 6.6 – Comparativo dos resultados para o Problema 3	80
Tabela 6.7 – Médias e desvios padrão das variáveis aleatórias do Problema 4	81
Tabela 6.8 – Comparativo dos resultados para o Problema 4	82
Tabela 6.9 – Médias e desvios padrão das variáveis aleatórias do Problema 5	83
Tabela 6.10 – Comparativo dos resultados para o Problema 5	84
Tabela 6.11 – Médias e desvios padrão das variáveis aleatórias do Exemplo 1	85
Tabela 6.12 – Comparativo dos resultados para o Exemplo 1	86
Tabela 6.13 – Médias e desvios padrão das variáveis aleatórias do Exemplo 2	87
Tabela 6.14 – Comparativo dos resultados para o Exemplo 2	88
Tabela 6.15 – Médias e desvios padrão das variáveis aleatórias do Exemplo 3	88
Tabela 6.16 – Comparativo dos resultados para o Exemplo 3	89
Tabela 6.17 – Médias e desvios padrão das variáveis aleatórias do Exemplo 4	90
Tabela 6.18 – Comparativo dos resultados para o Exemplo 4	91
Tabela 6.19 – Médias e desvios padrão das variáveis aleatórias do Exemplo 5	92
Tabela 6.20 – Comparativo dos resultados para o Exemplo 5	92
Tabela 6.21 – Médias e desvios padrão das variáveis aleatórias do Exemplo 6	93
Tabela 6.22 – Comparativo dos resultados para o Exemplo 6	93
Tabela 6.23 – Médias e desvios padrão das variáveis aleatórias do Exemplo 7	94
Tabela 6.24 – Comparativo dos resultados para o Exemplo 7	95
Tabela 6.25 – Médias e desvios padrão das variáveis aleatórias do Exemplo 8	95
Tabela 6.26 – Comparativo dos resultados para o Exemplo 8	96
Tabela 6.27 – Médias e desvios padrão das variáveis aleatórias do Exemplo 9	96
Tabela 6.28 – Comparativo dos resultados para o Exemplo 9	97
Tabela 6.29 – Médias e desvios padrão das variáveis aleatórias do Exemplo 10	98
Tabela 6.30 – Comparativo dos resultados para o Exemplo 10	98
Tabela 6.31 – Médias e desvios padrão das variáveis aleatórias do Exemplo 11	99

Tabela 6.32 – Comparativo dos resultados para o Exemplo 11	99
Tabela 6.33 – Comparativo dos resultados para o Exemplo 12	101
Tabela 6.34 – Médias e desvios padrão das variáveis aleatórias do Exemplo 12	101
Tabela 6.35 – Comparativo dos resultados para o Exemplo 13	103
Tabela 6.36 – Médias e desvios padrão das variáveis aleatórias do Exemplo 13	103
Tabela 6.37 – Comparativo dos resultados para o Exemplo 14	104
Tabela 6.38 – Médias e desvios padrão das variáveis aleatórias do Exemplo 14	105

LISTA DE SÍMBOLOS

A	matriz ortogonal, área [cm ²]
b	largura [cm]
B	matriz da derivada de segunda ordem
c	parâmetro de penalidade
C	matriz de correlação
d	distância, direção de busca, distância da face inferior do concreto em relação ao eixo das barras de aço [cm]
D	matriz diagonal
exp	exponencial
$E [g()]$	valor médio da função estado limite
f	tensão, resistência. [MPa]
$f()$	função densidade de probabilidade
$F()$	função de distribuição acumulada
$g()$	função estado limite
g	carga permanente [kN.m]
G	variável aleatória carga permanente
$h()$	função de amostragem
h	altura [cm]
H	matriz ortogonal
$I()$	função indicadora
J	matrizes Jacobianas
k	curvatura da superfície, valor característico
M	momento fletor [kN.m]
$p()$	função de mérito
P	probabilidade

q	carga acidental [kN/m]
Q	variável aleatória carga acidental
R	variável aleatória da força resistente, matriz de correlação
S	variável aleatória da força solicitante
u	parâmetro da distribuição Gumbel
V	vetor de médias
$Var [g()]$	variância da função estado limite
W	variável aleatória carga do vento
x^*	ponto de falha
X	variável aleatória, eixo X
y^*	ponto de falha
Y	variável aleatória, variável reduzida, eixo Y
Z	variável aleatória, eixo Z

Símbolos gregos

α	cosseno diretor
β	índice de confiabilidade
γ	Coefficiente de ponderação das ações, constante de Euler
θ	erro de modelo
λ	Média, multiplicador de Lagrange
μ	média
ζ	desvio padrão
Π	produto
ρ	correlação, taxa de armadura longitudinal
σ	desvio padrão
σ^2	variância
ϕ	função densidade de probabilidade

Φ	função de distribuição normal padronizada acumulada
Σ	somatório
χ	razão entre as cargas permanente e variável.
∂	derivada
$\int$	integral
∇	gradiente

Sub índices

c	concreto
f	falha
g	carga permanente
q	carga acidental
s	Segurança, armadura de aço
x	valor assumido pela variável aleatória de X .
y	escoamento do aço, valor assumido pela variável aleatória de Y .
z	valor assumido pela variável aleatória de Z .

LISTA DE ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
ACI	<i>American Concrete Institute</i>
AK-MCS	<i>Active learning Reliability Method combining Kriging and Monte Carlo Simulation</i>
ANSYS	<i>Analysis Systems Incorporated</i>
EGRA	<i>Efficient Global Reliability Analysis</i>
FDA	Função de Distribuição Acumulada
FDP	Função Densidade de Probabilidade
FIB	Fédération Internationale du Béton
FOSM	<i>First Order Second Moment</i>
FORM	<i>First Order Reliability Method</i>
HLRF	<i>Hasofer, Lind, Rackwitz e Fiessler</i>
iHLRF	<i>improved Hasofer, Lind, Rackwitz e Fiessler</i>
JCSS	<i>Joint Committee on Structural Safety</i>
MCAI	Monte Carlo com Amostragem por Importância
MCFB	Monte Carlo Força Bruta
NBR	Norma Brasileira
RSM	Método da Superfície de Resposta
SIS	<i>Sequential Importance Sampling</i>
SORM	<i>Second Order Reliability Method</i>

1. INTRODUÇÃO

Segundo Haldar e Mahadevan (2000), a maioria dos fenômenos observáveis no mundo contém uma certa quantidade de incerteza. Em geral, medições repetidas de fenômenos físicos geram múltiplos resultados. A ocorrência de múltiplos resultados sem nenhum padrão é descrita por termos como incerteza e aleatoriedade. Então, se uma amostra composta por vários elementos idênticos de uma barra de aço, por exemplo, fosse submetida ao carregamento até a falha, cada elemento possivelmente falharia com valores diferentes de carga. A capacidade de carga dessas barras é, portanto, uma quantidade aleatória, formalmente conhecida como variável aleatória.

Conforme Real (2000), as principais grandezas envolvidas em projeto de engenharia são, na realidade, variáveis aleatórias, que possuem uma determinada distribuição de probabilidade. Logo, a própria resposta da estrutura a um carregamento será também uma variável aleatória.

Dessa forma, garantir níveis de segurança adequados nas estruturas, proporcionando um bom desempenho, é um dos principais desafios da engenharia. Nas normas de projeto são utilizados modelos baseados em estados limites, onde é especificada uma série de critérios de segurança no processo de dimensionamento das estruturas. O efeito de incertezas na segurança das estruturas pode ser diminuído, analisando a probabilidade de falha de um estado limite (ALBUQUERQUE e REAL, 2020).

Para Verma *et al.* (2016), a confiabilidade é definida como probabilidade do desempenho sem problemas nas condições estabelecidas, ou ainda, pode ser considerada como a probabilidade de um elemento poder executar sua função pretendida por um tempo especificado, sob condições de projeto estabelecidas. Isso equivale à confiabilidade da missão, que é a probabilidade de o produto não deixar de concluir a missão, considerando todos os modos de operação possíveis.

Segundo Beck (2019), uma ferramenta que proporciona a determinação da probabilidade de uma estrutura falhar durante sua vida útil projetada é a análise de confiabilidade estrutural. Devido às incertezas dos parâmetros de projetos na engenharia é necessário que sejam realizadas análises de confiabilidade em função de possíveis falhas. A resposta nesse caso, é a determinação da probabilidade de falha ou índice de confiabilidade associado a uma função de estado limite que depende das variáveis aleatórias do problema. Dessa forma, os métodos de confiabilidade estrutural são

ferramentas importantes para essa análise, pois identificam e quantificam incertezas em variáveis aleatórias.

De acordo com Santos (2014), diversos métodos têm sido desenvolvidos ao longo de vários anos com o propósito de estimar valores para a probabilidade de falha e índice de confiabilidade. Dentre estes métodos destacam-se o FOSM (*First Order Second Moment*), FORM (*First Order Reliability Method*) e SORM (*Second Order Reliability Method*). Estes métodos de confiabilidade estrutural, também conhecidos como métodos de transformação, aproximam a função de estado limite no ponto de projeto por um hiperplano, no caso do FOSM e do FORM, ou por uma hipersuperfície quadrática, no caso do SORM. A aproximação desses métodos pode conduzir a erros, principalmente quando a função de estado limite é fortemente não linear. Dessa forma, pode-se fazer uso da simulação de Monte Carlo para se ter uma ideia da probabilidade de falha e do índice de confiabilidade corretos, por ser esse método considerado o mais exato.

1.1. Justificativa

A Associação Brasileira de Normas Técnicas (ABNT) é o principal órgão de normatização do país. As Normas Brasileiras (NBR) são documentos técnicos utilizadas para padronizar e garantir um padrão de qualidade. No Brasil ainda não existem especificações técnicas baseando-se nos princípios da confiabilidade estrutural, embora essa seja uma realidade em vários outros países. Logo, esse trabalho é justificado pela limitação do número de pesquisas sobre o tema no país e a omissão das NBR em relação à confiabilidade estrutural, que tornam essa investigação relevante para a engenharia estrutural no Brasil e ainda pode contribuir para o processo de desenvolvimento de uma norma nacional que inclua a confiabilidade como medida de segurança.

1.2. Objetivos

1.2.1. Objetivo geral

Este trabalho tem como objetivo comparar métodos de confiabilidade estrutural, aplicados a exemplos de funções de estado limite e, assim, analisar os resultados quanto à eficiência da aplicação do método, levando em consideração a segurança.

1.2.2. Objetivos específicos

Como objetivos específicos, busca-se:

- Apresentar um estudo comparativo dos métodos abordados em exemplos de funções de estado limite linear, não linear e implícita;
- Analisar a influência dos métodos nas soluções dos problemas de confiabilidade;
- Analisar como as não linearidades das funções de estado limite interferem na precisão dos resultados gerados pelos diferentes métodos de cálculo;
- Analisar a influência das diferentes distribuições de probabilidade das variáveis nos resultados dos diferentes métodos de confiabilidade.

1.3. Metodologia

A metodologia empregada na elaboração deste trabalho consiste no desenvolvimento de códigos para análise de confiabilidade estrutural, através dos métodos Monte Carlo Força Bruta (MCFB), Monte Carlo com Amostragem por Importância (MCAI), FOSM, FORM com os algoritmos HLRF (*Hasofer, Lind, Rackwitz e Fiessler*) e iHLRF (*improved Hasofer, Lind, Rackwitz e Fiessler*) e SORM.

A aplicação dos códigos programados foi inicialmente realizada em uma verificação dos métodos através de cinco problemas de funções estado limite linear, não linear e implícita, encontrados na literatura.

Após a verificação da programação foi apresentado um estudo comparativo entre os métodos, comparando os resultados do índice de confiabilidade e probabilidade de falha. Para a comparação entre os métodos são considerados catorze exemplos de funções estado limite linear, não linear e implícita, obtidas por alguns pesquisadores. Em seguida é analisado o desempenho e as deficiências de cada método.

Os métodos de confiabilidade estrutural foram programados na linguagem *Python*. *Python* é uma linguagem de programação interpretada, orientada a objetos, de alto nível e com semântica dinâmica. É uma das linguagens que mais tem crescido devido à sua compatibilidade e capacidade de auxiliar outras linguagens (PYTHON SOFTWARE FOUNDATION, 2019).

A linguagem de programação *Python* foi escolhida devido à simplicidade de seu uso e a sua rapidez e eficiência no processamento de operações matriciais. Além disso, a grande quantidade de pacotes disponíveis e o fato desses pacotes serem todos de código aberto (*Open Source*), o que permite que eles sejam estudados, modificados e adaptados

para que melhor se adéquem ao seu propósito final, foram determinantes na escolha dessa linguagem para elaboração deste trabalho.

Conforme Hynninen *et al.* (2016), códigos com mais ênfase na acessibilidade surgiram durante a última década. Uma estratégia para alcançar essa acessibilidade é construir uma interface de programação para o código, e *Python* se tornou uma linguagem de interface popular, pois tem capacidade de *script* poderosa.

Os métodos de confiabilidade estrutural implementados na linguagem *Python* foram verificados através de problemas de formulação explícita e implícita encontrados na literatura. No caso dos problemas com função estado limite implícita foi utilizado o *TrussPy* (DUTZLER, 2018) que é um programa escrito em *Python* capaz de considerar não linearidades físicas ou geométricas. Ele usa uma abordagem orientada a objetos para estruturação do código em classes. O programa *TrussPy* é utilizado como uma função dentro dos códigos dos métodos implementados no *Python*.

1.4. Organização do trabalho

Esta dissertação está estruturada em sete capítulos. Neste primeiro capítulo apresentam-se as considerações iniciais, justificativa, objetivos e metodologia.

O segundo capítulo contempla uma revisão bibliográfica sobre a utilização de métodos de confiabilidade estrutural, onde são citados alguns trabalhos fundamentais que inserem o leitor no universo dos métodos de transformação e simulação de Monte Carlo.

O terceiro capítulo aborda os conceitos referentes à probabilidade e estatística.

O quarto capítulo apresenta os conceitos relacionados à confiabilidade estrutural.

O quinto capítulo realiza uma descrição dos métodos de confiabilidade, dividindo-os em Simulação de Monte Carlo e métodos de transformação.

O sexto capítulo destina-se a apresentar os resultados obtidos para dezenove problemas de função estado limite linear, não linear e implícita, sendo cinco utilizados para verificação dos métodos e catorze para uma análise comparativa do índice de confiabilidade e probabilidade de falha.

O sétimo capítulo apresenta uma discussão dos aspectos considerados relevantes, sintetizando os resultados, tecendo as conclusões obtidas durante a realização deste trabalho e sugestões para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

Esse capítulo é destinado a sintetizar alguns trabalhos recentes e importantes que apresentam a utilização de métodos de confiabilidade estrutural, fazendo uma análise de confiabilidade nos exemplos apresentados nos seus resultados e, também, mostrando a importância do tema para a segurança estrutural.

2.1. Estado da arte

2.1.1. Artigos em periódicos

Santos *et al.* (2012) apresentaram três novos algoritmos de otimização. O primeiro, denominado nHLRF, foi desenvolvido especificamente para aplicação ao problema de confiabilidade estrutural. Esse algoritmo baseia-se no algoritmo HLRF, entretanto, utiliza uma nova função de mérito diferenciável para a seleção do comprimento do passo na busca linear. Os outros dois algoritmos propostos utilizam penalidades quadráticas e baseiam-se na estrutura dos métodos Lagrangeano aumentado empregados na resolução de problemas com restrições de desigualdade, implementados no Matlab. O desempenho e a robustez dos novos algoritmos também foram comparados na resolução de problemas de confiabilidade estrutural. O novo algoritmo nHLRF mostrou-se mais robusto que HLRF e tão eficiente quanto o algoritmo iHLRF (versão melhorada do HLRF). Os dois métodos Lagrangeano aumentado foram mais eficientes em comparação com nHLRF principalmente na resolução de problemas com um número maior de variáveis aleatórias.

Santos *et al.* (2015) apresentaram um estudo sobre técnicas de amostragem inteligentes e eficientes na simulação de Monte Carlo. Os autores compararam a simulação bruta de Monte Carlo com Amostragem por Importância, Amostragem Assintótica, Amostragem Aprimorada e Simulação por Subconjunto. Esses cinco esquemas de amostragem foram combinados com Amostragem Simples, Amostragem de Hipercubo Latino e Amostragem de Variáveis Antitéticas, resultando em quinze combinações de estratégias de amostragem. O desempenho dessas estratégias foi investigado por seis funções de estado limite linear e não linear. Os resultados mostram que a amostragem por importância usando pontos de projeto é extremamente eficiente para avaliar pequenas probabilidades de falha. No entanto, encontrar o ponto de projeto pode ser um problema. A simulação de subconjuntos apresentou desempenho muito bom para todas as funções estudadas nesse artigo. A Amostragem Aprimorada teve um desempenho melhor que a Amostragem Assintótica nas funções consideradas. Os autores

concluíram com esse estudo que a Simulação por Subconjunto, Amostragem Aprimorada e Amostragem por Importância, auxiliada pela amostragem Hipercubo Latino, são formas eficientes de resolver problemas de confiabilidade, com um número muito menor de elementos da amostra do que o necessário na Simulação de Monte Carlo força bruta.

Papaioannou *et al.* (2016) propuseram a aplicação de um método de amostragem por importância sequencial (SIS) para a estimativa da probabilidade de falha na confiabilidade estrutural. A ideia básica do SIS é converter gradualmente amostras da distribuição anterior em amostras da distribuição posterior através de uma operação corrigida sequencialmente. O SIS pode ser aplicado para produzir elementos de uma amostra com densidade de amostragem de importância aproximadamente ótima. Apresentaram e discutiram também dois algoritmos de Monte Carlo (Amostragem por Importância e Simulação de Subconjunto), sendo que esses foram utilizados como base dentro do procedimento do método SIS. Os autores ainda demonstraram, através de exemplos numéricos, a aplicação do SIS a problemas gerais com número pequeno a moderado de variáveis aleatórias e mostraram que o método é especialmente eficiente para resolver problemas de baixa e alta dimensão.

Huang *et al.* (2017a) apresentam uma visão geral da análise de confiabilidade estrutural, mostrando que o termo "métodos de confiabilidade local" refere-se aos métodos de análise de confiabilidade que usam a função aproximada local do estado limite no cálculo da probabilidade de falha. Os métodos baseados em pontos de projeto (FOSM, FORM, SORM e RSM) estão incluídos nos métodos locais de confiabilidade. Concluíram, através de exemplos, que o método FOSM só fornece resultados razoavelmente precisos quando a função estado limite é linear e as variáveis possuem distribuições normal. Essa limitação foi superada pelos métodos FORM, SORM e RSM, porém para funções altamente não lineares o FORM não apresenta resultado preciso. O SORM e RSM melhoram a estimativa de confiabilidade para as funções não lineares.

Huang *et al.* (2017b) apresentam uma visão geral da análise de confiabilidade estrutural, onde é apresentada a simulação de Monte Carlo que é a base para os métodos de amostragem da análise de confiabilidade. Os métodos de amostragem podem avaliar a probabilidade de falha definida pela função de desempenho explícita e implícita. Com número suficiente de amostras, os métodos de simulação podem fornecer resultados precisos. No entanto, para problemas complexos, há um elevado tempo de processamento. Para reduzir esse tempo com base em técnicas de redução de variância,

as técnicas de amostragem inteligentes são apresentadas como propostas para melhorar o custo computacional.

Huang *et al.* (2017c) apresentam uma visão geral da análise de confiabilidade estrutural, onde mostram que os métodos de confiabilidade global são baseados no modelo de aproximação global da função de desempenho usando o modelo de processo gaussiano. O modelo de processo gaussiano é a base para esses métodos de confiabilidade global. Essa categoria de métodos, em primeiro lugar, aproxima a função de desempenho pelo modelo de processo gaussiano e, em seguida, executa métodos de amostragem com base no modelo substituto construído para calcular a probabilidade de falha. O custo computacional para esses métodos é reduzido significativamente. Além disso, os métodos de confiabilidade global podem fornecer resultados precisos porque o modelo de processo gaussiano pode modelar adequadamente a função de estado limite não linear. Dois métodos globais de confiabilidade, EGRA e AK-MCS, são descritos e ilustrados por um exemplo.

Pereira *et al.* (2017) apresentam um estudo da confiabilidade estrutural de pilares mistos de aço-concreto, parcialmente revestidos e submetidos à flexo-compressão. Os pilares foram dimensionados de acordo com a norma brasileira que embasa o projeto estrutural de edifícios contendo elementos mistos (ABNT NBR 8800, 2008). Estatísticas são utilizadas para caracterizar as incertezas nas propriedades dos materiais constituintes e nas ações. Os índices de confiabilidade foram avaliados, via método FORM, para duas razões entre ação permanente e ação variável, e para alguns valores de excentricidade da ação vertical. Verificaram que o índice de confiabilidade dos pilares mistos diminui com o aumento da excentricidade de projeto. Estes valores ficaram acima de índices de confiabilidade mínimos sugeridos em normas internacionais, o que sugere que o projeto utilizando a norma NBR 8800 (ABNT, 2008) resulta em pilares mistos seguros.

Mapa *et al.* (2019) apresentaram procedimentos numéricos para avaliação da confiabilidade de pórticos. Considerando os efeitos da não linearidade geométrica e da flexibilidade das ligações, utilizaram o método FORM na avaliação da probabilidade de falha de funções de desempenho formuladas para estados limites últimos de resistência e estados limites de serviço, permitindo assim, avaliar os diferentes níveis de segurança de cada estado limite e também qual era o estado limite governante. Os resultados obtidos indicaram que os índices de confiabilidade dos exemplos de pórticos planos de aço analisados são significativamente afetados pela presença das ligações semirrígidas e pelos efeitos da não linearidade geométrica.

Scherer *et al.* (2019) avaliaram o índice de confiabilidade atingido por um conjunto de 81 vigas de concreto armado dimensionadas de acordo com a norma brasileira NBR 6118 (ABNT, 2014) em relação ao estado limite último de flexão. As principais estatísticas das resistências foram obtidas por simulações de Monte Carlo utilizando o software ANSYS e a ferramenta computacional *probabilistic design*. Os índices de confiabilidade foram calculados pelo método FORM, sendo realizado um estudo paramétrico para avaliar a influência das principais variáveis de projeto nos níveis de segurança atingidos. Das vigas avaliadas, um total de 54 apresentaram índices de confiabilidade inferiores ao índice alvo estipulado, sendo que esses resultados correspondem às combinações de ações nas quais a carga acidental é proporcionalmente igual ou maior que a carga permanente. Observaram ainda, que os índices de confiabilidade obtidos nesta pesquisa são resultados de aproximações, desde aquelas associadas ao comportamento mecânico dos materiais, até aquelas relacionadas às atribuições de distribuições teóricas para dada variável aleatória.

Wang *et al.* (2019) mostraram uma comparação entre o cálculo de fatores de segurança tradicional e o processo de otimização baseado em confiabilidade. Demonstraram que o peso da estrutura projetada pela otimização baseada na confiabilidade não é mais pesado que ao calculado pelo fator de segurança sob os mesmos requisitos de confiabilidade. A eficiência do tempo, estimativas de erro e análise de sensibilidade também foram comparadas para discutir as vantagens e desvantagens das duas abordagens. Através de exemplos observaram que o método de otimização pode obter uma solução mais eficiente no projeto estrutural devido à consideração abrangente do valor médio e fator de variação das variáveis de projeto.

Ghalenohvi *et al.* (2020) apresentam um método eficiente para avaliação da segurança de estruturas com pequena probabilidade de falha. O resultado é uma nova formulação de confiabilidade adequada para resolver problemas não lineares. As capacidades do método analisado resolvem vários problemas numéricos de engenharia de *benchmarks* envolvendo variáveis aleatórias não normais, funções de estado limite complexas e problemas não lineares. Em comparação com os métodos de confiabilidade convencionais, para todos os problemas resolvidos, é demonstrado que a abordagem proposta apresenta precisão próxima à simulação de Monte Carlo, enquanto o número necessário de avaliação da função de desempenho é próximo ao método de confiabilidade de primeira ordem (FORM).

Lu *et al.* (2020) propuseram uma técnica de transformação no quarto momento para transformar variáveis aleatórias não-normais correlacionadas em normais independentes. A técnica inclui duas etapas: primeiro, as variáveis aleatórias não-normais correlacionadas são transformadas em normais correlacionadas usando a transformação do quarto momento; segundo as variáveis aleatórias normais padrão correlacionadas são transformadas em variáveis normais padrão independentes usando a decomposição de *Cholesky*. Quatro exemplos numéricos incluindo funções de desempenho explícitas e implícitas são apresentados para demonstrar a eficiência e precisão do método proposto para avaliação da confiabilidade estrutural. Os autores concluíram que o método proposto fornece quase os mesmos resultados em comparação com o método FORM baseado na transformação de Nataf.

Straub *et al.* (2020) apresentaram quatro exemplos numéricos, para investigar e demonstrar o desempenho da confiabilidade de uma estrutura em deterioração sem ações de manutenção e reparo, através de métodos para avaliar com eficiência a confiabilidade ao longo do tempo. Para isso, utilizaram a combinação da Simulação Monte Carlo usando técnicas de amostragem e o método FORM do problema de confiabilidade do sistema em série que surge no cálculo de confiabilidade da vida útil. Em seguida, propuseram estratégias eficientes de cálculo através de aproximações analíticas, abordagens baseadas em amostragem e combinações das mesmas, apresentando estratégias computacionais para melhorar a eficiência das avaliações de confiabilidade invariantes no tempo e propondo uma abordagem para estimar a confiabilidade variante no tempo sem chamadas adicionais de função de estado limite.

2.1.2. Dissertações e teses

Magalhães (2014) analisou a influência dos casos de não conformidade no atendimento da resistência à compressão do concreto especificada em projeto no nível de segurança estrutural através do valor do índice de confiabilidade. A confiabilidade foi avaliada para o caso de pilares dimensionados de acordo com a NBR 6118 (ABNT, 2014), e a simulação do comportamento dos mesmos, e consequente determinação da carga de ruptura, foram realizadas através de um modelo numérico baseado no método dos elementos finitos. A simulação de Monte Carlo e o método FORM foram adotados para a determinação do índice de confiabilidade. Os resultados mostraram que, nos casos de verificação de não conformidade do concreto, vários fatores devem ser observados na busca de uma avaliação segura do real nível de comprometimento da segurança estrutural.

Através da abordagem via confiabilidade, que se demonstrou de grande utilidade, o processo permitiu uma visão mais sistêmica do elemento estrutural e suas características, levando a uma análise mais consistente.

Ferreira (2015) em sua tese apresentou o desenvolvimento de dois procedimentos matemáticos, via geometria diferencial, para o cálculo das curvaturas principais de uma superfície genérica, em um ponto qualquer, fazendo uma análise de confiabilidade feita através dos métodos FORM, SORM e SORM DG, sendo esse último desenvolvido pelo autor. Além desses métodos, a simulação Monte Carlo, também efetua essa análise, sendo utilizado apenas para aferição dos resultados, tendo em vista que conduz a valores confiáveis para a probabilidade de falha. Os resultados mostraram que o método SORM DG obteve, na maioria dos casos, maior acurácia do que o método SORM, em relação ao valor da probabilidade de falha. Comprovou também, que o método SORM DG efetuou a análise de confiabilidade estrutural de segunda ordem com eficácia, otimizando de maneira relevante os resultados fornecidos pelo método FORM.

Liberati (2019) em sua dissertação aplicou os conceitos das teorias de probabilidade e de confiabilidade na avaliação dos modelos de dimensionamento de ligação laje-pilar submetidas à punção, verificando as probabilidades de falha e índices de confiabilidade para cada estado limite de punção. Foram ensaiadas nove lajes lisas quadradas até a ruptura, para assim, verificar o comportamento desse sistema estrutural com ou sem a presença de aberturas. Além dos resultados experimentais obtidos nessa pesquisa, o estudo envolveu a análise de 240 resultados experimentais de carga de ruptura presentes na literatura de lajes lisas com diferentes características físicas e geométricas para que estatísticas da variável erro de modelo pudessem ser construídas. Os resultados obtidos foram comparados através de modelagem teórica e implementados computacionalmente, e utilizando os métodos de confiabilidade simulação de Monte Carlo e o FORM foram determinados as probabilidades de falha e índices de confiabilidade aos modelos mecânicos de previsão da carga de ruptura de lajes lisas. O autor concluiu que a teoria da confiabilidade pode ser aplicada com sucesso na verificação da segurança à punção das ligações laje-pilar em lajes lisas, permitindo um melhor conhecimento desse sistema estrutural.

Silva Junior (2019) em sua dissertação realizou um estudo de confiabilidade com base na norma americana ACI-318 para projeto de reforço à flexão de vigas de concreto armado. Utilizou nessa pesquisa modelagem computacional em elementos finitos no software ANSYS de uma viga de concreto armado reforçada com uma camada de fibra

de carbono, com destaque para a modelagem da região da interface concreto-reforço, que é a região onde geralmente ocorre a falha neste tipo de viga. O modelo adotado foi validado através de resultados experimentais disponíveis na literatura, mostrando bom funcionamento. Em seguida, utilizando o Método da Superfície de Resposta, foi obtida a função polinomial de resposta do sistema. Com esta função e considerando o erro de modelo, foram utilizados os métodos de confiabilidade FORM e simulação de Monte Carlo, onde por meio destes foram obtidos os índices de confiabilidade que foram comparados a um índice alvo. Os resultados mostraram que a confiabilidade da norma ACI-318 se mostrou adequada para a viga em estudo, o índice de confiabilidade alvo foi atingido em todas as situações quando utilizado o carregamento de projeto.

Ribeiro (2019) em sua dissertação propõe avaliação das respostas de esforços e deslocamentos de edifícios considerando a interação solo-fundação-estrutura usando o conceito de confiabilidade estrutural. Utilizou o método dos elementos finitos para a análise de estruturas e os métodos de simulação de Monte Carlo simples e FOSM para calcular sua segurança perante a inserção de variabilidade em certos parâmetros significativos no projeto, como as incertezas dos dados de rigidez do solo e a velocidade do vento. Realizou também a análise de estado limite último de pilares submetidos à flexo-compressão oblíqua com não linearidade física, de modo a avaliar o dimensionamento com fatores de segurança parciais e métodos de confiabilidade estrutural. Destacou a importância de um profissional capacitado para o controle das variáveis estatísticas do projeto, tendo em vista a grande importância do controle dos parâmetros estatísticos das variáveis aleatórias do problema. O autor comprovou, também, a eficiência dos resultados das distribuições estatísticas estabelecidas.

Kroetz (2019) em sua tese apresenta diferentes abordagens de otimização considerando incertezas e propõe os métodos Simulação de Monte Carlo e FORM para solução de problemas desse tipo. É apresentado também uma aplicação de otimização baseada em confiabilidade na calibração de coeficientes parciais de segurança. Inclui ainda um estudo sobre técnicas de metamodelagem, que são aplicados nos métodos propostos para a redução dos custos computacionais envolvidos na solução dos problemas de otimização. O autor utilizou o programa computacional StRAnD para determinar os resultados através dos métodos propostos. Os resultados foram apresentados em vários exemplos, demonstrando-se assim a eficiência das técnicas de metamodelagem nos métodos propostos. Concluiu que apesar do esforço empregado na obtenção de resultados de excelência, a revisão da literatura realizada, bem como uma avaliação das limitações

observadas nas metodologias propostas, torna claro que ainda há muito em que se avançar.

2.2. Relevância da pesquisa

O presente trabalho visa contribuir para o estado da arte, no sentido de analisar comparativamente os diversos métodos de avaliação de confiabilidade estrutural quanto à sua eficiência, e aplicados a exemplos de funções estado limite linear, não linear e implícita programados e simulados na linguagem *Python*. Visa, também, atender/contribuir para o processo de desenvolvimento de uma norma nacional brasileira que inclua a confiabilidade como fator de segurança e para trabalhos futuros que utilizem os métodos empregados.

3. CONCEITOS DE PROBABILIDADE E ESTATÍSTICA

Para Santos (2014), a Engenharia moderna considera a incerteza como inerente à natureza do problema que se está analisando. Dessa forma, o estudo de conceitos probabilísticos é parte integrante e fundamental da teoria da confiabilidade e na aplicação de seus métodos na análise de confiabilidade de estruturas.

Este capítulo pretende abordar os principais tópicos relacionados aos conceitos de probabilidade e estatística, que são de grande importância no desenvolvimento dos métodos a serem estudados.

3.1. Conceitos estatísticos

De acordo com Haldar e Mahadevan (2000), a Estatística é a quantificação Matemática de incertezas de uma variável.

Conforme Ferreira (2015), em um estudo estatístico, tem-se como maior interesse conhecer o comportamento das variáveis presentes no estudo. E os métodos estatísticos são essenciais no estudo de situações em que as variáveis de interesse estão sujeitas, inerentemente, a variações aleatórias.

Para Verma *et al.* (2016), a avaliação de várias propriedades estatísticas, como funções de distribuição de probabilidade e intervalos de confiança de respostas estruturais, desempenha um papel importante na análise de confiabilidade.

3.1.1. Média, variância e desvio padrão

Segundo Bussab e Morettin (2010), a média (μ) pode ser determinada como a soma de valores (x) divididos pela quantidade (n) em que ocorre em um determinado conjunto de dados, e é representada através de:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.1)$$

a variância (σ^2) é uma medida de dispersão que mostra o quão distantes (dispersos) os valores estão da média, como:

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2 \quad (3.2)$$

e o desvio padrão (σ) é a raiz quadrada positiva da variância, e é dado por:

$$\sigma = \sqrt{\sigma^2}. \quad (3.3)$$

3.2. Conceitos de Probabilidade

Segundo Haldar e Mahadevan (2000), a teoria da probabilidade usa informações estatísticas para calcular a probabilidade de eventos específicos. Um evento é um subconjunto contido no espaço amostral. Um espaço amostral equivale a pontos amostrais que são mutuamente exclusivos. Um espaço amostral pode ter valores discretos, contínuos, finitos e infinitos.

Segundo Beck (2019), a Função de Distribuição Acumulada (FDA) de uma variável aleatória X , corresponde à probabilidade de que essa variável assumira qualquer valor menor que x , e é representada através de:

$$F_X(x) = P[X \leq x], \text{ para todo } x. \quad (3.4)$$

Beck (2019) explica que a derivada em relação a x da FDA é chamada de Função de Densidade de Probabilidade (FDP), ou $f_X(x)$, tal que:

$$f_X(x) = \frac{d F_X(x)}{dx} \quad (3.5)$$

$$P[x < X \leq x + dx] = f_X(x)dx. \quad (3.6)$$

A relação conceitual entre FDA e FDP é mostrada na Fig. 3.1. Conforme Haldar e Mahadevan (2000), FDA deve ser 0 em $-\infty$ e 1 em $+\infty$, e é uma função crescente de uma variável aleatória. Já FDP deve ser uma função positiva, pode ser zero e, teoricamente seu intervalo varia de $-\infty$ a $+\infty$.

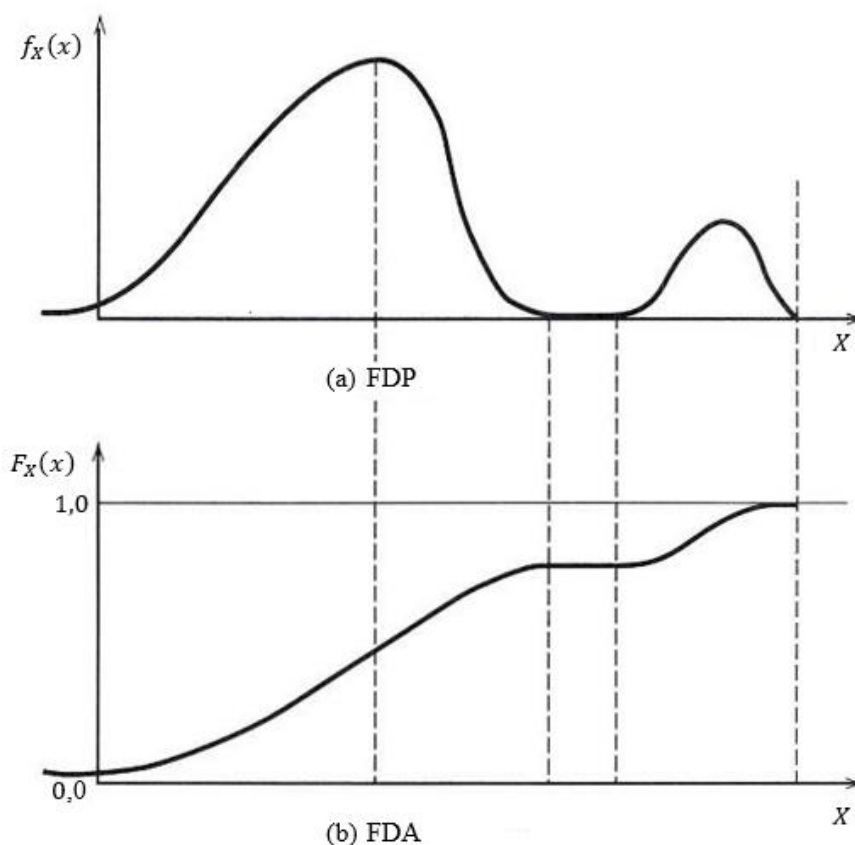


Figura 3.1 – Função Densidade de Probabilidade (a) e Função de Distribuição Acumulada (b)

Fonte: Haldar e Mahadevan (2000)

Segundo Haldar (2006), o conceito básico de confiabilidade estrutural em termos probabilísticos é determinado como a probabilidade de uma determinada estrutura ter seu desempenho de acordo com o que foi projetado. Essa determinação pode ser vista como um operador linear da função densidade de probabilidade da demanda aplicada à estrutura (cargas) com a função de distribuição cumulativa da capacidade ou fragilidade da estrutura (resistências).

3.2.1. Variáveis aleatórias

Para Haldar (2006), as incertezas associadas à previsão da capacidade de carga de uma estrutura, as intensidades das cargas que se espera serem aplicadas e os efeitos dessas cargas podem ser representados por variáveis aleatórias. O valor que uma variável aleatória pode assumir é descrito por uma função de densidade de probabilidade. Para algumas distribuições, pode ser definida apenas em termos da média e desvio padrão ou o coeficiente de variação dos dados do problema analisado.

Segundo Verma *et al.* (2016), variáveis aleatórias são usadas para associar um número ao resultado de um experimento aleatório. Uma variável aleatória é definida como uma função que associa um número real a cada resultado no espaço amostral de um experimento aleatório. Uma variável aleatória é indicada por uma letra maiúscula e o valor numérico que ela pode obter é representado por uma letra minúscula.

Conforme Ang e Tang (2007), os valores ou intervalos de valores de uma variável aleatória representam eventos, e dessa forma, são associados à probabilidade ou medidas de probabilidade específicas. Essas medidas podem ser atribuídas de acordo com as regras estabelecidas que são chamadas de distribuição de probabilidade.

3.2.2. Distribuição de probabilidade

De acordo com Verma *et al.* (2016), a distribuição de probabilidade de uma variável aleatória X é uma representação das probabilidades associadas aos possíveis valores de X . Para uma variável aleatória discreta, a distribuição normalmente é estabelecida por apenas uma lista dos valores possíveis, juntamente com a probabilidade de cada um. Existem diversas distribuições de probabilidade, sendo que neste trabalho serão abordadas três, que são as distribuições do tipo Normal, Lognormal e Gumbel.

A distribuição normal é a distribuição de probabilidade mais utilizada. Verma *et al.* (2016) afirmam que a distribuição Normal geralmente ocorre em aplicações práticas porque a soma de um grande número de variáveis aleatórias estatisticamente independentes converge para uma distribuição Normal (conhecida como Teorema do limite central).

Para Melchers e Beck (2018), a distribuição Normal é um modelo para observações de vários processos ou propriedades físicas. Também é aquela que surge frequentemente na prática como um caso de outras distribuições de probabilidade. A FDP da distribuição Normal é expressa pela Eq. (3.7) e sua FDA através da Eq. (3.8), e graficamente pela Fig. 3.2.

$$f_X(x) = \frac{1}{(2\pi)^{1/2} \sigma_x} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x} \right)^2 \right] \quad -\infty \leq x \leq \infty \quad (3.7)$$

$$F_X(x) = \int_{-\infty}^x \frac{1}{(2\pi)^{1/2} \sigma_x} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x} \right)^2 \right] dx \quad (3.8)$$

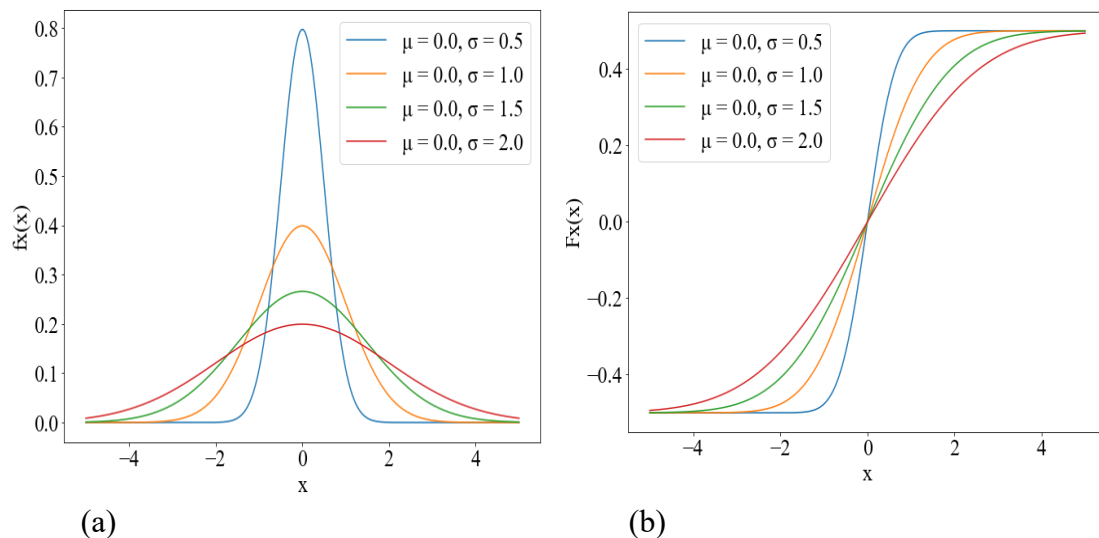


Figura 3.2 – Distribuição Normal: (a) função de densidade de probabilidade e (b) função de distribuição acumulada de probabilidade

Fonte: Autora

Haldar e Mahadevan (2000) afirmam que não é simples determinar probabilidades pela integração da Eq. (3.8). Este problema pode ser solucionado transformando a variável aleatória normal original em uma variável normal padrão, da seguinte forma:

$$Y = \frac{x - \mu_x}{\sigma_x} . \quad (3.9)$$

De acordo com El-Reedy (2013), pelo teorema do Limite Central a soma de um grande número de variáveis aleatórias, independente da sua distribuição tende a uma distribuição normal. Esse é um dos motivos porque a distribuição normal é utilizada em tantos testes estatísticos, podendo representar vários fenômenos naturais.

A distribuição Lognormal também é bastante utilizada. Para Verma *et al.* (2016), uma variável aleatória positiva contínua X é uma distribuição Lognormal se o seu logaritmo natural é normalmente distribuído. Da mesma forma, Haldar e Mahadevan (2000) explicam que em vários problemas de engenharia, uma variável não pode ter valores negativos devido às particularidades físicas do problema. Observando essa condição, modelar a variável como lognormal, é mais adequado para eliminar os valores negativos.

Conforme Melchers e Beck (2018), na distribuição Lognormal, o logaritmo da variável aleatória terá uma distribuição Normal. A FDP da distribuição Lognormal é expressa pela Eq. (3.10) e sua FDA através da Eq. (3.11), e graficamente pela Fig. 3.3.

$$f_X(x) = \frac{1}{(2\pi)^{1/2} \zeta x} \exp \left[-\frac{1}{2} \left(\frac{\ln x - \lambda}{\zeta} \right)^2 \right] \quad 0 \leq x \leq \infty \quad (3.10)$$

$$F_X(x) = \int_0^x f_X(u) du = \Phi \left(\frac{\ln x - \lambda}{\zeta} \right) \quad (3.11)$$

onde λ é a média e ζ é o desvio padrão de $\ln(X)$.

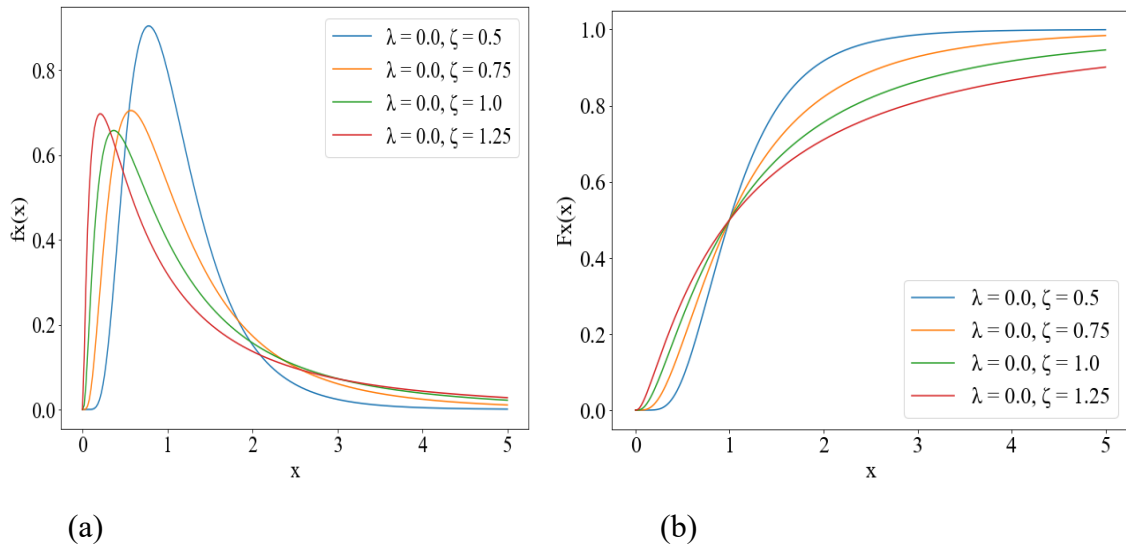


Figura 3.3 – Distribuição Lognormal: (a) função de densidade de probabilidade e (b) função de distribuição acumulada de probabilidade

Fonte: Autora

A Figura 3.3 apresenta o comportamento da distribuição Lognormal, sendo possível observar que a função de densidade de probabilidades admite somente valores positivos e não nulos. Segundo El-Reedy (2013), essa distribuição é muitas vezes empregada quando um fenômeno não pode assumir valores negativos. O JCSS (2001) sugere a utilização da distribuição Lognormal para representar modelos de resistência de aço e concreto.

Os momentos da variável Lognormal podem ser determinados por:

$$\mu_x = \exp [\lambda + 0,5\zeta^2] \quad (3.12)$$

$$\sigma_x = \mu_x \sqrt{(\exp[\zeta^2] - 1)} \quad (3.13)$$

e os parâmetros são calculados a partir dos momentos, de tal forma que:

$$\lambda = \ln(\mu_x) - 0,5\zeta^2 \quad (3.14)$$

$$\zeta = \sqrt{\ln(1 + (\sigma_x/\mu_x)^2)} . \quad (3.15)$$

A distribuição de Gumbel (valores extremos do tipo I) também será utilizada nessa pesquisa nos problemas estudados e analisados do capítulo 6. Segundo Haldar e Mahadevan (2000), seja X uma variável aleatória representada por uma função de distribuição de probabilidade $f_X(x)$, se houver n amostras para essa variável X , os valores extremos (máximo e mínimo) podem ser definidos. Se conjuntos de diferentes amostras do mesmo tamanho de n forem alcançados para X , cada conjunto terá valores distintos para mínimos e máximos. Usando todos esses conjuntos, funções de distribuições para os valores extremos podem ser definidas. A FDP da distribuição Gumbel é expressa pela Eq. (3.16) e sua FDA através da Eq. (3.17), e graficamente pela Fig. 3.4.

$$f_Y(y) = \alpha \exp[-\alpha(y-u) - e^{-\alpha(y-u)}] \quad -\infty < y < \infty \quad (3.16)$$

$$F_Y(y) = \exp[-e^{-\alpha(y-u)}] \quad -\infty < y < \infty \quad (3.17)$$

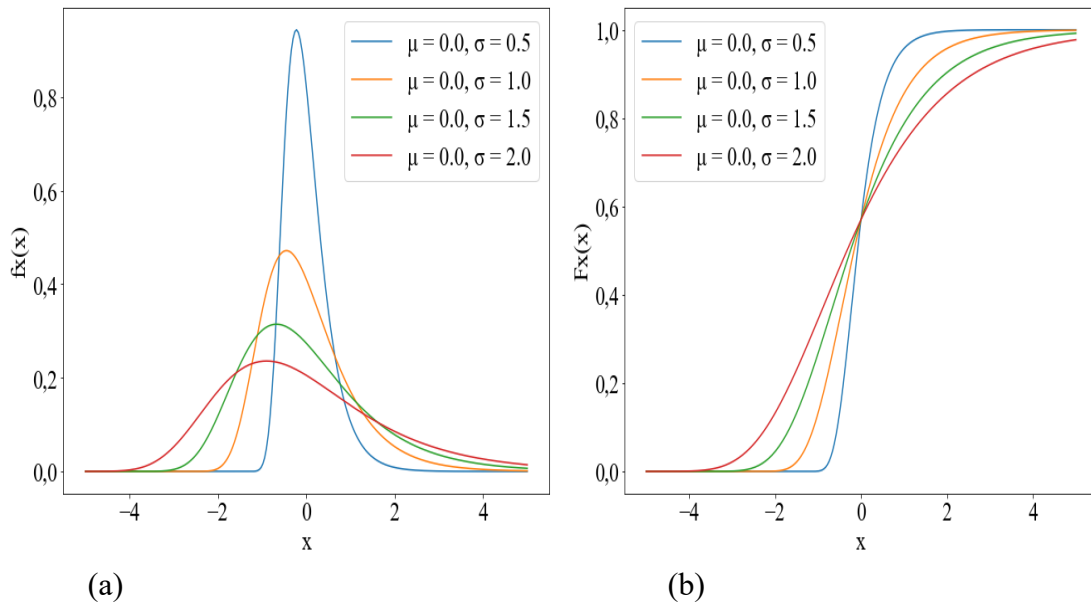


Figura 3.4 – Distribuição Gumbel: (a) função de densidade de probabilidade e (b) função de distribuição acumulada de probabilidade

Fonte: Autora

A Figura 3.4 apresenta o comportamento da distribuição de Gumbel para diferentes parâmetros. Na análise de confiabilidade estrutural a distribuição de Gumbel é comumente usada para representar cargas acidentais e a ação do vento (EL-REEDY, 2013).

Para Melchers e Beck (2018), os parâmetros dessa distribuição são u , que é a moda da distribuição e α , que é uma medida de sua dispersão. Tanto u quanto α podem ser obtidos, usando os momentos da distribuição, para o ajuste da curva aos dados observados, e representados por:

$$\mu_y = u + \gamma/\alpha \quad (3.18)$$

$$\sigma_y = \frac{\pi}{\alpha \sqrt{6}} \quad (3.19)$$

onde $\gamma = 0,577216$ é a constante de Euler. Conhecidos os momentos, os parâmetros são determinados, tal que:

$$u = \mu_y - \gamma/\alpha \quad (3.20)$$

$$\alpha = \frac{\pi}{\sigma_y \sqrt{6}}. \quad (3.21)$$

4. CONFIABILIDADE ESTRUTURAL

Conforme Nowak e Collins (2000), a confiabilidade pode ser descrita como a probabilidade de um sistema não falhar durante sua vida útil projetada, e dentro das condições estabelecidas para seu funcionamento. E é garantida a partir de modelos, cálculos e simulações através de métodos de transformação e de Monte Carlo que são baseados em variáveis aleatórias do elemento estrutural analisado.

A confiabilidade estrutural é uma medida do grau de confiança em que uma estrutura ou sistema estrutural cumpra aos requisitos técnicos de projeto, como: função, resistência e equilíbrio. Este grau de confiança reflete os modelos estatísticos utilizados na quantificação das incertezas, bem como os modelos físicos, matemáticos e de engenharia utilizados no cálculo (BECK, 2019).

4.1. Estados limites

Segundo Melchers e Beck (2018), a forma pela qual uma estrutura de engenharia responderá ao carregamento depende principalmente da carga aplicada e da resistência. Se a resposta é considerada positiva, isso inclui segurança da estrutura contra colapso, limitações de danos ou outros critérios. Cada um desses requisitos pode ser denominado estado limite. A ultrapassagem de um estado limite pode ser definido como a obtenção de uma condição indesejável para a estrutura.

Beck (2019) explica que os modos de falha e os estados limites são representados estritamente por falhas em estruturas. Os estados limites são basicamente distinguidos por duas categorias principais: último e serviço.

De acordo com a NBR 6118 (ABNT, 2014), os estados limites últimos são relacionados ao colapso, ou a qualquer outra forma de ruína estrutural, que determine a paralisação do uso da estrutura. A segurança de estruturas deve ser sempre verificada em relação aos estados limites últimos, tendo como exemplos:

- a) perda de equilíbrio da estrutura, admitida como corpo rígido;
- b) esgotamento da capacidade resistente da estrutura;
- c) colapso progressivo;
- d) mudança física ou geométrica da estrutura;
- e) ruptura ou instabilidade de peças estruturais.

Nesses casos, quando a estrutura está submetida a um estado limite último, isso caracteriza falha na estrutura, sendo necessário reparos ou substituições para que a segurança seja garantida.

Segundo a NBR 6118 (ABNT, 2014), os estados limites de serviço são aqueles determinados pelos requisitos de serviços e condições de uso, e estão relacionados à durabilidade, ao conforto do usuário, à aparência e à boa utilização da estrutura, sendo exemplos:

- a) formação de fissuras ou trincas exageradas;
- b) flechas excessivas em vigas e lajes;
- c) forte vibração da estrutura;
- d) deformação em nível não aceitável.

De acordo com Araújo (2014), uma estrutura não deve atingir nem um estado limite último, que corresponderia à ruína ou ao colapso; nem um estado limite de serviço, que tornaria inviável a sua utilização.

Beck (2019) afirma que cada forma distinta de chegar em um estado indesejável é chamada, genericamente, de um modo de falha. Onde cada modo de falha dá origem a uma equação de estado limite $g(X)$, que é função do vetor de variáveis aleatórias $X = \{X_1, X_2, \dots, X_n\}$, de tal forma que o estado limite fica determinado pela seguinte igualdade:

$$g(X) = g(X_1, X_2, \dots, X_n) = 0 \quad (4.1)$$

uma equação de estado limite $g(X)$ é escrita de forma a dividir o domínio do problema em domínios de falha Ω_f e de segurança Ω_s , tal que:

$$\Omega_f = \{X | g(X) \leq 0\} \quad (4.2)$$

$$\Omega_s = \{X | g(X) > 0\} \quad (4.3)$$

o domínio de falha Ω_f é constituído por todos os pontos do espaço amostral X que levam a falha da estrutura, como representado na Fig. 4.1. O domínio seguro Ω_s é o conjunto complementar ao domínio de falha.

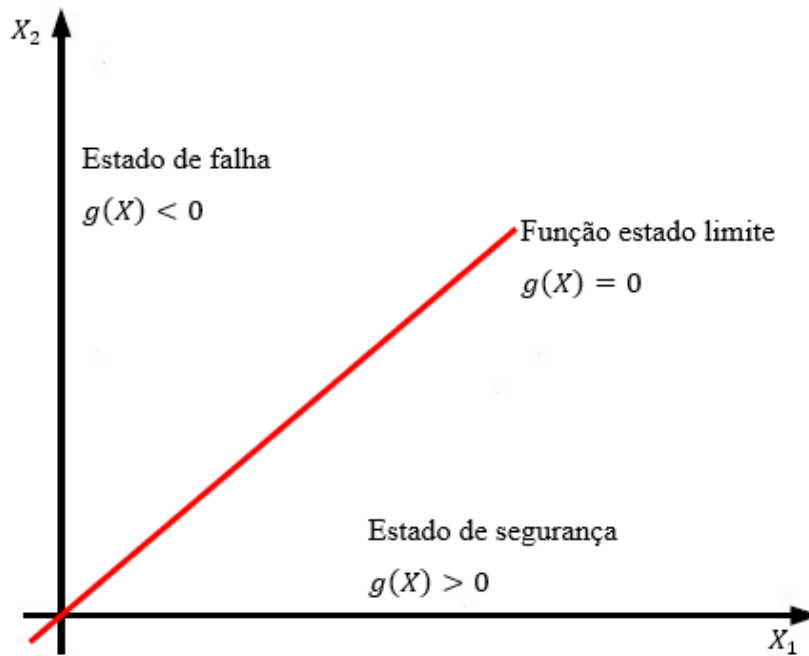


Figura 4.1 – Representação genérica para a função estado limite.

Fonte: Adaptado de Beck (2019)

4.2. Função estado limite

Conforme Melchers e Beck (2018), da mesma forma como existem incertezas em projetos de engenharia, há também algum risco de um sistema estrutural não atingir o desempenho projetado. Consequentemente, o desempenho desse sistema não pode ser garantido como totalmente seguro. Assim, para Santos (2014), é conveniente definir a função estado limite ou desempenho, considerando uma estrutura com resistência R e solicitação S , sendo possível, então, avaliar a probabilidade de falha. Uma função de estado limite em uma análise probabilística pode ser determinada por:

$$g(x) = R(x) - S(x) \quad (4.2)$$

onde x é o vetor de variáveis aleatórias do problema; $R(x)$ é a variável aleatória que representa a resistência da estrutura; $S(x)$ é a variável aleatória que representa a ação ou o esforço solicitante.

Sendo a probabilidade de falha (P_f) dessa estrutura definida quando a resistência é menor que a solicitação, e a probabilidade segura (P_s) é quando a resistência é maior ou igual à solicitação, sendo dadas pelas equações:

$$P_f = P[g(x) < 0] \quad (4.2)$$

$$P_s = 1 - P_f. \quad (4.3)$$

Em conformidade com o JCSS (2001), a probabilidade de falha deve ser calculada com base no tipo de distribuição conjunta das variáveis básicas e no formalismo distributivo padronizado de lidar com a incerteza do modelo e a incerteza estatística.

Na Figura 4.2 é apresentada uma superfície com região de falha, região segura e função estado limite para um sistema com duas variáveis.

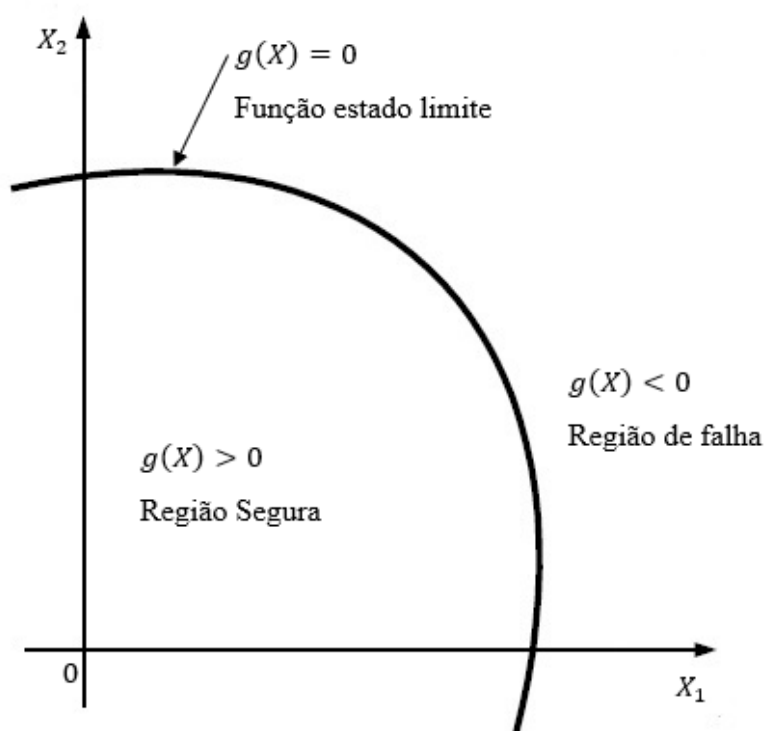


Figura 4.2 – Função estado limite para um sistema com duas variáveis.

Fonte: Autora

Segundo Haldar e Mahadevan (2000), para estruturas práticas e critérios de desempenho, é difícil calcular com precisão a probabilidade de falha. Porém, uma estimativa de primeira ordem emprega uma medida conhecida como índice de confiabilidade (denotado por β).

Conforme Sakamoto (2016), uma definição muito utilizada na prática por sua simplicidade é o índice de confiabilidade de Cornell (β_c), onde uma medida quantitativa

de confiabilidade com base no valor médio ou valor esperado $E[g(x)]$ e na variância $Var[g(x)]$ da função de estado limite linear $g(x)$, dado por:

$$\beta_c = \frac{E[g(x)]}{\sqrt{Var[g(x)]}}. \quad (4.4)$$

Utilizando a transformação proposta por Hasofer e Lind (1974), o índice de confiabilidade também pode ser obtido geometricamente como a menor distância da origem à função estado limite no espaço padronizado. Considerando a Eq. (4.1) com variáveis aleatórias R e S estatisticamente independentes e com distribuição de probabilidade normal, com médias μ_R e μ_S e desvios padrões σ_R e σ_S , pode-se transformar as variáveis R e S em variáveis reduzidas (Y_R e Y_S), com média zero e desvio padrão unitário, utilizando a Eq. (3.9), tem-se as equações:

$$Y_R = \frac{R - \mu_R}{\sigma_R} \quad (4.5)$$

$$Y_S = \frac{S - \mu_S}{\sigma_S} \quad (4.6)$$

substituindo as Eqs. (4.5) e (4.6) na Eq. (4.1), a função desempenho no estado limite linear no espaço padronizado pode ser descrita por:

$$g(S_R, S_S) = Y_R \sigma_R - Y_S \sigma_S + (\mu_R - \mu_S) \quad (4.7)$$

Para Beck (2019), o índice de confiabilidade calculado em problemas envolvendo variáveis aleatórias normais estatisticamente independentes e uma função de estado limite linear do tipo $g(x) = R - S$, corresponde à menor distância entre a origem e a função, e é dada através de:

$$\beta = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}}. \quad (4.8)$$

A probabilidade de falha pode ser obtida em função do índice de confiabilidade, se R e S tiverem distribuições normais, determinada por:

$$P_f = \Phi(-\beta). \quad (4.9)$$

onde Φ é a função de distribuição acumulada da variável normal padrão, definida por:

$$\Phi(\beta) = \int_{-\infty}^{\beta} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy. \quad (4.10)$$

Conforme Melchers e Beck (2018), para as funções de estado limite lineares, a probabilidade de falha pode ser obtida diretamente no índice de confiabilidade. Se a função for não linear, ainda poderá ser definida, mas somente com relação a um plano (hiper) tangente aproximado. Em cada um desses casos, β representa a menor distância entre a origem no espaço normal e o plano tangente, como mostra a Fig. 4.3.

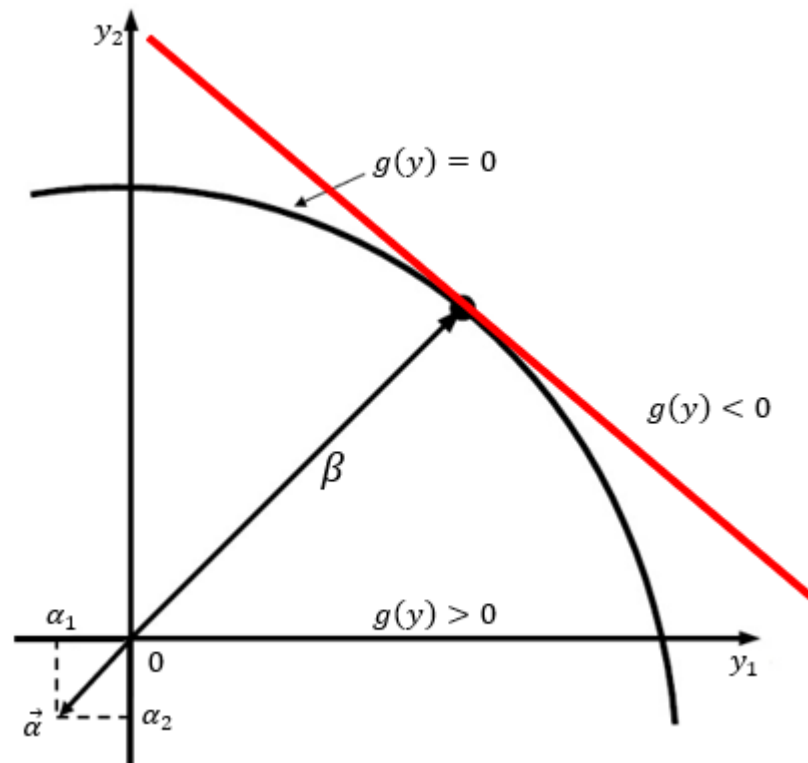


Figura 4.3 – Representação do índice de confiabilidade.

Fonte: Autora.

Observando a Fig. 4.3, e ainda segundo Melchers e Beck (2018), o ponto correspondente no hiperplano é denominado ponto de projeto. Esse ponto é o de maior densidade de probabilidade dentro do espaço abrangido pela região de falha.

5. MÉTODOS DE CONFIABILIDADE

Segundo Haldar e Mahadevan (2000), um projeto de engenharia consiste em propor os elementos essenciais para que um sistema satisfaça vários critérios como: desempenho, segurança, facilidade de manutenção e durabilidade. No entanto, existem inúmeras fontes de incerteza nos parâmetros desses critérios. Os métodos de confiabilidade apresentados nessa pesquisa pretendem identificar e quantificar incertezas em variáveis aleatórias apresentadas nos problemas do capítulo 6.

Conforme Beck (2019), os métodos de confiabilidade estrutural são divididos principalmente em dois grupos: métodos de simulação de Monte Carlo e métodos de transformação, e estão apresentados na Fig. 5.1.

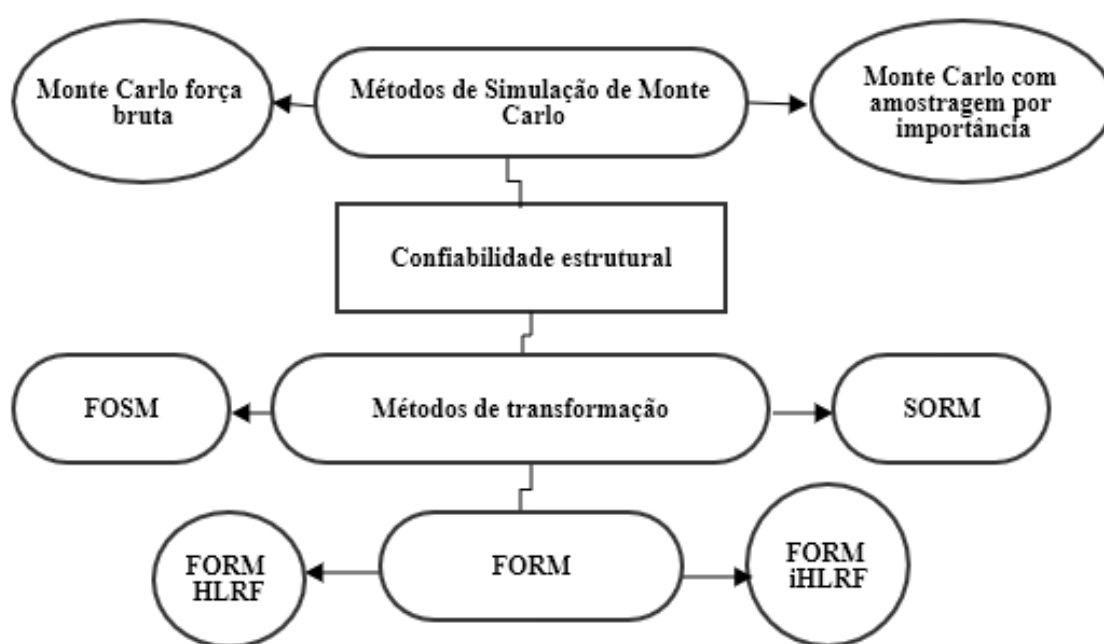


Figura 5.1 – Métodos de Confiabilidade estrutural

Fonte: Autora

5.1. Simulação de Monte Carlo

A simulação de Monte Carlo é um método numérico que se baseia em amostragens aleatórias. Segundo Beck (2019), em engenharia de estruturas, a simulação de Monte Carlo se adapta tão bem a problemas lineares como não lineares, estáticos ou dinâmicos. Em confiabilidade estrutural, a simulação resolve, com a mesma facilidade, problemas envolvendo uma única equação de estado limite, dependente do tempo ou não.

Beck (2019) explica que o método de simulação de Monte Carlo está baseado na geração das n amostras $X_k\{x_1, x_2, \dots, x_n\}_k^t$, $k = 1, \dots, n$. Cada uma dessas amostras contém n números aleatórios.

Segundo Santos (2014), em confiabilidade de estruturas, principalmente em simulação de Monte Carlo, um passo importante é a geração de amostras de variáveis aleatórias. Considerando que uma variável aleatória pode ser definida por sua função de densidade de probabilidade, a partir da função cumulativa de probabilidade $F_X(x)$, pode-se obter uma amostra aleatória da seguinte forma:

- Geração de um número aleatório u_1 com distribuição uniforme de probabilidade, que está entre 0 e 1;
- A partir da inversa da função cumulativa de probabilidade (FDA) se obtém o número aleatório com distribuição prescrita, como mostra:

$$x_i = F_X^{-1}(u_i). \quad (5.1)$$

Esse procedimento pode ser visualizado na Fig. (5.2).

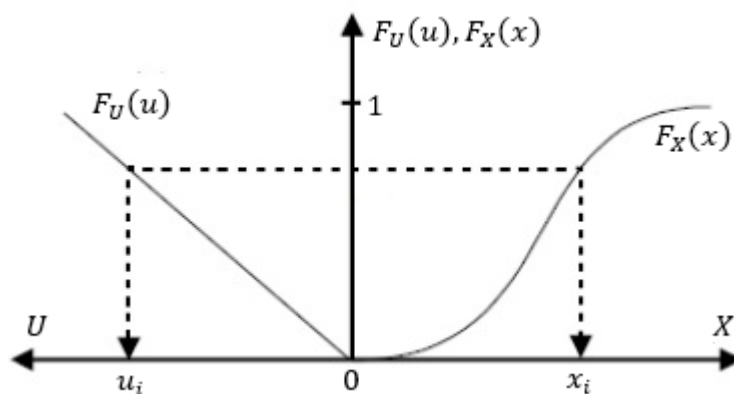


Figura 5.2 – Geração de números aleatórios com distribuição prescrita

Fonte: Adaptado Haldar e Mahadevan (2000)

O procedimento descrito não se adequa para distribuição Normal e consequentemente Lognormal, devido à não existência de solução analítica para a função cumulativa de probabilidade para essas distribuições. Dessa forma, é recomendada a utilização de algoritmos específicos. Pode-se adotar um par de números aleatórios u_1 e u_2 , independentes e com distribuição de probabilidade uniforme, no intervalo de $[0,1]$,

para gerar um par de números aleatórios y_1 e y_2 de uma variável normal padrão (SOONG e GRIGORIU, 1993). Os números y_1 e y_2 são obtidos por:

$$y_1 = \sqrt{-2 \ln(u_1)} \cos(2\pi u_2) \quad (5.2)$$

$$y_2 = \sqrt{-2 \ln(u_1)} \sin(2\pi u_2). \quad (5.3)$$

Transformando os números aleatórios do espaço normal padrão $\mathbb{Y}$ para o espaço $\mathbb{X}$, utiliza-se a transformação de Hasofer-Lind, dessa forma, têm-se:

$$x_i = y_i \sigma + \mu \quad (5.4)$$

onde μ e σ são a média e o desvio padrão, respectivamente, da variável X .

Se as variáveis aleatórias do problema são independentes, então a distribuição conjunta de probabilidade pode ser obtida através de:

$$f_X(x) = \prod_{i=1}^n f_{X_i}(x_i) \quad (5.5)$$

sendo $f_{X_i}(x_i)$ a função de densidade de probabilidade marginal da variável X_i . Isto significa que utilizando as Eqs. (5.1) e (5.4) os números aleatórios podem ser gerados de forma independente dos demais (BECK, 2019).

5.1.1. Monte Carlo força bruta

O método de simulação de Monte Carlo força bruta (MCFB) é um método numérico que se baseia em repetição do processo de simulação, usando em cada simulação um determinado conjunto de valores. Segundo Beck (2019), em engenharia de estruturas, a simulação de MCFB se adapta tão bem a problemas lineares como não lineares.

O método de simulação Monte Carlo é dito bruto ou força bruta quando este é apresentado em sua forma mais simples. A solução de problemas de confiabilidade através da simulação de MCFB é realizada utilizando uma função indicadora $I_g(x)$ tal que, $I_g(x) = 1$, se $g(x) < 0$; $I_g(x) = 0$, se $g(x) \geq 0$. Sendo, $g(x) < 0$ (domínio de

falha) e $g(x) \geq 0$ (domínio de segurança), podem-se estender os limites de integração sobre todo o domínio da função $g(x)$ das variáveis aleatórias para se calcular a probabilidade de falha, como mostrado em:

$$P_f = \int_{g(x)} I_g(x) f_X(x) dx. \quad (5.6)$$

Conforme Silva Junior (2019), o método de simulação de MCFB consiste em obter uma estimativa da distribuição de probabilidade da função $g(x)$, através da geração dos parâmetros das variáveis de entrada. A cada simulação, é verificada a condição da função $g(x)$ ser menor ou maior que zero. A probabilidade de falha e o índice de confiabilidade na simulação de MCFB quando as distribuições das variáveis aleatórias são do tipo mistas, são dados através das equações:

$$P_f = \frac{1}{N} \sum_{i=1}^N I_g(x_i) \quad (5.7)$$

$$\beta = -\Phi^{-1}(P_f) \quad (5.8)$$

onde N é o número de simulações, e Φ^{-1} é a função inversa de densidade de probabilidade acumulada da variável, também chamada de distribuição gaussiana inversa.

De acordo com Ang e Tang (2007), quanto menor a probabilidade de falha, maior o número de simulações para que pelo menos uma falha ocorra. Dessa forma, conhecer o erro correspondente a uma probabilidade estimada, ou ainda, saber quantas simulações (tamanho da amostra) são requeridas para se obter um certo intervalo de confiança, é dada a Eq. (5.9) para o erro percentual:

$$\%erro = 200 \sqrt{\frac{1 - P_f}{nP_f}} \quad (5.9)$$

onde P_f é a probabilidade de falha estimada e n é o tamanho da amostra. O número de simulações para algumas probabilidades de falha utilizando a Eq. (5.9) é apresentado na Tab. 5.1.

Tabela 5.1 – Número de simulações para diferentes probabilidades de falha

P_f	n		
	<i>erro</i> = 10%	<i>erro</i> = 5%	<i>erro</i> = 1%
1e-03	4e+05	2e+06	4e+07
1e-04	4e+06	2e+07	4e+08
1e-05	4e+07	2e+08	4e+09
1e-06	4e+08	2e+09	4e+10
1e-07	4e+09	2e+10	4e+11

De acordo com Haldar e Mahadevan (2000), a simulação de MCFB possui seis elementos essenciais, e o algoritmo do método é apresentado na Fig. 5.3.

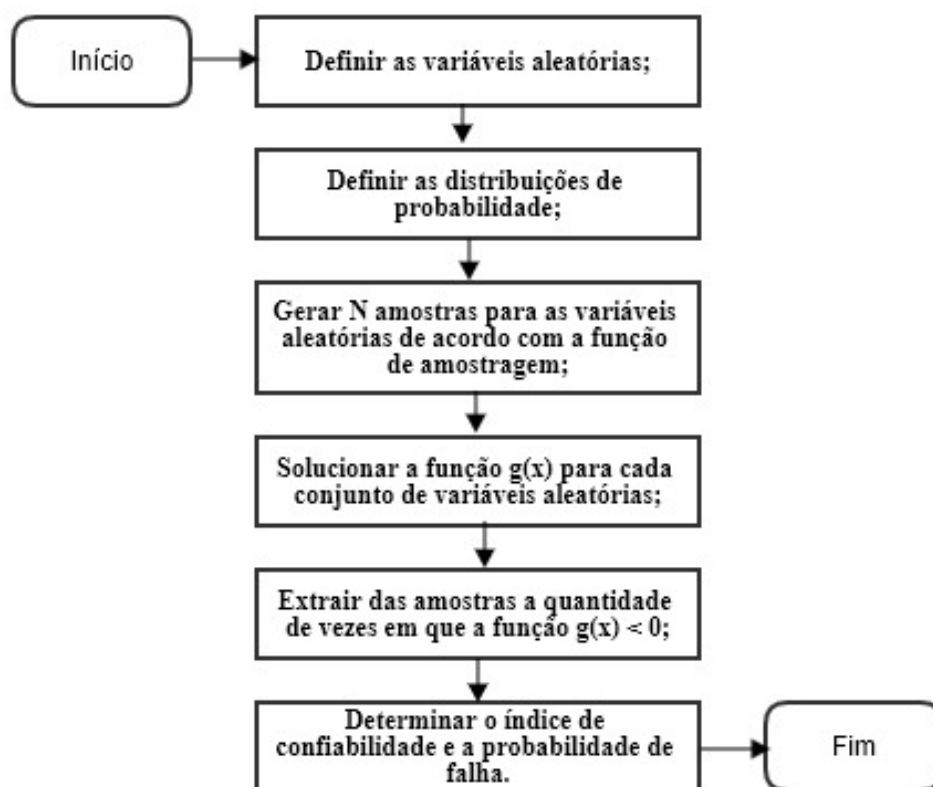


Figura 5.3 – Fluxograma do algoritmo do método Monte Carlo força bruta

Fonte: Autora

5.1.2. Monte Carlo com amostragem por importância

Uma das técnicas para acelerar a convergência da simulação de Monte Carlo é a utilização de amostragem por importância. A técnica consiste em determinar a probabilidade da falha na região de estado limite. Beck (2019) afirma que a técnica

amostragem por importância consiste em utilizar uma função de amostragem, a fim de evitar a geração de amostras longe da região de interesse, ou seja, longe do domínio de falha. No mesmo sentido Haldar e Mahadevan (2000) explicam que a ideia da amostragem de importância é concentrar a distribuição dos pontos de amostragem na região de maior importância. A Fig. 5.4 ilustra o funcionamento do Método MCAI, para um caso bidimensional.

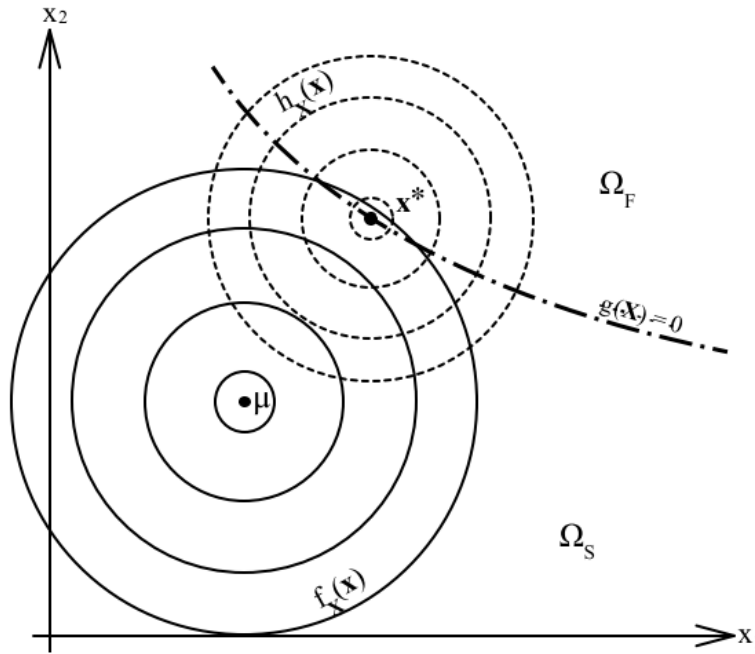


Figura 5.4 – Método de Monte Carlo com Amostragem por Importância

Fonte: TITELLO (2020)

Segundo Beck (2019), as técnicas de amostragem por importância estão entre as mais eficientes. Os pontos são gerados a partir de uma função de amostragem $h_X(x)$. Multiplicando e dividindo a Eq. (5.6) por $h_X(x)$, tem-se:

$$P_f = \int_{g(x)} \left[I_g(x) \frac{f_X(x)}{h_X(x)} \right] h_X(x) dx. \quad (5.10)$$

Ainda conforme Beck (2019), essa expressão corresponde ao valor esperado da função $I_g(x) \frac{f_X(x)}{h_X(x)}$, em relação à função de amostragem, $h_X(x)$. Este valor pode ser estimado com base em amostra de tamanho N , e representado através de:

$$P_f \approx \hat{P}_f = \frac{1}{N} \sum_{i=1}^N I_g(x_i) \frac{f_X(x_i)}{h_X(x_i)} \quad (5.11)$$

O ponto mostrado na Eq. (5.12) está associado a um peso $w_k \ll 1$, que é representado pela razão entre as funções $f_X(x)$ e $h_X(x)$. A variância da probabilidade de falha é estimada pela Eq. (5.13).

$$w_k = \frac{f_X(x)}{h_X(x)} \quad (5.12)$$

$$Var[\hat{P}_f] = \frac{1}{(N-1)} \sum_k^N (I_g[x_k]w_k - \hat{P}_f)^2 \quad (5.13)$$

De acordo com Haldar e Mahadevan (2000), o algoritmo do método MCAI é apresentado na Fig. 5.5.

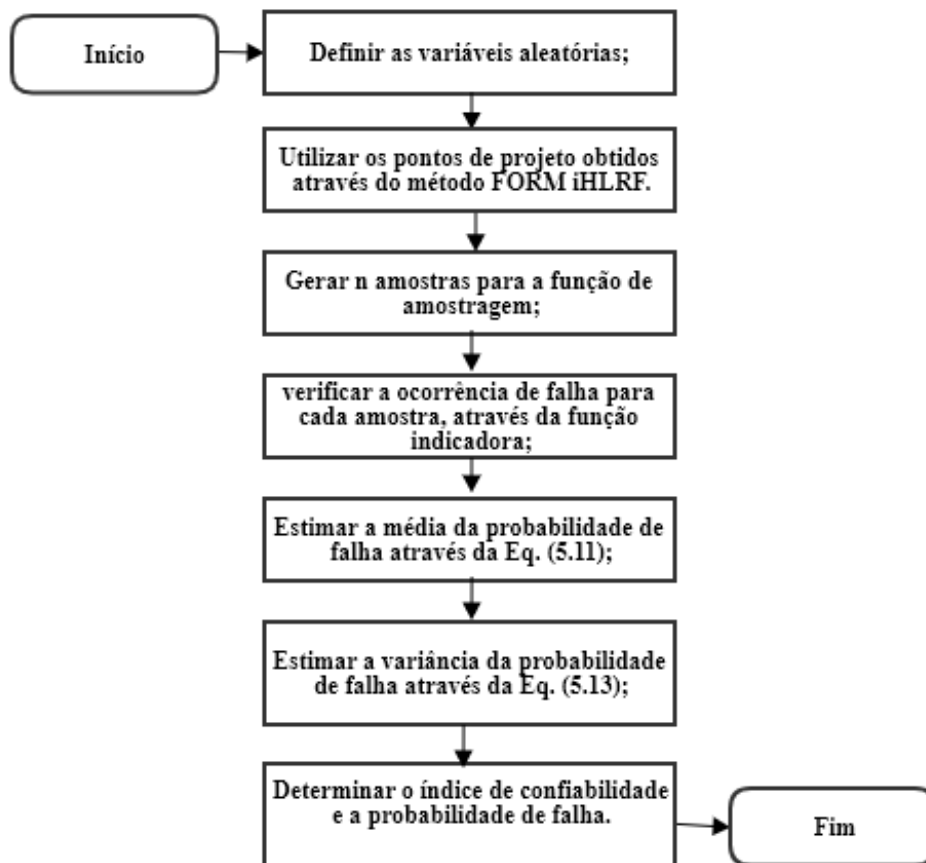


Figura 5.5 – Fluxograma do algoritmo do método Monte Carlo com amostragem por importância.

Fonte: Autora

5.2. Métodos de transformação

Para Beck (2019), os métodos de confiabilidade chamados de métodos de transformação são o FOSM, FORM e SORM. Esses métodos consistem na transformação de Hasofer-Lind reduzindo um vetor de variáveis aleatórias normais com médias e desvios padrão, a um vetor de variáveis aleatórias normais padrão com média zero e desvio padrão unitário não correlacionados.

Os métodos de transformação serão abordados de forma inicial, onde cada variável será interpretada por dois momentos, ou seja, média e desvio padrão, e ainda, pode ser representada por uma distribuição Normal (ou gaussiana).

Conforme Haldar e Mahadevan (2000), o FOSM e o FORM podem ser usados para avaliar uma função de variáveis normais e estatisticamente independentes. O FORM também pode representar uma função de estado limite não linear por uma aproximação de primeira ordem com variáveis normais equivalentes. E o SORM determina a probabilidade de falha aproximando a função estado limite não linear por uma representação de segunda ordem.

O FORM aproxima a função estado limite no ponto de projeto por um hiperplano, enquanto o SORM por uma hipersuperfície quadrática. Na Fig. 5.6 é mostrada a aproximação de primeira e segunda ordem.

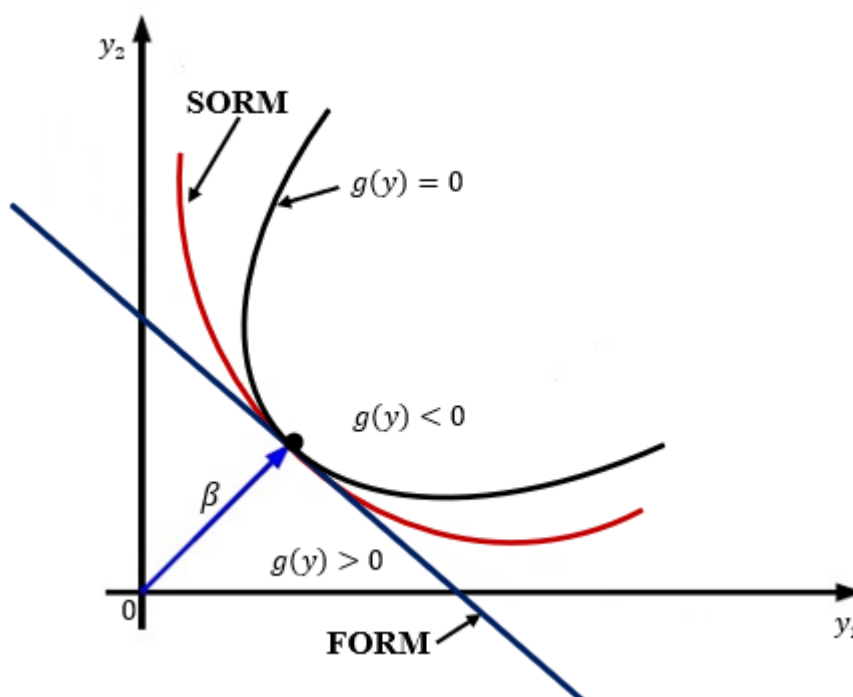


Figura 5.6 – Aproximação de primeira e segunda ordem (FORM e SORM)

Fonte: Autora

5.2.1. FOSM

Conforme Beck (2019), no método de primeira ordem e segundo momento (FOSM), a equação de estado limite é aproximada por uma função linear, e a informação para construção de $f_X(x)$ se limita aos momentos de até segunda ordem (média e covariância). Uma representação das variáveis do problema por seus momentos, corresponde a assumi-las como distribuição normal.

De acordo com Beck (2019), a solução de problemas via método FOSM é a transformação do espaço de projeto $\mathbb{X}$ para o espaço normal padrão $\mathbb{Y}$. A função de estado limite em geral é formulada e avaliada no espaço de projeto $\mathbb{X}$. Sendo, no espaço $\mathbb{Y}$ feita a busca pelo ponto de projeto. Neste caso, basta padronizar as variáveis de projeto por meio da transformação de Hasofer-Lind.

Para trabalhar com problemas envolvendo um grande número de variáveis aleatórias é conveniente usar a forma matricial. Dessa forma, denotando o vetor de médias por V e a matriz diagonal de desvios padrão por D , ou seja,

$$V = (\mu_{x_1}, \dots, \mu_{x_n})^t \quad \text{e} \quad D = \text{diag}(\sigma_{x_1}, \dots, \sigma_{x_n})$$

a transformação de Hasofer-Lind pode ser escrita de forma matricial através das equações:

$$y = D^{-1}\{x - V\} \tag{5.14}$$

$$x = Dy + V \tag{5.15}$$

onde D^{-1} é a matriz inversa de D .

Segundo Beck (2019), para implementação das transformações compostas do método de primeira ordem é adequado trabalhar com matrizes Jacobianas, definidas por:

$$J_{yx} = \left[\frac{\partial y_i}{\partial x_j} \right]_{i=1, \dots, n; j=1, \dots, n} = D^{-1} \tag{5.16}$$

$$J_{xy} = \left[\frac{\partial x_i}{\partial y_j} \right]_{i=1, \dots, n; j=1, \dots, n} = D \tag{5.17}$$

sendo n o número de variáveis aleatórias do problema. A transformação de Hasofer-Lind e inversa resultam nas equações:

$$y = J_{yx}\{x - V\} \quad (5.18)$$

$$x = J_{xy}y + V \quad (5.19)$$

como as funções de estado limite são escritas no espaço das variáveis originais ($\mathbb{X}$), as derivadas também podem ser avaliadas nesse mesmo espaço. Dessa forma, o vetor gradiente $\nabla g(x)$ é obtido através de:

$$\nabla g(y) = \left\{ \frac{\partial g}{\partial y_j} \right\}_{j=1, \dots, n} = \left\{ \sum_{i=1}^n \frac{\partial g}{\partial x_i} \frac{\partial x_i}{\partial y_j} \right\}_{j=1, \dots, n} = (J_{xy})^T \nabla g(x). \quad (5.20)$$

Segundo Beck (2019), não existe benefício aparente em trabalhar com matrizes jacobianas para realização das transformações necessárias no FOSM. Estes benefícios surgirão no contexto do método de primeira ordem (FORM).

O algoritmo de solução do método FOSM é apresentado a seguir (BECK, 2019).

Algoritmo 5.1. Método de confiabilidade FOSM

1. escolha do ponto de falha inicial x^k para $k=0$ (usualmente o ponto médio);
2. avaliação das matrizes Jacobianas (J_{yx} e J_{xy}) e do vetor de médias V ;
3. transformação do ponto x^k de $\mathbb{X} \rightarrow \mathbb{Y}$;
4. avaliação de $g(x^k)$;
5. cálculo do gradiente:
 - a. cálculo das derivadas parciais de $g(x)$ no espaço de projeto $\mathbb{X}$
 - b. transformação do gradiente para $\mathbb{Y}$
 - c. cálculo dos cossenos diretores $\alpha(y^k)$
6. Cálculo do índice de confiabilidade (β)
7. cálculo do novo ponto de falha y^{k+1} através da Eq. (5.21);

$$y^{k+1} = \mu_{X_i}^N - \alpha_{X_i} \beta \sigma_{X_i}^N \quad (5.21)$$

8. transformação de y^{k+1} em $\mathbb{X}$;
9. verificação do critério de convergência. Se:

$$1 - \frac{\nabla g(y^{k+1}) y^{k+1}}{\|\nabla g(y^{k+1})\| \|y^{k+1}\|} < \epsilon \quad \text{e} \quad |g(y^{k+1})| < \delta$$

sendo $\epsilon = 10^{-6}$; o algoritmo é interrompido ($y^* = y^{k+1}$); caso contrário, retorne ao item 4 com $k=k+1$ até atingir a convergência;

10. ao final, avaliação do índice de confiabilidade no ponto de projeto: $\beta = \|y^*\|$.

De acordo com Haldar e Mahadevan (2000), o método FOSM fornece a solução exata da probabilidade de falha apenas nos casos em que a função de estado limite é linear e as variáveis aleatórias são estatisticamente independentes e possuem distribuição Normal. Em situações práticas, essa hipótese é bastante limitante, mesmo assim, é um método importante, pois constitui a base para a construção dos métodos FORM e SORM, que podem ser usados na aplicação de resolução de problemas com variáveis aleatórias de qualquer distribuição de probabilidade.

5.2.2. FORM

Para Beck (2019), o método FORM tem sido amplamente empregado para a obtenção do índice de confiabilidade. No espaço das variáveis normais padronizadas, o índice de confiabilidade β pode ser obtido pela interpretação geométrica da função estado limite. Na Fig. 4.2 o índice de confiabilidade corresponde a menor distância entre a origem e a função de estado limite.

O algoritmo do FORM é igual ao do FOSM para um problema com distribuição de probabilidade normal das variáveis aleatórias. Quando as distribuições são mistas, é necessário calcular a média e o desvio padrão de uma distribuição normal equivalente. Segundo Beck (2019), o método FORM envolve a construção de uma função conjunta de probabilidades, $f_X(x)$, bem como uma transformação desta para o espaço normal padrão y . Esta transformação envolve a eliminação da correlação entre variáveis aleatórias e o cálculo de variáveis normais equivalentes, representado na Fig. 5.7.

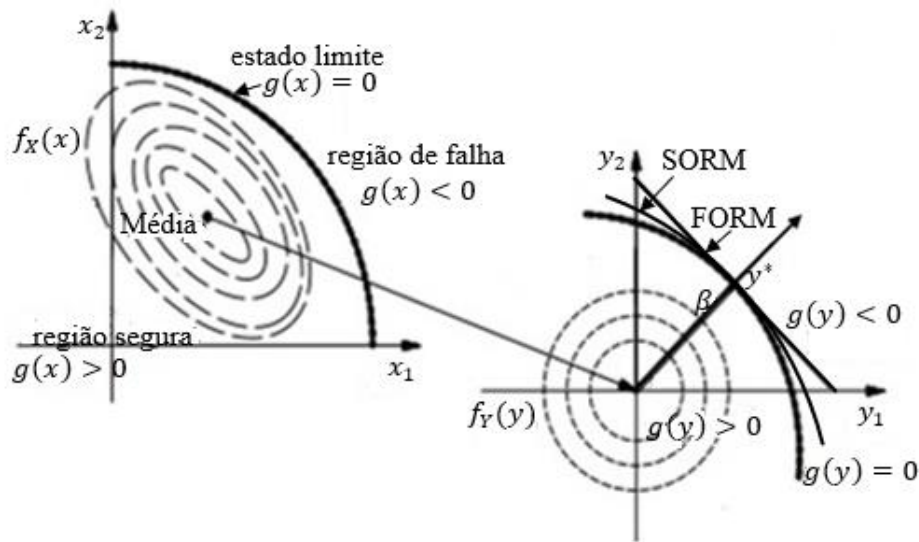


Figura 5.7 – Transformação do espaço normal para o normal equivalente

Fonte: Adaptado Youn *et al.* (2005)

Para Haldar e Mahadevan (2000), o FORM se baseia na construção da função de probabilidade $f_X(x)$ e sua transformação para o espaço normal padrão $f_Y(y)$. Essa construção envolve a eliminação da correlação entre variáveis aleatórias e o cálculo das variáveis normais equivalentes. Essa transformação corresponde a um mapeamento indireto do vetor $X \in \mathbb{X}$ para o espaço $\mathbb{Y}$, através de:

$$Y = \mathbb{M}[X], g_Y(Y, m) = g(\mathbb{M}^{-1}[Y], m) \quad (5.22)$$

Para a construção da função conjunta de probabilidade no espaço de projeto $\mathbb{X}$, a descrição estatística das variáveis de projeto é feita através das distribuições de probabilidades marginais:

$$f_{X_i}(x_i), \quad i = 1, 2, \dots, n,$$

e através de uma matriz de correlação R_X entre pares de variáveis, como mostrado em:

$$R_X = \begin{bmatrix} \rho_{X_{11}} & \rho_{X_{12}} & \cdots & \rho_{X_{1n}} \\ \rho_{X_{21}} & \rho_{X_{22}} & \cdots & \rho_{X_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{X_{n1}} & \rho_{X_{n2}} & \cdots & \rho_{X_{nn}} \end{bmatrix} \quad (5.23)$$

onde n é o número de variáveis aleatórias e $\rho_{X_{ii}} = 1$, com $i = 1, 2, \dots, n$ (BECK, 2019).

Segundo Beck (2019), a transformação de $f_X(x)$ para $f_Y(y)$ representa um mapeamento um-a-um que leva pontos do espaço original para o espaço reduzido. Teoricamente, esse mapeamento pode ser realizado por meio da Transformação de *Rosenblatt*, sendo inapropriada na prática, por envolver integrações multidimensionais e distribuições condicionais que dificilmente são conhecidas.

Conforme Melchers e Beck (2018), a busca do ponto de projeto y^* é uma das etapas fundamentais para a obtenção da probabilidade de falha pelo método FORM. A transformação de variáveis quaisquer, correlacionadas ou não, em variáveis normais equivalentes estatisticamente independentes é realizada através da transformação do modelo de *Nataf*.

De acordo com Ditlevsen e Madsen (2007), a transformação baseada no princípio da aproximação normal consiste no mapeamento realizado de $\mathbb{X} \rightarrow \mathbb{Y}$, e esse é dito indireto, pois se utiliza uma transformação da distribuição conjunta de probabilidade das variáveis $X \in \mathbb{X}$ em uma distribuição conjunta de variáveis normais equivalentes $Z \in \mathbb{Z}$, onde $\mathbb{Z}$ é o espaço de probabilidade das variáveis aleatórias normais equivalentes que mantêm possíveis correlações entre pares de variáveis aleatórias. Os parâmetros da distribuição Normal equivalente, média $\mu_{X_i}^{neq}$ e desvio padrão $\sigma_{X_i}^{neq}$, são obtidos impondo duas condições, através de:

$$F_{X_i}^{neq}(x_i^*) = F_X(x_i^*) \quad (5.24)$$

$$f_{X_i}^{neq}(x_i^*) = f_X(x_i^*). \quad (5.25)$$

Utilizando a transformação de Hasofer-Lind, Eq. (3.9), obtém-se um conjunto de variáveis $Z = (Z_1, Z_2, \dots, Z_n)$, possivelmente correlacionadas, com distribuição Normal padrão, mostrado em:

$$z_i^* = \frac{x_i^* - \mu_{X_i}^{neq}}{\sigma_{X_i}^{neq}} \quad (5.26)$$

Reescrevendo as Eqs. (5.24) e (5.25) em função de z_i^* , obtêm-se as equações:

$$F_{X_i}^{neq}(x_i^*) = \Phi\left(\frac{x_i^* - \mu_{X_i}^{neq}}{\sigma_{X_i}^{neq}}\right) = \Phi(z_i^*) \quad (5.27)$$

$$f_{X_i}^{neq}(x_i^*) = \frac{1}{\sigma_{X_i}^{neq} \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x_i^* - \mu_{X_i}^{neq}}{\sigma_{X_i}^{neq}}\right)^2\right] = \frac{\phi(z_i^*)}{\sigma_{X_i}^{neq}} \quad (5.28)$$

onde $\Phi(z_i^*)$ e $\phi(z_i^*)$ são, respectivamente, função de distribuição acumulada normal padrão e função densidade de probabilidade normal padrão da variável z_i^* .

Da Equação (5.28) obtém-se uma expressão para o desvio padrão da distribuição Normal equivalente, dada por:

$$\sigma_{X_i}^{neq} = \frac{\phi(z_i^*)}{f_X(x_i^*)} \quad (5.29)$$

e, utilizando a Eq. (5.26) obtém-se uma expressão para a média da distribuição Normal equivalente, dada por:

$$\mu_{X_i}^{neq} = x_i^* - z_i^* \sigma_{X_i}^{neq} \quad (5.30)$$

Beck (2019) explica que esta transformação é realizada para cada uma das distribuições marginais e é válida somente no ponto x^* . Dessa forma, à medida que o algoritmo de busca do ponto de projeto avança ($x_k \rightarrow x^*$), a transformação deve ser refeita. Assim como a transformação de Hasofer-Lind, a transformação de $\mathbb{X} \rightarrow \mathbb{Z}$ também pode ser escrita matricialmente a partir de um vetor de médias V^{neq} e de uma matriz diagonal de desvio padrão D^{neq} , contendo os parâmetros das distribuições Normais equivalentes, ou seja,

$$V^{neq} = (\mu_{X_1}^{neq}, \dots, \mu_{X_n}^{neq})^t \quad \text{e} \quad D^{neq} = \text{diag}(\sigma_{X_1}^{neq}, \dots, \sigma_{X_n}^{neq})$$

sendo, a inversa da matriz diagonal de desvios padrão denotado por $(D^{neq})^{-1}$.

Introduzindo as matrizes jacobianas, utilizando a Eq. (5.26), têm-se:

$$J_{zx} = \left[\frac{\partial z_i}{\partial x_j} \right] = \begin{cases} \frac{1}{\sigma_{x_i}^{neq}}, & \text{para } i = j \\ 0, & \text{para } i \neq j \end{cases} \quad (5.31)$$

$$J_{xz} = \left[\frac{\partial x_i}{\partial z_j} \right] = \begin{cases} \sigma_{x_i}^{neq}, & \text{para } i = j \\ 0, & \text{para } i \neq j \end{cases} \quad (5.32)$$

com $i = 1, 2, \dots, n$ e $j = 1, 2, \dots, n$.

As respectivas transformações $\mathbb{X} \rightarrow \mathbb{Z}$ e $\mathbb{Z} \rightarrow \mathbb{X}$, são dadas de forma matricial pelas equações.

$$z = J_{zx}\{x - V^{neq}\} \quad (5.33)$$

$$x = J_{xz}z + V^{neq}. \quad (5.34)$$

Para Santos (2012), é importante destacar que, caso exista correlação entre as variáveis originais X , a transformação dada pela Eq. (5.32) não elimina a correlação entre as variáveis normais padrão Z . Beck (2019) explica que a correlação entre pares de variáveis originais deve ser imposta na distribuição conjunta normal padrão $f_Z(z)$, isto é, $f_Z(z) = \phi(z, R_z)$ onde R_z é a matriz de correlação equivalente a ser determinada. Essa transformação pode ser feita por meio da transformação de Nataf.

Segundo Liu e Kiureghian (1986), a transformação proposta por Nataf consiste em construir uma aproximação para a função conjunta de densidade de probabilidade $f_X(x)$ a partir da função de densidade normal padrão com matriz de correlação R_z , como apresentado através de:

$$f_X(x) = \phi_n(z, R_z) \frac{f_{X_1}(x_1)f_{X_2}(x_2) \dots f_{X_n}(x_n)}{\phi_{Z_1}(z_1)\phi_{Z_2}(z_2) \dots \phi_{Z_n}(z_n)} \quad (5.35)$$

onde $f_{X_i}(x_i)$ é a distribuição marginal de X_i e $\phi_{Z_i}(z_i)$ é a função densidade de probabilidade normal padrão, com $i = 1, 2, \dots, n$. Os coeficientes de correlação equivalentes para as distribuições marginais normais podem ser obtidos de maneira iterativa por meio de:

$$\rho_{X_{ij}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} z_i z_j \phi_2(z_i, z_j, \rho_{Z_{ij}}) dz_i dz_j \quad (5.36)$$

onde $\phi_2(\cdot)$ é a distribuição normal padrão bivariada. A Eq. (5.36) permite calcular $\rho_{Z_{ij}}$ em função do $\rho_{X_{ij}}$, de forma a se construir uma matriz de correlação equivalente, porém calcular o termo desconhecido dentro da integral dupla, pode ser bastante trabalhoso.

Liu e Kiureghian (1986) desenvolveram um conjunto de fórmulas analíticas aproximadas que fornecem uma relação R_{ij} entre $\rho_{Z_{ij}}$ e $\rho_{X_{ij}}$ para várias combinações de distribuições, como mostrado por:

$$R_{ij} = \frac{\rho_{Z_{ij}}}{\rho_{X_{ij}}}. \quad (5.37)$$

Elimina-se a correlação entre os pares de variáveis aleatórias levando-as de $\mathbb{Z} \rightarrow \mathbb{Y}$. Essa eliminação pode ser obtida por meio da decomposição ortogonal ou da decomposição de Cholesky da matriz de covariância. Sendo, C_Y e C_Z , respectivamente, as matrizes de correlação em $\mathbb{Y}$ e $\mathbb{Z}$. Nas Eqs. (5.38) e (5.39) são apresentadas as matrizes jacobianas utilizando a decomposição ortogonal da matriz de correlação:

$$J_{yz} = A^T = (\bar{A} \Lambda^{-1/2})^t \quad (5.38)$$

$$J_{zy} = (A^T)^{-1} = (\Lambda^{1/2} \bar{A})^t \quad (5.39)$$

onde $\bar{A}$ é uma matriz ortogonal cujas colunas são os autovetores de $C_Z = R_X$, e Λ é a matriz cujos elementos da diagonal principal são as inversas das raízes dos autovalores de C_Z .

O custo computacional da decomposição ortogonal é considerável. Uma alternativa que se aplica com vantagens para matrizes de correlação não cheias é a decomposição de Cholesky da matriz de correlação C_Z (BECK, 2019). Utilizando a decomposição de Cholesky de C_Z , tem-se:

$$C_Z = L L^T \quad (5.40)$$

assim, as matrizes jacobianas obtidas por meio da decomposição de Cholesky de C_Z são dadas por:

$$J_{yz} = L^{-1} \quad (5.41)$$

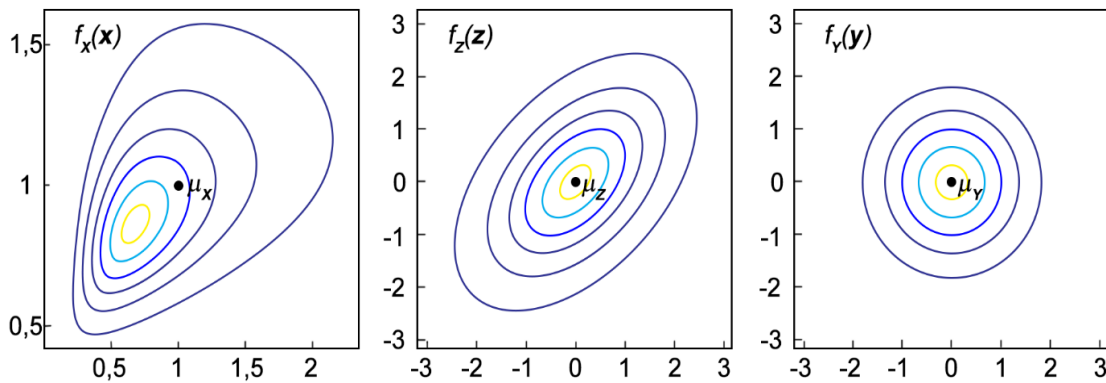
$$J_{zy} = L \quad (5.42)$$

e as transformações do espaço $\mathbb{Z} \rightarrow \mathbb{Y}$ e inversa são dadas por:

$$y = J_{yz} \cdot z \quad (5.43)$$

$$z = J_{zy} \cdot y. \quad (5.44)$$

A Figura 5.8 mostra as etapas da transformação composta considerando o caso em que as variáveis aleatórias X seguem a distribuição normal bivariada com correlação igual a 0,5.



5.8 – Transformação das variáveis X_1 e X_2 em variáveis normais padrão independentes

Fonte: Adaptado Beck (2019)

A implementação da transformação composta é dada através das matrizes jacobianas, obtidas por meio da aplicação da regra da cadeia, da seguinte forma:

$$J_{yx} = \left[\frac{\partial y_i}{\partial x_k} \right] = \left[\frac{\partial y_i}{\partial z_j} \frac{\partial z_j}{\partial x_k} \right] = J_{yz} \cdot J_{zx} \quad (5.45)$$

$$J_{xy} = \left[\frac{\partial x_i}{\partial y_k} \right] = \left[\frac{\partial x_i}{\partial z_j} \frac{\partial z_j}{\partial y_k} \right] = J_{xz} \cdot J_{zy} \quad (5.46)$$

sendo o fato de que $J_{xz} = D^{neq}$ e $J_{zx} = (D^{neq})^{-1}$. As matrizes jacobianas mostradas nas Eqs. (5.41) e (5.42) e utilizando a decomposição de Cholesky, tornam-se as equações:

$$J_{yx} = L^{-1} \cdot (D^{neq})^{-1} \quad (5.47)$$

$$J_{xy} = D^{neq} \cdot L. \quad (5.48)$$

As respectivas transformações $\mathbb{X} \rightarrow \mathbb{Y}$ e $\mathbb{Y} \rightarrow \mathbb{X}$ são dadas de forma matricial por:

$$y = J_{yx} \{x - V^{neq}\} \quad (5.49)$$

$$x = J_{xy} \cdot y + V^{neq}. \quad (5.50)$$

O algoritmo de solução do método FORM é apresentado a seguir (BECK, 2019).

Algoritmo 5.2. Método de confiabilidade FORM

1. determinação dos coeficientes de correlação equivalentes e da matriz de decomposição (ortogonal ou Cholesky); determinação das matrizes Jacobianas (J_{yx} e J_{xy});
2. escolha do ponto de falha inicial x^k para $k=0$ (usualmente o ponto médio);
3. atualização das matrizes Jacobianas J_{yx} e J_{xy} [Eqs. (5.47) e (5.48)];
4. transformação do ponto x^k de $\mathbb{X} \rightarrow \mathbb{Y}$ [Eqs. (5.49) ou (5.50)];
5. avaliação de $g(x^k)$;
6. cálculo do gradiente:
 - a. cálculo das derivadas parciais de $g(x)$ no espaço de projeto $\mathbb{X}$
 - b. transformação do gradiente para $\mathbb{Y}$ [Eq. (5.20)]

- c. cálculo dos cossenos diretores $\alpha(y^k)$
- 7. Cálculo do índice de confiabilidade (β)
- 8. cálculo do novo ponto de falha y^{k+1} [Eq. (5.21)];
- 9. transformação de y^{k+1} em $\mathbb{X}$ [Eqs. (5.49) ou (5.50)];
- 10. verificação do critério de convergência. Se:

$$1 - \frac{\nabla g(y^{k+1}) y^{k+1}}{\|\nabla g(y^{k+1})\| \|y^{k+1}\|} < \epsilon \quad \text{e} \quad |g(y^{k+1})| < \delta$$

sendo $\epsilon = 10^{-6}$; o algoritmo é interrompido ($y^* = y^{k+1}$); caso contrário, retorne ao item 3 com $k=k+1$ até atingir a convergência;

- 11. ao final, avaliação do índice de confiabilidade no ponto de projeto: $\beta = \|y^*\|$.

FORM HLRF

Para Beck (2019), a solução de problemas de confiabilidade via FORM, para funções de estado limites não lineares, envolve a solução de um problema de otimização para busca do ponto de projeto. O algoritmo mais utilizado é o Hasofer, Lind, Rackwitz e Fissler (HLRF). Este é muito popular, devido à sua simplicidade, ainda que apresente problemas de convergência em problemas fortemente não lineares. Para estados limites não lineares, o cálculo da distância mínima se torna um problema de otimização. A solução desse problema é apresentada por:

$$d = \|y\| = \sqrt{y^t y} \quad (5.51)$$

onde d é a distância entre y qualquer e a origem. Este é um típico problema de otimização sujeito a $g(y) = 0$. Neste caso, o índice de confiabilidade é a mínima distância, mostrado através de:

$$\beta \equiv \|y^*\| = \sqrt{y^{*t} y^*}. \quad (5.52)$$

Conforme Beck (2019), a solução do problema de otimização formulado na Eq. (5.51) pode ser feita através do método de Lagrange, aplicando-se um multiplicador de Lagrange λ . Assim, obtém-se um problema de otimização sem restrições, mostrado por:

$$\mathcal{L}(y, \lambda) = (y y^t)^{\frac{1}{2}} + \lambda g(y). \quad (5.53)$$

Um ponto estacionário da Eq. (5.53) é obtido nas Eqs. (5.54) e (5.55) derivando $\mathcal{L}(y, \lambda)$ em relação a y e a λ .

$$\frac{\partial \mathcal{L}}{\partial y} = y d^{-1} + \lambda \nabla g(y) = 0 \quad (5.54)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = g(y) = 0 \quad (5.55)$$

Ainda segundo Beck (2019), a Eq. (5.55) é satisfeita automaticamente se y está sobre a função estado limite. A Eq. (5.54) leva as coordenadas do ponto estacionário y^{est} da equação:

$$y^{est} = -\lambda d \nabla g(y) \quad (5.56)$$

e, assumindo que este ponto seja um ponto de mínimo, isto é, um ponto de projeto (y^*), usando $d = \sqrt{y y^t}$, tem-se:

$$d_{min} = -\frac{\nabla g(y)}{\|\nabla g(y)\|} \cdot y^* = -\alpha^t \cdot y^* \quad (5.57)$$

Conforme Hasofer e Lind (1974) e Rackwitz e Fiessler (1978), o algoritmo HLRF foi desenvolvido particularmente para solução do problema de otimização em confiabilidade estrutural. Seu algoritmo tem fórmula recursiva e está baseado na aproximação da função de estado limite por um hiperplano. Uma expansão da função de estado limite em série de Taylor de primeira ordem, em torno de um ponto inicial y^k , de forma que, o próximo iterando y^{k+1} é o ponto sobre a função de estado limite linearizada, dada por $g(y^k) + \nabla g(y^k)^t (y - y^k) = 0$.

A Figura 5.9 ilustra uma iteração do método FORM com o algoritmo HLRF, considerando uma função estado limite linear (a) e não linear (b). Observa-se que enquanto o ponto de projeto não é alcançado, a condição $g(y) = 0$ não é obtida. Ou seja, o ponto y^k não atinge a superfície de falha, mas sim uma superfície para a qual $g(y)$ é

constante (c). Portanto, o algoritmo de otimização deve começar do ponto y^k , que pode não estar no estado limite, e convergir para o ponto de distância mínimo y^* no estado limite.

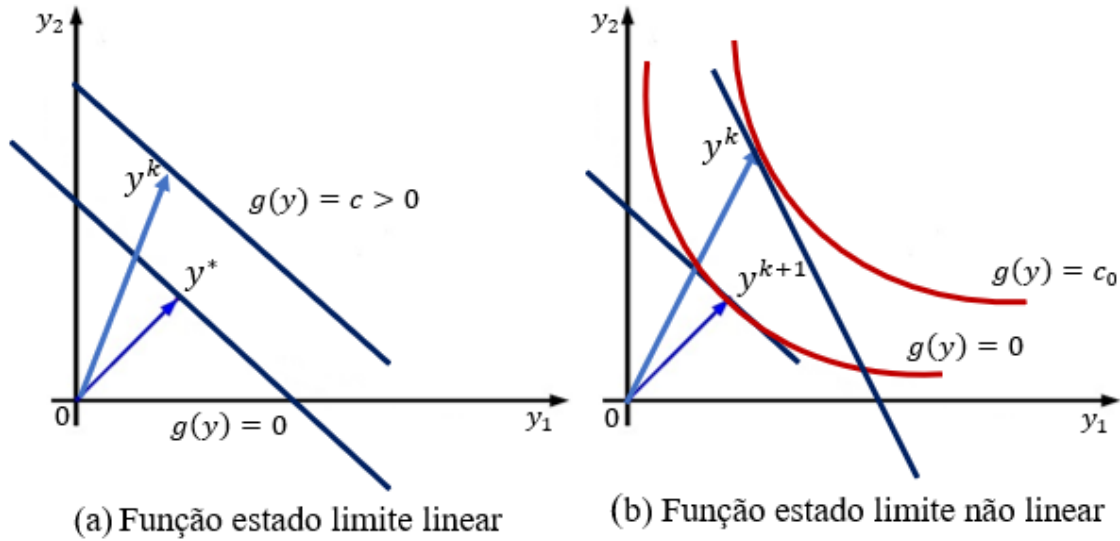


Figura 5.9 – Representação do método FORM com o algoritmo HLRF.

Fonte: Adaptado de Haldar e Mahadevan (2000).

A fórmula da recorrência para y^{k+1} pode ser obtida considerando a aproximação linear de $g(y)$, e é representado por:

$$\tilde{g}(y) = g(y^k) + \nabla g(y^k)^t (y - y^k) \quad (5.58)$$

substituindo a função de estado limite $g(y) = 0$ por sua linearização, $\tilde{g}(y)$, tem-se:

$$g(y^k) + \nabla g(y^k)^t (y - y^k) = 0. \quad (5.59)$$

Santos (2012) explica que o ponto sobre a função de estado limite linearizada mais próximo da origem é obtido pela interseção da Eq. (5.59) com a reta perpendicular a ela, dada por:

$$y = T \nabla g(y^k), \text{ com } T \in \mathbb{R} \quad (5.60)$$

calculando a interseção entre as Eqs. (5.59) e (5.60), tem-se:

$$T = \frac{\nabla g(y^k)^t y^k - g(y^k)}{\|\nabla g(y^k)\|^2}. \quad (5.61)$$

substituindo a Eq. (5.61) na Eq. (5.60), obtém-se a fórmula recursiva do algoritmo HLRF, através de:

$$y^{k+1} = \frac{[\nabla g(y^k)^t y^k - g(y^k)] \nabla g(y^k)}{\|\nabla g(y^k)\|^2}. \quad (5.62)$$

O algoritmo para o método FORM utilizando HLRF não será apresentado, pois o mesmo pode ser obtido simplesmente acrescentado sua fórmula recursiva, mostrada na Eq. (5.62), no passo 7 do algoritmo 5.2.

FORM iHLRF

De acordo Sudret e Der Kiureghian (2000), o algoritmo HLRF pode ser melhorado (iHLRF) através de uma busca linear para o ajuste. A direção de busca é dada pelo algoritmo HLRF original na equação:

$$d^k = y^{k+1} - y^k = \frac{\nabla g(y^k)^t y^k - g(y^k)}{\|\nabla g(y^k)\|^2} \nabla g(y^k) - y^k \quad (5.63)$$

considerando que a fórmula recursiva da Eq. (5.62) do algoritmo original HLRF é obtida quando o novo ponto é determinado através de um passo unitário $\alpha^k = 1$, tal que, $y^{k+1} = y^k + \alpha^k d^k$.

Conforme Zhang e Der Kiureghian (1995), a função de mérito não-diferenciável da Eq. (5.64) é usada para determinar α^k na busca linear:

$$p(y) = \frac{1}{2} y^t y + c |g(y)| \quad (5.64)$$

sendo que o passo adequado α^k é o valor que satisfaz uma condição de decréscimo suficiente na função de mérito, dada por:

$$p(y^{k+1}) \leq p(y^k) + m_1 \alpha^k \nabla p(y^k)^t d^k. \quad (5.65)$$

Os autores mostram que a direção d^k , mostrada na Eq. (5.63), é uma direção de descida para $p(y)$, dada na Eq. (5.64), em $y^k \in \mathbb{R}^n$ desde que a seguinte condição seja considerada, apresentada em:

$$c > \frac{\|y^k\|}{\|\nabla g(y^k)\|} \quad (5.66)$$

para que c seja atualizado, estabeleceram as seguintes regras:

Se $|g(y^k)| \geq \Delta$

$$c = \eta \max \left\{ \frac{\|y^k\|}{\|\nabla g(y^k)\|}, \frac{1}{2} \frac{\|y^k + d^k\|^2}{|g(y^k)|} \right\}$$

Senão,

$$c = \eta \frac{\|y^k\|}{\|\nabla g(y^k)\|}$$

O algoritmo de solução do método FORM iHLRF é apresentado a seguir (BECK, 2019).

Algoritmo 5.3. Método de confiabilidade FORM iHLRF

Dados: $y^k \in \mathbb{R}^n, \Delta > 0, \eta > 1, \alpha \in (0,1], 0 < m_1 < 1$

1. Calcular a direção de busca de d^k dada na Eq. (5.63);
2. Determinar o parâmetro de penalidade c da função de mérito da Eq. (5.64)

Se $|g(y^k)| \geq \Delta$

$$c = \eta \max \left\{ \frac{\|y^k\|}{\|\nabla g(y^k)\|}, \frac{1}{2} \frac{\|y^k + d^k\|^2}{|g(y^k)|} \right\}$$

Senão

$$c = \eta \frac{\|y^k\|}{\|\nabla g(y^k)\|};$$

3. Determinar α^k usando a desigualdade através da Eq. (5.65)

Enquanto $p(y^k + \alpha d^k) - p(y^k) > m_1 \alpha \nabla p(y^k)^t d^k$

$$\text{Fazer } \alpha = \frac{\alpha}{2}$$

$$\alpha^k = \alpha;$$

4. Atualizar o ponto y :

$$y^{k+1} = y^k + \alpha d^k.$$

5.2.3. SORM

O SORM aproxima a função estado limite no ponto de projeto por uma hipersuperfície quadrática. Beck (2019) afirma que esse método resulta em uma boa aproximação da probabilidade de falha quando a função estado limite, no espaço normal padrão, é linear ou fracamente não linear na vizinhança do ponto de projeto.

Para Beck (2019), a não linearidade da função de estado limite no espaço normal padrão surge quando a transformação para o espaço normal padrão também é não linear. Isto acontece quando existe correlação (forte dependência linear) entre as variáveis aleatórias ou quando as distribuições de probabilidade são fortemente não normais. O ajuste de uma curvatura à função estado limite é realizado através da determinação de um sistema de eixos ortogonais. Para um problema com duas variáveis aleatórias, a transformação é mostrada na Fig. 5.10.

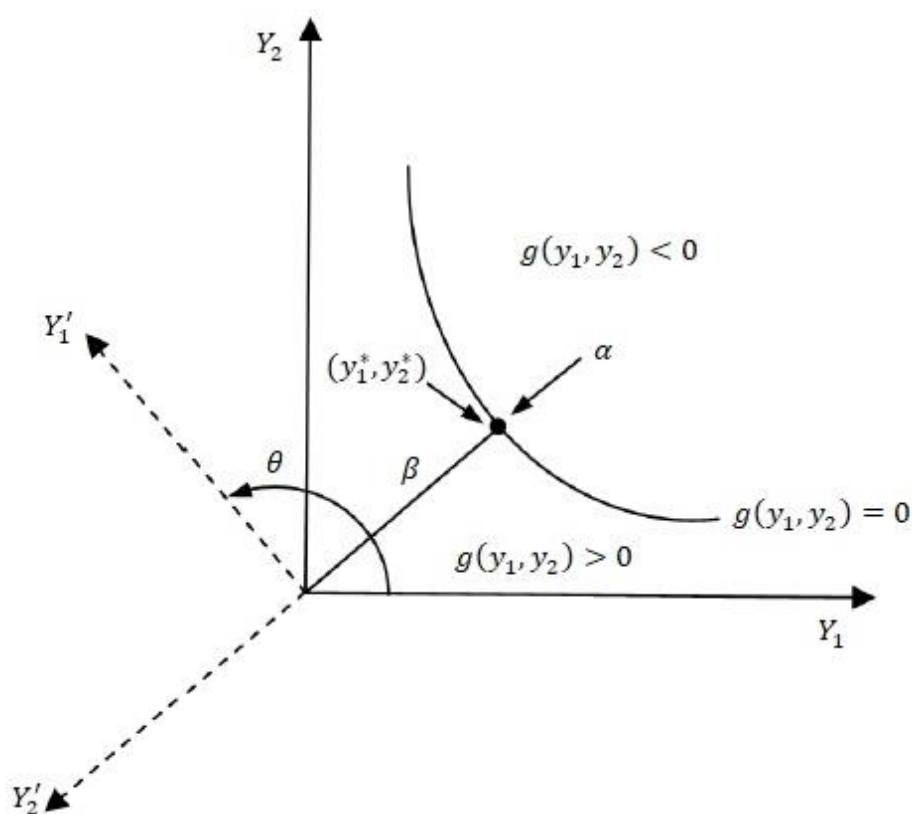


Figura 5.10 – Rotação das coordenadas

Fonte: Haldar e Mahadevan (2000)

Conforme Haldar e Mahadevan (2000), para cálculo dessas curvaturas, as variáveis aleatórias Y_i no espaço $\mathbb{Y}$ são rotacionadas em outro conjunto de variáveis denotadas como Y_i^* , tal que as últimas variáveis desta coincidam com o vetor α , sendo

este o vetor gradiente unitário do estado limite no ponto de distância mínima. Nesse caso, trata-se apenas de uma transformação de coordenadas.

Para Haldar e Mahadevan (2000), a transformação do espaço Y para o espaço Y^* é uma transformação ortogonal, dado em:

$$Y^* = HY \quad (5.67)$$

onde H é a matriz de rotação. Para um caso simples onde só existem duas variáveis aleatórias a matriz H pode ser escrita através de:

$$H = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (5.68)$$

onde θ é o ângulo de rotação. Quando o número de variáveis é maior de dois a obtenção da matriz H é dada em duas etapas. Primeiro constrói-se uma matriz H_0 da seguinte forma:

$$H_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{y_1} & \alpha_{y_2} & \cdots & \alpha_{y_n} \end{bmatrix} \quad (5.69)$$

onde $\alpha_{y_1}, \alpha_{y_2}, \dots, \alpha_{y_n}$ são os cossenos diretores, isto é, componentes do vetor unitário α como mostrado na Fig. 5.10. Na segunda etapa realiza-se uma ortogonalização de Gram-Schmidt na matriz H_0 que resultará na matriz H , onde suas linhas são determinadas através de:

$$\begin{cases} h_n = h_{0n} \\ h_k = h_{0k} - \sum_{j=k+1}^n \frac{h_j \cdot h_{0k}^T}{h_j \cdot h_j^T} \cdot h_j \end{cases} \quad (5.70)$$

uma vez que a matriz H é obtida, os elementos da matriz A podem ser calculados por:

$$A_{ij} = \frac{(HBH^t)}{|\nabla g(y^*)|} \quad i, j = 1, 2, \dots, n-1 \quad (5.71)$$

onde B é a matriz da derivada segunda $n \times n$ (matriz hessiana) da superfície de estado limite no espaço normal padrão avaliado no ponto de projeto, como mostra a Eq. (5.72). H é a matriz de rotação e $|\nabla g(y^*)|$ é o comprimento do vetor gradiente no espaço normal padrão.

$$B(y^*) = \left[\frac{\partial^2 g(y)}{\partial y_i \partial y_j} \right]_{i,j=1,2,\dots,n} \quad (5.72)$$

Ainda segundo Haldar e Mahadevan (2000), no espaço rotacionado, a última variável, Y_n , coincide com o vetor β calculado pelo método FORM. No próximo passo, a última linha e coluna da matriz A e a última linha do vetor Y' são descartadas para levar em conta esse fator. Assim, o estado limite pode ser reescrito em termos da aproximação de segunda ordem neste espaço padrão rotacionado, através de:

$$y'_n = \beta + \frac{1}{2} y'^t A y' \quad (5.73)$$

onde a matriz A tem agora dimensão $(n-1) \times (n-1)$.

Segundo Breitung (1984), a determinação da probabilidade de falha usando uma aproximação de segunda ordem é dada através de:

$$P_f = \Phi(-\beta) \prod_{j=1}^{n-1} (1 + \beta k_j)^{-\frac{1}{2}} \quad (5.74)$$

onde k_j são as curvaturas principais da superfície de falha no ponto y^* e n o número de variáveis aleatórias na análise.

A forma matricial da Eq. (5.74) é dada através de:

$$P_f = \Phi(-\beta) [\det(I + 2\beta A)]^{-\frac{1}{2}} \quad (5.75)$$

onde I é a matriz identidade com dimensão $n-1$.

De acordo com Breitung (1984), o algoritmo de método solução do método SORM é apresentado na Fig. 5.11.

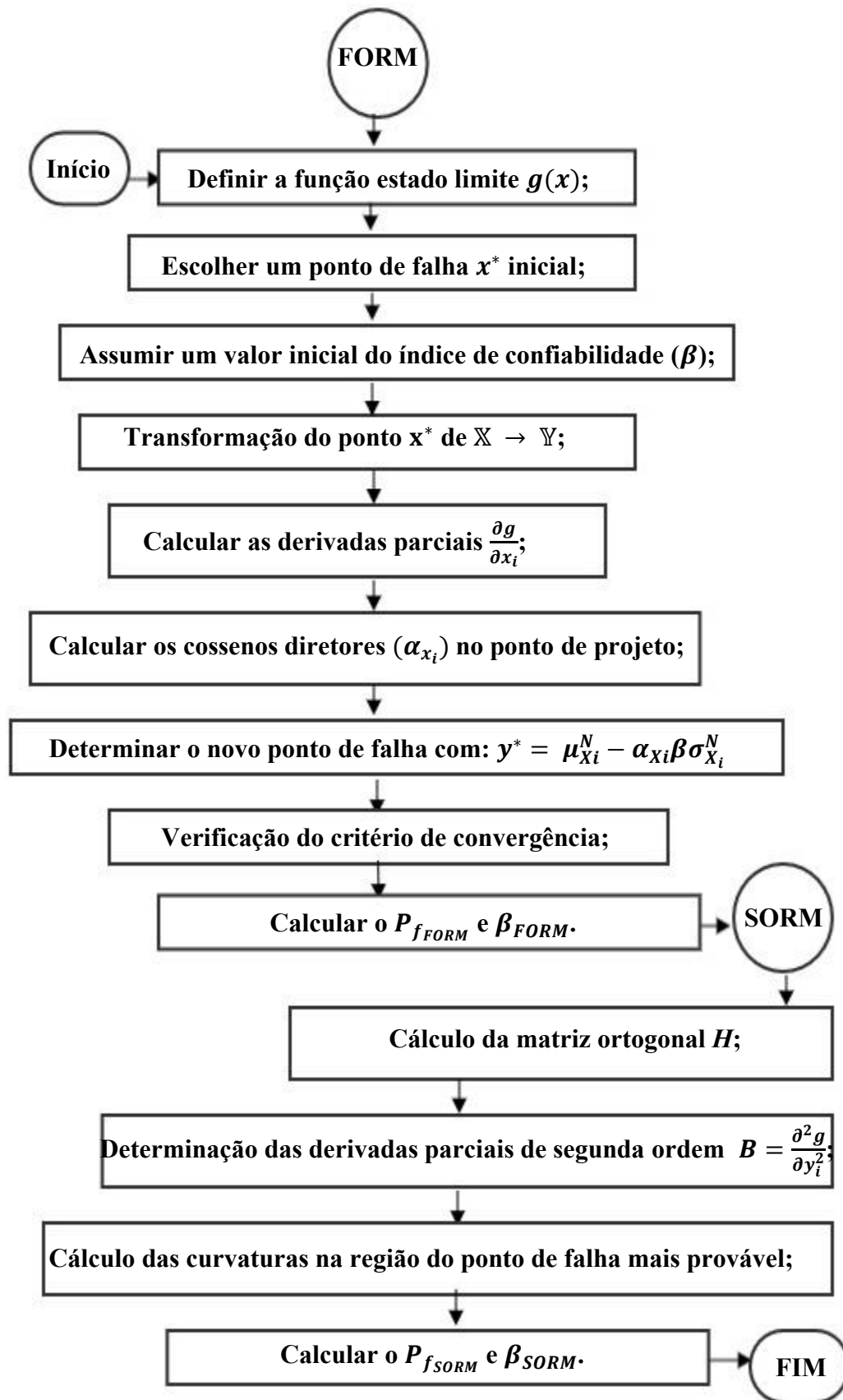


Figura 5.11 – Algoritmo 5.4. Fluxograma do método SORM.

Fonte: Autora

Conforme Haldar e Mahadevan (2000), a curvatura de qualquer função está relacionada às derivadas de segunda ordem em relação às variáveis aleatórias. Logo, o método de confiabilidade de segunda ordem (SORM) melhora o resultado do FORM incluindo informações adicionais sobre a curvatura do estado limite.

6. RESULTADOS

Neste capítulo são apresentados dezenove exemplos, sendo cinco para verificação dos métodos estudados e catorze para análise comparativa, com a aplicação da simulação de Monte Carlo e os métodos de transformação em problemas de confiabilidade estrutural. O estudo consiste na análise comparativa da convergência do índice de confiabilidade (β) e probabilidade de falha (P_f) de funções estado limite lineares, não lineares e implícitas encontradas na literatura.

Para efeitos de comparação da robustez e da eficiência dos seis métodos propostos, foram utilizados códigos programados na linguagem *Python*. Todos os exemplos apresentados foram desenvolvidos e simulados em um computador Intel® Core™ i5-265U 1,80 GHz 64 bits com 8 GB de memória RAM.

Os métodos de transformação são interrompidos quando encontram um ponto cujo valor da medida de estacionaridade é suficientemente pequeno. Dessa forma, são estabelecidas as seguintes condições:

$$1 - \frac{\nabla g(y^{k+1}) y^{k+1}}{\|\nabla g(y^{k+1})\| \|y^{k+1}\|} < \epsilon \quad \text{e} \quad |g(y^{k+1})| < \delta$$

sendo $\epsilon = 10^{-6}$. Se essas condições não forem atendidas até que o código atinja 100 iterações, a execução do método é interrompida.

Na implementação da simulação de Monte Carlo, o MCFB funciona bem para um número de simulação até 9×10^7 e o MCAI utiliza como amostragem um ponto pré determinado, sendo a utilização do ponto de projeto obtido através do método FORM iHLRF.

6.1. Verificação dos métodos

Nesta seção são analisadas funções estado limite, a fim de realizar a verificação dos métodos programados na linguagem *Python*. O índice de confiabilidade e probabilidade de falha obtidos nos cinco problemas a seguir são comparados a um valor alvo (β e P_f), obtidos através da literatura de referência.

6.1.1. Problema 1: Função estado limite não linear com variáveis Normais

Beck (2019) apresenta um exemplo que representa a formação de uma rótula plástica em uma viga de aço, devido à ação do momento fletor externo. A função estado limite desse exemplo é não linear e determinada por:

$$g(x) = x_1 x_2 - x_3 \quad (6.1)$$

onde x_1 é a tensão de escoamento do aço, x_2 é o módulo plástico da seção transversal da viga, e x_3 é o momento externo aplicado. Os parâmetros de projeto dessa rótula plástica são apresentados na Tab. 6.1.

Tabela 6.1 – Médias e desvios padrão das variáveis aleatórias do Problema 1.

Variável Aleatória	Unidade	Distribuição	Média	Desvio padrão
x_1	kN/cm ²	Normal	40	5,0
x_2	cm ³	Normal	50	2,5
x_3	kN.cm	Normal	1000	200,0

Fonte: Beck (2019)

A Figura 6.1 a seguir ilustra o histograma da função estado limite com distribuição de probabilidade Normal, para o problema 1.

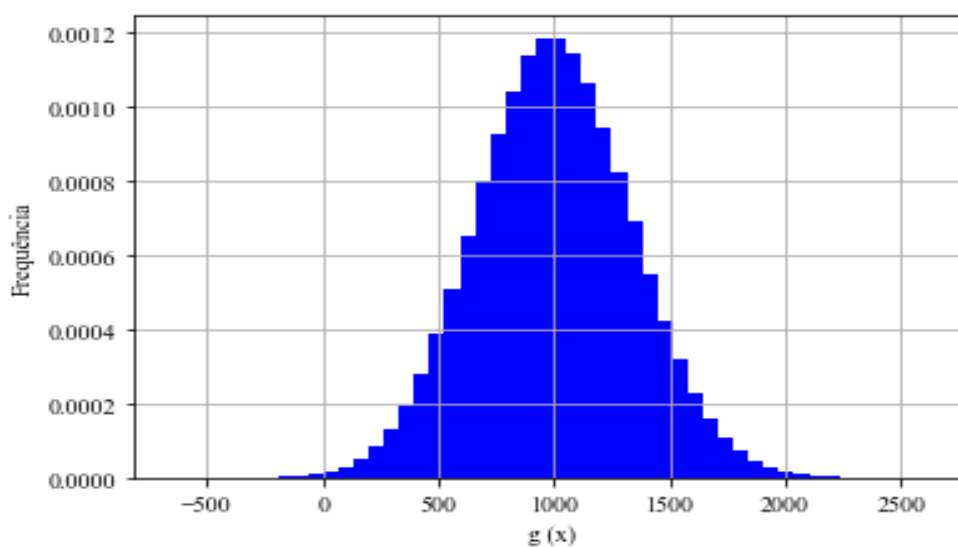


Figura 6.1 – Histograma da função estado limite do problema 1.

Fonte: Autora

Utilizando o método de simulação de Monte Carlo e os métodos de transformação estudados, realiza-se um comparativo entre os resultados dos métodos e o apresentado por Beck (2019) através do método FOSM, como mostrado na Tab. 6.2.

Tabela 6.2 – Comparativo dos resultados para o Problema 1.

MÉTODO	β	P_f
Beck (2019)	3,04	$1,17 \times 10^{-3}$
MCFB	3,04	$1,17 \times 10^{-3}$
MCAI	3,04	$1,17 \times 10^{-3}$
FOSM	3,04	$1,17 \times 10^{-3}$
FORM HLRF	3,04	$1,17 \times 10^{-3}$
FORM iHLRF	3,04	$1,17 \times 10^{-3}$
SORM	3,04	$1,17 \times 10^{-3}$

Fonte: Autora

Pode-se notar que os resultados dos métodos em análise da Tab. 6.2 tendem a convergir para o mesmo ponto, que é representado pelo índice de confiabilidade e probabilidade de falha. A partir dos resultados obtidos, pode-se afirmar que os métodos de simulação de Monte Carlo e transformação representaram adequadamente os valores encontrados pelo autor neste exemplo, já que os resultados foram iguais.

6.1.2. Problema 2: Função estado limite não linear com variáveis mistas

Santos *et al.* (2014) apresentam uma investigação da segurança de elementos estruturais submetidos à flexão simples, em concreto armado, aço e material misto, neste trabalho só foi analisada a viga de concreto armado. O estudo permite uma comparação da segurança relativa de vigas produzidas com estes materiais. Os resultados da investigação mostram que os índices de confiabilidade dos elementos de concreto armado são satisfatórios.

Neste problema é realizada uma análise de confiabilidade que consiste em determinar o índice de confiabilidade de uma viga de concreto armado à flexão.

O máximo momento fletor que esta viga poderá suportar é dado por:

$$M_R = \theta_R \cdot A_s \cdot f_y \left[h - d' - 0,5 \cdot \frac{A_s \cdot f_y}{0,85 \cdot f_c \cdot b} \right] \quad (6.2)$$

onde θ_R é o erro de modelo de resistência, A_s é a área de aço da armadura tracionada pelo momento fletor, f_y é a tensão de escoamento do aço empregado, d' é a distância da face inferior do concreto em relação ao eixo das barras tracionadas de aço, h é a altura da seção transversal, f_c é a resistência à compressão do concreto e b é a largura da seção transversal.

O momento fletor solicitante máximo é dado por:

$$M_S = \theta_S (M_g + M_q) \quad (6.3)$$

onde θ_S é o erro de modelo das ações, M_g é o momento da carga permanente e M_q é o momento da carga acidental.

Baseado nas Eqs. (6.2) e (6.3), tem-se a condição de resistência à flexão da seção central, que deve ser atendida pela viga, dada por:

$$g(x) = M_R - M_S = \theta_R \cdot A_s \cdot f_y \left[h - d' - 0,5 \cdot \frac{A_s \cdot f_y}{0,85 \cdot f_c \cdot b} \right] - \theta_S (M_g + M_q) \quad (6.4)$$

Para transformar o esforço solicitante em valores característicos de cada ação é necessário, primeiramente, definir a relação (χ) entre as cargas permanentes e variáveis que se desejar avaliar. Isso pode ser feito através da seguinte equação:

$$\chi = \frac{q_k}{g_k + q_k} \quad (6.5)$$

Na Tabela 6.3 é apresentada a seção transversal da viga de concreto armado adotada, que possui dimensões fixas. Esse elemento estrutural é comum em edifícios que costumam ter distância entre pilares de cinco a seis metros. Algumas taxas de armadura simples e cobrimentos são consideradas nesse estudo. Todas as vigas foram avaliadas com concreto C25 ($f_{ck} = 25$ MPa), e aço CA-50 ($f_{yk} = 500$ MPa). E ainda, é submetida a um carregamento distribuído característico, composto por uma parcela permanente e acidental. Os coeficientes de ponderação das ações permanentes e variáveis são $\gamma_g = 1,4$.

Tabela 6.3 – Viga de concreto armado.

Armadura	A_s (cm ²)	b (cm)	h (cm)	ρ (%)	d' (cm)	M_d (kN.m)
3 Φ 8 mm	1,50	20	50	0,15	3,9	29,36
4 Φ 10 mm	3,20	20	50	0,32	4,0	60,81
4 Φ 12,5 mm	5,00	20	50	0,50	4,1	92,00
4 Φ 16 mm	8,00	20	50	0,80	4,3	139,03
3 Φ 20 mm	9,45	20	50	0,95	4,5	159,14

Fonte: Santos *et al.* (2014)

As variáveis aleatórias adotadas assim como as suas respectivas distribuições e parâmetros relacionados estão apresentadas na Tab. 6.4.

Tabela 6.4 – Modelo probabilístico das variáveis aleatórias.

Categoria	Nome	Símbolo	Dimensão	Distribuição	μ_x	σ_x
Ações	Permanente	g	kN/m	Normal	g_k	$0,1 \mu_x$
	Acidental	q	kN/m	Gumbel	$0,93 q_k$	$0,2 \mu_x$
Resistência dos materiais	Concreto	f_c	MPa	Normal	$1,17 f_{ck}$	$0,15 \mu_x$
	Aço	f_y	MPa	Normal	$1,08 f_{yk}$	$0,05 \mu_x$
Dados geométricos	Altura	h	cm	Normal	50,00	2,25
	Largura	b	cm	Normal	20,00	1,20
	Cobrimento	d'	cm	Lognormal	d'	1,10
Incertezas do modelo	Resistência	θ_R	–	Lognormal	1,00	0,05
	Solicitação	θ_S	–	Lognormal	1,00	0,05

Fonte: Santos *et al.* (2014)

Santos *et al.* (2014) apresentam os resultados obtidos na análise da viga de concreto armado através do método FORM, como mostra a Fig. 6.2. Observa-se que o índice de confiabilidade depende da taxa de armadura da seção. A confiabilidade da viga de concreto armado aumenta com o aumento da taxa de armadura, uma vez que a variabilidade da resistência do aço é menor do que a variabilidade da resistência do concreto.

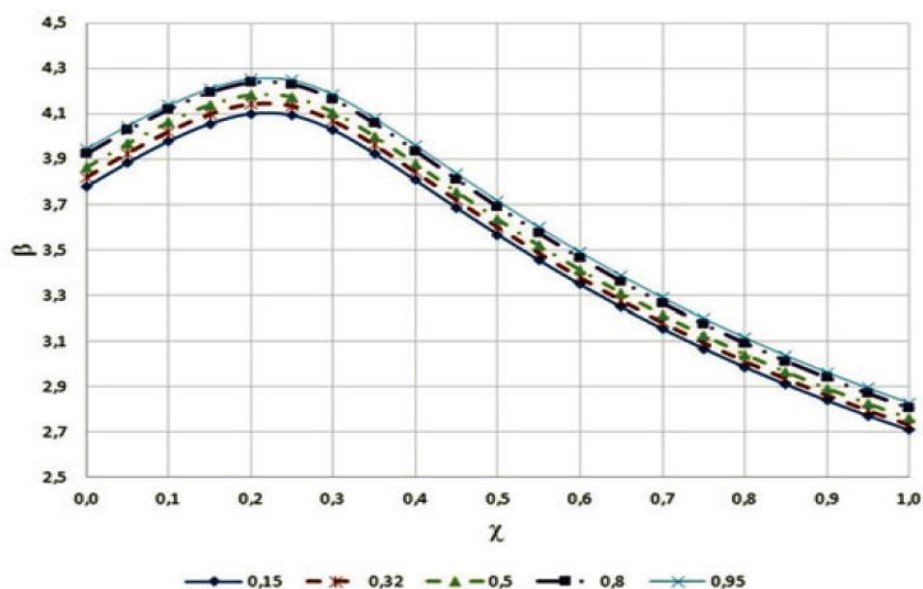


Figura 6.2 – Resultados dos índices de confiabilidade da viga de concreto armado.

Fonte: Santos *et al.* (2014)

Para efeitos de comparação, os métodos de confiabilidade estrutural estudados e os resultados obtidos na Fig. 6.2 do índice de confiabilidade para as taxas de armadura do aço ($\rho = 0,15\%$ e $0,95\%$) são sobrepostos e apresentados nas Figs. 6.3 e 6.4.

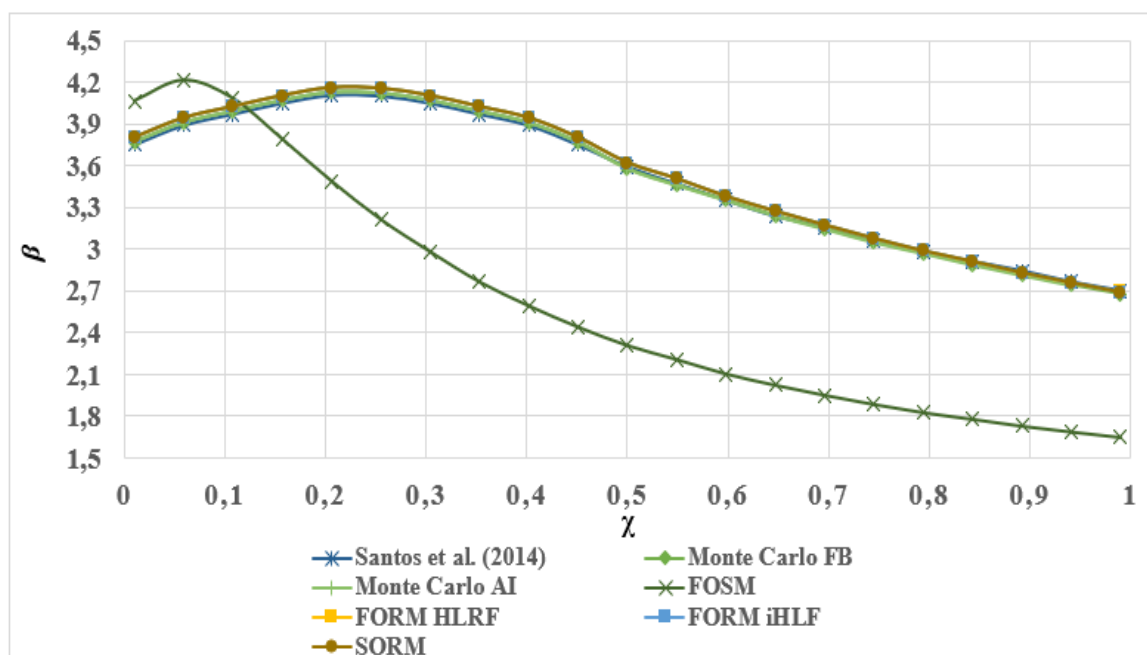


Figura 6.3 – Comparativo do índice de confiabilidade para a taxa de armadura do aço ($\rho = 0,15\%$).

Fonte: Autora

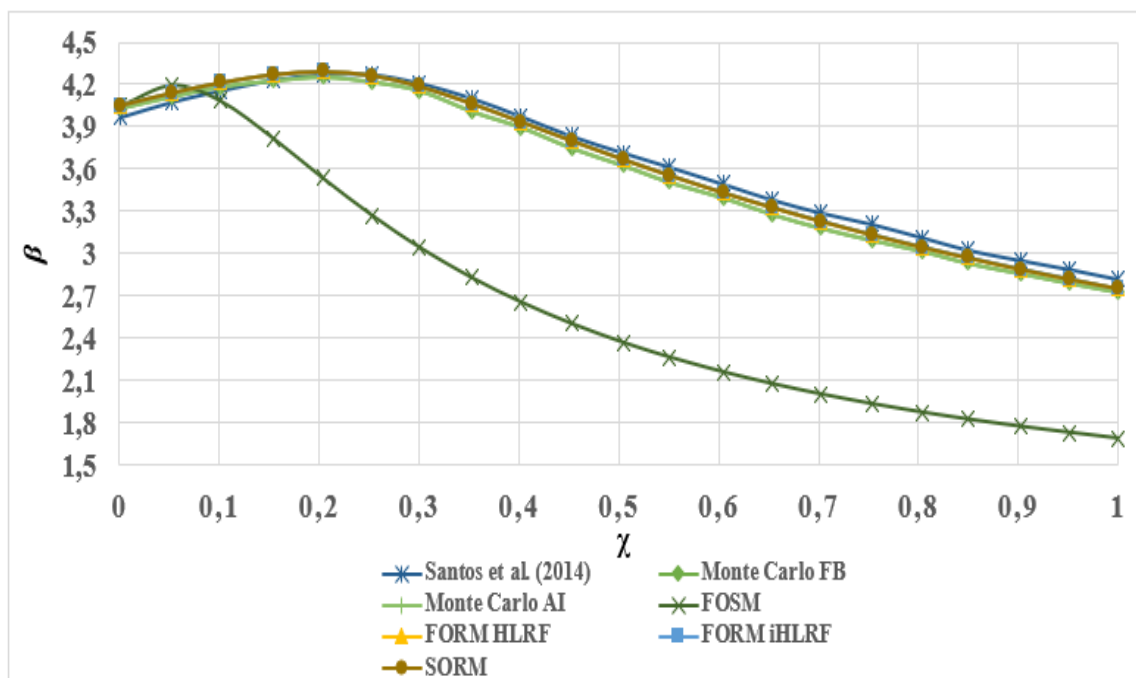


Figura 6.4 – Comparativo do índice de confiabilidade para a taxa de armadura do aço ($\rho = 0,95 \%$).

Fonte: Autora

A partir dos resultados obtidos nas Figs. 6.3 e 6.4, pode-se afirmar que os métodos de simulação de Monte Carlo e transformação representaram adequadamente os valores encontrados pelos autores, já que os resultados foram bastante próximos. É perceptível também que o método FOSM em relação aos demais métodos mostra resultado bem diferente e distante dos valores considerados, isso porque esse método só funciona para função estado limite linear e variáveis aleatórias com distribuição de probabilidade Normal.

6.1.3. Problema 3: Função estado limite altamente não linear com variáveis Normais

Shayanfar *et al.* (2017) apresentam a função estado limite altamente não linear deste exemplo, que é representada por um polinômio cúbico com termo misto e, é expressa da seguinte forma:

$$g(x) = x_1^3 + x_1^2 x_2 + x_2^3 - 18 \quad (6.6)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.5.

Tabela 6.5 – Médias e desvios padrão das variáveis aleatórias do Problema 3.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Normal	10,0	5,0
x_2	Normal	9,9	5,0

Fonte: Shayanfar *et al.* (2017)

Alguns autores fazem análise desta função utilizando o método FORM, como Gong *et al.* (2014) que encontram um índice de confiabilidade 2,2983, Keshtegar e Miri (2013) e Keshtegar (2016) obtiveram 2,2982 como índice de confiabilidade, Yang (2010) calculou o índice de 2,298, e Shayanfar *et al.* (2017) conseguiram 2,5274 como índice de confiabilidade para essa função, utilizando o MCFB com uma amostra de 10^6 elementos. E ainda, determinaram através do método FORM HLRF um índice de confiabilidade de 1,5370. Na Tab. 6.6 são apresentados os resultados obtidos por cada método estudado neste trabalho.

Tabela 6.6 – Comparativo dos resultados para o Problema 3.

MÉTODO	β	P_f
Shayanfar <i>et al.</i> (2017)	2,52	$5,84 \times 10^{-3}$
MCFB	2,52	$5,84 \times 10^{-3}$
MCAI	2,52	$5,84 \times 10^{-3}$
FOSM	1,54	$6,21 \times 10^{-2}$
FORM HLRF	1,54	$6,21 \times 10^{-2}$
FORM iHLRF	2,30	$1,08 \times 10^{-2}$
SORM	2,62	$4,32 \times 10^{-3}$

Fonte: Autora

A partir dos resultados obtidos, pode-se afirmar que os métodos estudados e programados para esse trabalho representaram adequadamente os resultados encontrados pelos autores. O FOSM e FORM HLRF não convergem para soluções estáveis para

funções de estado limite altamente não lineares, com uma diferença para a referência de 39%. Neste caso, o FORM iHLRF melhora o resultado diminuindo a diferença para a referência para aproximadamente 9% do índice de confiabilidade e probabilidade de falha, mas, ainda não consegue convergir para os resultados dos métodos MCFB, MCAI e SORM.

6.1.4. Problema 4: Função estado limite altamente não linear com variáveis mistas

Keshtegar e Chakraborty (2018) analisam este problema, que consiste em determinar o índice de confiabilidade e a probabilidade de falha, na resolução de uma função de estado limite altamente não linear, representada por um polinômio do quarto grau. O estado limite é definido por:

$$g(x) = x_1^4 + x_2^2 - 50. \quad (6.7)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.7.

Tabela 6.7 – Médias e desvios padrão das variáveis aleatórias do Problema 4.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Lognormal	5,0	1,0
x_2	Gumbel	10,0	10,0

Fonte: Keshtegar e Chakraborty (2018)

Keshtegar e Chakraborty (2018) determinaram 3,2593 como índice de confiabilidade baseado no método FORM iHLRF para essa função. Os resultados obtidos através da aplicação dos métodos estudados estão apresentados na Tab. 6.8.

Tabela 6.8 – Comparativo dos resultados para o Problema 4.

MÉTODO	β	P_f
Keshtegar e Chakraborty (2018)	3,2593	$1,51 \times 10^{-4}$
MCFB	3,56	$1,89 \times 10^{-4}$
MCAI	3,56	$1,89 \times 10^{-4}$
FOSM	1,51	$6,49 \times 10^{-2}$
FORM HLRF	1,84	$3,31 \times 10^{-2}$
FORM iHLRF	3,26	$1,53 \times 10^{-4}$
SORM	3,53	$1,93 \times 10^{-4}$

Fonte: Autora

A partir da Tab. 6.8, e assim, como nos resultados obtidos no problema 3, observa-se que os métodos FOSM e FORM HLRF distanciam-se do MCFB com 58% e 48%, respectivamente, mostrando com este resultado que realmente não funcionam bem para funções de estado limite altamente não lineares. O Método FORM iHLRF mostra o mesmo resultado dos autores.

6.1.5. Problema 5: Função de estado limite implícita

Li e Zhang (2013) apresentam uma análise da treliça plana com as dimensões e cargas mostradas na Fig. 6.5. O material das barras apresenta um comportamento elástico linear, com módulo de elasticidade $E = 200$ GPa. As seções transversais das barras da treliça possuem área de 25 cm^2 . A estrutura está sujeita a cargas $P = 200$ kN, aplicadas verticalmente nos nós 2 e 4.

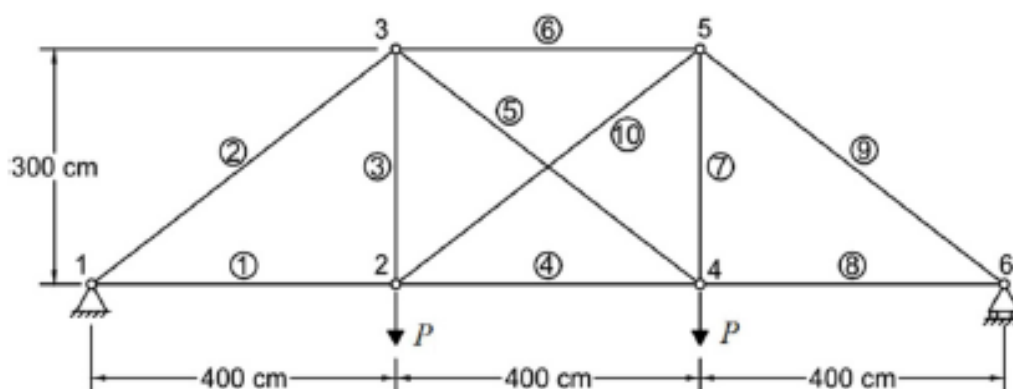


Figura 6.5 – Esquema estrutural da treliça plana

Fonte: Li e Zhang, 2013

As variáveis aleatórias da estrutura são a carga externa P e área da seção transversal A . Os parâmetros estatísticos (média e desvio padrão) de cada variável aleatória são apresentados na Tab. 6.9.

Tabela 6.9 – Médias e desvios padrão das variáveis aleatórias do Problema 5.

Variável Aleatória	Unidade	Distribuição	Média	Desvio padrão
A	m^2	Lognormal	0,0025	0,00025
P	kN	Gumbel	200,0	50,0

Fonte: Li e Zhang, 2013

O critério de falha definido para o exemplo é caracterizado pelo deslocamento máximo no nó 4. Desta forma, a função de estado limite implícita é determinada por:

$$g(A, P) = 3,0 - u_4(A, P) \quad (6.19)$$

onde $3,0 \text{ cm}$ é o deslocamento máximo permitido e $u_4(A, P)$ é o deslocamento na direção vertical no nó 4.

A Figura 6.6 a seguir ilustra o histograma da função estado limite com distribuição de probabilidade mistas.

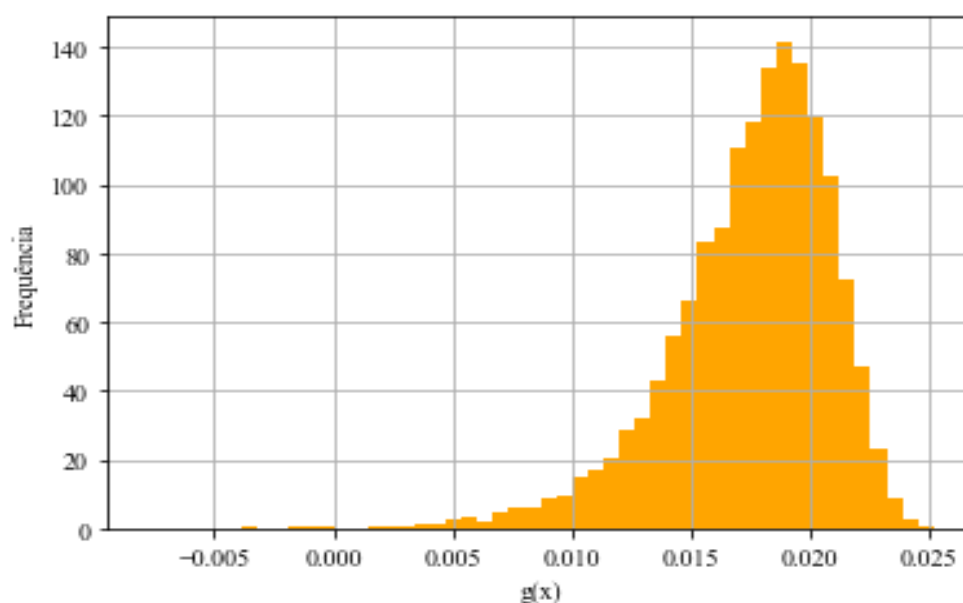


Figura 6.6 – Histograma da função estado limite da treliça plana

Fonte: Autora

Li e Zhang (2013) apresentam resultados para este exemplo através de diferentes métodos. Obtiveram 3,22 como índice de confiabilidade para essa função, utilizando o MC FB com uma amostra de 10^5 elementos. Os autores determinaram também, através do Método da Superfície de Resposta (RSM), um índice de confiabilidade de 3,19. Na Tab. 6.10 são apresentados os resultados obtidos por cada método, utilizando o programa *Trusspy* (DUTZLER, 2018).

Tabela 6.10 – Comparativo dos resultados para o Problema 5.

MÉTODO	β	P_f
Li e Zhang (2013)	3,22	$6,45 \times 10^{-4}$
MCFB	3,20	$6,86 \times 10^{-4}$
MCAI	3,20	$6,86 \times 10^{-4}$
FOSM	4,12	$1,89 \times 10^{-5}$
FORM HLRF	3,19	$6,98 \times 10^{-4}$
FORM iHLRF	3,19	$6,98 \times 10^{-4}$
SORM	3,19	$6,98 \times 10^{-4}$

Fonte: Autora

Utilizando o método MCFB com uma amostra de 10^6 elementos, e um tempo computacional de 16 min 35 s e MCAI com 10^4 elementos com um tempo de 9 s, e os métodos FOSM, FORM HLRF, FORM iHLRF e SORM com um tempo menor que 1 s foram determinados os índices de confiabilidade e as probabilidades da Tab. 6.10. O resultado do método FOSM é diferente dos demais métodos, pois, não leva em consideração as distribuições mistas. O MCAI mostrou um desempenho maior para esse problema, tanto em relação ao tempo computacional, quanto ao número de simulações. Os métodos de simulação de Monte Carlo, FORM e SORM obtiveram resultados bem próximos à referência, com uma diferença de menos de 1%. Dessa forma, pode-se afirmar que os métodos MCFB, MCAI, FORM HLRF, FORM iHLRF e SORM representaram adequadamente os valores encontrados pelos autores, já que os resultados foram bastante próximos.

6.2. Estudo comparativo

Esta seção apresenta um estudo da análise comparativa do índice de confiabilidade e probabilidade de falha de 14 exemplos da literatura que são apresentados na sequência. Todas as variáveis aleatórias envolvidas são estatisticamente independentes. Para cada exemplo, as distribuições de probabilidade e os momentos (média e desvio padrão) das variáveis aleatórias são apresentados, bem como a função estado limite $g(x)$.

Exemplo 1: Este exemplo consiste em determinar a probabilidade da resistência (R) exceder ou igualar a solicitação (S) ($R \geq S$) de um pilar de concreto armado, quando as distribuições estatísticas das variáveis aleatórias são conhecidas. A resistência é determinada pela capacidade resistente do pilar e a solicitação pelo somatório das cargas permanente (G), acidental (Q) e vento (W). A função estado limite desse problema é linear e determinada por:

$$g(R, S) = R - G - Q - W. \quad (6.9)$$

Os parâmetros de projeto desse pilar são apresentados na Tab. 6.11.

Tabela 6.11 – Médias e desvios padrão das variáveis aleatórias do Exemplo 1.

Variável Aleatória	Distribuição	Média (kN)	Desvio padrão (kN)	Coefficiente de variação
R	Normal	975	146,25	0,15
G	Normal	200	14,00	0,07
Q	Normal	300	36,00	0,12
W	Normal	150	30,00	0,20

Fonte: Adaptado de Ang e Tang (2007)

Considerando todas as variáveis aleatórias independentes estaticamente e com distribuição Normal de probabilidade, este exemplo pode ser resolvido analiticamente, uma vez que o índice de confiabilidade é definido através da Eq. (4.8):

$$\beta = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} = \frac{975 - (200 + 300 + 150)}{\sqrt{(146,25^2 + 14^2 + 36^2 + 30^2)}} = \frac{325}{154,21} = 2,1075$$

Este valor de β corresponde a uma probabilidade de falha de acordo com a distribuição Normal padrão acumulada, dada pela Eq. (4.9):

$$P_f = \Phi(-\beta) = 1,7429 \times 10^{-2}$$

sendo esse valor a probabilidade de falha de referência nas análises comparativas dos métodos apresentados.

Utilizando o método de simulação de Monte Carlo MCFB, com distribuição Normal e uma amostra de 10^6 elementos, os resultados obtidos do índice de confiabilidade e probabilidade de falha para o pilar de concreto armado estão representados na Tab. 6.12.

Tabela 6.12 – Comparativo dos resultados para o Exemplo 1.

MÉTODO	β	P_f
MCFB	2,10	$1,75 \times 10^{-2}$
MCAI	2,10	$1,75 \times 10^{-2}$
FOSM	2,10	$1,75 \times 10^{-2}$
FORM HLRF	2,10	$1,75 \times 10^{-2}$
FORM iHLRF	2,10	$1,75 \times 10^{-2}$
SORM	2,10	$1,75 \times 10^{-2}$

Fonte: Autora

Conforme esperado, os valores da Tab. 6.12 do índice de confiabilidade e a probabilidade de falha são iguais aos valores obtidos para a referência, de modo que o exemplo estudado representa uma função estado limite linear e com distribuição de probabilidade Normal, sendo assim, a aplicação dos métodos resultaria na solução exata do exemplo. Todos os métodos se mostraram significativos e eficientes para os resultados do Exemplo 1 em comparação com o valor analítico (referência).

Exemplo 2: Neste exemplo são utilizados os mesmos dados do Exemplo 1, com mudanças nas distribuições das variáveis, sendo que os novos parâmetros de projeto desse pilar de concreto armado são apresentados na Tab. 6.13.

Tabela 6.13 – Médias e desvios padrão das variáveis aleatórias do Exemplo 2.

Variável Aleatória	Distribuição	Média (kN)	Desvio padrão (kN)	Coefficiente de variação
R	Lognormal	975	146,25	0,15
G	Normal	200	14,00	0,07
Q	Gumbel	300	36,00	0,12
W	Gumbel	150	30,00	0,20

Fonte: Adaptado de Ang e Tang (2007)

A Figura 6.7 ilustra um comparativo através do histograma da função estado limite nas duas situações com variáveis aleatórias com distribuição Normal e mistas.

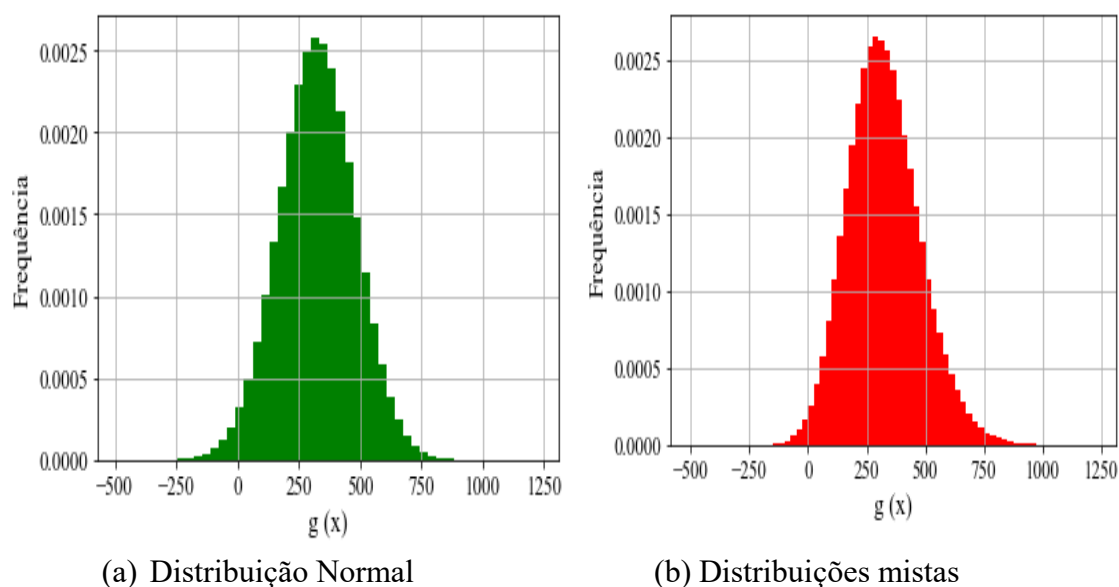


Figura 6.7 – Histograma comparativo da função estado limite dos exemplos 1 e 2.

Fonte: Autora

É perceptível que com a mudança das distribuições de probabilidade mostrada na Fig. 6.7, os resultados conduzem a uma qualidade de resultados distinta, que altera o índice de confiabilidade e probabilidade de falha, como mostrado na Tab. 6.14.

Tabela 6.14 – Comparativo dos resultados para o Exemplo 2.

MÉTODO	β	P_f
MCFB	2,35	$9,30 \times 10^{-3}$
MCAI	2,35	$9,30 \times 10^{-3}$
FOSM	2,10	$1,75 \times 10^{-2}$
FORM HLRF	2,45	$7,03 \times 10^{-3}$
FORM iHLRF	2,45	$7,03 \times 10^{-3}$
SORM	2,45	$7,03 \times 10^{-3}$

Fonte: Autora

Os resultados se mostram diferentes da referência, que nesse caso é o método de simulação de Monte Carlo. No método FOSM não houve alteração nas distribuições, então permaneceu com os mesmos resultados do Exemplo 1. Os métodos FORM HLRF, FORM iHLRF e SORM tiveram resultados próximos aos obtidos pela simulação de Monte Carlo, mostrando-se eficaz para este exemplo.

Exemplo 3: Este exemplo analisa a aplicação dos métodos estudados em uma função com variáveis aleatórias com distribuição de probabilidade Normal. A função estado limite deste exemplo é dada por (GHLEHNOVI *et al.*, 2020):

$$g(x) = \exp(0,2x_1 + 1,4) - x_2. \quad (6.10)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.15.

Tabela 6.15 – Médias e desvios padrão das variáveis aleatórias do Exemplo 3.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Normal	0,0	1,0
x_2	Normal	0,0	1,0

Fonte: Ghalehnovi *et al.* (2020)

Esta função é fracamente não linear e seu comportamento é ilustrado na Fig. 6.8.

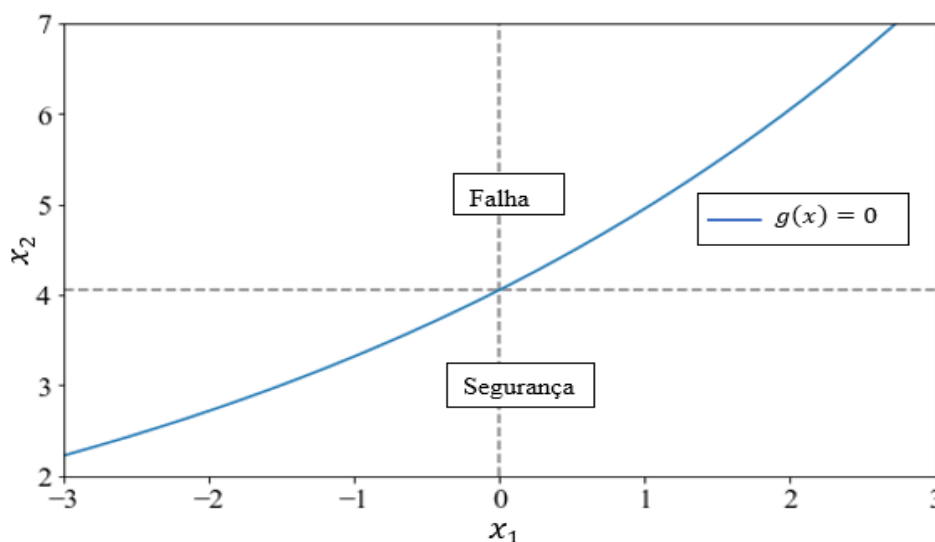


Figura 6.8 – Limite de falha e os domínios de falha e segurança, para o Exemplo 3.

Fonte: Autora

Ghalehnovi *et al.* (2020) apresentam resultados para este exemplo através de diferentes métodos. Obtiveram 3,38 como índice de confiabilidade para essa função, utilizando o MCFB com uma amostra de 10^6 elementos. Determinaram também através do método FORM um índice de confiabilidade de 3,35. Na Tab. 6.16 são apresentados os resultados obtidos por cada método estudado nesse trabalho.

Tabela 6.16 – Comparativo dos resultados para o Exemplo 3.

MÉTODO	β	P_f
MCFB	3,38	$3,74 \times 10^{-4}$
MCAI	3,38	$3,74 \times 10^{-4}$
FOSM	3,35	$4,05 \times 10^{-4}$
FORM HLRF	3,35	$4,05 \times 10^{-4}$
FORM iHLRF	3,35	$4,05 \times 10^{-4}$
SORM	3,38	$3,74 \times 10^{-4}$

Fonte: Autora

No que diz respeito aos resultados obtidos na Tab. 6.16, nota-se uma diferença nos valores do índice de confiabilidade dos métodos FOSM, FORM HLRF e FORM iHLRF, porém, as diferenças encontradas são aproximadamente 1% em comparação com os outros métodos. Esta diferença apresenta uma divergência relativamente pequena,

assim, pode-se afirmar que essa função é fracamente não linear, e que os métodos analisados são eficazes para determinar o índice de confiabilidade e probabilidade de falha desta função.

Exemplo 4: Este exemplo estuda a aplicação dos métodos apresentados, na resolução de um problema com uma função estado limite não linear. O estado limite é definido pela seguinte função (SANTOS, 2014):

$$g(p, q) = 3 - q + (4p)^4. \quad (6.11)$$

Esta função é fortemente não linear e seu comportamento é ilustrado na Fig. 6.9.

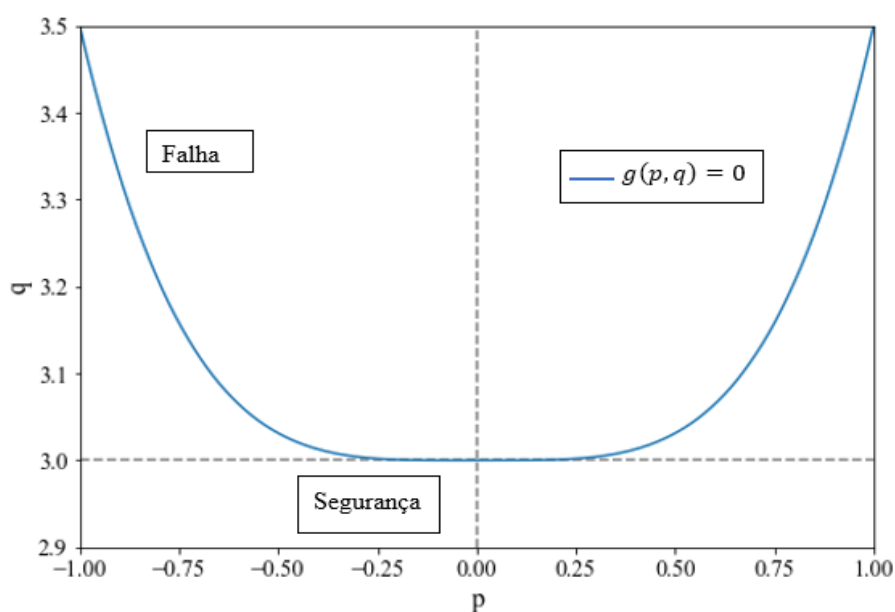


Figura 6.9 – Limite de falha e os domínios de falha e segurança, para o Exemplo 4.

Fonte: Autora

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.17.

Tabela 6.17 – Médias e desvios padrão das variáveis aleatórias do Exemplo 4.

Variável Aleatória	Distribuição	Média	Desvio padrão
p	Normal	0,0	1,0
q	Normal	0,0	1,0

Fonte: Santos (2014)

Santos (2014) apresenta resultados para este exemplo através de diferentes métodos. Obteve 3,5716 como índice de confiabilidade e uma probabilidade de falha de $1,774 \times 10^{-4}$ para essa função, utilizando o MCFB com uma amostra de 9×10^8 elementos. O autor determinou também através do método FORM uma probabilidade de falha de $1,35 \times 10^{-3}$ ($\beta = 3,0$). Na Tab. 6.18 são apresentados os resultados obtidos por cada método estudado neste trabalho.

Tabela 6.18 – Comparativo dos resultados para o Exemplo 4.

MÉTODO	β	P_f
MCFB	3,57	$1,76 \times 10^{-4}$
MCAI	3,57	$1,76 \times 10^{-4}$
FOSM	3,00	$1,35 \times 10^{-3}$
FORM HLRF	3,00	$1,35 \times 10^{-3}$
FORM iHLRF	3,00	$1,35 \times 10^{-3}$
SORM	3,00	$1,35 \times 10^{-3}$

Fonte: Autora

Utilizando o método MCFB como referência, e uma amostra com 2.10^6 elementos e MCAI com 10^4 elementos, e os métodos FOSM, FORM HLRF, FORM iHLRF e SORM foram determinados o índice de confiabilidade e a probabilidade de falha da Tab. 6.18. De acordo com os métodos FOSM, FORM e SORM, o ponto de projeto é $x^* = (0,00; 3,00)$, onde se encontra um índice de confiabilidade $\beta = 3,00$ e uma probabilidade de falha $P_f = 1,35 \times 10^{-3}$. Os métodos de transformação, apesar de apresentarem os mesmos resultados, convergem para um resultado distante 19% da referência (MCFB). O SORM não conseguiu melhorar o FORM, apesar da função ser não linear, as curvaturas na região de falha são nulas.

Exemplo 5: Uma função estado limite não linear envolvendo variáveis aleatórias Lognormal é considerada neste exemplo. O estado limite é definido por (MAHADEVAN e PAN, 2001):

$$g(x) = x_1 + 2x_2 + 3x_3 + x_4 - 5x_5x_6 \quad (6.12)$$

As variáveis aleatórias adotadas assim como as suas respectivas distribuições e parâmetros relacionados estão apresentadas na Tab. 6.19.

Tabela 6.19 – Médias e desvios padrão das variáveis aleatórias do Exemplo 5.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Lognormal	120	12
x_2	Lognormal	120	12
x_3	Lognormal	120	12
x_4	Lognormal	120	12
x_5	Lognormal	50	15
x_6	Lognormal	40	12

Fonte: Mahadevan e Pan (2001)

Mahadevan e Pan (2001) apresentam resultados para este exemplo através do Método FORM. Obtiveram 3,0483 como índice de confiabilidade para essa função. Os resultados da análise, correspondentes ao estado limite não linear acima, podem ser vistos na Tab. 6.20.

Tabela 6.20 – Comparativo dos resultados para o Exemplo 5.

MÉTODO	β	P_f
MCFB	2,94	$1,62 \times 10^{-3}$
MCAI	2,94	$1,62 \times 10^{-3}$
FOSM	3,65	$1,28 \times 10^{-4}$
FORM HLRF	3,04	$1,17 \times 10^{-3}$
FORM iHLRF	3,04	$1,17 \times 10^{-3}$
SORM	3,04	$1,17 \times 10^{-3}$

Fonte: Autora

Pode-se notar, que os resultados obtidos produziram resultados muito semelhantes, com exceção do método FOSM. Apenas o método FOSM não convergiu para o resultado dos demais métodos de transformação, isso porque não converge corretamente para distribuições não normais, e ainda obteve um β maior que os demais

métodos. Isso ocorreu porque essa função é fracamente não linear, e se os demais métodos tivessem distribuição de probabilidade Normal, convergiriam para o mesmo resultado.

Exemplo 6: Este exemplo analisa a aplicação dos métodos estudados em uma função estado limite altamente não linear com variáveis aleatórias com distribuição de probabilidade Normal. A função estado limite é dada por (GROOTEMAN, 2008):

$$g(x) = 2,5 - 0,2357(x_1 - x_2) + 0,0046(x_1 + x_2 - 20)^4 \quad (6.13)$$

As variáveis aleatórias adotadas, assim como as suas respectivas distribuições e parâmetros relacionados, estão apresentadas na Tab. 6.21.

Tabela 6.21 – Médias e desvios padrão das variáveis aleatórias do Exemplo 6.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Normal	10,0	3,0
x_2	Normal	10,0	3,0

Fonte: Grooteman (2008)

Grooteman (2008) apresentou resultados para este exemplo através do Método FORM. Obteve 2,50 como índice de confiabilidade para essa função. Determinou também através do método MCFB um $\beta = 2,75$. Os resultados da análise, correspondentes ao estado limite não linear acima, podem ser vistos na Tab. 6.22.

Tabela 6.22– Comparativo dos resultados para o Exemplo 6.

MÉTODO	β	P_f
MCFB	2,75	$2,98 \times 10^{-3}$
MCAI	2,75	$2,98 \times 10^{-3}$
FOSM	2,50	$6,21 \times 10^{-3}$
FORM HLRF	2,50	$6,21 \times 10^{-3}$
FORM iHLRF	2,50	$6,21 \times 10^{-3}$
SORM	2,50	$6,21 \times 10^{-3}$

Fonte: Autora

Utilizou-se o método MCFB como referência, com uma amostra com 10^6 elementos e MCAI com 10^4 elementos. Os métodos de transformação, apesar de apresentarem os mesmos resultados, convergem para um resultado distante 12% da referência (MC FB). Como no Exemplo 4, as curvaturas de segunda ordem $\left(B = \begin{bmatrix} 4.0051e - 07 & 3.3118e - 09 \\ 3.3118e - 09 & 4.0051e - 07 \end{bmatrix}\right)$ são nulas. Uma superfície de curvatura zero é um plano, portanto, neste caso, FORM e SORM obtiveram os mesmos resultados.

Exemplo 7: Este exemplo busca analisar uma função estado limite não linear com variáveis aleatórias definidas por diferentes distribuições de probabilidade. A função estado limite deste exemplo é dada por (SANTOS *et al.*, 2012):

$$g(x) = x_1 x_2 - 78,12 x_3. \quad (6.14)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.23.

Tabela 6.23 – Médias e desvios padrão das variáveis aleatórias do Exemplo 7.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Lognormal	$2,0 \times 10^7$	$0,5 \times 10^7$
x_2	Lognormal	10^{-4}	$0,2 \times 10^{-4}$
x_3	Gumbel	4,0	1,0

Fonte: Santos *et al.* (2012)

Santos *et al.* (2012) obtiveram 4,4282 como índice de confiabilidade para essa função, utilizando os métodos FORM HLRF e FORM iHLRF. Na Tab. 6.24 são apresentados os resultados obtidos por cada método estudado neste trabalho.

Tabela 6.24 – Comparativo dos resultados para o Exemplo 7.

MÉTODO	β	P_f
MCFB	4,43	$7,21 \times 10^{-6}$
MCAI	4,43	$7,21 \times 10^{-6}$
FOSM	3,29	$4,93 \times 10^{-4}$
FORM HLRF	4,43	$7,21 \times 10^{-6}$
FORM iHLRF	4,43	$7,21 \times 10^{-6}$
SORM	4,43	$7,21 \times 10^{-6}$

Fonte: Autora

Como era de se esperar, apenas o método FOSM não convergiu para o resultado dos demais métodos, devido às distribuições serem mistas. Dada a fraca não linearidade da função estado limite os métodos FORM HLRF e FORM iHLRF apresentam comportamento idêntico aos métodos MCFB, MCAI e SORM.

Exemplo 8: Este exemplo analisa a aplicação dos métodos estudados em uma função estado limite altamente não linear com variáveis aleatórias com distribuição de probabilidade Normal. A função estado limite deste exemplo é dada por (SANTOS *et al.*, 2012):

$$g(x) = x_1^3 + x_2^3 - 18 \quad (6.15)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.25.

Tabela 6.25 – Médias e desvios padrão das variáveis aleatórias do Exemplo 8.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Normal	10,0	5,0
x_2	Normal	10,0	5,0

Fonte: Santos *et al.* (2012)

Santos *et al.* (2012) obtiveram 2,2401 como índice de confiabilidade para essa função, utilizando os métodos FORM HLRF e FORM iHLRF. Na Tab. 6.26 são apresentados os resultados obtidos por cada método estudado neste trabalho.

Tabela 6.26 – Comparativo dos resultados para o Exemplo 8.

MÉTODO	β	P_f
MCFB	2,55	$5,33 \times 10^{-3}$
MCAI	2,55	$5,32 \times 10^{-3}$
FOSM	2,24	$1,25 \times 10^{-2}$
FORM HLRF	2,24	$1,25 \times 10^{-2}$
FORM iHLRF	2,24	$1,25 \times 10^{-2}$
SORM	2,63	$4,27 \times 10^{-3}$

Fonte: Autora

Pode-se constatar que a diferença entre os índices de confiabilidade dos métodos de Monte Carlo em comparação com o FOSM, FORM HLRF e FORM HLRF é de 12%. Na verdade, a alta não linearidade da função estado limite causa a divergência da eficiência dos métodos propostos. O MCAI se mostrou mais uma vez ser eficiente, em número de simulações (10^4 elementos) que o MCFB (10^6 elementos), conforme esperado. O SORM se aproxima do MCFB e MCAI, com uma diferença no índice de confiabilidade de apenas 3%.

Exemplo 9: Neste exemplo são utilizados os mesmos dados do exemplo 8, com mudança no x_2 de 10,0 para 9,9, sendo que os novos parâmetros dessa função são apresentados na Tab. 6.27.

Tabela 6.27 – Médias e desvios padrão das variáveis aleatórias do Exemplo 9.

Variável	Distribuição	Média	Desvio padrão
Aleatória			
x_1	Normal	10,0	5,0
x_2	Normal	9,9	5,0

Fonte: Santos *et al.* (2012)

Santos *et al.* (2012) apresentam resultados para este exemplo através do Método FORM. Obtiveram 1,1656 como índice de confiabilidade para essa função, utilizando o FORM HLRF. Determinaram também através do método FORM iHLRF um índice de confiabilidade de 2,2260. Na Tab. 6.28 são apresentados os resultados obtidos por cada método estudado nesse trabalho.

Tabela 6.28 – Comparativo dos resultados para o Exemplo 9.

MÉTODO	β	P_f
MCFB	2,55	$5,64 \times 10^{-3}$
MCAI	2,55	$5,67 \times 10^{-3}$
FOSM	1,16	$1,22 \times 10^{-1}$
FORM HLRF	1,16	$1,22 \times 10^{-1}$
FORM iHLRF	2,22	$1,30 \times 10^{-2}$
SORM	2,63	$4,27 \times 10^{-3}$

Fonte: Autora

Pode-se observar que com a mudança no x_2 de 10,0 para 9,9 os resultados dos métodos MCFB, MCAI e SORM permanecem iguais aos do exemplo 8. Claramente nota-se também que os resultados obtidos pelo FOSM, FORM HLRF e FORM iHLRF são menores que aqueles obtidos pela aplicação dos demais métodos. Pode-se constatar que a diferença entre os índices de confiabilidade dos métodos de Monte Carlo em comparação com o FOSM e FORM HLRF é de 54%. O FORM iHLRF aproxima-se dos métodos MCFB, MCAI e SORM, porém, com uma diferença de 13%, mostrando assim, ser mais robusto que o FORM HLRF, pois diminuiu em 41% a diferença do MCFB.

Exemplo 10: Este exemplo consiste em determinar o índice de confiabilidade e a probabilidade de falha na resolução de uma função de estado limite altamente não linear. O estado limite é definido por (CHOI *et al.*, 2007):

$$g(x) = x_1^4 + 2x_2^4 - 20 \quad (6.16)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.29.

Tabela 6.29 – Médias e desvios padrão das variáveis aleatórias do Exemplo 10.

Variável Aleatória	Distribuição	Média (kN)	Desvio padrão (kN)
x_1	Normal	10,0	5,0
x_2	Normal	10,0	5,0

Fonte: Choi *et al.* (2007)

Choi *et al.* (2007) obtiveram uma probabilidade de falha de $1,95 \times 10^{-3}$ ($\beta = 2,92$) utilizando o MCFB com uma amostra de 10^5 elementos. Determinaram através do método FORM iHLRF uma probabilidade de falha de $9,01 \times 10^{-3}$ ($\beta = 2,36$). E o SORM com uma probabilidade de falha de $2,22 \times 10^{-3}$ ($\beta = 2,84$).

Utilizando o método de simulação de Monte Carlo e os métodos de transformação estudados, é determinado o índice de confiabilidade e probabilidade de falha para a função do exemplo 10, sendo que seus resultados estão representados na Tab. 6.30.

Tabela 6.30 – Comparativo dos resultados para o Exemplo 10.

MÉTODO	β	P_f
MCFB	2,92	$1,95 \times 10^{-3}$
MCAI	2,92	$1,95 \times 10^{-3}$
FOSM	0,93	$1,77 \times 10^{-1}$
FORM HLRF	0,93	$1,17 \times 10^{-1}$
FORM iHLRF	2,36	$9,01 \times 10^{-3}$
SORM	2,84	$2,22 \times 10^{-3}$

Fonte: Autora

Os métodos FOSM e FORM HLRF apresentam uma divergência relativamente grande (68%) em relação aos valores de β e P_f do método de referência MCFB, que pode ser explicada pela alta não linearidade da função estado limite. O método SORM se mostrou adequado para este exemplo retornando valores muito próximo da referência (MCFB) e do método MCAI. O SORM melhora o resultado do FORM incluindo informações adicionais sobre a curvatura do estado limite. Os resultados obtidos são iguais aos da literatura de referência da função estado limite do exemplo 10, mostrando assim, a eficácia da implementação dos métodos em *Python*.

Exemplo 11: A fim de avaliar a eficiência dos métodos estudados, em um caso mais geral, considera-se um estado limite polinomial de quarto grau, representado por (GHALEHNOVI *et al.*, 2020):

$$g(x) = \frac{1}{40}x_1^4 + 2x_2^2 + x_3 + 3 \quad (6.17)$$

Os parâmetros das distribuições de probabilidade de cada variável aleatória são apresentados na Tab. 6.31.

Tabela 6.31 – Médias e desvios padrão das variáveis aleatórias do Exemplo 11.

Variável Aleatória	Distribuição	Média	Desvio padrão
x_1	Normal	0,0	1,0
x_2	Normal	0,0	1,0
x_3	Normal	0,0	1,0

Fonte: Ghalehnovi *et al.* (2020)

Ghalehnovi *et al.* (2020) obtiveram $\beta = 3,42$ para esta função, utilizando o MC FB com uma amostra de 10^7 elementos. Os autores determinaram através do método FORM um $\beta = 3,00$. Na Tab. 6.32 são apresentados os resultados obtidos por cada método.

Tabela 6.32 – Comparativo dos resultados para o Exemplo 11.

MÉTODO	β	P_f
MCFB	3,43	$3,23 \times 10^{-4}$
MCAI	3,43	$3,23 \times 10^{-4}$
FOSM	2,52	$5,91 \times 10^{-3}$
FORM HLRF	2,52	$5,91 \times 10^{-3}$
FORM iHLRF	3,00	$1,35 \times 10^{-3}$
SORM	3,37	$3,73 \times 10^{-4}$

Fonte: Autora

Conforme apresentado na Tab. 6.32, existem diferenças consideráveis entre os resultados dos métodos FOSM e FORM em relação à simulação de Monte Carlo. A alta não linearidade é a razão para essa diferença entre os métodos. O SORM se aproxima do MCFB e MCAI, com uma diferença no índice de confiabilidade de apenas 2%. Os resultados estão de acordo com os obtidos na literatura de referência da função estado limite do Exemplo 11.

Exemplo 12: Neste exemplo é analisada uma treliça de telhado sob carga uniforme, mostrada na Fig. 6.10. Os banzos superiores e barras de compressão da treliça são feitas de concreto armado, e os banzos inferiores e barras de tração são feitas de aço. A função estado limite que representa essa treliça pode ser dada da seguinte forma (KESHTEGAR e MENG, 2017):

$$g(x) = 0,03 - \left(ql^2/2\right) \cdot \left(3,81/A_c E_c\right) + \left(1,13/A_s E_s\right) \quad (6.18)$$

onde A_c e A_s são as áreas da seção transversal de concreto armado e barras de aço, respectivamente, E_c e E_s são seus módulos de elasticidade correspondentes, e l é o comprimento da treliça.

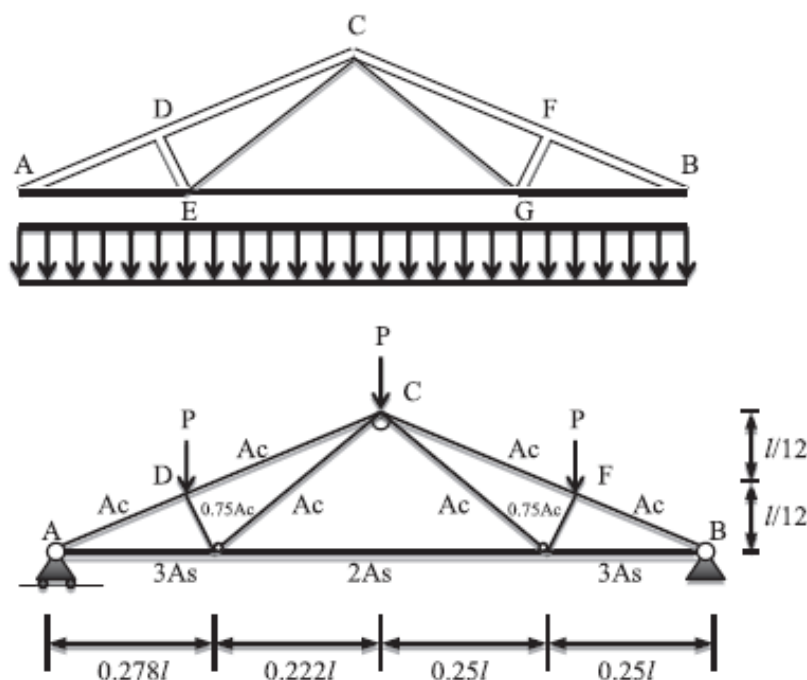


Figura 6.10 – Vista esquemática da treliça de telhado

Fonte: Keshtegar e Meng (2017)

Tabela 6.33 – Médias e desvios padrão das variáveis aleatórias do Exemplo 12.

Variável Aleatória	Unidade	Distribuição	Média	Desvio padrão
q	N/m	Normal	20000	1400
l	m	Normal	12	0,12
A_s	m ²	Normal	$9,82 \times 10^{-4}$	$5,9852 \times 10^{-5}$
A_c	m ²	Normal	0,04	0,0048
E_s	Pa	Lognormal	$1,0 \times 10^{11}$	$6,0 \times 10^9$
E_c	Pa	Gumbel	$2,0 \times 10^{10}$	$1,2 \times 10^9$

Fonte: Keshtegar e Meng (2017)

Keshtegar e Meng (2017) obtiveram 2,34697 como índice de confiabilidade para essa função, utilizando o MCFB com uma amostra de 10^6 elementos. Determinaram também através do método FORM iHLRF um índice de confiabilidade de 2,4197. Na Tab. 6.34 são apresentados os resultados obtidos neste exemplo por cada método.

Tabela 6.34 – Comparativo dos resultados para o Exemplo 12.

MÉTODO	β	P_f
MCFB	2,35	$9,81 \times 10^{-3}$
MCAI	2,35	$9,81 \times 10^{-3}$
FOSM	0,18	$4,25 \times 10^{-1}$
FORM HLRF	0,59	$2,77 \times 10^{-1}$
FORM iHLRF	2,42	$8,09 \times 10^{-3}$
SORM	2,42	$8,09 \times 10^{-3}$

Fonte: Autora

Utilizou-se o método MC FB com uma amostra com 10^7 elementos, MCAI com 10^4 elementos. A análise dos resultados da Tab. 6.34 foi realizada considerando o método MCFB como referência. O SORM e o FORM iHLRF se aproxima do MCFB e MCAI, com uma diferença no índice de confiabilidade de apenas 3%. Existem diferenças bem acentuadas entre os resultados dos métodos FOSM e FORM HLRF com 92% e 78%, respectivamente, em relação ao índice de confiabilidade do MCFB. A alta não linearidade é a razão para essa diferença.

Exemplo 13: Neste exemplo é apresentada uma estrutura de treliça de 10 barras da Fig. 6.11 com três diferentes áreas de seção transversal A_1 (vertical), A_2 (horizontal) e A_3 (diagonal). A estrutura está sujeita a duas cargas externas horizontais P . A ruptura da estrutura é considerada quando o deslocamento horizontal no canto superior direito excede o deslocamento permitido $d_0 = 0,1 \text{ m}$. A função estado limite implícita da treliça é dada, tal como (SHAYANFAR *et al.*, 2017):

$$g(A_1, A_2, A_3, B, P, E) = 0,1 - B \cdot u(A_1, A_2, A_3, P, E) \quad (6.8)$$

onde: E é o módulo de elasticidade. A variável aleatória B foi introduzida para as incertezas do modelo e $L = 9,0 \text{ m}$ é uma variável determinística. Supõe-se que A_1 , A_2 , A_3 , B , P , E são variáveis aleatórias estatisticamente independentes.

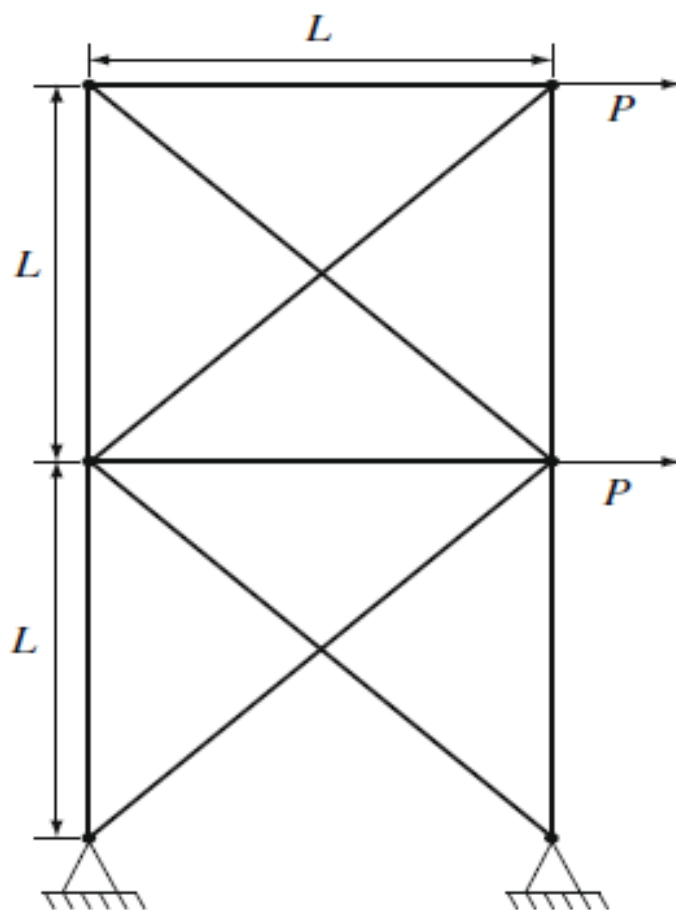


Figura 6.11 – Estrutura de treliça de 10 barras

Fonte: Shayanfar *et al.* (2017)

As variáveis aleatórias adotadas, assim como as suas respectivas distribuições e parâmetros relacionados, estão apresentadas na Tab. 6.35.

Tabela 6.35 – Médias e desvios padrão das variáveis aleatórias do Exemplo 13.

Variável Aleatória	Unidade	Distribuição	Média	Desvio padrão
A_1	m ²	Normal	10^{-2}	$5,0 \times 10^{-4}$
A_2	m ²	Normal	$1,5 \times 10^{-2}$	$7,5 \times 10^{-5}$
A_3	m ²	Normal	$6,0 \times 10^{-2}$	$3,0 \times 10^{-4}$
B		Normal	1,0	0,1
P	N	Lognormal	$25,0 \times 10^5$	$2,5 \times 10^4$
E	MPa	Gumbel	$6,9 \times 10^4$	$3,45 \times 10^3$

Fonte: Shayanfar *et al.* (2017)

Shayanfar *et al.* (2017) obtiveram 4,2994 como índice de confiabilidade através do método FORM usando o algoritmo iHLRF. Na Tab. 6.36 são apresentados os resultados obtidos por cada método, utilizando o programa *Trusspy* (DUTZLER, 2018).

Tabela 6.36 – Comparativo dos resultados para o Exemplo 13.

MÉTODO	β	P_f
MCFB	4,30	$8,54 \times 10^{-6}$
MCAI	4,30	$8,54 \times 10^{-6}$
FOSM	5,49	$2,03 \times 10^{-8}$
FORM HLRF	3,99	$3,32 \times 10^{-6}$
FORM iHLRF	4,30	$8,54 \times 10^{-6}$
SORM	4,30	$8,54 \times 10^{-6}$

Fonte: Autora

Esta treliça é um problema de referência bem conhecido, tendo sido estudado por alguns pesquisadores. E usando o método MCFB com uma amostra com 9×10^7 elementos, obteve um tempo computacional de 11h 45min 22 s, MCAI com 10^4 elementos com um tempo de 15 s, e os métodos FOSM, FORM HLRF, FORM iHLRF e SORM com um tempo menor que 5 s. A análise dos resultados da Tab. 6.36 foram

realizadas considerando o método MCFB como referência. O método MCAI mostrou mais uma vez que melhora em desempenho e eficiência o MCFB em relação ao número de simulações e o tempo computacional. Como esperado, o FOSM não convergiu para o resultado da referência. E o FORM HLRF se distanciou 7% do índice de confiabilidade alvo.

Exemplo 14: Neste exemplo é estudada uma viga contínua de três vãos, considerando os parâmetros intensidade da carga uniforme, módulo de elasticidade e altura da seção. A Fig. 6.12 ilustra a viga analisada.

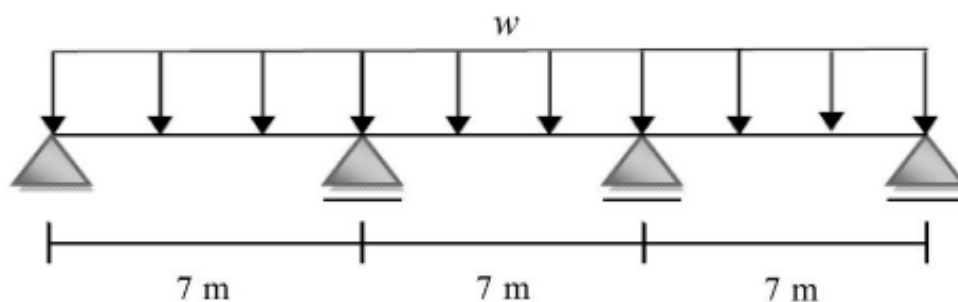


Figura 6.12 – Viga contínua de três vãos

Fonte: Shayanfar *et al.* (2017)

A função estado limite que representa essa viga pode ser dada da seguinte forma (SHAYANFAR *et al.*, 2017):

$$g(x) = (7/360) - 360 \left(x_1 / x_2 x_3^4 \right) \quad (6.20)$$

onde x_1 , x_2 e x_3 são a intensidade da carga uniforme, módulo de elasticidade e altura da seção, respectivamente. Todos estão definidos na Tab. 6.37.

Tabela 6.37 – Médias e desvios padrão das variáveis aleatórias do Exemplo 14.

Variável Aleatória	Unidade	Distribuição	Média	Desvio padrão
x_1	kN/m	Normal	10,0	0,4
x_2	kN/m ²	Normal	$2,0 \times 10^7$	$0,5 \times 10^7$
x_3	m	Normal	0,4	0,01

Fonte: Shayanfar *et al.* (2017)

Shayanfar *et al.* (2017) analisaram essa função através de três métodos. Utilizando o MCFB com uma amostra de 10^6 elementos obtiveram 3,4765 como índice de confiabilidade. Usando o método FORM iHLRF obtiveram um índice de confiabilidade de 2,5216. E o método FORM HLRF não convergiu. Na Tab. 6.38 são apresentados os resultados obtidos por cada método estudado nesse trabalho.

Tabela 6.38 – Comparativo dos resultados para o Exemplo 14.

MÉTODO	β	P_f
MCFB	2,52	$1,17 \times 10^{-3}$
MCAI	2,52	$1,17 \times 10^{-3}$
FOSM	Não convergiu	
FORM HLRF	Não convergiu	
FORM iHLRF	2,52	$1,17 \times 10^{-3}$
SORM	2,52	$1,17 \times 10^{-3}$

Fonte: Autora

Utilizou-se o método MC FB com uma amostra com 2×10^7 elementos e MCAI com 10^4 elementos. A análise dos resultados da Tab. 6.38 foi realizada considerando o método MCFB como referência. O método MCAI, assim, como nos exemplos anteriores possui o mesmo resultado do MCFB, demonstrando que utilizando a técnica com amostragem por importância, o método além de convergir para o resultado exato, melhora em relação ao número de simulações. Os métodos FOSM e FORM HLRF não conseguiram convergir para o resultado. FORM iHLRF e SORM obtiveram o mesmo dos métodos de simulação de Monte Carlo.

A Figura 6.13 apresenta um comparativo do índice de confiabilidade dos 14 exemplos analisados.

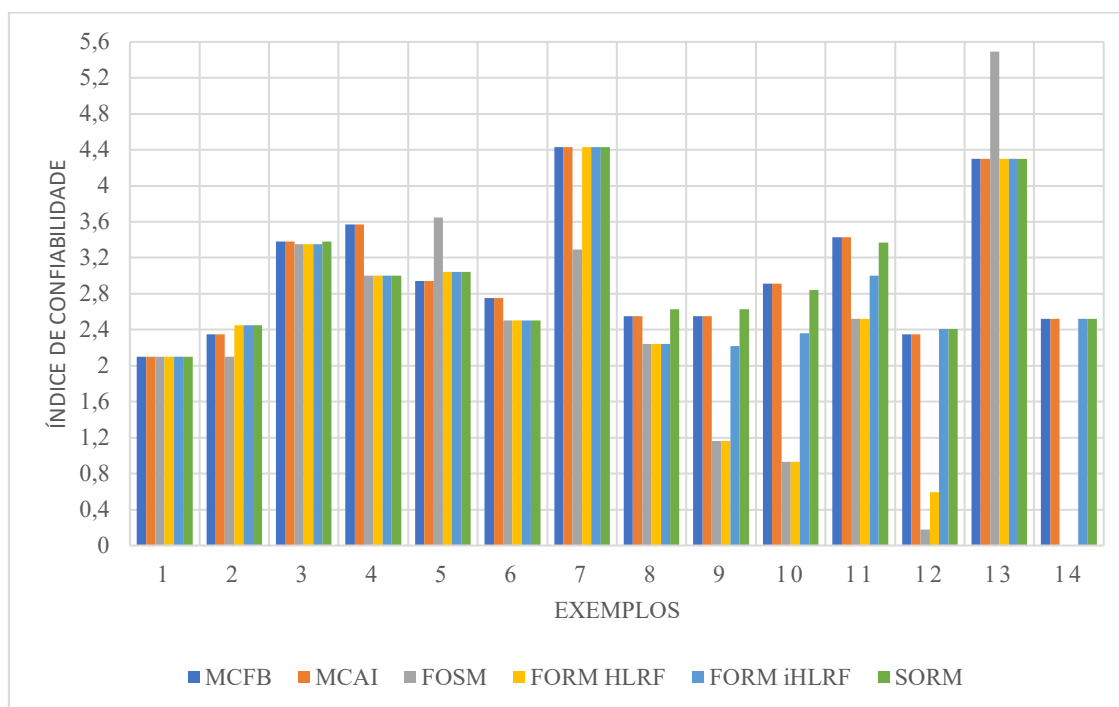


Figura 6.13 - Comparativo do índice de confiabilidade dos exemplos analisados.

Fonte: Autora

Analisando os resultados apresentados na Fig. 6.13 observa-se que o método SORM se mostrou adequado para a análise probabilística e retornou valores iguais ou muito próximo dos métodos MCFB e MCAI nos exemplos analisados. O SORM é calculado a partir da obtenção do índice de confiabilidade do método FORM. Dessa forma, o SORM se torna mais vantajoso porque melhora o resultado do FORM incluindo informações adicionais sobre a curvatura do estado limite. No caso dos exemplos 4 e 6, apesar da função ser não linear, as curvaturas na região de falha são nulas.

Ainda de acordo com a Fig. 6.13, observa-se que os métodos MCFB e MCAI resolveram com sucesso 100% dos exemplos, o SORM resolveu 85%, o FOSM, FORM HLRF e FORM iHLRF resolveram 14%, 36% e 50%, respectivamente. Logo, dos métodos de transformação, o SORM resolveu uma quantidade maior de exemplos do que os métodos FOSM e FORM.

Em relação ao desempenho dos métodos FOSM e FORM é considerado como critério o número de iterações. O FOSM e FORM HLRF atingiram o número máximo de iterações ($k = 100$) na resolução dos exemplos 7, 9, 10, 11 e 12, e o FORM iHLRF atingiu

56 como número máximo de iterações no exemplo 10. Logo, existe uma diferença significativa com base nesse critério de desempenho do FORM iHLRF em relação ao FORM HLRF.

7. CONCLUSÕES

Este trabalho apresentou um estudo comparativo da utilização de métodos de confiabilidade estrutural (MCFB, MCAI, FOSM, FORM HLRF, FORM iHLRF e SORM), levando em consideração os resultados do índice de confiabilidade e probabilidade de falha. O desempenho da implementação dos métodos na linguagem *Python* foram verificados através de cinco problemas de função de estado limite linear, não linear e implícita.

Os métodos de simulação de MCFB e MCAI funcionaram bem para todos os exemplos analisados. O método MCAI apresentou a mesma precisão do MCFB, mas, com uma maior eficiência, convergindo mais rápido por utilizar um menor número de simulações e com um custo computacional bastante inferior na análise das funções estado limite implícita.

Observou-se que a utilização do método FOSM é bastante limitada, por apresentar resultados bem distantes das referências, e assim, pode-se afirmar que o método não é adequado para a maioria dos exemplos analisados, por usar apenas os momentos de primeira e segunda ordem, não levando em conta a forma de distribuição de probabilidade. Se as variáveis aleatórias básicas não forem normais e a função estado limite linear, o nível de imprecisão cresce.

Os métodos FORM HLRF e FORM iHLRF não obtiveram bons resultados para as funções altamente não lineares analisadas, apesar desses métodos transformarem as distribuições que não são normais em normais equivalentes, trabalha com uma linearização da superfície de falha, se a função for altamente não linear no entorno do ponto de falha, os resultados desse método não são tão bons quanto os métodos MCFB e MCAI. O FORM iHLRF aproxima-se mais dos valores de referência para as funções analisadas. Logo, o algoritmo iHLRF melhora o HLRF através de uma busca linear para o ajuste.

O SORM apresentou resultados iguais ou muito próximo em relação à referência (MCFB). A aplicação desse método se mostrou eficiente em doze dos catorze exemplos analisados, e efetuou a análise de confiabilidade estrutural de segunda ordem para as funções com eficácia, otimizando de maneira relevante os resultados obtidos pelo método FORM iHLRF. O SORM é mais vantajoso porque melhora o resultado do FORM incluindo informações adicionais sobre a curvatura do estado limite no ponto de projeto.

Nos exemplos 4 e 6, as curvaturas de segunda ordem são nulas. Uma superfície de curvatura zero é um plano, neste caso, FORM e SORM obtiveram os mesmos resultados.

Os códigos programados e simulados na linguagem *Python* se mostraram eficientes para todos os métodos, pois os resultados foram bastante precisos para a análise realizada.

Conclui-se que os métodos de simulação de MCFB e MCAI funcionaram bem para funções lineares e não lineares com distribuição normal ou mista. Foram os métodos mais eficientes de todos. Dos métodos de transformação o FOSM realmente só obteve bons resultados para funções lineares com distribuição normal. O FORM HLRF e FORM iHLRF funcionaram bem para funções lineares com distribuição normal e mista. Dos métodos de transformação o SORM foi o método que resolveu uma quantidade maior de exemplos, e, portanto, pode ser considerado mais robusto do que os métodos FOSM e FORM.

Para trabalhos futuros sugere-se a utilização da otimização baseada em confiabilidade, para minimizar a função de custo de estruturas sujeitas a restrições probabilísticas, descritas por funções de estado limite, a fim de determinar soluções ótimas e seguras.

8. REFERÊNCIAS

- ALBUQUERQUE, C. S.; REAL, M. V. Estudo comparativo de métodos de confiabilidade estrutural aplicados a estruturas de concreto, **62º Congresso Brasileiro do concreto**, 2020.
- ANG, A. H-S.; TANG, W. H. **Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering**. – 2. ed. – John Wiley & Sons, 2007.
- ARAÚJO, J. M. **Curso de Concreto Armado. 4 vols.** Rio Grande, Editora Dunas, 2014.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6118**: Projeto de estruturas de concreto – Procedimento. Rio de Janeiro: ABNT, 2014.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 8800**: Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edifícios. Rio de Janeiro: ABNT, 2008.
- BECK, A. T. **Confiabilidade e Segurança das Estruturas** – 1. ed. – Rio de Janeiro: Elsevier, 2019.
- BREITUNG, K. Asymptotic approximations for multinormal integrals. **Journal of Engineering Mechanics**, vol. 110, 357–366, 1984.
- BUSSAB, W. O.; MORETTIN, P. A. **Estatística Básica** – 6. ed. – São Paulo : Saraiva, 2010.
- CHOI, S. K.; GRANDHI, R. V.; CANFIELD, R. A. **Reliability-based Structural Design**. Springer-Verlag London Limited, 2007.
- DITLEVSEN, O; MADSEN, H. O. **Structural Reliability Methods**. John Wiley & Sons, 2007.
- DUTZLER, A. **TrussPy - Truss Solver orientado a objetos para Python**, Copyright (C), 2018. Disponível em: <<https://github.com/adtzlr/trusspy>>.
- EL-REEDY, M.A. **Reinforced concrete structural reliability**. CRC Press, 2013.
- FERREIRA, E. G. **Análise de confiabilidade estrutural via método SORM DG**. 2015. Tese de Doutorado - Universidade Federal de Ouro Preto, Minas Gerais, 2015.
- GHALEHNOVI, M.; RASHKI, M.; AMERYAN, A. First order control variates algorithm for reliability analysis of engineering structures. **Applied Mathematical Modelling**, v. 77, p. 829–847, 2020.
- GONG J. X.; YI, P.; ZHAO, N. Non-gradient-based algorithm for structural reliability analysis. **J Eng Mech**, 140(6): 1–16, 2014.
- GROOTEMAN, F. Adaptive radial-based importance sampling method for structural reliability. **Structural Safety**, 30, 533-542, 2008.

HALDAR, A. **Recent developments in reliability-based civil engineering**. World Scientific Publishing Co. Pte. Ltd., 2006.

HALDAR, A.; MAHADEVAN, S. **Reliability assessment using stochastic finite element analysis**. John Wiley & Sons, 2000.

HASOFER, A. M.; LIND, N. C. Exact and invariant second-moment code format. **Journal of the engineering mechanics division**, p. 111-121, 1974.

HUANG, C.; HAMI, A. E.; RADI, B. Overview of structural reliability analysis methods – Part I: Local Reliability Methods. **ISTE OpenScience**, 2017a.

HUANG, C.; HAMI, A. E.; RADI, B. Overview of structural reliability analysis methods – Part II: Sampling Methods. **ISTE OpenScience**, 2017b.

HUANG, C.; HAMI, A. E.; RADI, B. Overview of structural reliability analysis methods – Part III: Global Reliability Methods. **ISTE OpenScience**, 2017c.

HYNNINEN, T.; HIMANEN L.; PARKKINEN, V.; MUSSO, T.; CORANDER, J.; FOSTER, A. S. An object oriented Python interface for atomistic simulations. **Computer Physics Communications**, 2016.

JOINT COMMITTEE ON STRUCTURAL SAFETY, **JCSS: Probabilistic Model Code**, 2001.

KESHTEGAR, B. Chaotic conjugate stability transformation method for structural reliability analysis. **Comput Methods Appl Mech Engrg**, 310: 866–885, 2016.

KESHTEGAR, B.; CHAKRABORTY, S. A hybrid self-adaptive conjugate first order reliability method for robust structural reliability analysis. **Applied Mathematical Modelling**, 53: 319-333, 2018.

KESHTEGAR, B.; MENG, Z. A hybrid relaxed first-order reliability method for efficient structural reliability analysis, **Structural Safety**. 66, 84–93, 2017.

KESHTEGAR, B.; MIRI, M. An enhanced HL–RF method for the computation of structural failure probability based on relaxed approach. **Civ Eng Infrastruct**, 1(1): 69–80, 2013.

KROETZ, H. M. **Otimização Estrutural sob incertezas: Métodos e Aplicações**, 2019. Tese de Doutorado, São Carlos, SP. Escola de Engenharia de São Carlos da Universidade de São Paulo.

LIBERATI, E. A. P. **Análise de Confiabilidade de lajes lisas de concreto armado submetidas à punção**. 2019. Dissertação de mestrado - Universidade Estadual de Campinas, São Paulo, 2019.

LI, Y.; D. Zhang, D. Plane Truss Reliability Numerical Simulation Based on MATLAB. **Applied Mechanics and Materials**, Vols. 256-259, p. 1091-1096, 2013.

LIU, P. L.; KIUREGHIAN, A. D. **Optimization algorithms for structural reliability analysis**. Technical Report UCB/SESM 86/09, Department of Civil Engineering, University of California, Berkeley, 1986.

LU, Z. H.; CAI, C. H.; ZHAO, Y. G.; *et al.* Normalization of correlated random variables in structural reliability analysis using fourth-moment transformation. **Structural Safety**, v. 82, p. 101888, 2020.

MAGALHÃES, F. C. **A problemática dos concretos não conformes e sua influência na confiabilidade de pilares de concreto armado**. 2014. Tese de Doutorado - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

MAPA, D. L. S.; FREITAS, M. S. R.; SILVEIRA, R. A. M.; *et al.* Confiabilidade estrutural de pórticos de aço. **Revista da Estrutura do Aço**, v. 8, n. 1, p.2-21, 2019.

MAHADEVAN S.; PAN S. Multiple linearization method for nonlinear reliability analysis. **Journal of Engineering Mechanics**, 127(11):1165–1173, 2001.

MELCHERS, R. E; BECK, A. T. **Structural Reliability Analysis and Prediction**. John Wiley & Sons, 2018.

NOWAK, A. S.; COLLINS, K. R. **Reliability of Structures**, McGraw-Hill Companies, 2000.

PAPAIIOANNOU, I.; PAPADIMITRIOU, C.; STRAUB, D. Sequential importance sampling for structural reliability analysis. **Structural Safety**, v. 62, p. 66–75, 2016.

PEREIRA, M. F.; BECK, A. T.; EL DEBS, A. L. H. C. Reliability of partially encased steel-concrete composite columns under eccentric loading. **Revista IBRACON de Estruturas e Materiais**, v. 10, n. 2, p. 298–316, 2017.

PYTHON SOFTWARE FOUNDATION. **Python Language** Site: Documentation, 2019. Disponível em: <<https://www.python.org/doc/>>.

RACKWITZ, R.; FIESSLER, B. Structural reliability under combined load sequences. **Computers & Structures**, p. 489–494, 1978.

REAL, M.V. **Análise probabilística de estruturas de concreto, submetidas a um estado plano de tensão, através do método dos elementos finitos**. Porto Alegre, Tese de Doutorado em Engenharia (Estruturas). PPGEC-UFRGS, 2000.

RIBEIRO, M. R. R. **Aplicação da teoria de confiabilidade na análise estrutural de edifícios considerando a interação solo-estrutura**. 2019. Dissertação de Mestrado - Escola Politécnica da Universidade de São Paulo, São Paulo, 2019.

SAKAMOTO, M. L. **Análise de confiabilidade estrutural utilizando o método FORM com múltiplos modos de falha para edifício alto submetido à ação do vento**. Dissertação de mestrado, Universidade Federal Tecnológica do Paraná, 2016.

SANTOS, D. M.; BECK, A. T. Reliability of beams designed in accordance with Brazilian codes. **Revista IBRACON de Estruturas e Materiais**, v. 7, n. 5, p. 723–746, 2014.

SANTOS, K. R. M. dos; STRUCCHI, F. R.; BECK, A. T. A benchmark study on intelligent sampling techniques in Monte Carlo simulation. **Latin American Journal of Solids and Structures**, v. 12, n. 4, p. 624–648, 2015.

SANTOS, K. R. M. dos. **Técnicas de amostragem inteligente em simulação de Monte Carlo**. 2014. Dissertação de Mestrado - Universidade de São Paulo - Escola de Engenharia São Carlos, São Paulo, 2014.

SANTOS, S. R. **Desenvolvimento de algoritmos matemáticos aplicados a confiabilidade estrutural**. 2012. Tese de Doutorado - Universidade Federal do Paraná, Curitiba-PR, 2012.

SANTOS, S. R.; MATIOLI, L. C.; BECK, A. T. New optimization algorithms for structural reliability analysis. **Computer Modeling in Engineering and Science**, v. 83, p. 23–56, 2012.

SCHERER, M.; MORSCH, I. B.; REAL, M. V. Reliability of reinforced concrete beams designed in accordance with Brazilian code NBR-6118:2014. **Revista IBRACON de Estruturas e Materiais**, v. 12, n. 5, p. 1086–1125, 2019.

SHAYANFAR, M. A.; BARKHORDARI, M. A.; ROUDAK, M. A. An efficient reliability algorithm for locating design point using the combination of importance sampling concepts and response surface, **Communications in Nonlinear Science and Numerical Simulation**. (47) 223–237, 2017.

SILVA JUNIOR, F. E. S. **Estudo da confiabilidade de vigas de concreto armado reforçadas com fibra de carbono através de modelo computacional**. 2019. Dissertação de Mestrado - Universidade Federal do Rio Grande, Rio Grande -RS, 2019.

SOONG, T. T.; GRIGORIU, M. **Random Vibration of Mechanical and Structural Systems**. Prentice Hall, Englewood Cliffs, New Jersey, 2000.

STRAUB, D.; SCHNEIDER, R.; BISMUT, E.; *et al.* Reliability analysis of deteriorating structural systems. **Structural Safety**, v. 82, p. 101877, 2020.]

SUDRET, B.; DER KIUREGHIAN A. Stochastic Finite Element Methods and Reliability: A state-of-the-art Report. Research report No. UCB/SEMM-2000/08, Dep. of Civil & Env. Eng., University of California, Berkeley, 2000.

TITELLO, E.P. **Análise da confiabilidade de vigas em concreto armado reforçado com fibras de aço em relação aos esforços transversais**. 2020. Dissertação de Mestrado em Engenharia Civil - Universidade Federal do Rio Grande do Sul, Porto Alegre -RS, 2020.

VERMA, A. K.; AJIT, S.; DURGA, R. K. **Reliability and Safety Engineering**. – 2. ed. – Springer-Verlag London Ltd., 2016.

WANG, X.; SHI, Q.; FAN, W.; *et al.* Comparison of the reliability-based and safety factor methods for structural design. **Applied Mathematical Modelling**, v. 72, p. 68–84, 2019.

YOUN, B. D.; CHOI, K. K.; DU, L. Adaptive probability analysis using an enhanced hybrid mean value method. **Struct Multidisc Optim** 29, p. 134–148, 2005.

ZHANG, Y.; DER KIUREGHIAN A. **Two Improved Algorithms for Reliability Analysis**. Department of Civil and Environmental Engineering, University of California, Berkeley, 1995.

APÊNDICE 1 – MCFB implementado na linguagem Python

```
#Exemplo3
import numpy as np
from scipy.stats import norm
import pandas as pd

n = int(2)
nsimul = int(1_000_000)
xvar = list(range(n))
x = np.zeros((nsimul, n))
g = np.zeros(nsimul)
index = np.zeros(nsimul)
xvar = [
    {'varname': 'x1', 'vardist': 'normal', 'varmean': 0.0, 'varstd': 1.0},
    {'varname': 'x2', 'vardist': 'normal', 'varmean': 0.0, 'varstd': 1.0},
    i = -1
for var in xvar:
    i += 1
    var['varcov'] = float(var['varstd'])/float(var['varmean'])
    print(xvar[i])
    namedist = var['vardist']
    if namedist.lower() in ['norm', 'normal', 'gauss']:
        mux = float(var['varmean'])
        sigmax = float(var['varstd'])
        x[:, i] = np.random.normal(mux, sigmax, nsimul)
    elif namedist.lower() in ['uniform', 'uniforme', 'const']:
        a = float(var['varmean'])
        b = float(var['varcov'])
        x[:, i] = np.random.uniform(a, b, nsimul)
    elif namedist.lower() in ['lognormal', 'lognorm', 'log']:
        mux = float(var['varmean'])
        sigmax = float(var['varstd'])
        zetax = np.sqrt(np.log(1.00+(sigmax/mux)**2))
        lambdax = np.log(mux)-0.5*zetax**2
        x[:, i] = np.random.lognormal(lambdax, zetax, nsimul)
    elif namedist.lower() in ['gumbel', 'extvalue1', 'evt1max']:
        mux = float(var['varmean'])
        sigmax = float(var['varstd'])
        alphan = (np.pi / np.sqrt(6.00)) / (sigmax)
        un = mux - np.euler_gamma / alphan
        betan = 1.00/alphan
        x[:, i] = np.random.gumbel(un, betan, nsimul)
g=np.exp(0.2*x[:, 0] + 1.4)- x[:, 1];
index = np.where(g <= 0.00, 1, 0)
nfail = sum(index)
pf = nfail/nsimul
betaf = -norm.ppf(pf)
```

```

error = 200.00 * np.sqrt((1 - pf)/(nsimul * pf))
print(pf)
print(betaf)
print(error)
falhasSeries = pd.Series(index)
mean_pf = np.array(falhasSeries.rolling(nsimul, min_periods=1).mean())
mean_pf = np.nan_to_num(mean_pf)
meanSeries = pd.Series(mean_pf)
std_pf = np.array(meanSeries.rolling(nsimul, min_periods=1).std())
std_pf = np.nan_to_num(std_pf)
pf_inf = mean_pf - 1.96 * std_pf
pf_sup = mean_pf + 1.96 * std_pf
beta = -norm.ppf(mean_pf)
beta_sup = -norm.ppf(pf_inf)
beta_inf = -norm.ppf(pf_sup)
print('Valor médio de Pf={0:0.5e}'.format(mean_pf[-1]))
print('Desvio padrão de Pf={0:0.5e}'.format(std_pf[-1]))
print('Beta = {0:0.4f}'.format(beta[-1]))

```

APÊNDICE 2 – MCAI implementado na linguagem Python

```
#Exemplo5
import numpy as np
from scipy.stats import uniform
from scipy.stats import norm
#from scipy.stats import lognorm
from scipy.stats import gumbel_r
from scipy.stats import invweibull
from scipy.stats import weibull_min
from scipy.special import gamma
import scipy.optimize
import pandas as pd
def fkapa(kapa, deltax, gsignal):
    fk = 1.00 + deltax ** 2 - gamma(1.00 + gsignal*2.00 / kapa) / gamma(1.00 +
    gsignal*1.00 / kapa) ** 2
    return fk
xvar = [
    {'varname': 'x1', 'vardist': 'lognorm', 'varmean': 120.0, 'varcov': 0.1, 'varhmean': 117.04},
    {'varname': 'x2', 'vardist': 'lognorm', 'varmean': 120.0, 'varcov': 0.1, 'varhmean': 114.81},
    {'varname': 'x3', 'vardist': 'lognorm', 'varmean': 120.0, 'varcov': 0.1, 'varhmean': 112.71},
    {'varname': 'x4', 'vardist': 'lognorm', 'varmean': 120.0, 'varcov': 0.1, 'varhmean': 117.04},
    {'varname': 'x5', 'vardist': 'lognorm', 'varmean': 50.0, 'varcov': 0.3, 'varhmean': 100.85},
    {'varname': 'x6', 'vardist': 'lognorm', 'varmean': 40.0, 'varcov': 0.3, 'varhmean': 59.52}]
n = len(xvar)
nsimul = int(10_000)
x = np.zeros((nsimul, n))
weight = np.ones(nsimul)
i = -1
for var in xvar:
    i += 1
    var['varstd'] = float(var['varcov'])*float(var['varmean'])
    print(xvar[i])
    namedist = var['vardist']
    if namedist.lower() in ['norm', 'normal', 'gauss']:
        mux = float(var['varmean'])
        sigmax = float(var['varstd'])
        muhx = float(var['varhmean'])
        x[:, i] = np.random.normal(mux, sigmax, nsimul)
        fx = norm.pdf(x[:, i], mux, sigmax)
        hx = norm.pdf(x[:, i], muhx, sigmax)
        weight = weight * (fx / hx)
    elif namedist.lower() in ['uniform', 'uniforme', 'const']:
        a = float(var['a'])
        b = float(var['b'])
        x[:, i] = np.random.uniform(a, b, nsimul)
        fx = uniform.pdf(x[:, i], a, b)
        hx = uniform.pdf(x[:, i], a, b)
        weight = weight * (fx / hx)
```

```

elif namedist.lower() in ['lognormal', 'lognorm', 'log']:
    mux = float(var['varmean'])
    sigma = float(var['varstd'])
    muhx = float(var['varhmean'])
    zetax = np.sqrt(np.log(1.00+(sigma/mux)**2))
    lambdax = np.log(mux)-0.5*zetax**2
    lambdahx = np.log(muhx)-0.5*zetax**2
    x[:, i] = np.random.lognormal(lambdahx, zetax, nsimul)
    fx = norm.pdf(np.log(x[:, i]), lambdax, zetax)
    hx = norm.pdf(np.log(x[:, i]), lambdahx, zetax)
    weight = weight * (fx / hx)
elif namedist.lower() in ['gumbel', 'extvalue1', 'evt1max']:
    mux = float(var['varmean'])
    sigma = float(var['varstd'])
    muhx = float(var['varhmean'])
    alphan = np.pi / np.sqrt(6.00) / sigma
    un = mux - np.euler_gamma / alphan
    betan = 1.00/alphan
    uhn = muhx - np.euler_gamma / alphan
    x[:, i] = np.random.gumbel(uhn, betan, nsimul)
    fx = gumbel_r.pdf(x[:, i], un, betan)
    hx = gumbel_r.pdf(x[:, i], uhn, betan)
    weight = weight * (fx / hx)
elif namedist.lower() in ['frechet', 'extvalue2', 'evt2max']:
    mux = float(var['varmean'])
    sigma = float(var['varstd'])
    muhx = float(var['varhmean'])
    deltax = sigma / mux
    kapa0 = 2.50
    gsinal = -1.00
    kapa = scipy.optimize.newton(fkapa, kapa0, args=(deltax, gsinal))
    vn = mux / gamma(1.00 - 1.00 / kapa)
    vhn = muhx / gamma(1.00 - 1.00 / kapa)
    x[:, i] = invweibull.rvs(kapa, size=nsimul)*vhn
    ynf = x[:, i] / vn
    ynh = x[:, i] / vhn
    fx = invweibull.pdf(ynf, kapa) / vn
    hx = invweibull.pdf(ynh, kapa) / vhn
    weight = weight * (fx / hx)
elif namedist.lower() in ['weibull', 'extvalue3', 'evt3min']:
    mux = float(var['varmean'])
    sigma = float(var['varstd'])
    epsilon = float(var['varinf'])
    muhx = float(var['varhmean'])
    deltax = sigma / (mux-epsilon)
    kapa0 = 2.50
    gsinal = 1.00
    kapa = scipy.optimize.newton(fkapa, kapa0, args=(deltax, gsinal))
    w1 = (mux-epsilon) / gamma(1.00 + 1.00 / kapa)+epsilon
    w1h = (muhx-epsilon) / gamma(1.00 + 1.00 / kapa)+epsilon

```

```

x[:, i] = weibull_min.rvs(kapa, size=nsimul) * (w1h - epsilon) + epsilon
ynf = (x[:, i] - epsilon) / (w1 - epsilon)
ynh = (x[:, i] - epsilon) / (w1h - epsilon)
fx = weibull_min.pdf(ynf, kapa) / (w1-epsilon)
hx = weibull_min.pdf(ynh, kapa) / (w1h-epsilon)
weight = weight * (fx / hx)
g = x[:, 0]+2*x[:, 1]+3*x[:, 2]+x[:, 3]-5*x[:, 4]-5*x[:, 5]
index = np.where(g <= 0.00, 1, 0) * weight
nfail = sum(index)
pf = nfail/nsimul
cvpf1 = 1./(pf*np.sqrt(nsimul*(nsimul-1)))*np.sqrt(sum(index**2)-pf**2)
betaf = -norm.ppf(pf)
print(pf)
print(betaf)
print(cvpf1)
falhasSeries = pd.Series(index)
mean_pf = np.array(falhasSeries.rolling(nsimul, min_periods=1).mean())
mean_pf = np.nan_to_num(mean_pf)
meanSeries = pd.Series(mean_pf)
std_pf = np.array(meanSeries.rolling(nsimul, min_periods=1).std())
std_pf = np.nan_to_num(std_pf)
cvpf2 = std_pf[-1]/mean_pf[-1]
pf_inf = mean_pf-1.96*std_pf
pf_sup = mean_pf+1.96*std_pf
beta = -norm.ppf(mean_pf)
beta_sup = -norm.ppf(pf_inf)
beta_inf = -norm.ppf(pf_sup)
print('Método de Monte Carlo com Amostragem por Importância')
print('Número de simulações = {0:0.5e}'.format(nsimul))
print('Valor médio de Pf={0:0.5e}'.format(mean_pf[-1]))
print('Desvio padrão de Pf={0:0.5e}'.format(std_pf[-1]))
print('Coef. de Variação de Pf = {0:0.4e}'.format(cvpf2))
print('Beta = {0:0.4f}'.format(beta[-1]))

```

APÊNDICE 3 – FOSM implementado na linguagem Python

```
#Problema1
import numpy as np
from scipy.stats import norm
from scipy import optimize
def gfunc(x):
    g=(x[0]*x[1])-x[2]
    return g
def mfunc(y,g,c):
    m=1./2. *(np.linalg.norm(y))**2+c*np.abs(g)
    return m
n=int(3)
mux=np.zeros(n)
sigmax=np.zeros(n)
xk=np.zeros(n)
yk=np.zeros(n)
yk1=np.zeros(n)
gradxk=np.zeros(n)
gradyk=np.zeros(n)
alpha=np.zeros(n)
mux=np.array([40.0,50.0,1000.0])
sigmax=np.array([5.0,2.5,200.0])
xk=mux
I=np.eye(n)
D=sigmax*I
Jyx=np.linalg.inv(D)
Jxy=D
y1=Jyx.dot(xk-mux)
epsilon=1e-6
delta=1e-6*gfunc(xk)
erro1=1000.00
erro2=1000.00
kiter=0
eps=1.e-8
print('k      yk      alphak      Beta      erro1      g(x)      xk')
while (erro1>epsilon or erro2>delta) and kiter<100:
    kiter+=1
    yk=yk1
    xk=mux+Jxy.dot(yk)
    gradxk=optimize.approx_fprime(xk, gfunc,eps)
    gradyk=Jxy.dot(gradxk)
    normgradyk=np.linalg.norm(gradyk)
    alpha=gradyk/normgradyk
    normyk=np.linalg.norm(yk)
    gxk=gfunc(xk)
    beta=np.linalg.norm(yk)
    prod=normgradyk*normyk
    if prod>0:
        erro1=1.-np.abs(np.dot(gradyk,yk)/(normgradyk*normyk))
```

```

else:
    erro1=1000.00
    erro2=np.abs(gxk)
    np.set_printoptions(formatter={'float': '{: 0.4f}'.format})
    print(kiter,yk,alpha,'{0:10.4f} {1:10.4e} {2:10.4f}'.format(beta,erro1,erro2),xk)
    dk=((np.dot(gradyk,yk)-gxk)/normgradyk**2)*gradyk-yk
    lambdak=1.00
    yk1=yk+lambdak*dk
    print("\nbeta ={:0.4f}'.format(beta))
    pf = norm.cdf(-beta)
    print("\nProbability of Failure Pf = {0:0.4e}'.format(pf))

```

APÊNDICE 4 – FORM HLRF implementado na linguagem Python

```

#Exemplo4
import numpy as np
from scipy import optimize
from scipy.stats import norm
def gfunc1(x):
    g=3-x[1]+(4*x[0]**2)
    return g
def mfunc(normy,g,c):
    my=1./2.*normy**2+c*np.abs(g)
    return my
def normeqv(xval,xpar1,xpar2,xpar3,xpar4,namedist):
    if namedist.lower() in ['norm','normal','gauss']:
        mux=xpar1
        sigmax=xpar2
        muxneq=mux
        sigmaxneq=sigmax
    elif namedist.lower() in ['uniform','uniforme','const']:
        a=xpar1
        b=xpar2
        c=(b-a)
        pdfx=1./c
        cdfx=(xval-a)/c
        zval=norm.ppf(cdfx)
        sigmaxneq=(norm.pdf(zval))/pdfx
        muxneq=xval-zval*sigmaxneq
    elif namedist.lower() in ['lognormal','lognorm','log']:
        mux=xpar1
        sigmax=xpar2
        zetax=np.sqrt(np.log(1.+(sigmax/mux)**2))
        lambdax=np.log(mux)-0.50*zetax**2
        sigmaxneq=zetax*xval
        muxneq=xval*(1.-np.log(xval)+lambdax)
    elif namedist.lower() in ['gumbel','extvalue1','evt1max']:
        mux=xpar1
        sigmax=xpar2
        alphan=(np.pi/np.sqrt(6.00))/(sigmax)
        un=mux-np.euler_gamma/alphan
        cdfx=np.exp(-np.exp(-alphan*(xval-un)))
        pdfx=alphan*np.exp(-alphan*(xval-un))*cdfx
        zval=norm.ppf(cdfx)
        sigmaxneq=norm.pdf(zval)/pdfx
        muxneq=xval-zval*sigmaxneq
    n=int(2)
    muxneqk=np.zeros(n)
    sigmaxneqk=np.zeros(n)
    namevar=np.array(['x1','x2'])
    mux0=np.array([0.0,0.0])
    sigmax0=np.array([1.0,1.0])

```

```

Imatrix=np.eye(n)
D=sigmax0*Imatrix
Jyx=np.linalg.inv(D)
Jxy=np.copy(D)
yk1=np.zeros(n)
xk1=mux0+Jxy.dot(yk1)
epsilon=1e-6
delta=1e-6*np.abs(gfunc1(xk1))
erro1=1000.00
erro2=1000.00
kiter=0
eps=1.e-8
while (erro1>epsilon or erro2>delta) and kiter<100:
    kiter+=1
    xk=np.copy(xk1)
    for i in range(n):
        xval=xk[i]
        mux=mux0[i]
        sigmax=sigmax0[i]
        namedist=dist[i]
        muxneqk[i],sigmaxneqk[i]=normeqv(xval,mux,sigmax,0,0,namedist)
    Dneq=sigmaxneqk*Imatrix
    Jyx=np.linalg.inv(Dneq)
    Jxy=np.copy(Dneq)
    yk=Jyx.dot(xk-muxneqk)
    normyk=np.linalg.norm(yk)
    beta=np.linalg.norm(yk)
    gxk=gfunc1(xk)
    gradxk=optimize.approx_fprime(xk, gfunc1,eps)
    gradyk=np.transpose(Jxy).dot(gradxk)
    normgradyk=np.linalg.norm(gradyk)
    alpha=gradyk/normgradyk
    dk=((np.dot(gradyk,yk)-gxk)/normgradyk**2)*gradyk-yk
    lambdak=1.00
    yk1=yk+lambdak*dk
    xk1=muxneqk+Jxy.dot(yk1)
    prod=normgradyk*normyk
    if np.abs(prod)>eps:
        erro1=1.-np.abs(np.dot(gradyk,yk)/(normgradyk*normyk))
    else:
        erro1=1000.00
    erro2=np.abs(gxk)
    print("\nIteration number = {0:d} g(x)={1:0.5e} erro1={2:0.5e} Beta={3:0.4f}'
        .format(kiter,gxk,erro1,beta))
    datadict={'xvar':namevar,'prob_dist':dist,'mux':muxneqk,'sigmax':sigmaxneqk,
        'xk':xk,'yk':yk,'alpha':alpha}
    data=pd.DataFrame(datadict)
    print(data)
pf = norm.cdf(-beta)
print("\nProbability of Failure Pf = {0:0.4e}'.format(pf))

```

APÊNDICE 5 – FORM iHLRF implementado na linguagem Python

```
#Exemplo5
import numpy as np
from scipy import optimize
from scipy.stats import norm
import scipy.linalg
import pandas as pd

def gfunc1(x):
    g=x[0]+2*x[1]+3*x[2]+x[3]-5*x[4]*x[5]
    return g
def mfunc(normy,g,c):
    my=1./2.*normy**2+c*np.abs(g)
    return my
def normeqv(xval,xpar1,xpar2,xpar3,xpar4,namedist):
    if namedist.lower() in ['norm','normal','gauss']:
        mux=xpar1
        sigmax=xpar2
        muxneq=mux
        sigmaxneq=sigmax
    elif namedist.lower() in ['uniform','uniforme','const']:
        a=xpar1
        b=xpar2
        c=(b-a)
        pdfx=1./c
        cdfx=(xval-a)/c
        zval=norm.ppf(cdfx)
        sigmaxneq=(norm.pdf(zval))/pdfx
        muxneq=xval-zval*sigmaxneq
    elif namedist.lower() in ['lognormal','lognorm','log']:
        mux=xpar1
        sigmax=xpar2
        zetax=np.sqrt(np.log(1.+(sigmax/mux)**2))
        lambdax=np.log(mux)-0.50*zetax**2
        sigmaxneq=zetax*xval
        muxneq=xval*(1.-np.log(xval)+lambdax)
    elif namedist.lower() in ['gumbel','extvalue1','evt1max']:
        mux=xpar1
        sigmax=xpar2
        alphan=(np.pi/np.sqrt(6.00))/(sigmax)
        un=mux-np.euler_gamma/alphan
        cdfx=np.exp(-np.exp(-alphan*(xval-un)))
        pdfx=alphan*np.exp(-alphan*(xval-un))*cdfx
        zval=norm.ppf(cdfx)
        sigmaxneq=norm.pdf(zval)/pdfx
        muxneq=xval-zval*sigmaxneq
n=int(6)
```

```

muxneqk=np.zeros(n)
sigmaxneqk=np.zeros(n)
namevar=np.array(['x1','x2','x3','x4','x5','x6' ])
mux0=np.array([120.0,120.0,120.0,120.0,50.0,40.0])
sigmax0=np.array([12.0,12.0,12.0,12.0,15.0,12.0])
dist=['lognormal','lognormal','lognormal','lognormal','lognormal','lognormal']
Rz=np.array([[1.00,0.00,0.00,0.00,0.00,0.00],[0.00,1.00,0.00,0.00,0.00,0.00],\
             [0.00,0.00,1.00,0.00,0.00,0.00], [0.00,0.00,0.00,1.00,0.00,0.00],\
             [0.00,0.00,0.00,0.00,1.00,0.00],[0.00,0.00,0.00,0.00,0.00,1.00]])
Imatrix=np.eye(n)
L=scipy.linalg.cholesky(Rz, lower=True)
Jzy=np.copy(L)
Jyz=np.linalg.inv(L)
eigvals, eigvecs =scipy.linalg.eig(Rz)
Lambdamatrix=eigvals.real*Imatrix
Amatrix=np.transpose(eigvecs.real)
Jzy=np.transpose(np.dot(Lambdamatrix**(1./2.),Amatrix))
Jyz=np.transpose(np.dot(Amatrix,(np.linalg.inv(Lambdamatrix)**1./2.)))
Imatrix=np.eye(n)
D=sigmax0*Imatrix
Jzx=np.linalg.inv(D)
Jyx=np.dot(Jyz,Jzx)
Jxz=np.copy(D)
Jxy=np.dot(Jxz,Jzy)
yk1=np.zeros(n)
xk1=mux0+Jxy.dot(yk1)
epsilon=1e-6
delta=1e-6*np.abs(gfunc1(xk1))
erro1=1000.00
erro2=1000.00
kiter=0
eps=1.e-8
iHLRF='True'
while (erro1>epsilon or erro2>delta) and kiter<100:
    kiter+=1
    xk=np.copy(xk1)
    for i in range(n):
        xval=xk[i]
        mux=mux0[i]
        sigmax=sigmax0[i]
        namedist=dist[i]
        muxneqk[i],sigmaxneqk[i]=normeqv(xval,mux,sigmax,0,0,namedist)
    Dneq=sigmaxneqk*Imatrix
    Jzx=np.linalg.inv(Dneq)
    Jyx=np.dot(Jyz,Jzx)
    Jxz=np.copy(Dneq)
    Jxy=np.dot(Jxz,Jzy)
    yk=Jyx.dot(xk-muxneqk)
    normyk=np.linalg.norm(yk)
    beta=np.linalg.norm(yk)

```

```

gxk=gfunc1(xk)
gradxk=optimize.approx_fprime(xk, gfunc1,eps)
gradyk=np.transpose(Jxy).dot(gradxk)
normgradyk=np.linalg.norm(gradyk)
alpha=gradyk/normgradyk
dk=((np.dot(gradyk,yk)-gxk)/normgradyk**2)*gradyk-yk
lambdak=1.00
yk1=yk+lambdak*dk
if iHLRF=='True':
    gamma=2.0
    a=0.1
    b=0.5
    gyk=gxk
    normyk=np.linalg.norm(yk)
    normyk1=np.linalg.norm(yk1)
    c1=normyk/normgradyk
    if erro2>delta:
        c2=0.5*normyk1**2/np.abs(gyk)
        ck=gamma*np.max([c1,c2])
    else:
        ck=gamma*c1
    k=-1
    f1=1.00
    f2=0.00
    while f1>f2 and k<10:
        k+=1
        lambdak=b**k
        yk1=yk+lambdak*dk
        xk1=muxneqk+Jxy.dot(yk1)
        gyk1=gfunc1(xk1)
        normyk1=np.linalg.norm(yk1)
        f1=mfunc(normyk1,gyk1,ck)-mfunc(normyk,gyk,ck)
        gradm=yk+ck*gradyk*np.sign(gyk)
        normgradm=np.linalg.norm(gradm)
        f2=a*lambdak*np.dot(gradm,dk)
    yk1=yk+lambdak*dk
    xk1=muxneqk+Jxy.dot(yk1)
    prod=normgradyk*normyk
    if np.abs(prod)>eps:
        erro1=1.-np.abs(np.dot(gradyk,yk)/(normgradyk*normyk))
    else:
        erro1=1000.00
    erro2=np.abs(gxk)
    print("\nIteration number = {0:d} g(x)={1:0.5e} erro1={2:0.5e} Beta={3:0.4f}'
        .format(kiter,gxk,erro1,beta))
    datadict={'xvar':namevar,'prob_dist':dist,'mux':muxneqk,'sigmax':sigmaxneqk,
        'xk':xk,'yk':yk,'alpha':alpha}
    data=pd.DataFrame(datadict)
    print(data)

```

```
pf = norm.cdf(-beta)
print("\nProbability of Failure Pf = {0:0.4e}'.format(pf))
```

APÊNDICE 6 – SORM implementado na linguagem Python

```

#Exemplo10
import numpy as np
from scipy import optimize
from scipy.stats import norm
import pandas as pd

def g(x):
    g=x[0]**4+2*x[1]**4-20.00
    return g
def orthogonalize(A, n):
    b = A[:, n][:, np.newaxis]
    # se for o primeiro vetor, somente normalize (norma igual a um)
    if n == 0:
        return b / np.linalg.norm(b)
    else:
        X = A[:, range(0, n)]
        # matriz identidade nxn (n = número de linhas)
        I = np.eye(X.shape[0])
        M = (I - X @ np.linalg.inv(X.T @ X) @ X.T)
        e = M.dot(b)
        return e / np.linalg.norm(e)
def q(A,n):
    qmatrix=np.zeros((n,n))
    hmatrix=np.zeros((n,n))
    qmatrix=np.concatenate([orthogonalize(A, i) for i in range(n)], axis=1)
    for i in range(n):
        hmatrix[:,i]=qmatrix[:,n-1-i]
    return hmatrix
def second_order_derivative(x,i,j):
    epsilon=1.e-4 # tolerance for the increments
    h1=epsilon # increments: h1 and h2, when i is not equal to j
    h2=epsilon # different increments can be adopted
    h=epsilon # increment h
    a=x[i] # reference value for x[i]
    b=x[j] # reference value for x[j]
    if i==j:
        x0=np.copy(x)
        x0[i]=a-h
        g10=g(x0)
        x0[i]=a
        g00=g(x0)
        x0[i]=a+h
        g20=g(x0)
        d2g=(g10-2.*g00+g20)/h**2 # second order derivative: d2g/dxi2
    else:

```

```

x0=np.copy(x)
x0[i]=a+h1
x0[j]=b+h2
g22=g(x0)
x0[i]=a+h1
x0[j]=b-h2
g21=g(x0)
x0[i]=a-h1
x0[j]=b+h2
g12=g(x0)
x0[i]=a-h1
x0[j]=b-h2
g11=g(x0)
d2g=(g22-g21-g12+g11)/(4.*h1*h2) # second order derivative: d2g/dxidxj
return d2g
def mfunc(normy,g,c):
    my=1./2.*normy**2+c*np.abs(g)
    return my
def dsigmaxneqv(xval,xpar1,xpar2,xpar3,xpar4,namedist):
    epsilon=1.e-6
    h=epsilon
    a=xval
    muxneq=0.00
    muxneq,f1= normeqv(a,xpar1,xpar2,xpar3,xpar4,namedist)
    muxneq,f2= normeqv(a+h,xpar1,xpar2,xpar3,xpar4,namedist)
    dsigmaxn=(f2-f1)/h
    if namedist.lower() in ['norm','normal','gauss']:
        dsigmaxn=1.00
    return dsigmaxn
    if namedist.lower() in ['norm','normal','gauss']:
        mux=xpar1
        sigma=xpar2
        muxneq=mux
        sigmaxneq=sigma
    elif namedist.lower() in ['uniform','uniforme','const']:
        a=xpar1
        b=xpar2
        c=(b-a)
        pdfx=1./c
        cdfx=(xval-a)/c
        zval=norm.ppf(cdfx)
        sigmaxneq=(norm.pdf(zval))/pdfx
        muxneq=xval-zval*sigmaxneq
    elif namedist.lower() in ['lognormal','lognorm','log']:
        mux=xpar1
        sigma=xpar2
        zetax=np.sqrt(np.log(1.+(sigma/mux)**2))
        lambdax=np.log(mux)-0.50*zetax**2
        sigmaxneq=zetax*xval
        muxneq=xval*(1.-np.log(xval)+lambdax)

```

```

elif namedist.lower() in ['gumbel','extvalue1','evt1max']:
    mux=xpar1
    sigmax=xpar2
    alphan=(np.pi/np.sqrt(6.00))/(sigmax)
    un=mux-np.euler_gamma/alphan
    cdfx=np.exp(-np.exp(-alphan*(xval-un)))
    pdfx=alphan*np.exp(-alphan*(xval-un))*cdfx
    zval=norm.ppf(cdfx)
    sigmaxneq=norm.pdf(zval)/pdfx
    muxneq=xval-zval*sigmaxneq
n=int(2)
muxneqk=np.zeros(n)
sigmaxneqk=np.zeros(n)
namevar=np.array(['X1','X2'])
mux0=np.array([10.00,10.00])
sigmax0=np.array([5.00,5.00])
Imatrix=np.eye(n)
D=sigmax0*Imatrix
Jyx=np.linalg.inv(D)
Jxy=np.copy(D)
yk1=np.zeros(n)
xk1=mux0+Jxy.dot(yk1)
epsilon=1e-6
delta=1e-6*np.abs(g(xk1))
erro1=1000.00
erro2=1000.00
kiter=0
eps=1.e-8
while (erro1>epsilon or erro2>delta) and kiter<100:
    kiter+=1
    xk=np.copy(xk1)
    for i in range(n):
        xval=xk[i]
        mux=mux0[i]
        sigmax=sigmax0[i]
        namedist=dist[i]
        muxneqk[i],sigmaxneqk[i]=normeqv(xval,mux,sigmax,0,0,namedist)
    Dneq=sigmaxneqk*Imatrix
    Jyx=np.linalg.inv(Dneq)
    Jxy=np.copy(Dneq)
    yk=Jyx.dot(xk-muxneqk)
    normyk=np.linalg.norm(yk)
    beta=np.linalg.norm(yk)
    gxk=g(xk)
    gradxk=optimize.approx_fprime(xk, g,eps)
    gradyk=np.transpose(Jxy).dot(gradxk)
    normgradyk=np.linalg.norm(gradyk)
    alpha=gradyk/normgradyk
    dk=((np.dot(gradyk,yk)-gxk)/normgradyk**2)*gradyk-yk
    lambdak=1.00

```

```

yk1=yk+lambdak*dk
gamma=2.0
a=0.1
b=0.5
gyk=gxk
normyk=np.linalg.norm(yk)
normyk1=np.linalg.norm(yk1)
c1=normyk/normgradyk
if erro2>delta:
    c2=0.5*normyk1**2/np.abs(gyk)
    ck=gamma*np.max([c1,c2])
else:
    ck=gamma*c1
k=-1
f1=1.00
f2=0.00
while f1>f2 and k<10:
    k+=1
    lambdak=b**k
    yk1=yk+lambdak*dk
    xk1=muxneqk+Jxy.dot(yk1)
    gyk1=g(xk1)
    normyk1=np.linalg.norm(yk1)
    f1=mfunc(normyk1,gyk1,ck)-mfunc(normyk,gyk,ck)
    gradm=yk+ck*gradyk*np.sign(gyk)
    normgradm=np.linalg.norm(gradm)
    f2=a*lambdak*np.dot(gradm,dk)
    yk1=yk+lambdak*dk
    xk1=muxneqk+Jxy.dot(yk1)
    prod=normgradyk*normyk
    if np.abs(prod)>eps:
        erro1=1.-np.abs(np.dot(gradyk,yk)/(normgradyk*normyk))
    else:
        erro1=1000.00
        erro2=np.abs(gxk)
    print("\nIteration number = {0:d} g(x)={1:0.5e} erro1={2:0.5e} Beta={3:0.4f}'
        .format(kiter,gxk,erro1,beta))
    datadict={'xvar':namevar,'prob_dist':dist,'mux':muxneqk,'sigmax':sigmaxneqk,
        'xk':xk,'yk':yk,'alpha':alpha}
    data=pd.DataFrame(datadict)
    print(data)
print("\nSORM results:")
pfform=norm.cdf(-beta)
bmatrix=np.zeros((n,n))
d2gmatrix=np.zeros((n,n))
amatrix=np.eye(n)
hmatrix=np.zeros((n,n))
qmatrix=np.zeros((n,n))
np.set_printoptions(precision=4)
for i in range(n):

```

```

    for j in range(n):
        d2gmatrix[i,j]=second_order_derivative(xk,i,j)
print("\nHessian matrix:")
print(d2gmatrix)
variancematrix=Imatrix*sigmaxneqk**2
print("\nVariance matrix:")
print(variancematrix)
dsigmaxneq=np.zeros(n)
for i in range(n):
    xval=xk[i]
    mux=mux0[i]
    sigmax=sigmax0[i]
    namedist=dist[i]
    dsigmaxn=dsigmaxneqv(xval,mux,sigmax,0,0,namedist)
    dsigmaxneq[i]=dsigmaxn*sigmaxneqk[i]
bmatrix=1./normgradyk*d2gmatrix.dot(variancematrix)
print("\nNorm of the gradient of g(y) =",normgradyk)
print("\nB matrix:")
print(bmatrix)
amatrix[:,0]=alpha
hmatrix=q(amatrix,n)
print("\nH matrix:")
print(hmatrix)
kmatrix=hmatrix.dot(bmatrix.dot(hmatrix.T))
print("\nK = curvatures matrix:")
print(kmatrix)
factor=1.00
for i in range(n-1):
    factor=factor*1./np.sqrt(1.00+beta*kmatrix[i,i])
pfsorm=pfform*factor
BS=-norm.ppf(pfsorm)
print("\nnpfFORM =",pfform)
print("\nfactor =",factor)
print("\nnpfSORM =",pfsorm)
print("\nBS =",BS)

```