



UNIVERSIDADE FEDERAL DO MARANHÃO - UFMA
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIAS - CCET
COORDENAÇÃO DO CURSO DE ENGENHARIA CIVIL - CCEC

FLÁVIO CONCEIÇÃO FREITAS RUBIM FERREIRA

**DETERMINAÇÃO DE COEFICIENTES PARCIAIS DE SEGURANÇA EM VIGA DE
CONCRETO ARMADO PELO MÉTODO DE CONFIABILIDADE DE PRIMEIRA
ORDEM**

São Luís – MA

2026

FLÁVIO CONCEIÇÃO FREITAS RUBIM FERREIRA

**DETERMINAÇÃO DE COEFICIENTES PARCIAIS DE SEGURANÇA EM VIGA DE
CONCRETO ARMADO PELO MÉTODO DE CONFIABILIDADE DE PRIMEIRA
ORDEM**

Trabalho de Conclusão de Curso de Graduação,
apresentado à Disciplina de Trabalho de Conclusão de
Curso II, do Curso de Engenharia Civil da Universidade
Federal do Maranhão – UFMA, como requisito parcial
para a obtenção do título de Engenheiro Civil.

Orientador: Prof. Dr. Felipe Alexander Vargas
Bazán.

São Luís – MA

2026

FLÁVIO CONCEIÇÃO FREITAS RUBIM FERREIRA

**DETERMINAÇÃO DE COEFICIENTES PARCIAIS DE SEGURANÇA EM VIGA DE
CONCRETO ARMADO PELO MÉTODO DE CONFIABILIDADE DE PRIMEIRA
ORDEM**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Engenharia Civil da Universidade Federal do Maranhão, como requisito parcial exigido pelo curso de Graduação em Engenharia Civil.

Banca Examinadora:

Prof. Felipe Alexander Vargas Bazán, Dr.
Orientador
Universidade Federal do Maranhão

Prof. Dalmo Inácio Galdêz Costa, Dr.
Universidade Federal do Maranhão

Prof. Mikhail Luczynski, Msc.
Universidade Federal do Maranhão

AGRADECIMENTOS

A Deus, pela vida, pela saúde e pelas oportunidades concedidas ao longo desta caminhada acadêmica.

Aos meus pais e familiares, pelo apoio incondicional e pelo incentivo constante em todas as etapas da formação.

Ao meu orientador, Prof. Dr. Felipe Alexander Vargas Bazán, pela dedicação, pela paciência e pelos ensinamentos transmitidos durante o desenvolvimento deste trabalho, sendo fundamental para a conclusão desta pesquisa.

Aos professores do Curso de Engenharia Civil da UFMA, pela formação sólida proporcionada ao longo da graduação.

Aos colegas de curso, pelo companheirismo e pelos momentos de troca de conhecimento que tornaram esta trajetória mais rica.

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).

Diretoria Integrada de Bibliotecas/UFMA

Rubim Ferreira, Flávio Conceição Freitas.

Determinação de coeficientes parciais de segurança em viga de concreto armado pelo método de confiabilidade de primeira ordem / Flávio Conceição Freitas Rubim Ferreira. - 2026.

58 f.

Orientador(a): Felipe Alexander Vargas Bazán.

Monografia (Graduação) - Curso de Engenharia Civil, Universidade Federal do Maranhão, São Luís/ma, 2026.

1. Confiabilidade Estrutural. 2. Método Form. 3. Algoritmo Hlrf. 4. Coeficientes Parciais de Segurança. 5. Viga de Concreto Armado. I. Vargas Bazán, Felipe Alexander. II. Título.

RESUMO

A confiabilidade estrutural é uma abordagem probabilística que permite quantificar a probabilidade de falha de elementos estruturais, incorporando explicitamente as incertezas associadas às propriedades dos materiais e às ações atuantes. O presente trabalho tem como objetivo determinar os coeficientes parciais de segurança de uma viga de concreto armado, biapoiada e submetida à flexão simples, a partir da aplicação do Método de Confiabilidade de Primeira Ordem (FORM – First Order Reliability Method), associado ao algoritmo iterativo de Hasofer, Lind, Rackwitz e Fiessler (HLRF). As variáveis geométricas e a área de armadura foram tratadas como determinísticas, sendo as incertezas concentradas nas quatro variáveis de maior relevância probabilística para o problema: a resistência à compressão do concreto (f_c), a tensão de escoamento do aço (f_y), a ação permanente (g) e a ação variável (q), com distribuições de probabilidade Normal, Lognormal e Gumbel para máximos, respectivamente. O momento fletor resistente da seção foi calculado por um módulo físico baseado no equilíbrio de forças internas e na compatibilidade de deformações. O algoritmo FORM-HLRF foi implementado na linguagem de programação Python, com auxílio das bibliotecas NumPy e SciPy, determinando o ponto de projeto, o índice de confiabilidade (β), a probabilidade de falha (P_f) e os fatores de sensibilidade (α^2). O algoritmo convergiu em dez iterações, com $\beta = 3,9285$ e $P_f \approx 4,27 \times 10^{-5}$. A análise de sensibilidade evidenciou que a resistência à compressão do concreto é a variável de maior influência sobre a probabilidade de falha (64,19%), seguida pela ação variável (34,92%). A partir do ponto de projeto, foram estimados os coeficientes parciais de segurança para as quatro variáveis aleatórias.

Palavras-chave: Confiabilidade Estrutural. Método FORM. Algoritmo HLRF. Coeficientes Parciais de Segurança. Viga de Concreto Armado.

ABSTRACT

Structural reliability is a probabilistic approach that allows quantifying the failure probability of structural elements, explicitly incorporating the uncertainties associated with material properties and applied actions. This work aims to determine the partial safety factors of a simply supported reinforced concrete beam subjected to simple bending, through the application of the First Order Reliability Method (FORM), combined with the iterative Hasofer-Lind-Rackwitz-Fiessler (HLRF) algorithm. The geometric variables and reinforcement area were treated as deterministic, concentrating the uncertainties on the four variables of greatest probabilistic relevance: compressive strength of concrete (f_c), steel yield strength (f_y), permanent action (g), and variable action (q), modeled by Normal, Lognormal, and Gumbel (maxima) distributions, respectively. The resisting bending moment was calculated by a physical module based on internal force equilibrium and strain compatibility. The FORM-HLRF algorithm was implemented in Python, using NumPy and SciPy, determining the design point, reliability index (β), failure probability (P_f), and sensitivity factors (α^2). The algorithm converged in ten iterations, with $\beta = 3.9285$ and $P_f \approx 4.27 \times 10^{-5}$. Sensitivity analysis showed that the compressive strength of concrete is the most influential variable on the failure probability (64.19%), followed by the variable action (34.92%). Partial safety factors were estimated from the design point.

Keywords: Structural Reliability. FORM Method. HLRF Algorithm. Partial Safety Factors. Reinforced Concrete Beam.

SUMÁRIO

1 INTRODUÇÃO	9
1.1 JUSTIFICATIVA	12
1.2 OBJETIVOS	13
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
3 FUNDAMENTAÇÃO TEÓRICA	14
3.1 Concreto Armado e Estados Limites	14
3.2 Diagramas de Tensão-Deformação e Domínios de Deformação	14
3.3 Confiabilidade Estrutural: Conceitos Fundamentais	20
3.4 Variáveis Aleatórias e Distribuições de Probabilidade	21
3.5 O Problema Generalizado da Confiabilidade	23
3.6 Métodos de Transformação: FOSM e FORM	23
3.7 Coeficientes Parciais de Segurança	30
3.8 Exemplo Teórico Ilustrativo	31
4 METODOLOGIA	35
4.1 Viga Estudada: Modelo Físico e Geométrico	35
4.2 Módulo Físico: Momento Fletor Resistente	35
4.3 Variáveis Aleatórias e Variáveis Determinísticas	36
4.4 Algoritmo FORM-HLRF Implementado	37
4.5 Ferramenta Computacional	40
5 RESULTADOS E DISCUSSÕES	42
5.1 Convergência do Algoritmo	42
5.2 Ponto de Projeto e Índice de Confiabilidade	43
5.3 Fatores de Sensibilidade (α^2)	43
5.4 Verificação do Estado Limite no Ponto de Projeto	44
5.5 Coeficientes Parciais de Segurança Obtidos	45
5.6 Discussão Geral	46
REFERÊNCIAS	47
Apêndice a — código-fonte do algoritmo form-hlrf, quatro variáveis (python)	49
Apêndice b — código-fonte do algoritmo form-hlrf, oito variáveis (python)	55

1 INTRODUÇÃO

A determinação dos coeficientes parciais de segurança é um aspecto essencial no projeto e no dimensionamento de estruturas, visando garantir sua confiabilidade e segurança ao longo de toda a vida útil. Esses coeficientes, estabelecidos por normas técnicas como a NBR 6118 (ABNT, 2026) e a NBR 8681 (ABNT, 2025), resultam de um processo de calibração que incorpora, de forma simplificada, as incertezas associadas às propriedades dos materiais e às ações atuantes. Por se tratar de valores genéricos, calibrados para um amplo espectro de situações de projeto, tais coeficientes podem não refletir com precisão o nível de risco efetivo de um elemento estrutural específico.

Historicamente, o dimensionamento de estruturas de concreto armado evoluiu de métodos puramente determinísticos, em que todas as grandezas envolvidas (resistências, dimensões e ações) eram tratadas como valores fixos e conhecidos, para métodos semiprobabilísticos, nos quais coeficientes de segurança majoram as ações e minoram as resistências, cobrindo de forma indireta as incertezas do problema. Esse formato semiprobabilístico, embora consagrado e de aplicação relativamente simples, não evidencia ao projetista qual é o nível de risco real associado a uma determinada estrutura, tampouco quais variáveis mais contribuem para esse risco. É nessa lacuna que a confiabilidade estrutural se insere: um arcabouço matemático capaz de tratar explicitamente as incertezas do problema e de quantificar diretamente a probabilidade de falha de um elemento estrutural.

A confiabilidade estrutural constitui uma abordagem complementar que trata explicitamente as incertezas inerentes ao problema de projeto. Por meio de métodos probabilísticos é possível calcular a probabilidade de falha de um elemento estrutural e, a partir do ponto de projeto (o ponto de maior probabilidade de ocorrência de falha sobre a superfície de falha), obter coeficientes parciais de segurança fisicamente coerentes com as incertezas do problema analisado. Ao contrário dos valores normativos de aplicação geral, esses coeficientes refletem a distribuição de segurança específica do caso estudado e podem evidenciar que determinada variável é bem mais relevante para a segurança da estrutura do que as demais, o que representa informação de grande valor prático para o engenheiro projetista. No âmbito do Curso de Engenharia Civil da UFMA, essa linha de pesquisa já havia sido explorada por Coelho (2022) que, sob a mesma orientação, avaliou a confiabilidade de vigas de

concreto armado por meio da Simulação de Monte Carlo e identificou, em um estudo de caso real em São Luís/MA, situações em que os índices de confiabilidade alvo recomendados pela literatura não eram atingidos para determinados modos de falha. O presente trabalho dá continuidade a essa linha de investigação, adotando, contudo, o método FORM associado ao algoritmo HLRF como abordagem alternativa de transformação, em vez da simulação direta.

Dentre os métodos disponíveis para essa análise, destaca-se o Método de Confiabilidade de Primeira Ordem (FORM – First Order Reliability Method), associado ao algoritmo iterativo de Hasofer, Lind, Rackwitz e Fiessler (HLRF). Fundamentado na transformação de Hasofer e Lind (1974), o FORM converte o espaço das variáveis aleatórias originais em um espaço normal padrão, no qual o índice de confiabilidade (β) corresponde geometricamente à menor distância entre a origem e a superfície de falha. Ao contrário do método FOSM (First Order Second Moment), que considera todas as variáveis como normais, o FORM permite incorporar distribuições de probabilidade reais (Normal, Lognormal, Gumbel para máximos, entre outras), o que produz resultados mais acurados para problemas de engenharia estrutural. Essa característica torna o método especialmente adequado para a análise de elementos de concreto armado, nos quais grandezas como a resistência à compressão do concreto, a tensão de escoamento do aço e as ações variáveis apresentam distribuições de probabilidade reconhecidamente distintas da distribuição normal.

O presente trabalho aplica o método FORM, implementado computacionalmente na linguagem Python, à análise de confiabilidade de uma viga de concreto armado biapoiada, submetida à flexão simples. Antes de apresentar o estudo principal, é discutido, na fundamentação teórica, um exemplo teórico ilustrativo em que todas as oito grandezas relevantes ao problema (largura, altura total, altura útil, área de armadura, resistência do concreto, tensão de escoamento do aço e as ações permanente e variável) são tratadas como variáveis aleatórias, permitindo observar, por meio dos fatores de sensibilidade do método FORM, qual a contribuição relativa de cada uma dessas grandezas para a probabilidade de falha da estrutura.

Essa constatação preliminar é amplamente respaldada pela literatura técnica consultada (BECK, 2019; MELCHERS; BECK, 2018), de modo que o estudo principal deste trabalho concentra as incertezas nas quatro variáveis aleatórias de maior relevância probabilística para o problema: a resistência à compressão do concreto (f_c), a tensão de escoamento do aço (f_y), a ação permanente (g) e a ação variável (q),

enquanto as dimensões geométricas da seção transversal e a área de armadura são tratadas como determinísticas, fixadas em seus valores médios. Essa simplificação metodológica, devidamente justificada e quantificada na Seção 4.8 deste trabalho, não compromete a qualidade dos resultados obtidos, permitindo, ao mesmo tempo, uma análise mais direta e objetiva do problema central de pesquisa: a determinação dos coeficientes parciais de segurança da viga estudada.

A partir da função de estado limite definida pela diferença entre o momento fletor resistente e o momento fletor solicitante, o algoritmo FORM-HLRF determina iterativamente o ponto de projeto, o índice de confiabilidade, a probabilidade de falha e os fatores de sensibilidade de cada variável aleatória, possibilitando, ao final, a estimativa dos coeficientes parciais de segurança da estrutura analisada.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta a justificativa do estudo, e a Seção 3, os objetivos geral e específicos. A Seção 4 traz a fundamentação teórica, abrangendo os conceitos de concreto armado e estados limites, os métodos de transformação FOSM e FORM, as distribuições de probabilidade empregadas e o exemplo teórico ilustrativo com oito variáveis aleatórias. A Seção 5 descreve a metodologia adotada, incluindo o modelo físico da viga estudada e o algoritmo computacional implementado. Por fim, a Seção 6 apresenta e discute os resultados obtidos para o problema com quatro variáveis aleatórias, seguida das referências bibliográficas e dos apêndices contendo o código-fonte desenvolvido.

1.1 JUSTIFICATIVA

Em um projeto estrutural, o objetivo central é garantir que a estrutura cumpra, de forma satisfatória, todas as funções exigidas ao longo de sua vida útil: resistir aos carregamentos impostos, proteger-se das intempéries e manter desempenho adequado. Segurança e desempenho são os pilares técnicos fundamentais do projeto de engenharia. A confiabilidade estrutural tem assumido papel cada vez mais relevante nesse contexto, à medida que diversas normas internacionais passaram a embasar seus coeficientes de segurança em fundamentos probabilísticos, superando os coeficientes puramente empíricos determinados pela experiência acumulada de projetistas (MRÁZIK; KRIZMA, 1997).

No desenvolvimento de projetos de engenharia, as questões de segurança estrutural decorrem das incertezas inerentes às variáveis do problema, sejam elas relativas às propriedades dos materiais, às dimensões geométricas dos elementos ou à magnitude e frequência das ações. Quantificar essas incertezas é fundamental para uma avaliação rigorosa da confiabilidade. Normas como o Eurocode (2023) e as brasileiras NBR 6118 (ABNT, 2023) e NBR 8681 (ABNT, 2025) avaliam a segurança por meio de coeficientes parciais de segurança, calibrados de forma semiprobabilística para um conjunto representativo de situações de projeto.

Este trabalho justifica-se pela relevância de investigar, de forma direta e transparente, como os métodos de confiabilidade estrutural podem ser aplicados a um elemento específico de concreto armado, gerando coeficientes parciais de segurança coerentes com as incertezas reais do problema. A implementação computacional do algoritmo FORM-HLRF em Python constitui, ainda, uma ferramenta didática e replicável, de uso potencial tanto no ensino quanto em estudos futuros de calibração de normas no contexto brasileiro.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Determinar os coeficientes parciais de segurança de uma viga de concreto armado biapoiada, submetida à flexão simples, por meio da aplicação do Método de Confiabilidade de Primeira Ordem.

1.2.2 Objetivos Específicos

- Implementar computacionalmente o algoritmo FORM-HLRF em Python, com distribuições Normal (f_c , g), Lognormal (f_y) e Gumbel para máximos (q);
- Implementar computacionalmente os métodos iterativos para o cálculo do momento solicitante e gradiente numérico;
- Obter o ponto de projeto, o índice de confiabilidade (β) e a probabilidade de falha (P_f) associados ao estado limite último de flexão;
- Determinar os coeficientes parciais de segurança para as variáveis aleatórias a partir do ponto de projeto obtido pelo FORM;

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Concreto Armado e Estados Limites

O concreto armado é um material estrutural composto, essencialmente, por dois materiais distintos e complementares: o concreto e o aço. Enquanto o concreto apresenta boa resistência aos esforços de compressão, o aço é eficiente na resistência aos esforços de tração, de modo que ambos trabalham solidariamente para resistir às solicitações impostas ao elemento estrutural. Segundo Chust (2014), o concreto armado resulta da associação entre o concreto simples e uma armadura passiva convenientemente posicionada. Essa associação somente é possível devido à aderência entre o aço e o concreto (ARAÚJO, 2023): as deformações das barras de aço são praticamente iguais às do concreto que as envolve, e os esforços de tração, após a fissuração do concreto na região tracionada, passam a ser absorvidos integralmente pela armadura.

De acordo com a ABNT NBR 6118 (2023), o dimensionamento e a verificação de estruturas de concreto armado devem considerar dois grupos de estados limites: o Estado Limite Último (ELU) e o Estado Limite de Serviço (ELS). O ELU caracteriza-se pelo colapso ou pelo esgotamento da capacidade resistente do elemento estrutural, enquanto o ELS está associado ao conforto, à durabilidade e à boa utilização da estrutura. O presente trabalho concentra-se no ELU de flexão simples, verificando se o momento fletor resistente da seção transversal é superior ao momento fletor solicitante pela combinação de ações atuantes. Segundo Araújo (2023), a segurança estrutural pode ser verificada pela relação entre esforços resistentes e solicitantes:

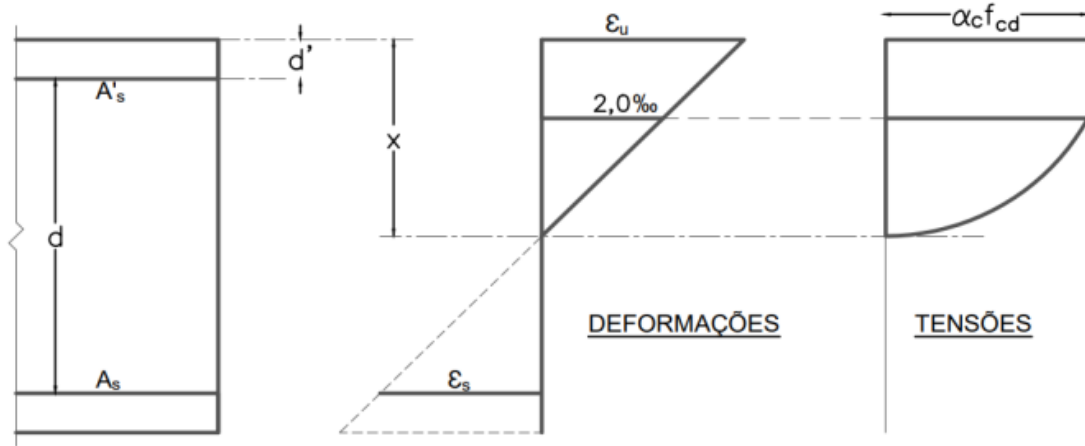
$$R_d \geq S_d \quad (1)$$

em que R_d representa o esforço resistente de cálculo e S_d , o esforço solicitante. No contexto probabilístico adotado neste trabalho, essa margem é tratada pela função de estado limite $g(X) = M_r - M_s$, discutida na Seção 4.3.

2.2 Diagramas de Tensão-Deformação e Domínios de Deformação

Segundo Fusco (2017), as distribuições de tensões de compressão em peças sujeitas a solicitações normais podem ser representadas pelo diagrama parabólico-retangular, conforme ilustrado na Figura 1.

Figura 1 – Diagrama parábola-retângulo.



Fonte: Fusco, 2017.

A distribuição de tensões, como se pode perceber, consiste em uma parábola de segundo grau prolongada por um retângulo. A parábola tem seu vértice localizado no ponto em que a deformação da peça atinge o valor de 2 ‰, enquanto o retângulo se estende até a face superior, onde a deformação máxima da peça (ϵ_u) ocorre. Além disso, é possível deduzir que a ordenada máxima correspondente à parte retangular da distribuição de tensões representa a tensão de compressão, que é definida pela seguinte equação:

$$\sigma_{cd} = \alpha_c f_{cd} \quad (2)$$

Onde:

f_{cd} Resistência à compressão de cálculo do concreto;

α_{cd} Tensão de cálculo no concreto;

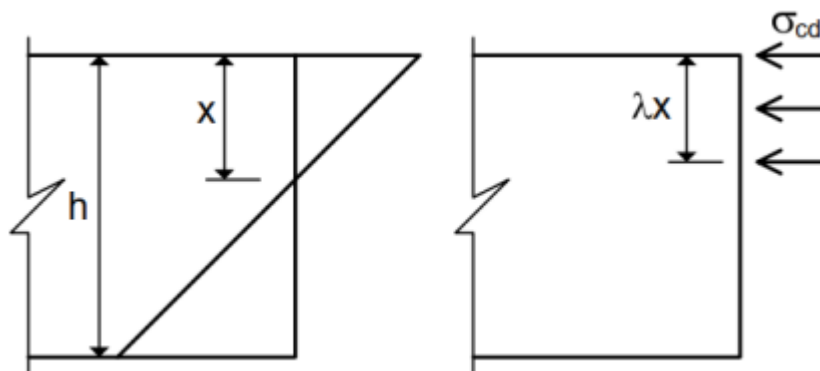
α_c Coeficiente com valor definido pela equação a seguir:

$$\alpha_c = \begin{cases} 0,85 & , \quad f_{ck} \leq 50\text{MPa} \\ 0,85 \left[1 - \frac{(f_{ck}-50)}{200} \right] & , \quad f_{ck} > 50\text{MPa} \end{cases} \quad (3)$$

O diagrama parabólico-retangular pode ser ainda simplificado para um diagrama retangular, conforme ilustrado na Figura 2. Essa simplificação gera

resultados praticamente idênticos aos obtidos pelo primeiro diagrama (ARAÚJO, 2023).

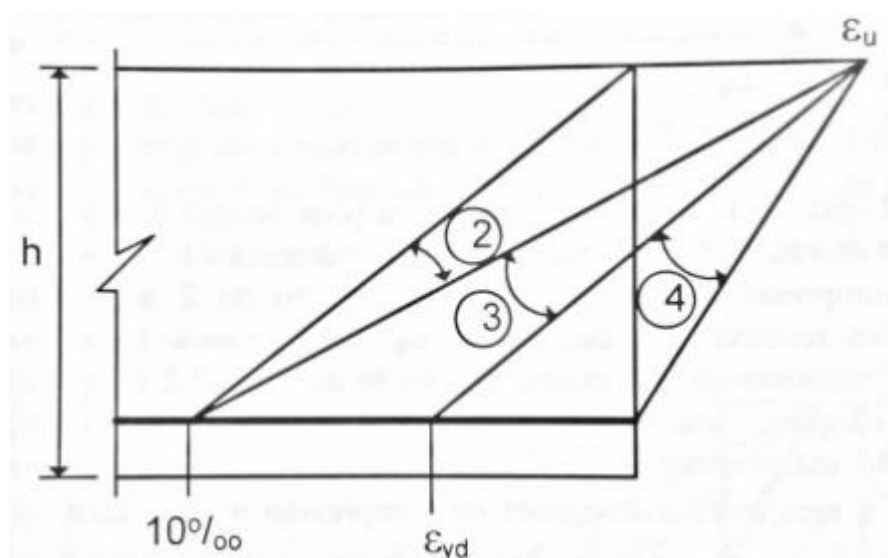
Figura 2 – Diagrama retangular de distribuição de tensões.



Fonte: ARAÚJO, 2023 (adaptado)

Segundo Araújo (2023), com a simplificação, é possível admitir que a tensão no concreto (σ_{cd}) seja correspondente a 85% da resistência de cálculo do concreto (f_{cd}), desde que a resistência característica do concreto (f_{ck}) seja igual ou inferior a 50 MPa, ou então, para um f_{ck} maior do que 50 MPa, necessita-se de um cálculo diferenciado, como foi visto na Equação 3. Dessa forma, o presente trabalho levará em conta o diagrama retangular de tensões para a realização dos cálculos associados à confiabilidade estrutural.

Figura 3 – Domínios de deformação para flexão simples.



Fonte: Araújo, 2023 (adaptado)

O domínio de deformação 2 como mostrado na Figura 3, ocorre quando a armadura de uma peça submetida à flexão tem seu máximo alongamento atingido ($\epsilon_s = 10\%$), sem que haja ruptura à compressão do concreto. Estruturas que rompem nesse domínio possuem uma baixa taxa de armadura, ocasionando o rompimento por deformação excessiva da armadura (MONTTOYA *et al*, 2009; ARAÚJO, 2023).

O domínio de deformação 3 (ARAÚJO, 2023; FUSCO, 2017) é dado quando o escoamento da armadura, de uma peça sujeita à flexão, ocorre simultaneamente à ruptura por compressão do concreto, ou seja, quando a deformação da armadura é maior que a sua deformação de escoamento ($\epsilon_s \geq \epsilon_{yd}$) e a deformação do concreto é maior que a sua deformação máxima admissível ($\epsilon_c \geq \epsilon_u$). Estruturas que rompem nesse domínio são normalmente armadas, ou seja, têm uma taxa de armadura adequada em relação a sua seção.

O domínio de deformação 4 ocorre quando o concreto, de uma peça submetida à flexão, rompe devido à compressão ($\epsilon_c \geq \epsilon_u$) sem que haja escoamento da armadura tracionada. Estruturas que rompem nesse domínio têm uma taxa excessiva de armadura, dessa forma acontece o rompimento do concreto antes que a armadura possa escoar (MONTTOYA *et al*, 2009; ARAÚJO, 2023).

Para o cálculo de estruturas submetidas à flexão simples, levando em conta unicamente o requisito de segurança, considera-se apenas o Estado Limite Último para os devidos cálculos, mais especificamente nos domínios de deformação 2, 3 e 4. Com base nessa delimitação e considerando que as vigas estudadas terão taxas normais ou baixas de armadura, um método de cálculo para vigas fletidas nos domínios de deformação 2 e 3, levantando informações sobre a resistência dos materiais, a combinação das cargas atuantes, os coeficientes de ponderação, bem como as respectivas equações necessárias.

Naturalmente, os materiais que compõem uma estrutura possuem diferentes características de resistências a esforços que atuam sobre eles. Quando uma estrutura é composta por esses materiais, as suas características de resistência passam a ser definidas pela forma como eles trabalham em conjunto, de maneira que, se as solicitações que atuam sobre essa estrutura forem menores do que essas “resistências conjuntas”, a estrutura poderá ser considerada segura. Essa afirmação pode ser verificada na seguinte equação, prevista na ABNT NBR 6118 (2023) e

amplamente divulgada na literatura, por exemplo, nos trabalhos de Araújo (2023), Fusco (2017) e Santos *et al.* (2014).

$$R_d \geq S_d \quad (4)$$

Onde:

R_d Esforços resistentes do estado limite último da estrutura;

S_d Esforços solicitantes de cálculo.

O concreto é o material mais comum da construção civil e tem grande importância em estruturas de concreto armado devido a sua capacidade de resistência à compressão. A ABNT NBR 6118 (2023) prevê a adoção da resistência à compressão de cálculo (f_{cd}) com base na sua resistência característica (f_{ck}) aos 28 dias de cura, pela Equação 5. Essa resistência característica é obtida por meio de ensaios de corpos de prova regulamentados pela ABNT NBR 5738 (2015) (moldagem) e pela ABNT NBR 5739 (2018) (rompimento). A classificação do concreto quanto a sua resistência característica pode ser vista na Tabela 1.

$$f_{cd} = \frac{f_{ck}}{\gamma_c} \quad (5)$$

onde:

f_{cd} Resistência à compressão de cálculo do concreto;

f_{ck} Resistência característica à compressão do concreto;

γ_c Coeficiente de ponderação de resistência do concreto;

Tabela 1 – Classes de resistência do concreto.

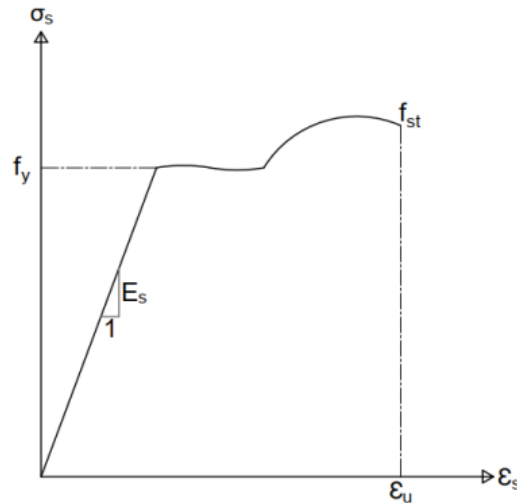
Classe de resistência	C20	C25	C30	C35	C40	C45	C50	C60	C70	C80	C90
f_{ck} (MPa)	≥20	≥25	≥30	≥35	≥40	≥45	≥50	≥60	≥70	≥80	≥90

Fonte: NBR 6118/2023 (adaptado)

O aço é um material essencial na produção de estruturas de concreto armado devido a sua grande capacidade de resistência a esforços de tração, o aço mais comum utilizado e considerado dos cálculos de dimensionamento e verificação de

estruturas é o aço Classe A. Segundo Fusco (2017), o aço classe A é caracterizado pela sua linearidade até atingir o seu limite de escoamento, dentro do diagrama tensão-deformação (Figura 4). Ainda segundo Fusco (2017), o módulo de deformação (E_s) considerado é de 210 GPa.

Figura 4 – Diagrama tensão-deformação de tração.



Fonte: Araújo, 2023 (adaptado).

Podemos definir ainda as resistências características de escoamento (f_{yk}) mais usuais (Tabela 2). As mesmas são utilizadas para se definir as resistências a escoamento de cálculo (f_{yd}), conforme a equação 6. Por fim, as bitolas das barras de aço CA-50 usualmente utilizadas, bem como suas áreas de seção, podem ser vistas na Tabela 3.

$$f_{yd} = \frac{f_{yk}}{\gamma_s} \quad (6)$$

onde:

f_{yd} Resistência à tração de cálculo do aço;

f_{yk} Resistência característica à tração do aço;

γ_s Coeficiente de ponderação de resistência do aço;

Tabela 2 – Classes de resistência do aço

Classe de resistência	CA-25	CA-50	CA-60
f_{yk} (Mpa)	250	500	600

Fonte: Araújo, 2023 (adaptado).

Tabela 3 – Classes de resistência do concreto

$\varnothing$ (mm)	8.0	10.0	12.5	16.0	20.0	22.5	25.0	32.0
$A\varnothing$ (cm ²)	0,5	0,8	1,2	2,0	3,1	4,0	4,9	8,0

Fonte: NBR 6118/2023 (adaptado)

2.3 Confiabilidade Estrutural: Conceitos Fundamentais

A confiabilidade estrutural pode ser entendida como a extensão probabilística dos métodos determinísticos de projeto. Enquanto nesses métodos todas as variáveis são tratadas como exatas, a abordagem probabilística reconhece que propriedades dos materiais, dimensões geométricas e ações atuantes apresentam incertezas de natureza física, estatística e de modelo, que devem ser incorporadas explicitamente à análise (MELCHERS; BECK, 2018). Segundo Beck (2019), o termo confiabilidade estrutural refere-se ao grau de confiança de que um sistema não falhe dentro de determinado período de tempo, respeitando as condições de operação previstas.

4.3.1 O problema fundamental R–S

O problema fundamental da confiabilidade estrutural envolve duas variáveis aleatórias: a resistência (R) e a solicitação (S). A diferença entre essas variáveis define a margem de segurança (M):

$$M = R - S \quad (7)$$

A função de falha $g(R,S) = R - S$ é denominada função de estado limite: valores de $g(X) \leq 0$ indicam falha; $g(X) > 0$ indica segurança; $g(X) = 0$ corresponde à superfície de falha. Coelho (2022) descreve esse mesmo problema básico de confiabilidade a partir da separação entre os domínios de falha e de segurança, formalismo também adotado neste trabalho na definição do domínio de falha D_f , apresentada na Seção 4.5. A probabilidade de falha é:

$$P_f = P\{R - S \leq 0\} = P\{M \leq 0\} \quad (8)$$

Quando R e S são variáveis normais independentes, a probabilidade de falha pode ser calculada analiticamente a partir da média e do desvio padrão da margem de segurança, sendo $\Phi(\cdot)$ a função de distribuição cumulativa normal padrão e β o índice de confiabilidade:

$$P_f = \Phi(-\beta) \quad (9)$$

$$\beta = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \quad (10)$$

A transformação de Hasofer e Lind (1974) converte as variáveis R e S com médias e desvios padrão quaisquer em variáveis normais padrão Y1 e Y2 (médias nulas, desvios padrão unitários). No espaço transformado, β corresponde geometricamente à menor distância entre a origem e a superfície de falha $g(y_1, y_2) = 0$. O ponto da superfície de falha mais próximo da origem é o ponto de projeto ou ponto mais provável de falha, e as coordenadas nesse espaço satisfazem:

$$(y_1^*, y_2^*) = -(\alpha_1, \alpha_2) \cdot \beta \quad (11)$$

em que os cossenos diretores (α_i) quantificam a influência de cada variável sobre o índice de confiabilidade base dos fatores de sensibilidade α^2 , empregados ao longo de todo este trabalho, inclusive no exemplo teórico de múltiplas variáveis apresentado na Seção 4.8.

2.4 Variáveis Aleatórias e Distribuições de Probabilidade

Uma variável aleatória pode ser definida como uma função que associa um número a cada resultado de um espaço amostral de um experimento aleatório (BECK, 2019; MONTGOMERY; RUNGER, 2024). Os momentos estatísticos fundamentais de uma variável aleatória contínua média (μ), variância (σ^2), desvio padrão (σ) e coeficiente de variação (C.V.) são dados, respectivamente, por:

$$E[X] = \mu_X = \int x \cdot f_X(x) dx \quad (12)$$

$$Var[X] = \sigma^2 = \int (x - \mu_X)^2 \cdot f_X(x) dx \quad (13)$$

$$\sigma = \sqrt{Var[X]} \quad (14)$$

$$C.V. = \frac{\sigma}{\mu} \quad (15)$$

O coeficiente de variação (C.V.) é especialmente útil para comparar a dispersão relativa de variáveis com diferentes magnitudes e unidades, sendo amplamente utilizado na caracterização estatística de variáveis em estudos de confiabilidade estrutural. As três distribuições empregadas neste trabalho são apresentadas a seguir.

2.4.1 Distribuição Normal $N(\mu, \sigma)$

A distribuição normal é o modelo probabilístico mais utilizado na engenharia, sendo parametrizada pela média (μ) e pelo desvio padrão (σ). Sua função densidade de probabilidades é dada por:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \quad (16)$$

Neste trabalho, são modeladas como variáveis normais a resistência à compressão do concreto (f_c) e a ação permanente (g). Para a padronização de uma variável $X \sim N(\mu, \sigma)$, aplica-se $Y = (X - \mu)/\sigma$, obtendo-se $Y \sim N(0,1)$, o que possibilita o uso das tabelas e funções da normal padrão.

2.4.2 Distribuição Lognormal $LN(\lambda, \xi)$

Uma variável X tem distribuição lognormal quando seu logaritmo natural, $Y = \ln(X)$, tem distribuição normal. Essa distribuição é adequada para grandezas estritamente positivas, como a tensão de escoamento do aço (f_y). Os parâmetros λ (média de $\ln X$) e ξ (desvio padrão de $\ln X$) relacionam-se à média e ao coeficiente de variação de X por:

$$\xi = \sqrt{\ln(1 + C.V.^2)} \quad (17)$$

$$\lambda = \ln(\mu) - \frac{1}{2}\xi^2 \quad (18)$$

2.4.3 Distribuição de Gumbel para máximos $EVI(un, \beta)$

A distribuição de Gumbel para máximos (Tipo I) é amplamente utilizada para representar variáveis associadas a valores extremos ao longo de um período de referência, como a ação variável (q). Coelho (2022) emprega a mesma distribuição para modelar a ação accidental em vigas de concreto armado analisadas por Simulação de Monte Carlo, destacando sua forma exponencial dupla como adequada à representação de máximos. Seus parâmetros un (moda) e β (parâmetro de escala) relacionam-se à média e ao desvio padrão por:

$$\beta_G = \frac{\pi}{\sqrt{6}} \cdot \frac{1}{\sigma} \quad (19)$$

$$u_n = \mu - \frac{\gamma}{\beta_G} \quad (20)$$

em que $\gamma \approx 0,577216$ é a constante de Euler-Mascheroni. As funções cumulativa e densidade correspondentes são:

$$F_Y(y) = \exp[-\exp(-\beta_G(y - u_n))] \quad (21)$$

$$f_Y(y) = \beta_G \cdot \exp[-\beta_G(y - u_n) - \exp(-\beta_G(y - u_n))] \quad (22)$$

2.5 O Problema Generalizado da Confiabilidade Estrutural

Como visto anteriormente, nem sempre podemos reduzir as variáveis da equação de estado limite para apenas duas. Nos casos em que não é possível realizar tal procedimento, o que corresponde à imensa maioria de casos na prática, é necessária uma generalização da formulação.

Em uma primeira análise, ocorre a definição das variáveis envolvidas no problema de confiabilidade. De modo geral, são utilizadas as variáveis envolvidas no dimensionamento da estrutura: dimensões geométricas, pesos específicos, resistência dos materiais, etc. É conveniente que as variáveis aleatórias escolhidas sejam independentes, nos casos em que isso não é possível, utiliza-se uma matriz de correlação (MELCHERS & BECK, 2018).

2.6 Métodos de Transformação: FOSM e FORM

A resolução de problemas de confiabilidade estrutural independentes do tempo requer a determinação (ou uma aproximação) da função conjunta de densidade de probabilidades, $f_X(x)$, além de aproximações para o domínio de integração. Diversos métodos de solução, como os métodos de transformação FOSM (*First Order Second Moment*) e o FORM (*First Order Reliability Method*), utilizam diferentes formas de aproximar a função e o domínio de integração. Para este estudo, não iremos abordar a simulação de Monte Carlo (Beck, 2019).

Na prática, não dispomos de observações que permitam determinar diretamente a função de densidade conjunta $f_X(x)$. Isso significa que essa função deve ser construída com base nas

informações disponíveis, que, na maioria dos casos, se restringem às funções de distribuição marginais de cada variável, e, em alguns casos, os coeficientes de correlação entre pares de variáveis aleatórias.

No método de Primeira Ordem e Segundo Momento (FOSM - *First Order Second Moment*), a equação de estado limite é aproximada por uma função linear, e os dados estatísticos necessários para a construção de $f_X(x)$ se limitam aos momentos até a segunda ordem (média e covariância). Representar as variáveis aleatórias com seus momentos de primeira e segunda ordem equivale a considerá-las com distribuição normal. Essa suposição impõe restrições significativas na solução de problemas práticos. No entanto, a solução conhecida como FOSM serve como base teórica para o desenvolvimento de métodos mais completos e com maior aplicabilidade prática. O FOSM é o fundamento sobre o qual os demais métodos de transformação são construídos (Beck, 2019).

No método de confiabilidade de primeira ordem (FORM - *First Order Reliability Method*), toda a informação estatística relacionada às variáveis aleatórias do problema é considerada. Isso inclui tanto distribuições marginais não normais quanto os coeficientes de correlação entre os pares de variáveis. O domínio de integração continua sendo aproximado por uma função linear. Já no método de confiabilidade de segunda ordem (SORM - *Second Order Reliability Method*), a mesma informação estatística é utilizada para construir a função conjunta de densidades, porém a equação de estado limite é aproximada por uma equação quadrática, entretanto não iremos abordar este método.

Os métodos de transformação (FOSM e FORM) possuem como base a transformação de Hasofer e Lind, aplicada a cada uma das variáveis aleatórias individualmente (Beck, 2019):

$$Y_i = \frac{X_i - \mu_{xi}}{\sigma_{xi}}, i = 1, \dots, n. \quad (23)$$

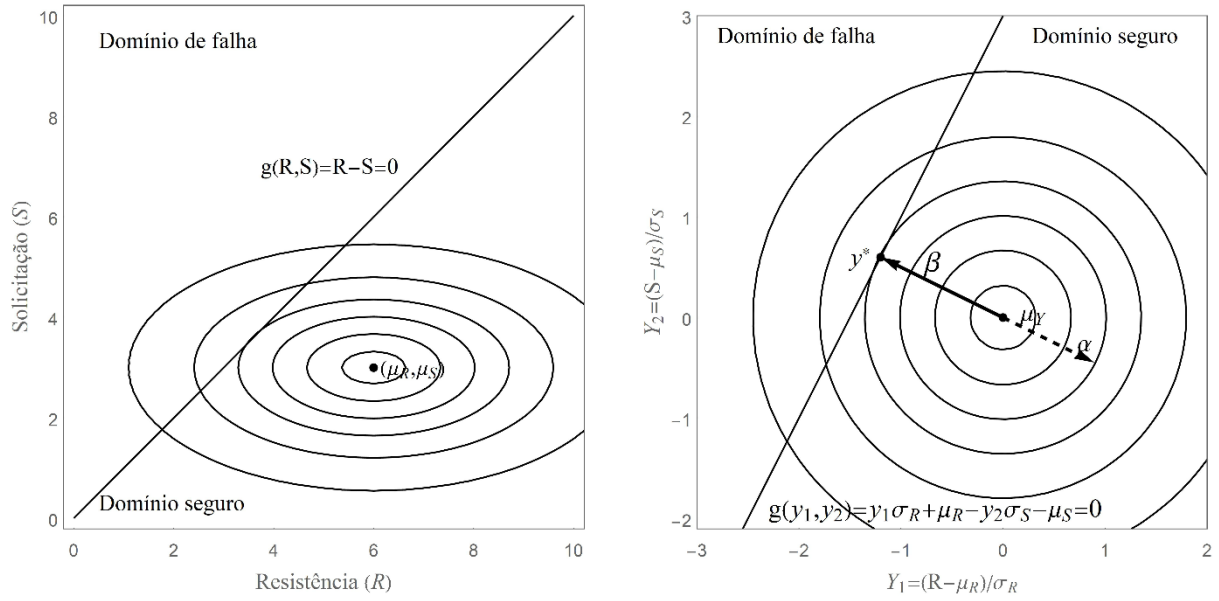
A equação transforma um conjunto de vetores em variáveis normais com médias e desvios-padrão quaisquer, em um conjunto de variáveis aleatórias normais padrão, com média nula e desvio padrão unitário.

Uma interpretação do ponto de vista geométrico do índice de confiabilidade β é obtida aplicando a transformação de Hasofer e Lind às variáveis R e S do problema fundamental da confiabilidade, resolvendo o seguinte problema:

Encontrar o ponto (y_1^*, y_2^*) , que minimiza $d^2 = y_1^2 + y_2^2$, sujeito a $g(y_1, y_2) = 0$ (24)

Substituindo a margem de segurança M pela equação de estado limite G, e convertendo as variáveis R e S para Y1 e Y2 por meio da Eq. (23), obtemos uma nova expressão para a equação de estado limite no espaço normalizado:

Figura 5 - Transformação das variáveis R e S em variáveis normais padrão.



Fonte: Beck, 2019.

$$G = g(R, S) = R - S \quad (25)$$

$$g(y_1, y_2) = g(r(y_1), s(y_2)) = y_1\sigma_R + \mu_R - y_2\sigma_S - \mu_S \quad (26)$$

Fazendo $g(y_1, y_2) = 0$ e isolando y_2 na Eq. (26), temos (Beck, 2019):

$$y_2 = \frac{y_1\sigma_R + \mu_R - \mu_S}{\sigma_S} \quad (27)$$

O quadrado da distância entre um ponto qualquer do plano (y_1, y_2) e a origem é $d^2 = y_1^2 + y_2^2$. Fazendo a derivada em relação a y_1 e igualando a mesma a 0, temos a condição mínima, porém não suficiente:

$$2y_1 + 2y_2 \frac{\partial y_2}{\partial y_1} = 0, \quad (28)$$

$$2y_1 + 2y_2 \frac{\sigma_R}{\sigma_S} = 0. \quad (29)$$

Utilizando a Eq. (28), conseguimos obter a coordenada y_1^* do ponto sobre a equação $g(y_1, y_2) = 0$ que mais se aproxima da origem:

$$y_1^* = y_2 \frac{\sigma_R}{\sigma_S} = -\frac{\sigma_R(\mu_R - \mu_S)}{\sigma_R^2 + \sigma_S^2}. \quad (30)$$

Tomando como base novamente $d^2 = y_1^2 + y_2^2$, e fazendo a derivada em relação a y_2 e igualando a zero, e utilizando também a Eq. (29), temos a coordenada y_2^* do ponto sobre $g(y_1, y_2) = 0$ mais próximo da origem:

$$y_2^* = y_1 \frac{\sigma_R}{\sigma_S} = \frac{\sigma_S(\mu_R - \mu_S)}{\sigma_R^2 + \sigma_S^2}. \quad (31)$$

Resumindo as duas últimas expressões, temos as coordenadas do ponto sobre $g(y_1, y_2) = 0$ mais próximo da origem:

$$(y_1^*, y_2^*) = \frac{(\mu_R - \mu_S)}{\sigma_R^2 + \sigma_S^2} (-\sigma_R, \sigma_S). \quad (32)$$

O ponto y^* sobre a superfície de falha mais próximo da origem é também o ponto sobre o domínio de falha com maior valor de probabilidade de ocorrência, ou seja, o ponto sobre o domínio de falha com maior valor da função densidade conjunta de probabilidades y_1, y_2 . Este ponto é normalmente chamado de ponto de projeto ou ponto mais provável de falha.

Podemos também substituir o resultado em $d^2 = y_1^2 + y_2^2$ e tomando a raiz quadrada, obtendo uma expressão para a mínima distância entre a equação $g(y_1, y_2) = 0$ e a origem:

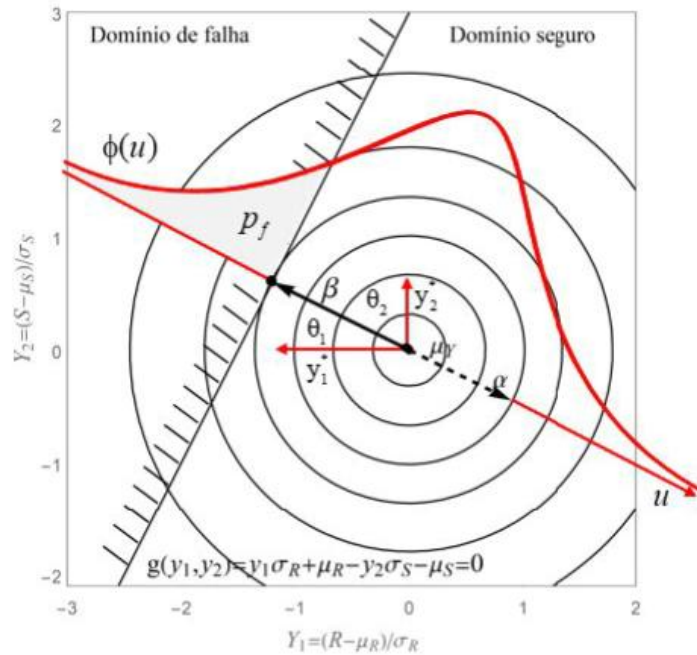
$$d_{min} = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \quad (33)$$

Desta forma, podemos generalizar este resultado, constatando que o índice de confiabilidade corresponde à mínima distância entre a equação limite e a origem do espaço normal padrão (Beck, 2019):

$$\beta = d_{min} = \frac{\mu_R - \mu_S}{\sqrt{\sigma_R^2 + \sigma_S^2}} \quad (34)$$

A probabilidade de falha P_f corresponde ao volume de probabilidades na região atrás do hiper-plano, área hachurada na Figura 6.

Figura 06 – Relação entre a P_f e o índice de confiabilidade β



Fonte: Beck, 2019.

E pode ser calculada como:

$$P_f = \Phi(-\beta) \quad (35)$$

Portanto, estes resultados mostram que o índice de confiabilidade β pode ser visto como uma medida geométrica da probabilidade de falha, e permite formular a solução de problemas multi-dimensionais de confiabilidade independente do tempo como problemas de otimização.

Após a determinação do ponto de projeto, que é obtido por meio de métodos de confiabilidade estrutural, como o FORM (*First Order Reliability Method*), os coeficientes parciais de segurança são determinados para a verificação real das condições da estrutura.

Os coeficientes parciais de segurança (γ) são definidos com base no índice de confiabilidade (β), que é calculado a partir do ponto de projeto. O índice de confiabilidade representa a distância, em termos probabilísticos, entre o ponto de projeto e a superfície de falha no espaço normalizado. Quanto maior o valor de (β), menor é a probabilidade de falha.

A relação básica entre o ponto de projeto e os coeficientes parciais de segurança é dada por:

$$\gamma_R = \frac{R_d}{R_K} \quad (36)$$

$$\gamma_S = \frac{S_d}{S_K} \quad (37)$$

onde:

R_d e S_d são os valores de resistência e de carregamento de projeto, respectivamente;

R_K e S_K são os valores característicos de resistência e de carregamento;

γ_R é o coeficiente parcial de segurança para a resistência, e γ_S é o coeficiente parcial de segurança para o carregamento.

4.6.3 Algoritmo HLRF

O algoritmo HLRF (Hasofer, Lind, Rackwitz e Fiessler) é o procedimento iterativo de busca do ponto de projeto, aplicado em ambos os métodos FOSM e FORM. Partindo do vetor de médias μ como ponto inicial ($x_k=0 = \mu$), o algoritmo executa, a cada iteração k , os seguintes passos:

- Passo 1 — Determinação da matriz de decomposição de Cholesky ($R_z = I \rightarrow L = I$) e das matrizes jacobianas;
- Passo 2 — Atualização das normais equivalentes no ponto corrente x_k (Eqs. 32–34), calculando as jacobianas J_{xy} e J_{yx} ;

- Passo 3 — Transformação do ponto corrente para o espaço normal padrão: $y_k = J_{yx} \cdot (x_k - \mu_{neq})$;
- Passo 4 — Avaliação da função de estado limite: $g(x_k)$;
- Passo 5 — Cálculo do gradiente de $g(x)$ no espaço original por diferenças finitas progressivas e transformação para o espaço Y: $\nabla g(y) = J_{xy}^T \cdot \nabla g(x)$; determinação dos fatores de sensibilidade: $\alpha = -\nabla g(y) / \|\nabla g(y)\|$;
- Passo 6 — Cálculo do novo ponto no espaço Y pelo algoritmo HLRF: $y_{k+1} = [(\nabla g(y_k) \cdot y_k - g(x_k)) / \|\nabla g(y_k)\|^2] \cdot \nabla g(y_k)$;
- Passo 7 — Avaliação do índice de confiabilidade: $\beta_{k+1} = \|y_{k+1}\|$;
- Passo 8 — Retransformação para o espaço original: $x_{k+1} = J_{xy} \cdot y_{k+1} + \mu_{neq}$;
- Passo 9 — Verificação do critério de convergência: $|\beta_{k+1} - \beta_k| \leq \text{TOL}$; se satisfeito, encerra; caso contrário, retorna ao Passo 2.

Ao final da convergência, a probabilidade de falha é estimada por:

$$P_f = \Phi(-\beta) \quad (38)$$

2.7 Coeficientes Parciais de Segurança

Após a determinação do ponto de projeto x^* pelo método FORM, os coeficientes parciais de segurança são obtidos a partir da relação entre o valor de projeto (valor da variável no ponto de projeto) e o valor característico correspondente. Para variáveis de resistência, o valor característico é definido como o quantil de 5% da distribuição de probabilidades:

$$X_{k,res} = F_X^{-1}(0,05) \quad (39)$$

$$\gamma_R = \frac{X_k}{x^*} \quad (40)$$

Para a ação permanente (g), de baixa variabilidade, o valor característico é tomado como a média da distribuição, em conformidade com a NBR 8681 (ABNT, 2025). Para a ação variável (q), modelada pela distribuição de Gumbel para máximos, adota-se o quantil de 95%:

$$q_k = F_Q^{-1}(0,95) \quad (41)$$

$$\gamma_S = \frac{x^*}{X_{k,sol}} \quad (42)$$

Os coeficientes parciais obtidos pelo FORM para um caso específico refletem a distribuição real da segurança naquele problema, podendo divergir dos valores genéricos normalizados, divergência que constitui informação relevante para a calibração racional dos coeficientes de segurança em normas técnicas.

Complementarmente aos coeficientes parciais, o JCSS (2001), em seu Probabilistic Model Code — Part 1: Basis of Design, estabelece valores-alvo tentativos para o índice de confiabilidade β , referidos a um período de referência de um ano, em função da classe de consequência da falha (menor, moderada ou grande) e do custo relativo das medidas de segurança adotadas (grande, normal ou pequeno). A Tabela 4 reproduz esses valores-alvo, também expressos em termos da probabilidade de falha (Pf) correspondente.

Tabela 4 — Índices de confiabilidade-alvo β (e Pf associada) para período de referência de 1 ano, segundo o JCSS (2001).

Custo relativo da medida de segurança	Consequências menores	Consequências moderadas	Consequências grandes
Grande (A)	$\beta = 3,1 (10^{-3})$	$\beta = 3,3 (5 \cdot 10^{-4})$	$\beta = 3,7 (10^{-4})$
Normal (B)	$\beta = 3,7 (10^{-4})$	$\beta = 4,2 (10^{-5})$	$\beta = 4,4 (5 \cdot 10^{-6})$
Pequeno (C)	$\beta = 4,2 (10^{-5})$	$\beta = 4,4 (5 \cdot 10^{-6})$	$\beta = 4,7 (10^{-6})$

Fonte: adaptado de JCSS (2001), Probabilistic Model Code — Part 1: Basis of Design, Tabela 1.

Para situações de projeto usuais (custo normal de segurança e consequências moderadas a grandes, como é o caso de edificações correntes de concreto armado), os valores-alvo recomendados situam-se na faixa de $\beta = 4,2$ a $4,4$. Essa faixa de referência é retomada na Seção 6.2, na discussão do índice de confiabilidade obtido para a viga estudada neste trabalho.

2.8 Exemplo Teórico Ilustrativo: Aplicação do FORM com Oito Variáveis Aleatórias

Antes de apresentar o estudo principal deste trabalho, no qual as incertezas do problema são concentradas em quatro variáveis aleatórias (Seção 5), esta seção apresenta um exemplo teórico ilustrativo em que todas as oito grandezas relevantes à determinação do momento fletor resistente e solicitante da viga são tratadas como variáveis aleatórias: a largura (bw), a altura total (h), a altura útil (d) e a área de armadura (As) da seção transversal, além da resistência à compressão do concreto

(f_c), da tensão de escoamento do aço (f_y) e das ações permanente (g) e variável (q). O objetivo é evidenciar, de forma quantitativa e completa, por meio do mesmo algoritmo FORM-HLRF descrito na Seção 4.6, a contribuição relativa de cada uma dessas oito grandezas para a probabilidade de falha da estrutura, fundamentando a simplificação metodológica adotada no restante do trabalho.

A viga, a função de estado limite $g(X) = M_r - M_s$ e o algoritmo FORM-HLRF empregados neste exemplo são idênticos aos descritos, respectivamente, nas Seções 5.1, 5.2 e 5.4 deste trabalho, diferindo apenas quanto ao número e à natureza das variáveis aleatórias consideradas. O código-fonte correspondente a este exemplo de oito variáveis está disponível, na íntegra, no Apêndice B.

2.8.1 Variáveis aleatórias do exemplo

A Tabela 5 apresenta as oito variáveis aleatórias consideradas neste exemplo teórico, com suas distribuições de probabilidade, médias e coeficientes de variação.

Tabela 5 — Variáveis aleatórias do exemplo teórico com oito grandezas.

Variável	Descrição	Distribuição	Média (μ)	C.V.
bw	Largura da seção	Normal	15,00 cm	6,7%
h	Altura total da seção	Normal	45,00 cm	4,5%
d	Altura útil da seção	Normal	40,00 cm	2,75%
As	Área de armadura	Normal	12,00 cm ²	1,5%
f_c	Resistência do concreto	Normal	2,50 kN/cm ² (25 MPa)	15,0%
f_y	Tensão de escoamento do aço	Lognormal	50,00 kN/cm ² (50 Mpa)	5,0%
g	Ação permanente	Normal	0,10 kN/cm	10,0%
q	Ação variável	Gumbel-máx.	0,15 kN/cm	20,0%

Fonte: elaborado pelo autor (2026), com base nos parâmetros de entrada do algoritmo FORM-HLRF (Apêndice B).

Observa-se, desde já, uma diferença substancial entre os coeficientes de variação das quatro variáveis geométricas (bw, h, d, As), todos inferiores a 7%, e os das quatro variáveis de resistência e de ações (f_c , f_y , g, q), que chegam a 20% no caso da ação variável (q). Essa disparidade é o ponto de partida para a discussão dos resultados deste exemplo.

2.8.2 Convergência do algoritmo

O algoritmo foi inicializado no vetor de médias ($x_k=0 = \mu$), com tolerância de convergência de 10^{-4} na variação do índice de confiabilidade β entre iterações consecutivas. A convergência foi atingida na 10ª iteração, conforme apresentado na Tabela 6.

Tabela 6 — Histórico de convergência do algoritmo FORM-HLRF para o exemplo com oito variáveis.

Iter.	β	Pf	g(x) [kN·m]	fc* [kN/cm²]	fy* [kN/cm²]	q* [kN/cm]
1	6,5792	$2,365 \times 10^{-11}$	105,4044	1,213	43,511	0,2491
2	3,7073	$1,048 \times 10^{-4}$	-33,5611	1,513	49,505	0,2515
3	3,7984	$7,280 \times 10^{-5}$	3,304494	1,457	49,936	0,2449
4	3,7983	$7,283 \times 10^{-5}$	0,043425	1,443	49,938	0,2427
5	3,7981	$7,291 \times 10^{-5}$	-0,00208	1,437	49,938	0,2415
6	3,7980	$7,293 \times 10^{-5}$	-0,00034	1,434	49,938	0,2409
7	3,7980	$7,294 \times 10^{-5}$	-0,000086	1,432	49,938	0,2405
8	3,7980	$7,294 \times 10^{-5}$	-0,000024	1,431	49,938	0,2403
9	3,7980	$7,294 \times 10^{-5}$	-0,000006	1,430	49,938	0,2402
10	3,7980	$7,294 \times 10^{-5}$	-0,000001	1,430	49,938	0,2402

Fonte: saída do algoritmo FORM-HLRF, elaborado pelo autor (2026).

Os resultados globais obtidos ao final da convergência foram: índice de confiabilidade $\beta = 3,7980$; probabilidade de falha $Pf = 7,2941 \times 10^{-5}$; e 10 iterações até a convergência. Esses valores são próximos, mas não idênticos, aos obtidos no estudo principal deste trabalho (Seção 6), com apenas quatro variáveis aleatórias ($\beta = 3,9285$; $Pf = 4,2742 \times 10^{-5}$); essa diferença é discutida a seguir, na Seção 4.8.3.

2.8.3 Ponto de projeto e fatores de sensibilidade

Tabela 7 — Ponto de projeto (x^*) e fatores de sensibilidade (α^2) das oito variáveis aleatórias.

Variável	Distribuição	Média (μ)	C.V.	x^* (projeto)	α^2 (%)
bw [cm]	Normal	15,0000	6,7%	14,2274	4,10%
h [cm]	Normal	45,0000	4,5%	45,0000	0,00%
d [cm]	Normal	40,0000	2,75%	39,2741	3,02%
As [cm²]	Normal	12,0000	1,5%	11,9952	0,01%
fc [kN/cm²]	Normal	2,5000	15,0%	1,4301	56,43%

f_y [kN/cm ²]	Lognormal	50,0000	5,0%	49,9376	0,00%
g [kN/cm]	Normal	0,1000	10,0%	0,1037	0,94%
q [kN/cm]	Gumbel-máx.	0,1500	20,0%	0,2402	35,50%

Fonte: saída do algoritmo FORM-HLRF, elaborado pelo autor (2026).

Os resultados da Tabela 7 revelam que, ao contrário do que a disparidade de coeficientes de variação da Tabela 5 poderia sugerir isoladamente, a resistência à compressão do concreto (f_c) é a variável de maior influência sobre a probabilidade de falha, respondendo por 56,43% da variabilidade total do problema, seguida pela ação variável (q), com 35,50%. As quatro variáveis geométricas (b_w , h , d , A_s), somadas, contribuem com apenas 7,13% da variabilidade total (4,10% + 0,00% + 3,02% + 0,01%), enquanto a tensão de escoamento do aço (f_y) apresentou contribuição praticamente nula (0,00%), e a ação permanente (g), com apenas 0,94%. A altura total (h) apresentou contribuição nula, uma vez que, no ponto de projeto, essa variável permaneceu exatamente em seu valor médio, evidenciando sensibilidade praticamente nula do momento resistente a pequenas variações dessa grandeza, dentro do modelo físico adotado.

É particularmente relevante observar que, entre as quatro variáveis geométricas, cujos coeficientes de variação variam de 1,5% a 6,7%), a largura da seção (b_w) se destaca com a maior contribuição (4,10%), o que é coerente com seu papel direto na determinação da força resultante de compressão no concreto. Ainda assim, mesmo essa contribuição é bem menor do que a da resistência do concreto (f_c) e da ação variável (q).

Os resultados deste exemplo teórico, tomados em conjunto, mostram que o índice de confiabilidade obtido com oito variáveis aleatórias ($\beta = 3,7980$) é próximo, embora não idêntico, ao obtido no estudo principal deste trabalho com apenas quatro variáveis aleatórias ($\beta = 3,9285$), e que o padrão de sensibilidade se mantém qualitativamente semelhante entre os dois casos, com a resistência do concreto e a ação variável respondendo, juntas, pela quase totalidade da variabilidade total do problema. Essa constatação fundamenta a simplificação metodológica adotada no restante deste trabalho (Seções 5 e 6), que trata as quatro variáveis geométricas (b_w , h , d , A_s) como determinísticas, concentrando a análise probabilística nas quatro variáveis de maior relevância para a segurança da estrutura.

3 METODOLOGIA

3.1 Viga Estudada: Modelo Físico e Geométrico

Foi analisada uma viga de concreto armado biapoiada, de seção retangular, submetida a carregamento uniformemente distribuído ao longo do vão. O vão da viga (L) e o módulo de elasticidade do aço (Es) foram tratados como grandezas determinísticas:

- Vão: L = 500 cm (5,0 m);
- Módulo de elasticidade do aço: Es = 21.000 kN/cm² (210 GPa), em conformidade com Fusco (2017).

As dimensões da seção transversal (largura bw, altura total h, altura útil d e área de armadura As) foram fixadas em seus valores médios, tratadas como variáveis determinísticas, conforme justificado na Seção 5.3. As quatro variáveis aleatórias do problema (fc, fy, g, q) caracterizam as incertezas nas resistências dos materiais e nas ações atuantes, sendo descritas na Seção 5.3.

3.2 Módulo Físico: Momento Fletor Resistente

O cálculo do momento fletor resistente (Mr) baseia-se no equilíbrio de forças internas e na compatibilidade de deformações, com a formulação de Venturini e Rodrigues (1987) para os três domínios de deformação (Seção 4.2). A posição da linha neutra (x) é determinada pela condição de esforço normal resistente nulo (flexão simples):

$$N_r(x) = -0,68 \cdot f_c \cdot b_w \cdot x + A_s \cdot \sigma_s = 0 \quad (43)$$

em que σ_s é a tensão na armadura, calculada pelo modelo elastoplástico perfeito em função da deformação compatibilizada com a posição x. A equação $N_r(x) = 0$ é resolvida iterativamente pelo método da secante, com tolerância de 10^{-6} . Uma vez determinado xf, o momento fletor resistente é:

$$M_r = \frac{0,34 \cdot f_c \cdot b_w \cdot x_f \cdot (h - 0,8 \cdot x_f) + (d - 0,5h) \cdot A_s \cdot \sigma_{s,f}}{100} [kN \cdot m] \quad (44)$$

O momento fletor solicitante (Ms) é determinado pela combinação das ações permanente (g) e variável (q) para a viga biapoiada:

$$M_s = \frac{(g+q) \cdot L^2}{800} [kN \cdot m] \quad (45)$$

A função de estado limite do problema é, portanto:

$$g(X) = M_r(f_c, f_y) - M_s(g, q) \quad (46)$$

sendo $X = (f_c, f_y, g, q)$ o vetor das variáveis aleatórias. O valor $g(X) \leq 0$ indica que o momento solicitante supera o resistente, caracterizando falha por flexão.

3.3 Variáveis Aleatórias e Variáveis Determinísticas

A Tabela 8 apresenta as quatro variáveis aleatórias do problema, com suas distribuições de probabilidade, médias e coeficientes de variação adotados. Os valores foram definidos com base em parâmetros típicos de vigas de concreto armado, compatíveis com os intervalos reportados na literatura de confiabilidade estrutural (BECK, 2019; MELCHERS; BECK, 2018).

Tabela 8 — Variáveis aleatórias: distribuições de probabilidade e parâmetros estatísticos.

Variável	Descrição	Distribuição	Média (μ)	C.V.
f_c	Resistência à compressão do concreto	Normal	2,50 kN/cm ²	15,0%
f_y	Tensão de escoamento do aço	Lognormal	50,00 kN/cm ²	5,0%
g	Ação permanente	Normal	0,10 kN/cm	10,0%
q	Ação variável (sobrecarga)	Gumbel-máx.	0,15 kN/cm	20,0%

Fonte: elaborado pelo autor (2026), com base nos parâmetros de entrada do algoritmo FORM-HLRF.

O coeficiente de variação de 15% adotado para a resistência do concreto (f_c) é compatível com o controle de qualidade usual em obras de médio padrão. O coeficiente de 5% para a tensão de escoamento do aço (f_y) reflete a baixa dispersão característica dos aços estruturais CA-50. A ação variável (q) apresenta o maior coeficiente de variação (20%), refletindo a incerteza inerente às sobrecargas, conforme observado na literatura (BECK, 2019).

As variáveis geométricas e a área de armadura foram tratadas como determinísticas, iguais a seus valores médios, conforme a Tabela 9. Essa escolha fundamenta-se no fato de que, para coeficientes de variação típicos dessas grandezas (da ordem de 2% a 5%), sua contribuição à probabilidade de falha de elementos fletidos de concreto armado é muito menor do que a da ação variável (MELCHERS;

BECK, 2018), como demonstrado de forma quantitativa e completa pelo exemplo teórico ilustrativo com oito variáveis aleatórias, apresentado na Seção 4.8.

Tabela 9 — Variáveis determinísticas e seus valores adotados.

Variável	Descrição	Valor	Unidade	Justificativa
bw	Largura da seção transversal	15,0	cm	Tratada como determinística
h	Altura total da seção	45,0	cm	Tratada como determinística
d	Altura útil da seção	40,0	cm	Tratada como determinística
As	Área de armadura longitudinal	12,0	cm ²	Tratada como determinística
Es	Módulo de elasticidade do aço	21.000	kN/cm ²	Parâmetro determinístico
L	Vão da viga	500,0	cm	Parâmetro determinístico

Fonte: elaborado pelo autor (2026).

3.4 Algoritmo FORM-HLRF Implementado

O algoritmo FORM-HLRF foi implementado em Python, organizado em quatro blocos principais:

3.4.1 Bloco 1 — Módulo físico (classe *VigaPrismatica*)

A classe *VigaPrismatica* concentra toda a lógica de cálculo do momento fletor resistente. O método privado `_calcular_nr(x, bw, h, d, fc, fy, As)` recebe uma posição tentativa x para a linha neutra e retorna o esforço normal resistente correspondente, $Nr(x) = -0,68 \cdot fc \cdot bw \cdot x + As \cdot \sigma_s(x)$ (Eq. 43), em que a tensão no aço $\sigma_s(x)$ é obtida a partir da deformação ϵ_s compatível com x , segundo o modelo elastoplástico perfeito, com a região de deformação (I, II ou III, Seção 4.2) determinada automaticamente a partir da posição relativa de x em relação aos limites $0,259d$ e h .

A posição de equilíbrio x_f , para a qual $Nr(x_f) = 0$, é determinada pelo método público `calcular_mr` por meio de uma busca iterativa pelo método da secante. O processo parte de uma estimativa inicial $x_0 = As \cdot fy / (0,68 \cdot fc \cdot bw)$, obtida supondo o escoamento pleno da armadura; caso $Nr(x_0)$ já esteja dentro da tolerância (10^{-6}), a busca é encerrada de imediato. Caso contrário, adota-se um segundo ponto $x_1 = 0,259d$, correspondente à fronteira entre as Regiões I e II (Seção 4.2), e o algoritmo itera a recorrência da secante $x_{n+1} = [x_{n-1} \cdot Nr(x_n) - x_n \cdot Nr(x_{n-1})] / [Nr(x_n) - Nr(x_{n-1})]$,

atualizando a cada passo o par de pontos mais recente, por até 100 iterações ou até que $|Nr(x)| \leq 10^{-6}$. Optou-se pelo método da secante, em vez do método de Newton-Raphson, por dispensar o cálculo analítico da derivada de $Nr(x)$ em relação a x — derivada que mudaria de expressão a cada transição de região de deformação, tornando sua implementação mais complexa e sujeita a descontinuidades. Uma vez convergida a posição x_f e a tensão σ_s correspondente, o momento resistente é obtido diretamente pela Eq. 44.

O método `g_function` encapsula esse procedimento completo e implementa a função de estado limite $g(X) = M_r - M_s$ (Eq. 46): impõe limites físicos mínimos às variáveis de entrada (evitando valores nulos ou negativos que inviabilizariam o método da secante durante a busca do ponto de projeto, quando o algoritmo FORM-HLRF avalia pontos temporariamente distantes da região física do problema), chama `calcular_mr` para obter x_f , e retorna $g(X)$, os momentos resistente e solicitante, a posição da linha neutra e o domínio de deformação correspondente — informações reaproveitadas tanto pelo laço iterativo (Seção 5.4.3) quanto pelo relatório final (Seção 5.4.4).

3.4.2 Bloco 2 — Gradiente numérico

O gradiente $\nabla g(x)$ é calculado numericamente por diferenças finitas progressivas, com perturbação relativa $\delta = 10^{-5}$:

$$\frac{\partial g}{\partial x_i} \approx \frac{g(x + \Delta x_i \cdot e_i) - g(x)}{\Delta x_i} \quad (47)$$

$$\Delta x_i = \delta \cdot |x_i| \quad (48)$$

A função `gradiente_numerico` avalia inicialmente $g_base = g(x_k)$ uma única vez; em seguida, para cada uma das n variáveis aleatórias do problema ($n = 4$ no estudo principal; $n = 8$ no exemplo teórico da Seção 4.8), perturba-se apenas a coordenada correspondente em Δx_i (ou $\Delta x_i = \delta$, caso $x_i = 0$), recalcula-se g no ponto perturbado e estima-se a derivada parcial pela razão incremental da Eq. 47. Cada chamada à função requer, portanto, $n + 1$ avaliações completas da função de estado limite — uma para g_base e uma por variável perturbada —, cada qual envolvendo, internamente, uma busca completa pelo método da secante (Seção 5.4.1). A opção por diferenças progressivas, em vez de diferenças centrais (que exigiriam $2n$ avaliações), reduz o custo computacional pela metade a cada iteração do laço HLRF, sem perda relevante de precisão dentro de uma mesma região de deformação, onde a função $g(x)$ se

comporta de forma suave. O uso de um passo relativo ($\delta \cdot |x_i|$), em vez de um passo absoluto fixo, garante ainda que a perturbação seja proporcional à ordem de grandeza de cada variável — f_c na casa de poucos kN/cm^2 e A_s , no exemplo de oito variáveis, na casa de dezenas de cm^2 —, evitando tanto o cancelamento numérico (passo pequeno demais) quanto o viés de truncamento (passo grande demais) na estimativa do gradiente.

3.4.3 Bloco 3 — Laço iterativo HLRF

O laço iterativo principal, no bloco while not convergiu, implementa os Passos 1 a 9 descritos na Seção 4.6.3. A cada iteração k , calculam-se as normais equivalentes e as matrizes jacobianas J_{xy} e J_{yx} no ponto corrente x_k (Eqs. 32–34); transforma-se x_k para o espaço normal padrão, obtendo y_k ; avalia-se $g(x_k)$; chama-se a função gradiente_numerico (Seção 5.4.2) para obter $\nabla g(x)$, que é então mapeado para o espaço Y por J_{xy}^T e normalizado para fornecer o vetor de cossenos diretores α ; o novo ponto y_{k+1} é obtido pela fórmula de projeção do HLRF (Passo 6), da qual decorrem diretamente o índice de confiabilidade $\beta_{k+1} = \|y_{k+1}\|$ e, por retransformação, o ponto x_{k+1} no espaço original. A verificação da convergência, originalmente baseada apenas na estabilização relativa de β ($|\beta_{k+1} - \beta_k| / \beta_{k+1} \leq 10^{-4}$), foi complementada neste trabalho por um segundo critério: exige-se também que a própria função de estado limite, reavaliada no novo ponto x_{k+1} , satisfaça $|g(x_{k+1})| \leq 10^{-6} \text{ kN}\cdot\text{m}$. A convergência só é declarada quando ambos os critérios são satisfeitos simultaneamente. Essa exigência adicional se justifica porque o critério de β , por ser relativo, pode estabilizar antes que o ponto encontrado esteja, de fato, suficientemente próximo da superfície de falha real ($g(x) = 0$): verificou-se que, no ponto em que o critério de β isoladamente já era satisfeito, $g(x)$ ainda se encontrava na casa de $10^{-4} \text{ kN}\cdot\text{m}$, uma ordem de grandeza distante da tolerância absoluta desejada. Com o critério duplo, o algoritmo passou a executar iterações adicionais — de 5 para 10, no estudo principal (Seção 6.1) — refinando o ponto de projeto até que $g(x)$ efetivamente se aproxime de zero dentro da tolerância especificada, sem alterar de forma prática os valores finais de β e P_f (variação inferior a 0,0001 em β). Um limite de segurança de 100 iterações evita a ocorrência de laços infinitos em eventuais casos de não convergência.

5.4.4 Bloco 4 — Relatório final e cálculo dos coeficientes parciais de segurança

Ao final da convergência, a função `_imprimir_resultados` organiza o relatório com o número de iterações, β , P_f , o ponto de projeto x^* , os fatores de sensibilidade α^2 (Seção 4.6.3) e a verificação do estado limite no ponto de projeto (M_r , M_s , posição da linha neutra e domínio de deformação), obtidos por uma última chamada a `g_function` (Seção 5.4.1).

O cálculo dos coeficientes parciais de segurança é realizado em um bloco específico, imediatamente após a convergência do laço iterativo. Para cada variável aleatória, calcula-se primeiro o valor característico X_k , empregando a função quantil apropriada à sua distribuição de probabilidades (Seção 4.4): para as resistências (f_c e f_y), X_k corresponde ao 5º percentil inferior — $f_{c_k} = \mu_{f_c} - 1,645 \cdot \sigma_{f_c}$, para a distribuição normal, e $f_{y_k} = \exp(\lambda f_y - 1,645 \cdot \zeta f_y)$, para a distribuição lognormal, reaproveitando os parâmetros λ e ζ já calculados no início do script (Seção 5.3); para a ação permanente g , adota-se o próprio valor médio, $g_k = \mu_g$; e para a ação variável q , calcula-se o 95º percentil da distribuição de Gumbel, $q_k = \mu_n - (1/\alpha_q) \cdot \ln[-\ln(0,95)]$, reaproveitando os parâmetros μ_n e α_q da distribuição ajustada. O coeficiente parcial de cada variável é então obtido de forma direta, pela razão entre o valor no ponto de projeto x^* (já disponível ao final do laço iterativo) e o respectivo valor característico: $\varphi_c = x^* f_c / f_{c_k}$ e $\varphi_s = x^* f_y / f_{y_k}$ para as resistências, e $\gamma_g = x^* g / g_k$ e $\gamma_q = x^* q / q_k$ para as ações. Esse procedimento é, em essência, o mesmo tanto no script de quatro variáveis (Apêndice A) quanto no de oito variáveis (Apêndice B), neste último estendido às quatro variáveis geométricas (b_w , h , d , A_s), cujos coeficientes γ são calculados de forma análoga, porém tomando como referência o próprio valor nominal (médio) de cada grandeza geométrica, já que a NBR 6118 não define valores característicos reduzidos para dimensões de projeto.

4.5 Ferramenta Computacional

A implementação foi realizada em Python 3, utilizando a biblioteca NumPy para operações matriciais (incluindo a decomposição de Cholesky da matriz de correlação e operações de álgebra linear) e a biblioteca SciPy, especificamente o módulo `scipy.stats`, para o cálculo das funções de distribuição cumulativa e densidade de probabilidades da distribuição normal padrão, necessárias ao princípio da normal equivalente (Eqs. 32–34). O código completo está disponível no Apêndice A.

5 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados e discutidos os resultados obtidos pela execução do algoritmo FORM-HLRF (Apêndice A) aplicado à viga descrita na Seção 5, com as quatro variáveis aleatórias e os parâmetros da Tabela 8.

5.1 Convergência do Algoritmo

O algoritmo foi inicializado com $x_k=0 = \mu$ (vetor de médias) e tolerância de convergência de 10^{-4} em $|\beta_{k+1} - \beta_k|$. A convergência foi atingida na 10ª iteração, conforme a Tabela 10.

Tabela 10 — Histórico de convergência do algoritmo FORM-HLRF.

Iter.	β	Pf	$g(x)$ [kN·m]	f_c^* [kN/cm ²]	f_y^* [kN/cm ²]	q^* [kN/cm]
1	7,5136	$2,876 \times 10^{-14}$	105,4044	0,822	41,555	0,2808
2	4,0049	$3,102 \times 10^{-5}$	-52,5683	1,324	49,191	0,2516
3	3,9292	$4,261 \times 10^{-5}$	-2,0066	1,336	49,932	0,2476
4	3,9287	$4,271 \times 10^{-5}$	-0,00095	1,329	49,938	0,2459
5	3,9285	$4,273 \times 10^{-5}$	-0,00031	1,325	49,938	0,2449
6	3,9285	$4,274 \times 10^{-5}$	-0,00021	1,322	49,938	0,2444
7	3,9285	$4,274 \times 10^{-5}$	0,00002	1,321	49,938	0,2441
8	3,9285	$4,274 \times 10^{-5}$	0,00001	1,320	49,938	0,2439
9	3,9285	$4,274 \times 10^{-5}$	0,00001	1,320	49,938	0,2438
10	3,9285	$4,274 \times 10^{-5}$	0,0000	1,320	49,938	0,2438

Fonte: saída do algoritmo FORM-HLRF, elaborado pelo autor (2026).

A convergência em dez iterações ainda é rápida para os padrões do método FORM, embora menos abrupta do que em problemas mais lineares: a primeira iteração produziu um índice de confiabilidade ($\beta = 7,5136$) bastante distante do valor final, refletindo o quanto o ponto de partida ($x_k=0 = \mu$) está afastado da superfície de falha real. A partir da segunda iteração, o algoritmo já se aproxima fortemente do valor convergido (variação de apenas 1,91% entre as iterações 2 e 5), com o índice de confiabilidade estabilizado nas quatro primeiras casas decimais já na 5ª iteração. As cinco iterações adicionais (6ª a 10ª) decorrem do critério de parada duplo adotado neste trabalho (Seção 5.4.3): além da estabilização relativa de β , exige-se que a própria função de estado limite $g(x)$ se aproxime de zero com tolerância absoluta de 10^{-6} kN·m, o que refina a posição do ponto de projeto sobre a superfície de falha sem

alterar de forma prática os resultados finais (variação inferior a 0,0001 em β entre a 5ª e a 10ª iteração).

5.2 Ponto de Projeto e Índice de Confiabilidade

Os resultados globais obtidos ao final da convergência foram:

- Índice de confiabilidade: $\beta = 3,9285$;
- Probabilidade de falha: $P_f = 4,2742 \times 10^{-5}$ (aproximadamente 43 falhas em cada 1.000.000 vigas nominalmente idênticas);
- Número de iterações: 10.

O índice de confiabilidade obtido ($\beta = 3,9285$) situa-se acima dos valores-alvo usualmente recomendados para o estado limite último de elementos estruturais dúcteis, tipicamente entre $\beta = 3,7$ e $3,8$ para um período de referência de 50 anos (NBR 8681, 2025; JCSS, 2001), embora com uma margem mais modesta do que a observada em versões anteriores deste estudo. O resultado indica que, com os parâmetros estatísticos adotados (Tabela 8), a viga apresenta nível de segurança adequado, ainda que próximo do limite inferior usualmente recomendado, compatível com as práticas de projeto normatizadas.

5.3 Fatores de Sensibilidade (α^2)

Tabela 11 — Ponto de projeto (x^*) e fatores de sensibilidade (α^2) das variáveis aleatórias.

Variável	Distribuição	Média (μ)	C.V.	x^* (projeto)	α^2 (%)
f_c [kN/cm ²]	Normal	2,5000	15,0%	1,31970	64,19%
f_y [kN/cm ²]	Lognormal	50,0000	5,0%	49,93762	0,00%
g [kN/cm]	Normal	0,1000	10,0%	0,10371	0,89%
q [kN/cm]	Gumbel-máx.	0,1500	20,0%	0,24379	34,92%

Fonte: saída do algoritmo FORM-HLRF, elaborado pelo autor (2026).

Os resultados da Tabela 11 evidenciam com clareza que a resistência à compressão do concreto (f_c) é a principal responsável pela probabilidade de falha da viga, respondendo por 64,19% da variabilidade total do problema, seguida pela ação variável (q), com 34,92%. Esse resultado reflete tanto o coeficiente de variação relativamente elevado adotado para f_c (15%) quanto sua influência direta e não linear sobre o momento resistente da seção. A tensão de escoamento do aço, f_y (0,00%), e

a ação permanente, g (0,89%), têm papel praticamente irrelevante para a probabilidade de falha, a primeira por seu baixo coeficiente de variação (5%) combinado à posição usualmente pouco crítica da armadura no equilíbrio da seção, e a segunda por sua baixa variabilidade (C.V. = 10%).

Em termos dos valores no ponto de projeto, verifica-se que: a resistência do concreto f_c deslocou-se do valor médio (2,50 kN/cm²) para 1,31970 kN/cm², uma redução de quase 47% em relação à média, evidenciando que a falha tende a ocorrer justamente nas realizações mais desfavoráveis dessa variável; a ação variável q deslocou-se do valor médio (0,15 kN/cm) para 0,24379 kN/cm, cerca de 1,6 vez seu valor médio; ao passo que f_y e g deslocaram-se apenas marginalmente em relação às suas médias, confirmando a baixa sensibilidade da probabilidade de falha às incertezas dessas duas variáveis para os coeficientes de variação adotados.

Esse padrão de sensibilidade, com a resistência do concreto e a ação variável respondendo juntas por cerca de 99% da variabilidade total do problema, é consistente com a literatura de confiabilidade de elementos fletidos de concreto armado (MELCHERS; BECK, 2018) e reforça a pertinência da hipótese adotada de tratar as variáveis geométricas e a área de armadura como determinísticas.

5.4 Verificação do Estado Limite no Ponto de Projeto

Tabela 12 — Estado limite no ponto de projeto.

Grandeza	Valor	Observação
Momento Resistente (M_r)	108,5938 kN·m	—
Momento Solicitante (M_s)	108,5938 kN·m	—
$g(x^*) = M_r - M_s$	0,0000 kN·m	≈ ponto sobre a superfície de falha
Posição da linha neutra (x)	28,0187 cm	—
Domínio de deformação	Região II	Equivalente ao Domínio 3 (NBR 6118)

Fonte: saída do algoritmo FORM-HLRF, elaborado pelo autor (2026).

O valor de $g(x^*) = 0,0000$ kN·m confirma que o algoritmo convergiu corretamente para a superfície de falha ($g(x) = 0$), validando a consistência numérica da implementação. O domínio de deformação no ponto de projeto corresponde à Região II (Domínio 3 da NBR 6118), caracterizado pelo escoamento da armadura tracionada simultaneamente à ruptura por compressão do concreto, domínio

recomendado para o dimensionamento de vigas (ARAÚJO, 2023; FUSCO, 2017), indicando adequação física do problema modelado.

5.5 Coeficientes Parciais de Segurança Obtidos

A partir do ponto de projeto (Tabela 11) e da metodologia descrita na Seção 4.7, foram calculados os valores característicos e os coeficientes parciais de segurança das quatro variáveis aleatórias, apresentados na Tabela 13.

Tabela 13 — Coeficientes parciais de segurança: valores obtidos pelo FORM e valores normativos.

Variável	X _k (caract.)	x* (projeto)	γ — FORM
f _c	1,883 kN/cm ²	1,320 kN/cm ²	0,7008
f _y	45,997 kN/cm ²	49,938 kN/cm ²	1,0857
g	0,100 kN/cm	0,1037 kN/cm	1,0371
q	0,206 kN/cm	0,2438 kN/cm	1,1836

Fonte: elaborado pelo autor (2026).

Os resultados da Tabela 13 mostram que os coeficientes obtidos pelo FORM concentram a maior parte da margem de segurança na resistência do concreto ($\varphi_c = 0,7008$) e, secundariamente, na ação variável ($\gamma_q = 1,1836$), refletindo o mesmo padrão de sensibilidade identificado na Seção 6.3 ($\alpha^2 f_c = 64,19\%$; $\alpha^2 q = 34,92\%$).

Para a resistência do concreto, o coeficiente $\varphi_c = 0,7008$ (inferior à unidade) indica que, no ponto de projeto, o valor de f_c está abaixo até mesmo do valor característico (5º percentil, $X_k = 1,883$ kN/cm²): por ser a variável de maior sensibilidade (Seção 6.3), f_c se afasta substancialmente de sua média no ponto de projeto (redução de quase 47%), superando a margem já embutida no próprio percentil característico. Em contraste, a tensão de escoamento do aço (f_y) e a ação permanente (g), por apresentarem sensibilidade praticamente nula, permanecem próximas de seus valores médios no ponto de projeto, resultando em coeficientes ($\varphi_s = 1,0857$; $\gamma_g = 1,0371$) próximos da unidade. Já a ação variável (q), segunda variável mais sensível do problema, apresenta um coeficiente ($\gamma_q = 1,1836$) moderadamente superior à unidade, coerente com sua contribuição relevante, embora secundária à do concreto, para a probabilidade de falha.

5.6 Discussão Geral

Os coeficientes normativos resultam de um processo de calibração conduzido para um conjunto amplo e representativo de elementos estruturais, situações de carregamento e pares de materiais, com o objetivo de fornecer um nível médio de segurança adequado para a generalidade dos projetos. Por esse motivo, eles não necessariamente coincidem com os coeficientes obtidos para um problema específico, mas devem ser entendidos como valores médios que asseguram um índice de confiabilidade-alvo na média dos casos projetados.

O presente estudo, ao analisar uma única viga com parâmetros estatísticos específicos, captura a distribuição de segurança particular daquele caso. Os coeficientes parciais apresentados na Tabela 13 (Seção 6.5), recalculados com o modelo físico e o critério de convergência atuais ($\beta = 3,9285$; Tabela 11), mostram que o FORM concentra a margem de segurança predominantemente na resistência do concreto ($\varphi_c = 0,7008$) e, secundariamente, na ação variável ($\gamma_q = 1,1836$), refletindo diretamente a análise de sensibilidade da Seção 6.3, que indica que a estrutura é mais sensível à resistência do concreto ($\alpha^2 = 64,19\%$) do que à ação variável ($\alpha^2 = 34,92\%$). Essa distribuição difere da adotada pelos coeficientes normativos gerais ($\gamma_c = 1,40$; $\gamma_q = 1,40$), que repartem a margem de segurança de forma mais equilibrada entre resistências e ações — o que é esperado, já que os coeficientes normativos são calibrados para um conjunto amplo de situações de projeto, e não para o caso particular desta viga.

Os resultados confirmam a aplicabilidade e a robustez do método FORM para a análise de confiabilidade de vigas de concreto armado, e ilustram como a análise de sensibilidade pode fornecer ao engenheiro projetista informações valiosas sobre quais incertezas mais influenciam a segurança da estrutura, informação diretamente utilizável em decisões de projeto, controle de qualidade e calibração de normas.

O resultado mais conservador obtido neste trabalho, com β superior ao alvo normativo, reforça que o nível de segurança de uma viga de concreto armado é fortemente dependente das características específicas de cada elemento, sendo a comparação direta entre métodos de transformação, como o FORM, e métodos de simulação, como o de Monte Carlo, uma direção promissora para trabalhos futuros, retomada a seguir.

REFERÊNCIAS

- ABNT. NBR 6118: Projeto de estruturas de concreto – Procedimento. Rio de Janeiro: ABNT, 2023.
- ABNT. NBR 5738: Concreto – Procedimento para moldagem e cura de corpos de prova. Rio de Janeiro: ABNT, 2015.
- ABNT. NBR 5739: Concreto – Ensaio de compressão de corpos de prova cilíndricos. Rio de Janeiro: ABNT, 2018.
- ABNT. NBR 8681: Ações e segurança nas estruturas – Procedimento. Rio de Janeiro: ABNT, 2025.
- ARAÚJO, J. M. Curso de Concreto Armado. v. 1-4. 5. ed. Rio Grande: Dunas, 2023.
- BECK, A. T. Confiabilidade e Segurança das Estruturas. Rio de Janeiro: Elsevier, 2019.
- CHUST, R. C. Curso Básico de Concreto Armado. 2. ed. São Paulo: Pini, 2014.
- COELHO, B. R. C. Simulação de Monte Carlo Aplicada à Avaliação da Confiabilidade Estrutural de Vigas de Concreto Armado. 2022. 81 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) — Universidade Federal do Maranhão, São Luís, 2022.
- ELLINGWOOD, B. R. Reliability-based condition assessment and LRFD for existing structures. *Structural Safety*, v. 18, n. 2-3, p. 67-80, 1996.
- EUROCODE. EN 1990: Basis of Structural and Geotechnical Design. Brussels: European Committee for Standardization, 2023.
- FABER, Michael H.; VROUWENVELDER, A. C. W. M. Interpretation of Uncertainties and Probabilities in Civil Engineering Decision Analysis. Background Document nº 2. [S.l.]: JCSS, 2008.
- FUSCO, Péricles Brasiliense; ONISHI, Minoru. Introdução à engenharia de estruturas de concreto. São Paulo: Cengage Learning, 2017. 264 p.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, v. 585, p. 357-362, 2020.
- HASOFER, A. M.; LIND, N. C. Exact and Invariant Second-Moment Code Format. *Journal of the Engineering Mechanics Division, ASCE*, v. 100, n. 1, p. 111-121, 1974.
- JOINT COMMITTEE ON STRUCTURAL SAFETY (JCSS). Probabilistic Model Code — Part 1: Basis of Design. [S.l.]: JCSS, 2001.

JOINT COMMITTEE ON STRUCTURAL SAFETY (JCSS). Risk Assessment in Engineering: Principles, System Representation & Risk Criteria. Ed. M. H. Faber. [S.l.]: JCSS, 2008.

KROON, Inger B.; MAES, Marc A. Theoretical Framework for Risk Assessment and Evaluation. Background Document nº 1. [S.l.]: JCSS, 2008.

LEONHARDT, F.; MÖNNIG, E. Construções de Concreto: Princípios Básicos do Dimensionamento de Estruturas de Concreto Armado. v. 1. Rio de Janeiro: Interciência, 1982.

MELCHERS, R. E.; BECK, A. T. Structural Reliability Analysis and Prediction. 3. ed. Chichester: John Wiley & Sons, 2018.

MONTGOMERY, D. C.; RUNGER, G. C. Estatística Aplicada e Probabilidade para Engenheiros. 7. ed. Rio de Janeiro: LTC, 2024.

MONTOYA, P. J.; MESEGUER, A. G.; CABRÉ, F. M. Hormigón Armado. 15. ed. Barcelona: Gustavo Gili, 2009.

MRÁZIK, A.; KRIZMA, M. Probability-based design standards of structures. Journal of Constructional Steel Research, v. 42, n. 1, p. 1-18, 1997.

NOWAK, A. S.; COLLINS, K. R. Reliability of Structures. 2. ed. Boca Raton: CRC Press, 2013.

RACKWITZ, R.; FIESSLER, B. Structural reliability under combined random load sequences. Computers & Structures, v. 9, n. 5, p. 489-494, 1978.

VENTURINI, W. S.; RODRIGUES, R. O. Dimensionamento de Peças Retangulares de Concreto Armado Solicitadas à Flexão Normal Composta. São Carlos: EESC/USP, 1987.

VIRTANEN, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, v. 17, p. 261-272, 2020.

APÊNDICE A — CÓDIGO-FONTE DO ALGORITMO FORM-HLRF, QUATRO VARIÁVEIS (PYTHON)

A seguir é apresentado, na íntegra, o código-fonte em linguagem Python do algoritmo FORM-HLRF desenvolvido e executado para a obtenção dos resultados apresentados na Seção 6 deste trabalho. O código considera quatro variáveis aleatórias (f_c, f_y, g, q) e as demais grandezas como determinísticas.

```
"""
=====
ALGORITMO FORM (First Order Reliability Method)

Prof. Felipe Alexander Vargas Bazán - UFMA

Problema: Viga de concreto armado simplesmente apoiada
          g(X) = Mr - Ms [kN.m]

Variáveis aleatórias (n=4):
  Ordem: [fc, fy, g, q]
  Distribuições: Normal (fc, g), Lognormal (fy), Gumbel-máx (q)

Variáveis determinísticas:
  bw, h, d, As
=====
"""

import numpy as np
from scipy.stats import norm
import traceback

# =====
# BLOCO 1: MÓDULO FÍSICO - Viga Prismática de Concreto Armado
# =====

class VigaPrismatica:
    """Calcula o Momento Resistente (MR) de uma viga de concreto armado."""

    def __init__(self, Es, L):
        self.Es = Es      # Módulo de elasticidade do aço [kN/cm²]
        self.L = L        # Vão da viga [cm]
        self.TOL = 1e-6   # Tolerância para equilíbrio da seção

    def _determinar_regiao(self, x, h, d):
        """Domínio de deformação (Venturini & Rodrigues)."""
        if x <= 0.259 * d: return "Região I"
        elif x <= h:       return "Região II"
        else:              return "Região III"

    def _calcular_nr(self, x, bw, h, d, fc, fy, As):
        """Esforço Normal Resistente para uma dada posição da linha neutra x."""
        x = max(1e-6, min(x, 1.25 * h))

        if x <= 0.259 * d:
            eps_s = 0.010
            eps_c = eps_s * x / (x - d) if abs(x - d) > 1e-6 else 0.0
        elif x <= h:
            eps_cu = -0.0035
            eps_s = eps_cu * (x - d) / x
            eps_c = eps_cu
        else:
            eps_c2 = -0.002
```

```

        alpha = 3.0 / 7.0
        denom = (x - alpha * h) if abs(x - alpha * h) > 1e-6 else 1e-6
        eps_s = eps_c2 * (x - d) / denom
        eps_c = eps_c2 * x / denom

    eps_y = fy / self.Es
    if eps_s <= -eps_y: sigma_s = -fy
    elif eps_s <= eps_y: sigma_s = self.Es * eps_s
    elif eps_s <= 0.010: sigma_s = fy
    else: sigma_s = 0.0

    nr = -0.68 * fc * bw * x + As * sigma_s
    return nr, sigma_s

def calcular_mr(self, bw, h, d, fc, fy, As):
    """
    Determina a posição da linha neutra (x) por secante e calcula MR.
    Retorna: (mr [kN.cm], x_final [cm], região)
    """
    x0 = (As * fy) / max(0.68 * fc * bw, 1e-6)
    nr0, _ = self._calcular_nr(x0, bw, h, d, fc, fy, As)

    if abs(nr0) <= self.TOL:
        x_final = x0
        _, sigma_s_final = self._calcular_nr(x_final, bw, h, d, fc, fy, As)
    else:
        x1 = 0.259 * d
        nr1, _ = self._calcular_nr(x1, bw, h, d, fc, fy, As)
        x_final, sigma_s_final = x1, 0.0

        for _iter in range(100):
            diff = nr1 - nr0
            if abs(diff) < 1e-12:
                break
            x2 = (x0 * nr1 - x1 * nr0) / diff
            nr2, s2 = self._calcular_nr(x2, bw, h, d, fc, fy, As)
            if abs(nr2) <= self.TOL:
                x_final, sigma_s_final = x2, s2
                break
            x0, nr0 = x1, nr1
            x1, nr1 = x2, nr2
            x_final, sigma_s_final = x1, s2

        _, sigma_s_final = self._calcular_nr(x_final, bw, h, d, fc, fy, As)

    mr = 0.34 * fc * bw * x_final * (h - 0.8 * x_final) \
        + (d - 0.5 * h) * As * sigma_s_final
    return mr, x_final, self._determinar_regiao(x_final, h, d)

def g_function(self, bw, h, d, fc, fy, As, g_carga, q_carga):
    """
    Função de Estado Limite: G(X) = Mr - Ms [kN.m]
    Ms = (g+q)*L²/8 (viga simplesmente apoiada, carga distribuída)
    """
    bw = max(bw, 1.0)
    h = max(h, 1.0)
    d = max(d, 1.0)
    fc = max(fc, 0.1)
    fy = max(fy, 0.1)
    As = max(As, 0.1)

    mr_kncm, x_f, reg = self.calcular_mr(bw, h, d, fc, fy, As)
    ms_kncm = (g_carga + q_carga) * self.L**2 / 8.0

    mr_knm = mr_kncm / 100.0
    ms_knm = ms_kncm / 100.0

    return mr_knm - ms_knm, mr_knm, ms_knm, x_f, reg

```



```

# =====
# BLOCO 2: GRADIENTE NUMÉRICO (diferenças finitas progressivas)
# =====

def gradiente_numerico(x_k, viga, deterministico):
    """
    Calcula  $\nabla g(x)$  no espaço original X (apenas para n=4)
    """
    n = len(x_k)
    grad_gx = np.zeros(n)
    h_fd = 1e-5

    # Desempacota variáveis
    bw_d, h_d, d_d, As_d = deterministico
    fc, fy, g, q = x_k

    g_base, *_ = viga.g_function(bw_d, h_d, d_d, fc, fy, As_d, g, q)

    for i in range(n):
        xp = np.copy(x_k)
        delta = h_fd * (abs(x_k[i]) if x_k[i] != 0 else 1.0)
        xp[i] += delta
        fc_p, fy_p, g_p, q_p = xp

        g_pert, *_ = viga.g_function(bw_d, h_d, d_d, fc_p, fy_p, As_d, g_p, q_p)
        grad_gx[i] = (g_pert - g_base) / delta

    return grad_gx, g_base

# =====
# BLOCO 3: ALGORITMO FORM-HLRF
# =====

def executar_form():
    print("=" * 70)
    print("    ALGORITMO FORM-HLRF | CCEC0039 - Confiabilidade Estrutural")
    print("    (Versão com n=4 Variáveis Aleatórias e 4 Determinísticas)")
    print("=" * 70)

    # -----
    # Dados do problema (determinísticos)
    # -----
    Es_det = 21000.0
    L_det = 500.0
    viga = VigaPrismatica(Es=Es_det, L=L_det)

    # As 4 variáveis que antes eram aleatórias, agora são fixas aqui
    bw_det = 15.0
    h_det = 45.0
    d_det = 40.0
    As_det = 12.0
    deterministico = (bw_det, h_det, d_det, As_det)

    n = 4 # Apenas fc, fy, g, q

    # Médias e desvios padrão - Ordem: [fc, fy, g, q]
    mu = np.array([ 2.5, 50.0, 0.10, 0.15 ])
    sigma = np.array([ 0.375, 2.5, 0.01, 0.03 ])

    # Lognormal para fy (índice 1)
    cv_fy = sigma[1] / mu[1]
    zeta_fy = np.sqrt(np.log(1.0 + cv_fy**2))
    lam_fy = np.log(mu[1]) - 0.5 * zeta_fy**2

    # Gumbel-máximos para q (índice 3)
    alpha_q = np.pi / (sigma[3] * np.sqrt(6.0))
    u_q = mu[3] - np.euler_gamma / alpha_q

```

```

def fxcdf_pdf(val, i):
    eps = 1e-15
    if i == 1: # fy (Lognormal)
        v = max(val, eps)
        cd = norm.cdf((np.log(v) - lam_fy) / zeta_fy)
        pd = (1.0 / (v * zeta_fy * np.sqrt(2.0 * np.pi))) * np.exp(-0.5 * ((np.log(v) - lam_fy) / zeta_fy)**2)
    elif i == 3: # q (Gumbel)
        zg = alpha_q * (val - u_q)
        cd = np.exp(-np.exp(-zg))
        pd = alpha_q * np.exp(-zg - np.exp(-zg))
    else: # fc e g (Normal)
        cd = norm.cdf(val, mu[i], sigma[i])
        pd = norm.pdf(val, mu[i], sigma[i])
    return max(pd, eps), cd

Rz = np.eye(n)
L_mat = np.linalg.cholesky(Rz)
L_inv = np.linalg.inv(L_mat)

x_k = np.copy(mu)
beta_k = 0.0
TOL = 1e-4 # tolerância relativa em beta (critério original)
TOL_G = 1e-6 # tolerância absoluta em g(x) [kN.m] (novo critério de parada)

print(f"\n{'Iter':<5} {'Beta':<10} {'Pf':<12} {'g(x) [kN.m]':<14} "
      f"{'fc*':<9} {'fy*':<9} {'g*':<9} {'q*':<9}")
print("-" * 82)

beta_final = 0.0
pf_final = 1.0
iter_final = 0
alpha_final = np.zeros(n)

k = 0
convergiu = False

while not convergiu:
    z_k = np.zeros(n)
    phi_z = np.zeros(n)
    pdf_x = np.zeros(n)
    mu_neq = np.zeros(n)

    for i in range(n):
        pd_i, cd_i = fxcdf_pdf(x_k[i], i)
        cd_clip = np.clip(cd_i, 1e-15, 1.0 - 1e-15)
        z_k[i] = norm.ppf(cd_clip)
        phi_z[i] = norm.pdf(z_k[i])
        pdf_x[i] = pd_i
        sigma_neq_i = phi_z[i] / pd_i
        mu_neq[i] = x_k[i] - z_k[i] * sigma_neq_i

    D_neq = np.diag(phi_z / pdf_x)
    J_xy = D_neq @ L_mat
    J_yx = L_inv @ np.linalg.inv(D_neq)

    y_k = J_yx @ (x_k - mu_neq)

    fc_k, fy_k, g_k_val, q_k_val = x_k
    g_val, mr_k, ms_k, x_ln_k, reg_k = viga.g_function(
        bw_det, h_det, d_det, fc_k, fy_k, As_det, g_k_val, q_k_val)

    grad_g_x, _ = gradiente_numerico(x_k, viga, deterministico)

    grad_g_y = J_xy.T @ grad_g_x
    norm_grad = np.linalg.norm(grad_g_y)
    if norm_grad < 1e-9:
        norm_grad = 1e-9

```

```

alpha = -grad_g_y / norm_grad

y_next = ((np.dot(grad_g_y, y_k) - g_val) / norm_grad**2) * grad_g_y

beta_next = np.linalg.norm(y_next)

x_next      = J_xy @ y_next + mu_neq
pf_next     = norm.cdf(-beta_next)

# Avalia g(x) no novo ponto (x_next) para o critério de parada g(x) ≈ 0
fc_n, fy_n, g_n_val, q_n_val = x_next
g_next, *_ = viga.g_function(
    bw_det, h_det, d_det, fc_n, fy_n, As_det, g_n_val, q_n_val)

print(f"{k+1:<5} {beta_next:<10.4f} {pf_next:<12.4e} {g_val:<14.4f} "
      f"{x_next[0]:<9.4f} {x_next[1]:<9.4f} {x_next[2]:<9.4f} {x_next[3]:<9.4f}")

if beta_next > 1e-12:
    criterio_beta = abs(beta_next - beta_k) / beta_next
else:
    criterio_beta = abs(beta_next - beta_k)

criterio_g = abs(g_next)

if k > 0 and (criterio_beta <= TOL and criterio_g <= TOL_G):
    beta_final = beta_next
    pf_final   = pf_next
    iter_final = k
    alpha_final = alpha
    x_k        = x_next
    convergiu  = True
else:
    x_k        = x_next
    beta_k     = beta_next
    k         += 1
    if k >= 100:
        beta_final = beta_next; pf_final = pf_next
        iter_final = 99; alpha_final = alpha; x_k = x_next
        convergiu = True

_imprimir_resultados(x_k, alpha_final, beta_final, pf_final,
                    iter_final, viga, Es_det, L_det, deterministico)

# =====
# CÁLCULO DOS COEFICIENTES PARCIAIS DE SEGURANÇA (Apenas n=4)
# =====
print(f"{'COEFICIENTES PARCIAIS DE SEGURANÇA':^75}")
print("-" * 75)

fc_k = mu[0] - 1.645 * sigma[0]
fy_k = np.exp(lam_fy - 1.645 * zeta_fy)

g_k = mu[2]
q_k = u_q - (1.0 / alpha_q) * np.log(-np.log(0.95))

phi_c = x_k[0] / fc_k
phi_s = x_k[1] / fy_k

gamma_g = x_k[2] / g_k
gamma_q = x_k[3] / q_k

print(f"  fc (Concreto)      | φ_c = {phi_c:.4f}")
print(f"  fy (Aço)           | φ_s = {phi_s:.4f}")
print(f"  g (Perm.)          | γ_g = {gamma_g:.4f}")
print(f"  q (Variável)       | γ_q = {gamma_q:.4f}")
print("-" * 75 + "\n")

```

```

# =====
# BLOCO 4: RELATÓRIO FINAL
# =====

def _imprimir_resultados(x_star, alpha, beta, pf, iters, viga, Es, L, deterministico):
    nomes = ["fc (Concreto)", "fy (Aço)", "g (Perm.)", "q (Variável)"]
    unids = ["kN/cm²", "kN/cm²", "kN/cm", "kN/cm"]
    distribs = ["Normal", "Lognormal", "Normal", "Gumbel-máx"]

    sep = "-" * 75

    print(f"\n{sep}")
    print(f"{'CONVERGÊNCIA - MÉTODO FORM-HLRF':^75}")
    print(sep)
    print(f"  Iterações até convergência : {iters+1}")
    print(f"  Índice de Confiabilidade  $\beta$  : {beta:.6f}")
    print(f"  Probabilidade de Falha Pf : {pf:.4e}")
    print(sep)

    print(f"\n{'PONTO DE PROJETO x* e FATORES DE SENSIBILIDADE  $\alpha^2$ ':^75}")
    print("-" * 75)
    print(f"{'Variável':<16} {'Distribuição':<12} {'x* (valor)':<12} "
          f"{'Unidade':<9} {' $\alpha^2$  (%)':<12}")
    print("-" * 75)
    for i, (nome, dist, val, unid) in enumerate(zip(nomes, distribs, x_star, unids)):
        contrib = alpha[i]**2 * 100.0
        print(f"  {nome:<14} {dist:<12} {val:<12.5f} {unid:<9} {contrib:>6.2f}%")

    print("-" * 75)
    print(f"  [Variáveis Determinísticas Utilizadas]")
    bw_det, h_det, d_det, As_det = deterministico
    print(f"  bw = {bw_det} cm | h = {h_det} cm | d = {d_det} cm | As = {As_det} cm²")
    print(f"  Es = {Es} kN/cm² | L = {L} cm")

    fc, fy, g_c, q_c = x_star
    gl, mr, ms, x_ln, reg = viga.g_function(bw_det, h_det, d_det, fc, fy, As_det, g_c, q_c)

    print(f"\n{'ESTADO LIMITE NO PONTO DE PROJETO':^75}")
    print("-" * 75)
    print(f"  Linha Neutra x : {x_ln:>12.4f} cm")
    print(f"  Domínio de deformação : {reg}")
    print(sep + "\n")

if __name__ == "__main__":
    try:
        executar_form()
    except Exception:
        print("\n[ERRO CRÍTICO NA EXECUÇÃO]")
        traceback.print_exc()
    input("\nPressione Enter para fechar...")

```

APÊNDICE B — CÓDIGO-FONTE DO ALGORITMO FORM-HLRF, OITO VARIÁVEIS (PYTHON)

A seguir é apresentado, na íntegra, o código-fonte em linguagem Python do algoritmo FORM-HLRF utilizado na obtenção dos resultados do exemplo teórico ilustrativo apresentado na Seção 4.8 deste trabalho. O código considera as oito variáveis aleatórias do problema ($b_w, h, d, A_s, f_c, f_y, g, q$).

```
"""
=====
ALGORITMO FORM (First Order Reliability Method)

Prof. Felipe Alexander Vargas Bazán - UFMA

Problema: Viga de concreto armado simplesmente apoiada
          g(X) = Mr - Ms [kN.m]

Variáveis aleatórias (n=8):
  Ordem: [bw, h, d, As, fc, fy, g, q]
  Distribuições: Normal (bw, h, d, As, fc, g), Lognormal (fy), Gumbel-máx (q)
=====
"""

import numpy as np
from scipy.stats import norm
import traceback

# =====
# BLOCO 1: MÓDULO FÍSICO - Viga Prismática de Concreto Armado
# =====

class VigaPrismatica:
    """Calcula o Momento Resistente (MR) de uma viga de concreto armado."""

    def __init__(self, Es, L):
        self.Es = Es          # Módulo de elasticidade do aço [kN/cm²]
        self.L = L            # Vão da viga [cm]
        self.TOL = 1e-6       # Tolerância para equilíbrio da seção

    def _determinar_regiao(self, x, h, d):
        """Domínio de deformação (Venturini & Rodrigues)."""
        if x <= 0.259 * d: return "Região I"
        elif x <= h: return "Região II"
        else: return "Região III"

    def _calcular_nr(self, x, bw, h, d, fc, fy, As):
        """Esforço Normal Resistente para uma dada posição da linha neutra x."""
        x = max(1e-6, min(x, 1.25 * h))

        if x <= 0.259 * d:
            eps_s = 0.010
            eps_c = eps_s * x / (x - d) if abs(x - d) > 1e-6 else 0.0
        elif x <= h:
            eps_cu = -0.0035
            eps_s = eps_cu * (x - d) / x
            eps_c = eps_cu
        else:
            eps_c2 = -0.002
            alpha = 3.0 / 7.0
            denom = (x - alpha * h) if abs(x - alpha * h) > 1e-6 else 1e-6
            eps_s = eps_c2 * (x - d) / denom
            eps_c = eps_c2 * x / denom
```

```

        eps_y = fy / self.Es
        if eps_s <= -eps_y: sigma_s = -fy
        elif eps_s <= eps_y: sigma_s = self.Es * eps_s
        elif eps_s <= 0.010: sigma_s = fy
        else: sigma_s = 0.0

        nr = -0.68 * fc * bw * x + As * sigma_s
        return nr, sigma_s

def calcular_mr(self, bw, h, d, fc, fy, As):
    """
    Determina a posição da linha neutra (x) por secante e calcula MR.
    Retorna: (mr [kN.cm], x_final [cm], região)
    """
    x0 = (As * fy) / max(0.68 * fc * bw, 1e-6)
    nr0, _ = self._calcular_nr(x0, bw, h, d, fc, fy, As)

    if abs(nr0) <= self.TOL:
        x_final = x0
        _, sigma_s_final = self._calcular_nr(x_final, bw, h, d, fc, fy, As)
    else:
        x1 = 0.259 * d
        nr1, _ = self._calcular_nr(x1, bw, h, d, fc, fy, As)
        x_final, sigma_s_final = x1, 0.0

        for _iter in range(100):
            diff = nr1 - nr0
            if abs(diff) < 1e-12:
                break
            x2 = (x0 * nr1 - x1 * nr0) / diff
            nr2, s2 = self._calcular_nr(x2, bw, h, d, fc, fy, As)
            if abs(nr2) <= self.TOL:
                x_final, sigma_s_final = x2, s2
                break
            x0, nr0 = x1, nr1
            x1, nr1 = x2, nr2
            x_final, sigma_s_final = x1, s2

        _, sigma_s_final = self._calcular_nr(x_final, bw, h, d, fc, fy, As)

    mr = 0.34 * fc * bw * x_final * (h - 0.8 * x_final) \
        + (d - 0.5 * h) * As * sigma_s_final
    return mr, x_final, self._determinar_regiao(x_final, h, d)

def g_function(self, bw, h, d, fc, fy, As, g_carga, q_carga):
    """
    Função de Estado Limite: G(X) = Mr - Ms [kN.m]
    Ms = (g+q)*L²/8 (viga simplesmente apoiada, carga distribuída)
    """
    bw = max(bw, 1.0)
    h = max(h, 1.0)
    d = max(d, 1.0)
    fc = max(fc, 0.1)
    fy = max(fy, 0.1)
    As = max(As, 0.1)

    mr_kncm, x_f, reg = self.calcular_mr(bw, h, d, fc, fy, As)
    ms_kncm = (g_carga + q_carga) * self.L**2 / 8.0

    mr_knm = mr_kncm / 100.0
    ms_knm = ms_kncm / 100.0

    return mr_knm - ms_knm, mr_knm, ms_knm, x_f, reg

# =====
# BLOCO 2: GRADIENTE NUMÉRICO (diferenças finitas progressivas)
# =====

```

```

def gradiente_numerico(x_k, viga):
    n = len(x_k)
    grad_gx = np.zeros(n)
    h_fd = 1e-5

    bw, h, d, As, fc, fy, g, q = x_k
    g_base, *_ = viga.g_function(bw, h, d, fc, fy, As, g, q)

    for i in range(n):
        xp = np.copy(x_k)
        delta = h_fd * (abs(x_k[i]) if x_k[i] != 0 else 1.0)
        xp[i] += delta
        bw_p, h_p, d_p, As_p, fc_p, fy_p, g_p, q_p = xp
        g_pert, *_ = viga.g_function(bw_p, h_p, d_p, fc_p, fy_p, As_p, g_p, q_p)
        grad_gx[i] = (g_pert - g_base) / delta

    return grad_gx, g_base

# =====
# BLOCO 3: ALGORITMO FORM-HLRF
# =====

def executar_form():
    print("=" * 70)
    print("    ALGORITMO FORM-HLRF | CCEC0039 - Confiabilidade Estrutural")
    print("=" * 70)

    Es_det = 21000.0
    L_det = 500.0
    viga = VigaPrismatica(Es=Es_det, L=L_det)
    n = 8

    mu = np.array([15.0, 45.0, 40.0, 12.0, 2.5, 50.0, 0.10, 0.15])
    sigma = np.array([1.005, 2.025, 1.10, 0.18, 0.375, 2.5, 0.01, 0.03])

    cv_fy = sigma[5] / mu[5]
    zeta_fy = np.sqrt(np.log(1.0 + cv_fy**2))
    lam_fy = np.log(mu[5]) - 0.5 * zeta_fy**2

    alpha_q = np.pi / (sigma[7] * np.sqrt(6.0))
    u_q = mu[7] - np.euler_gamma / alpha_q

    def fxcdf_pdf(val, i):
        eps = 1e-15
        if i == 5:
            v = max(val, eps)
            cd = norm.cdf((np.log(v) - lam_fy) / zeta_fy)
            pd = (1.0 / (v * zeta_fy * np.sqrt(2.0 * np.pi))) * np.exp(-0.5 * ((np.log(v) - lam_fy) / zeta_fy)**2)
        elif i == 7:
            zg = alpha_q * (val - u_q)
            cd = np.exp(-np.exp(-zg))
            pd = alpha_q * np.exp(-zg - np.exp(-zg))
        else:
            cd = norm.cdf(val, mu[i], sigma[i])
            pd = norm.pdf(val, mu[i], sigma[i])
        return max(pd, eps), cd

    Rz = np.eye(n)
    L_mat = np.linalg.cholesky(Rz)
    L_inv = np.linalg.inv(L_mat)

    x_k = np.copy(mu)
    beta_k = 0.0
    TOL = 1e-4 # tolerância relativa em beta (critério original)
    TOL_G = 1e-6 # tolerância absoluta em g(x) [kN.m] (novo critério de parada)

```

```

print(f"\n{'Iter':<5} {'Beta':<10} {'Pf':<12} {'g(x) [kN.m]':<14} "
      f"{'fc*':<8} {'fy*':<8} {'q*':<8}")
print("-" * 70)

beta_final = 0.0
pf_final   = 1.0
iter_final = 0
alpha_final = np.zeros(n)

k          = 0
convergiu  = False

while not convergiu:
    z_k     = np.zeros(n)
    phi_z   = np.zeros(n)
    pdf_x   = np.zeros(n)
    mu_neq  = np.zeros(n)

    for i in range(n):
        pd_i, cd_i = fxcdf_pdf(x_k[i], i)
        cd_clip    = np.clip(cd_i, 1e-15, 1.0 - 1e-15)
        z_k[i]     = norm.ppf(cd_clip)
        phi_z[i]   = norm.pdf(z_k[i])
        pdf_x[i]   = pd_i
        sigma_neq_i = phi_z[i] / pd_i
        mu_neq[i]  = x_k[i] - z_k[i] * sigma_neq_i

    D_neq = np.diag(phi_z / pdf_x)
    J_xy  = D_neq @ L_mat
    J_yx  = L_inv @ np.linalg.inv(D_neq)

    y_k = J_yx @ (x_k - mu_neq)

    bw_k, h_k, d_k, As_k, fc_k, fy_k, g_k_val, q_k_val = x_k
    g_val, mr_k, ms_k, x_ln_k, reg_k = viga.g_function(
        bw_k, h_k, d_k, fc_k, fy_k, As_k, g_k_val, q_k_val)

    grad_g_x, _ = gradiente_numerico(x_k, viga)

    grad_g_y = J_xy.T @ grad_g_x
    norm_grad = np.linalg.norm(grad_g_y)
    if norm_grad < 1e-9:
        norm_grad = 1e-9

    alpha = -grad_g_y / norm_grad

    y_next = ((np.dot(grad_g_y, y_k) - g_val) / norm_grad**2) * grad_g_y

    beta_next = np.linalg.norm(y_next)

    x_next     = J_xy @ y_next + mu_neq
    pf_next    = norm.cdf(-beta_next)

    # Avalia g(x) no novo ponto (x_next) para o critério de parada g(x) ≈ 0
    bw_n, h_n, d_n, As_n, fc_n, fy_n, g_n_val, q_n_val = x_next
    g_next, *_ = viga.g_function(
        bw_n, h_n, d_n, fc_n, fy_n, As_n, g_n_val, q_n_val)

    print(f"{'k+1':<5} {'beta_next':<10.4f} {'pf_next':<12.4e} {'g_val':<14.4f} "
          f"{'x_next[4]':<8.3f} {'x_next[5]':<8.3f} {'x_next[7]':<8.4f}")

    if beta_next > 1e-12:
        criterio_beta = abs(beta_next - beta_k) / beta_next
    else:
        criterio_beta = abs(beta_next - beta_k)

    criterio_g = abs(g_next)

```



```

        if k > 0 and (criterio_beta <= TOL and criterio_g <= TOL_G):
            beta_final = beta_next
            pf_final = pf_next
            iter_final = k
            alpha_final = alpha
            x_k = x_next
            convergiu = True
        else:
            x_k = x_next
            beta_k = beta_next
            k += 1
            if k >= 100:
                beta_final = beta_next; pf_final = pf_next
                iter_final = 99; alpha_final = alpha; x_k = x_next
                convergiu = True

_imprimir_resultados(x_k, alpha_final, beta_final, pf_final,
                    iter_final, viga, Es_det, L_det)

# =====
# CÁLCULO DOS COEFICIENTES PARCIAIS DE SEGURANÇA ( $\gamma$  e  $\phi$ )
# =====
print(f"{'COEFICIENTES PARCIAIS DE SEGURANÇA':^75}")
print("-" * 75)

bw_k = mu[0]
h_k = mu[1]
d_k = mu[2]
As_k = mu[3]

fc_k = mu[4] - 1.645 * sigma[4]
fy_k = np.exp(lam_fy - 1.645 * zeta_fy)

g_k = mu[6]
q_k = u_q - (1.0 / alpha_q) * np.log(-np.log(0.95))

gamma_bw = x_k[0] / bw_k
gamma_h = x_k[1] / h_k
gamma_d = x_k[2] / d_k
gamma_As = x_k[3] / As_k

phi_c = x_k[4] / fc_k
phi_s = x_k[5] / fy_k

gamma_g = x_k[6] / g_k
gamma_q = x_k[7] / q_k

print(f" bw (Largura) |  $\gamma_{bw}$  = {gamma_bw:.4f}")
print(f" h (Altura total) |  $\gamma_h$  = {gamma_h:.4f}")
print(f" d (Altura útil) |  $\gamma_d$  = {gamma_d:.4f}")
print(f" As (Área de aço) |  $\gamma_{As}$  = {gamma_As:.4f}")
print(f" fc (Concreto) |  $\phi_c$  = {phi_c:.4f}")
print(f" fy (Aço) |  $\phi_s$  = {phi_s:.4f}")
print(f" g (Perm.) |  $\gamma_g$  = {gamma_g:.4f}")
print(f" q (Variável) |  $\gamma_q$  = {gamma_q:.4f}")
print("=" * 75 + "\n")

# =====
# BLOCO 4: RELATÓRIO FINAL
# =====

def _imprimir_resultados(x_star, alpha, beta, pf, iters, viga, Es, L):
    nomes = ["bw (Largura)", "h (Altura total)", "d (Altura útil)",
            "As (Área de aço)", "fc (Concreto)", "fy (Aço)",
            "g (Perm.)", "q (Variável)"]
    unids = ["cm", "cm", "cm", "cm²", "kN/cm²", "kN/cm²", "kN/cm", "kN/cm"]
    distribs = ["Normal", "Normal", "Normal", "Normal", "Normal",
            "Normal", "Lognormal", "Normal", "Gumbel-máx"]

```

```

sep = "=" * 75

print(f"\n{sep}")
print(f"{'CONVERGÊNCIA - MÉTODO FORM-HLRF':^75}")
print(sep)
print(f"  Iterações até convergência : {iters+1}")
print(f"  Índice de Confiabilidade  $\beta$  : {beta:.6f}")
print(f"  Probabilidade de Falha Pf : {pf:.4e}")
print(sep)

print(f"\n{'PONTO DE PROJETO x* e FATORES DE SENSIBILIDADE  $\alpha^2$ ':^75}")
print("-" * 75)
print(f"{'Variável':<22} {'Distribuição':<14} {'x* (valor)':<14} "
      f"{'Unidade':<9} {' $\alpha^2$  (%)':<14}")
print("-" * 75)
for i, (nome, dist, val, unid) in enumerate(zip(nomes, distribs, x_star, unids)):
    contrib = alpha[i]**2 * 100.0
    print(f"  {nome:<20} {dist:<14} {val:<14.5f} {unid:<9} {contrib:>6.2f}%")
print("-" * 75)
print(f"  {'Es (det.)':<20} {'-':<14} {'Es:<14.1f} {'kN/cm²'}")
print(f"  {'L (det.)':<20} {'-':<14} {'L:<14.1f} {'cm'}")

bw, h, d, As, fc, fy, g_c, q_c = x_star
gl, mr, ms, x_ln, reg = viga.g_function(bw, h, d, fc, fy, As, g_c, q_c)

print(f"  Linha Neutra x          : {x_ln:>12.4f}  cm")
print(f"  Domínio de deformação    : {reg}")
print(sep + "\n")

if __name__ == "__main__":
    try:
        executar_form()
    except Exception:
        print("\n[ERRO CRÍTICO NA EXECUÇÃO]")
        traceback.print_exc()
    input("\nPressione Enter para fechar...")

```