

INSTITUTO FEDERAL DE EDUCAÇÃO,
CIÊNCIA E TECNOLOGIA DA PARAÍBA

COORDENAÇÃO DO CURSO SUPERIOR DE
BACHARELADO EM ENGENHARIA CIVIL



MONALIZA ARAÚJO PARNAÍBA

**ANÁLISE DA ESTABILIDADE DE COLUNAS COM VARIAÇÃO DA
SEÇÃO TRANSVERSAL E/OU MÓDULO DE ELASTICIDADE VIA
MÉTODO DAS DIFERENÇAS FINITAS**

Cajazeiras
2018

MONALIZA ARAÚJO PARNAÍBA

**ANÁLISE DA ESTABILIDADE DE COLUNAS COM VARIAÇÃO DA
SEÇÃO TRANSVERSAL E/OU MÓDULO DE ELASTICIDADE VIA
MÉTODO DAS DIFERENÇAS FINITAS**

Trabalho de Conclusão de Curso submetido à
Coordenação do Curso de Bacharelado em
Engenharia Civil do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba,
como parte dos requisitos para a obtenção do
grau de nível Superior em Engenharia Civil

Orientador: Prof. Me. Patrício Luiz de
Andrade

Cajazeiras
2018

MONALIZA ARAÚJO PARNAÍBA

**ANÁLISE DA ESTABILIDADE DE COLUNAS COM VARIAÇÃO DA
SEÇÃO TRANSVERSAL E/OU MÓDULO DE ELASTICIDADE VIA
MÉTODO DAS DIFERENÇAS FINITAS**

Trabalho de Conclusão de Curso submetido à
Coordenação do Curso de Bacharelado em
Engenharia Civil do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba,
como parte dos requisitos para a obtenção do
grau de nível Superior em Engenharia Civil

BANCA EXAMINADORA

Prof. Me. Patrício Luiz de Andrade - (Orientador)
Unidade Acadêmica de Indústria do IFPB

Prof. Me. Robson Arruda dos Santos - (Membro)
Unidade Acadêmica de Indústria do IFPB.

Prof. Me. Sebastião Simão da Silva - (Membro)
Unidade Acadêmica de Indústria do IFPB.

Cajazeiras, 13 de dezembro de 2018

Dedico este trabalho aos meus pais, José Quaresma Parnaíba e Evanilma Araújo Gomes Parnaíba, e a minha irmã, Mércia Araújo Parnaíba, que com muito amor não mediram esforços para que eu alcançasse mais esta etapa na minha vida.

Agradecimentos

A Deus, que pela sua infinita bondade me permitiu concluir este trabalho de conclusão de curso.

A toda minha família, pelo apoio e confiança que em mim foram depositados. À vocês, todo meu amor e gratidão eterna.

Aos meus colegas: John Williams Ferreira de Souza, Francisco Gonçalves Matias, Hayanne Macêdo de Mello e Elissandra Cheu Pereira do Nascimento por o todo suporte e companheirismo que foram fundamentais em minha trajetória acadêmica.

Ao meu professor orientador, Patrício Luiz de Andrade, pela orientação incansável, empenho e confiança. E além disso, pelo brilhante exemplo de docente e pesquisador.

A todos os docentes que em meu percurso acadêmico de graduação tive o privilégio de estudar, por todo o conhecimento técnico repassado servindo assim de ferramenta para minha formação profissional.

A instituição de ensino IFPB, pelo ambiente inspirador. E pela oportunidade de me permitir conhecer e apresentar uma das minha pesquisas no Instituto de Matemática Pura e Aplicada (IMPA), no Rio de Janeiro.

Ao departamento de Matemática, pela disponibilidade do espaço físico necessário para as orientações diárias.

Resumo

A análise estrutural é um campo específico da Engenharia Civil que trata da determinação dos efeitos de cargas sobre estruturas físicas e seus componentes. No dimensionamento de colunas, esta passa necessariamente por uma análise de estabilidade à flambagem. Por ser considerado um dos assuntos mais complexos da mecânica das estruturas, a flambagem é um fenômeno grave em estruturas esbeltas que precisa ser evitado, uma vez que a estrutura pode sofrer colapso sem aviso prévio. Os componentes estruturais, como colunas, com seções transversais variáveis e mudanças de materiais ao longo do seu comprimento são comuns em edifícios e pontes. Nesse sentido, torna-se de grande relevância o desenvolvimento de uma ferramenta numérica para análise de coluna com variação de seção transversal e módulo de elasticidade submetida à compressão axial. Com isso, este trabalho visa construir um simulador computacional via Método das Diferenças Finitas, implementado em linguagem de programação PYTHON, capaz de calcular a carga crítica de flambagem, ou seja, a capacidade de suporte de uma estrutura esbelta, solicitada por um esforço axial de compressão. Para tanto, como fundamentação teórica, inicialmente foi tomado por base o clássico Método de Leonard Euler, sendo posteriormente modificadas as considerações da coluna ideal abordada pelo o mesmo. Através dos resultados obtidos com as aplicações feitas para a validação da ferramenta computacional, verificou-se que a mesma foi capaz de resolver os problemas analisados, fornecendo resultados condizentes com os esperados.

Lista de Figuras

FIGURA 1.1 – Armadura de pilares com mudança de seção transversal.	16
FIGURA 2.1 – Composto original.	19
FIGURA 2.2 – Composto perturbado.	20
FIGURA 2.3 – Coluna suportada por pinos.	21
FIGURA 2.4 – Coluna defletida e diagrama de corpo livre.	22
FIGURA 2.5 – Coluna com (E) e (I) variando ao longo do comprimento.	24
FIGURA 3.1 – Definição de derivada.	26
FIGURA 3.2 – Aproximação da derivada por diferenças finitas progressivas.	26
FIGURA 3.3 – Aproximação da derivada por diferenças finitas regressivas.	27
FIGURA 3.4 – Aproximação da derivada por diferenças finitas centradas.	27
FIGURA 4.1 – Etapas e processos de aplicação do MDF.	38
FIGURA 4.2 – Coluna bi-rotulada de seção circular.	40
FIGURA 4.3 – Divisão da coluna bi-rotulada de seção circular para $n = 3$	40
FIGURA 4.4 – Resultados fornecidos pelo simulador para coluna bi-rotulada de seção circular com $n = 3$	41
FIGURA 4.5 – Coluna de pinos fixos com seção transversal cônica.	42
FIGURA 4.6 – Divisão da coluna de pinos fixos com seção transversal cônica para $n = 3$	42
FIGURA 4.7 – Resultados fornecido pelo simulador para coluna de seção transversal cônica com $n = 3$	43
FIGURA 4.8 – Coluna bi-rotulada de seção quadrada.	44

FIGURA 4.9 – Divisão da coluna bi-rotulada de seção quadrada de três tramos com $n = 4$	44
FIGURA 4.10 – Resultados fornecidos pelo simulador da coluna bi-rotulada de seção quadrada de três tramos com $n = 4$	46
FIGURA 5.1 – Gráfico de cargas da coluna de seção circular dois tramos.	48
FIGURA 5.2 – Gráfico de cargas da coluna de seção transversal cônica.	48
FIGURA 5.3 – Gráfico de cargas da coluna de seção quadrada de três tramos.	49
FIGURA A.1 – Coluna de seção circular de dois tramos.	53
FIGURA A.2 – Coluna com seção transversal cônica.	54
FIGURA A.3 – Coluna de seção quadrada de três tramos.	55
FIGURA A.4 – Fatoração LU.	56
FIGURA A.5 – Coluna de seção circular de dois tramos - Fatoração LU.	57
FIGURA A.6 – Coluna de seção transversal cônica - Fatoração LU.	58
FIGURA A.7 – Coluna de seção quadrada de três tramos - Fatoração LU.	59
FIGURA A.8 – Método da Secante.	60
FIGURA A.9 – Coluna de seção circular de dois tramos - Matrix A com $n = 23$	61
FIGURA A.10 – Coluna de seção circular de dois tramos - Vetor X com $n = 23$	62
FIGURA A.11 – Coluna de seção circular de dois tramos - Função y com $n = 23$	63
FIGURA A.12 – Coluna de seção circular de dois tramos - Carga Crítica com $n = 23$	64
FIGURA A.13 – Coluna com seção transversal cônica - Matrix A com $n = 27$	65
FIGURA A.14 – Coluna com seção transversal cônica - Vetor X com $n = 27$	66
FIGURA A.15 – Coluna com seção transversal cônica - Função y com $n = 27$	67
FIGURA A.16 – Coluna com seção transversal cônica - Carga Crítica com $n = 27$	68
FIGURA A.17 – Coluna de seção quadrada de três tramos - Matriz A com $n = 25$	69
FIGURA A.18 – Coluna de seção quadrada de três tramos - Vetor X com $n = 25$	70
FIGURA A.19 – Coluna de seção quadrada de três tramos - Função y com $n = 25$	71
FIGURA A.20 – Coluna de seção quadrada de três tramos - Carga Crítica com $n = 25$	72

Lista de Abreviaturas e Siglas

CAS	Álgebra computacional
EDF	Equação de Diferenças Finitas
EDO	Equação Diferencial Ordinária
GPL	General Public License
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
MDF	Método das Diferenças Finitas
PSF	Python Software Foundation
PVC	Problema de valor de contorno

Lista de Símbolos

P_{cr}	Carga crítica
$\Delta\theta$	Ângulo de deslocamento
P e P'	Força vertical
I	Momento de inércia
E	Módulo de elasticidade
K	Rigidez da mola
P_x	Componente horizontal
L	Comprimento da barra
v	Deflexão
M	Momento interno
F	Força restauradora
δ	Distância

Sumário

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Metodologia	15
1.4	Justificativa	15
1.5	Organização do trabalho	17
2	MODELAGEM FÍSICA E MATEMÁTICA	18
2.1	Instabilidade estrutural: o fenômeno de flambagem	18
2.2	Método de Euler para o cálculo da carga crítica	19
2.3	Fórmula da carga crítica com variação da seção transversal e/ou módulo de elasticidade	23
3	MÉTODO DAS DIFERENÇAS FINITAS	25
3.1	Aproximação da Derivada por Diferenças Finitas	25
3.1.1	Fórmulas de Diferença Progressiva, Regressiva e Central para a deri- vada primeira	26
3.2	Fórmulas de Diferenças Finitas Usando a Expansão em Série de Talyor	27
3.2.1	Fórmulas de Diferenças Finitas para a derivada primeira	28
3.2.2	Fórmulas de Diferenças Finitas para a derivada segunda	32
3.3	Resolução da EDO em estudo via Método das Diferenças Finitas	34

4	SIMULAÇÕES NUMÉRICAS	36
4.1	Linguagem de Programação PYTHON	36
4.2	Formulação numérica para utilizar o Método das Diferenças Finitas	37
4.3	Exemplos numéricos	39
4.3.1	Coluna de seção circular de dois tramos com variação de módulo de elasticidade (E) e momento de inércia (I).	40
4.3.2	Coluna com seção transversal cônica e módulo de elasticidade (E) constante.	42
4.3.3	Coluna de seção quadrada de três tramos com variação do momento de inércia (I)	44
5	RESULTADOS E DISCUSSÕES	47
5.1	Gráficos das cargas críticas	47
5.2	Fórmula de extrapolação	49
6	CONCLUSÃO	51
	REFERÊNCIAS	52
	APÊNDICE A – CÓDIGOS-FONTE	53

1 Introdução

1.1 Motivação

Os elementos estruturais podem ser classificados em lineares, bidimensionais e tridimensionais. Esta classificação advém de sua geometria, comparando a ordem de grandeza das três dimensões principais do elemento, que são elas comprimento, altura e espessura.

Os elementos lineares são aqueles em que o comprimento longitudinal é maior em pelo menos três vezes a maior dimensão da seção transversal, estes elementos também são chamados de “barras”. Os exemplos mais comuns são as vigas e as colunas.

As colunas são elementos estruturais lineares submetidos predominantemente a carregamentos axiais de compressão. A carga aplicada nas colunas poderá ser grande o suficiente para provocar um deslocamento ou uma oscilação lateral, denominada flambagem.

A flambagem é um fenômeno grave que ocorre comumente nas estruturas esbeltas em torno do eixo de sua seção transversal que tem menor momento de inércia, isso devido a um esforço de compressão axial. Esse fenômeno deve ser evitado, uma vez que resulta em “uma falha repentina e dramática de uma estrutura ou mecanismos e, por isso, é preciso dedicar especial atenção ao projeto de colunas para que estas possam suportar com segurança as cargas pretendidas”. (HIBBELER; SILVA, 2010)

A carga axial máxima que uma coluna pode suportar quando está na iminência de sofrer flambagem é denominada carga crítica (P_{cr}). As expressões que comumente calculam essa carga, por exemplo a expressão da carga crítica de Euler, são usadas para colunas cuja seção transversal e material não variam ao longo do comprimento. Casos em que há mudança brusca ou mesmo contínua do momento de inércia ou do módulo de elasticidade longitudinal do material não são contemplados pela expressão da carga crítica de Euler.

Caso deseje-se calcular a carga crítica cuja seção transversal e material variam ao longo do comprimento, precisaríamos, portanto, resolver uma EDO demasiadamente complicada. Essa análise pode ser possível se utilizarmos os métodos numéricos, encontrando assim uma solução aproximada para a EDO de interesse (BASTOS *et al.*, 2017)

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um código capaz de determinar a carga crítica de uma coluna com variação de seção transversal e módulo de elasticidade ao longo do comprimento.

1.2.2 Objetivos Específicos

- Descrever o fenômeno de flambagem em colunas;
- Compreender a linguagem de programação Python a ser utilizada na implementação;
- Resolver a EDO em estudo via Método das Diferenças Finitas;
- Desenvolver um código e a partir do mesmo obter valores de cargas críticas cuja coluna em análise apresente seção transversal e/ou módulo de elasticidade variáveis ao longo do comprimento.

1.3 Metodologia

Trata-se de uma pesquisa descritiva, envolvendo um levantamento bibliográfico sobre o fenômeno de flambagem segundo os autores (ARBABI; ASCE, 1991) e (BASTOS *et al.*, 2017), também sobre o Método das Diferenças Finitas segundo (GILAT; SUBRAMANIAM, 2008.) e sobre a linguagem de programação PYTHON segundo o autor (ROSSUM, 2005). Após a construção de um arcabouço bibliográfico, foi possível resolver a EDO em estudo via Método das Diferenças Finitas e desenvolver o código capaz de calcular a carga crítica de flambagem de colunas variação de momento de inércia e/ou módulo de elasticidade. Para validação do código, foi extraído exemplos do autor (BASTOS *et al.*, 2017) e realizada a plotagem dos gráficos das cargas críticas. Por último, foi calculada a carga crítica segundo a fórmula de extrapolação.

1.4 Justificativa

Segundo (HIBBELER; SILVA, 2010), quando se projeta um elemento estrutural, é imprescindível que o mesmo satisfaça requisitos específicos de resistência, deslocamento e estabilidade. Dessa forma, para o dimensionamento de colunas será indispensável analisar a estabilidade elástica. Isto é, sua capacidade de suportar determinado carregamento sem sofrer uma mudança abrupta em sua configuração.

Para (ARBABI; ASCE, 1991) “componentes com seções transversais variáveis são comuns em edifícios e pontes, bem como em peças de máquinas”. Nesse sentido, (LONGO, 2000) afirma que na prática, as seções transversais dos pilares em edifícios podem sofrer variações. Ou seja, pode-se adotar dimensões maiores nos andares de baixo e menores para os de cima, graças a carga vertical que é maior nos primeiros andares e menor nos últimos. Assim, tomamos como exemplo, um pilar de um prédio com 12 andares que pode ter 40 cm x 60 cm do 1° ao 4° pavimento, 30 cm x 50 cm do 5° ao 8° e 20 cm x 40 cm do 9° ao 12° pavimento, como podemos ver na Figura 1.1.

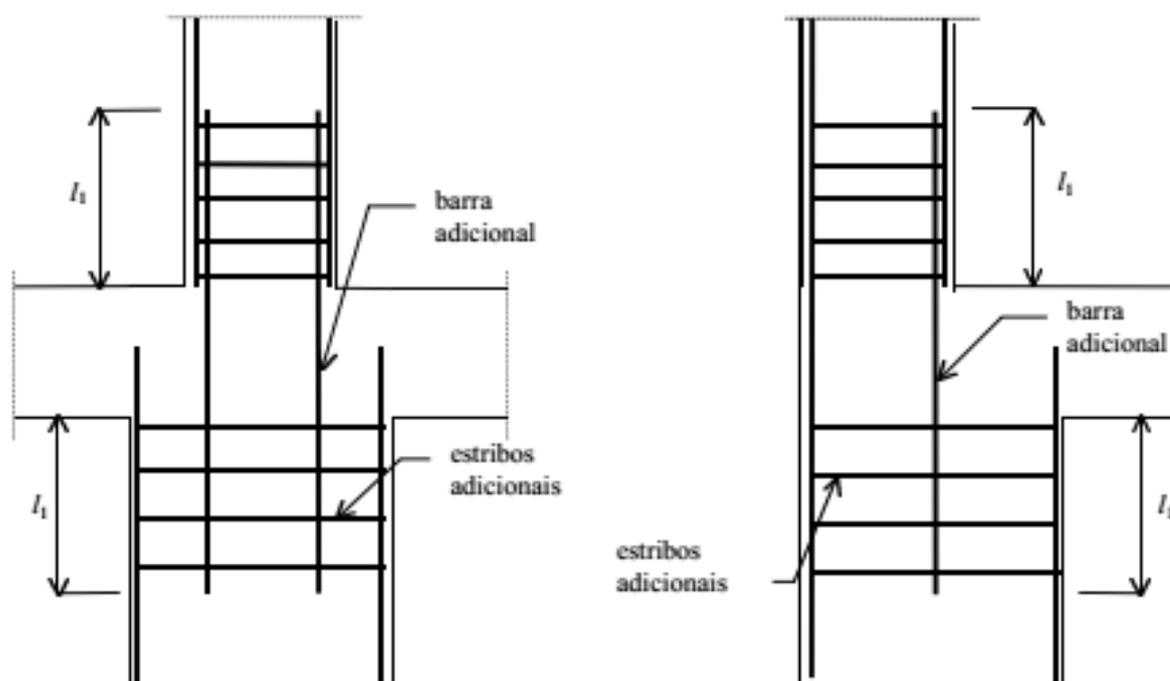


FIGURA 1.1 – Armadura de pilares com mudança de seção transversal.

Fonte: Longo (2000)

As colunas são elementos estruturais considerados de sustentação, e quase sempre verticais. Ao longo da história da arquitetura, assumiu as formas mais variadas e diversos ornamentos, podendo ser de pedra, alvenaria, madeira ou metal. Assim, torna-se importante a construção de uma ferramenta computacional capaz de calcular a carga crítica em elementos esbeltos cujo módulo de elasticidade (E) e o momento de inércia (I) variam ao longo da coluna. E a partir do cálculo, obter ganho de produtividade e redução de falhas estruturais.

O dimensionamento de colunas passa necessariamente por uma análise de estabilidade à flambagem, fenômeno geralmente relacionado a cargas axiais compressivas excessivas que pode conduzir essas estruturas ao súbito colapso. Logo, sua prevenção é de extrema importância e pode ser feita através da determinação do esforço axial máximo que o elemento estrutural suporta, a carga crítica.

É notável também que o uso de ferramentas computacionais para a resolução de problemas de engenharia se tornou imprescindível, pois com elas se ganha em termos de produtividade e garante-se a redução de falhas das operações de cálculo manuais. Logo, podem prestar grande auxílio na resolução do problema referido acima.

Portanto, esse trabalho trata de implementar na linguagem de programação PYTHON um código capaz de fornecer o valor da carga crítica de flambagem.

1.5 Organização do trabalho

O trabalho é composto por seis capítulos. O primeiro consiste de uma introdução que norteia a discussão sobre o fenômeno de instabilidade lateral de colunas. Além disso, consta com a justificativa do estudo, os objetivos (geral e específicos) e a organização do trabalho.

O segundo capítulo apresenta brevemente o fenômeno de flambagem e o Método de Euler para o cálculo da carga crítica em colunas ideais. Em seguida, é discutido sobre a obtenção da carga crítica quando tratamos a seção transversal e o módulo de elasticidade variáveis.

O terceiro capítulo aborda sobre o Método das Diferenças Finitas como ferramenta para a solução da EDO em estudo, cujo procedimento numérico consiste na discretização do contínuo em um sistema discreto composto.

O capítulo quatro apresenta as simulações numéricas utilizando exemplos da literatura a título de validação da ferramenta desenvolvida.

Já o capítulo cinco apresenta a discussão dos principais resultados obtidos a partir das análises feitas através do programa referido e por último, o capítulo seis consiste em considerações finais e recomendações para trabalhos futuros.

2 Modelagem Física e Matemática

Este capítulo aborda brevemente o fenômeno da instabilidade lateral em colunas e a sua relevância para estudo do dimensionamento desse tipo de estrutura. Nele também é apresentado o Método de Euler para o cálculo da carga crítica. Por fim, é discutido sobre a obtenção da carga crítica quando tratamos de colunas com a seção transversal e/ou módulo de elasticidade variáveis, diferente das considerações feitas por Leonhard Euler para uma coluna ideal.

2.1 Instabilidade estrutural: o fenômeno de flambagem

As colunas são elementos estruturais de seção delgada que comumente são submetidos a esforços axiais de compressão. Quando esses esforços levam à flexão da coluna dizemos que ocorreu instabilidade estrutural, comumente chamado de fenômeno de flambagem ou curvatura. Nesse sentido, segundo (BRACAL; JUNIOR, 2012):

A maneira como a flambagem ocorre depende do modo como a estrutura é carregada, de suas propriedades geométricas e materiais. O conceito de estabilidade está diretamente ligado ao conceito de equilíbrio de uma estrutura, que é caracterizada pelos valores dos deslocamentos dos seus nós. O surgimento da instabilidade é caracterizado por uma bifurcação de equilíbrio (instabilidade bifurcacional) ou pela ocorrência de um ponto limite (instabilidade por snap-through). A instabilidade por ‘snap-through’ ocorre quando a carga crítica é alcançada e mantida sobre a estrutura, dessa forma a estrutura se deforma e entra em colapso instantaneamente.

A curvatura é considerada uma instabilidade elástica, quando a peça pode perder sua estabilidade sem que o material já tenha atingido a sua tensão de escoamento. Este colapso ocorrerá sempre na direção do eixo de menor momento de inércia de sua seção transversal. A tensão crítica para ocorrer a flambagem não depende da tensão de escoamento do material, mas do seu módulo de Young (E).

Os sistemas mecânicos e estruturas em geral quando estão submetidos a carregamentos, podem falhar de várias formas, o que vai depender do material usado, do tipo de estrutura, das condições de apoio, entre outras considerações. Assim, quando se projeta um elemento, é necessário que ele satisfaça requisitos específicos de tensão, deslocabilidade e estabilidade.

2.2 Método de Euler para o cálculo da carga crítica

A carga axial máxima que uma coluna pode suportar quando está na iminência de sofrer flambagem é denominada carga crítica, (P_{cr}). Qualquer carga adicional provocará flambagem na coluna e, portanto, deslocamento lateral.

Para entender melhor a natureza dessa instabilidade, considere um modelo simplificado de duas barras rígidas AC e CB conectadas em C por um pino e uma mola de torção de constante K , como pode ser visto na Figura 2.1. (BEER *et al.*, 2011)

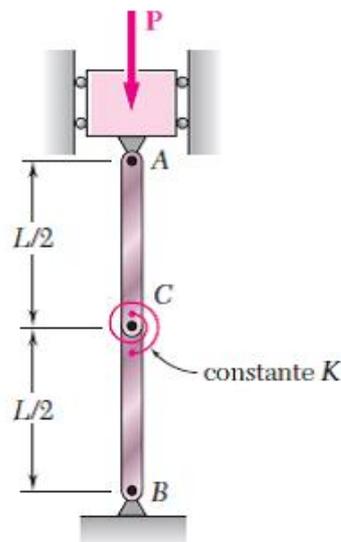


FIGURA 2.1 – Composto original.

Fonte: Beer, *et al.* (2011)

Quando as barras estão na posição vertical, a mola, de rigidez K , não está esticada e duas pequenas forças verticais P e P' , cujo os valores em módulo são equivalente, são aplicadas nas extremidades da coluna, conforme a Figura 2.2.

Suponha que movamos C ligeiramente para a direita, até uma pequena distância δ de maneira que cada barra forme agora um pequeno ângulo $\Delta\theta$ com a vertical. Quando as barras são deslocadas, a mola produz uma força de recuperação $F = K\delta$. Como $\text{tg}\Delta\theta = \frac{\delta}{L}$ e visto que $\Delta\theta$ é muito pequeno, então $\text{tg}\Delta\theta = \Delta\theta$, assim, $\delta = \frac{L\Delta\theta}{2}$. Logo, a

força de restauração da mola torna-se: $F = K \frac{L\Delta\theta}{2}$.

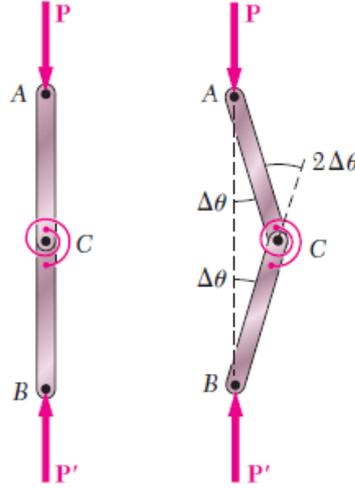


FIGURA 2.2 – Composto perturbado.

Fonte: Beer, *et al.* (2011)

Além disso, as cargas P e P' desenvolvem componentes horizontais do tipo: $P_x = Ptg\Delta\theta$ e $P'_x = P'tg\Delta\theta$. Como $P = P'$, assim $P_x = P'_x$ e fazendo o equilíbrio de força horizontais, tem-se $F = 2P_x$. Logo, $K \left(\frac{L\Delta\theta}{2} \right) = 2Ptg\Delta\theta$.

Se a força de restauração for maior que a força perturbadora, isto é, $K \left(\frac{L\Delta\theta}{2} \right) > 2Ptg\Delta\theta$, e observando que θ é cancelado, poderemos encontrar P , o que dá:

$$P < \frac{KL}{4} \quad (2.1)$$

Essa é uma condição para equilíbrio estável, visto que a força desenvolvida pela mola seria adequada para devolver as barras a suas respectivas posições verticais. Por outro lado, se $K \left(\frac{L\Delta\theta}{2} \right) < 2Ptg\Delta\theta$, o mecanismo estaria em equilíbrio instável, ou seja,

$$P > \frac{KL}{4} \quad (2.2)$$

Em outras palavras, se essa carga P for aplicada e ocorrer um leve deslocamento em C , o mecanismo tenderá a sair do equilíbrio e não retornar a sua posição original. O valor intermediário de P , definido pelo requisito $K \left(\frac{L\Delta\theta}{2} \right) = 2Ptg\Delta\theta$, é a carga crítica. Aqui,

$$P = \frac{KL}{4} \quad (2.3)$$

Essa carga representa um caso de mecanismo que está em equilíbrio neutro. Como P_{cr} é independente do pequeno deslocamento $\Delta\theta$ das barras, qualquer leve perturbação

adicional aplicada ao mecanismo fará com que ele se afaste mais do equilíbrio, e não retorne a sua posição original. Neste caso, as barras permanecerão na posição defletida.

Determinaremos agora a carga crítica de flambagem para uma coluna suportada por pinos, como mostra a Figura 2.3. Consideraremos a coluna perfeitamente reta antes da carga, feita de material homogêneo e na qual a carga é aplicada no centróide da seção transversal. Além disso, supomos que o material comporta-se de uma maneira linear elástica e que a coluna sofre flambagem ou flexão em um único plano.

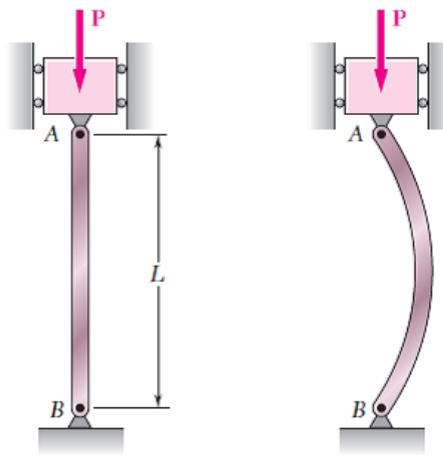


FIGURA 2.3 – Coluna suportada por pinos.

Fonte: Beer, *et al.* (2011)

O fato da coluna continuar estável ou torna-se estável quando sujeita a uma carga axial, dependerá de sua capacidade de restauração, que é baseada em sua resistência à flexão.

Por consequência, para determinar a carga crítica e a forma da coluna quando flambada, aplicaremos a Equação (2.4) também chamada de equação da linha elástica que relaciona o momento interno M na coluna com sua forma defletida, isto é, a coluna sofre flambagem ou flexão em um único plano.

$$EI \frac{d^2v}{dx^2} = M \quad (2.4)$$

A Equação (2.4) considera que a inclinação da curva elástica seja pequena e que as deflexões ocorrem somente por flexão. Quando a coluna está em posição fletida como pode ser visto na Figura 2.4, o momento fletor interno pode ser determinado pelo método das seções. O diagrama de corpo livre de um segmento na posição fletida também é mostrado na Figura 2.4.

Tanto a deflexão v quanto o momento interno M são mostrados na direção positiva, de acordo com a convenção de sinal usada para estabelecer a Equação (2.4). Somando momentos, o momento interno é $M = -Pv$.

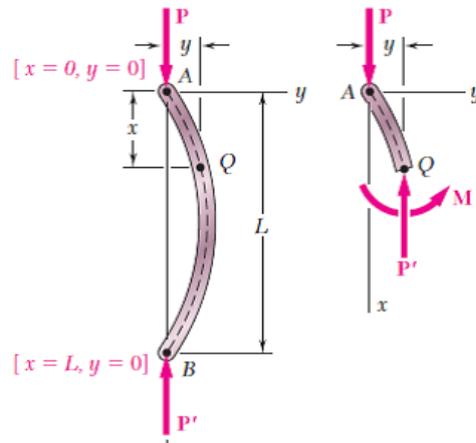


FIGURA 2.4 – Coluna defletida e diagrama de corpo livre.

Fonte: Beer, *et al.* (2011)

Assim,

$$EI \frac{d^2v}{dx^2} = -Pv \quad (2.5)$$

$$\frac{d^2v}{dx^2} + \left(\frac{P}{EI} v \right) = 0 \quad (2.6)$$

Essa é uma equação diferencial linear homogênea de segunda ordem com coeficientes constantes, cuja dedução pode ser encontrada, por exemplo, em (ZILL; CULLEN, 2012) ou (BOYCE; DIPRIMA, 2015). A solução geral dessa da Equação (2.6) é:

$$v = C_1 \sin \left(\sqrt{\frac{P}{EI}} x \right) + C_2 \cos \left(\sqrt{\frac{P}{EI}} x \right) \quad (2.7)$$

As duas constantes de integração são determinadas pelas condições de contorno nas extremidades da coluna. Visto que $v = 0$ em $x = 0$, então $C_2 = 0$. E, considerando $v = 0$ em $x = L$, obtemos:

$$C_1 \sin \left(\sqrt{\frac{P}{EI}} L \right) = 0 \quad (2.8)$$

Essa equação é satisfeita se $C_1 = 0$, porém, neste caso, $v = 0$, o que é uma solução trivial que exige que a coluna permanecerá sempre reta, ainda que a carga faça com que a coluna torne-se instável. A outra possibilidade é:

$$\sin \left(\sqrt{\frac{P}{EI}} L \right) = 0 \quad (2.9)$$

que é satisfeita se:

$$\left(\sqrt{\frac{P}{EI}} L \right) = n\pi \quad (2.10)$$

ou

$$P_{cr} = \frac{n^2 \pi^2 EI}{L^2}, n = 0, 1, 2, 3, \dots \quad (2.11)$$

Se considerarmos $n = 0$ significa que nenhuma carga está sendo aplicada, assim, o menor valor de P é obtido quando $n = 1$, de modo que a carga crítica para a coluna é, portanto,

$$P_{cr} = \frac{\pi^2 EI}{L^2} \quad (2.12)$$

sendo:

P_{cr} = carga crítica ou carga axial máxima na coluna imediatamente antes do início da flambagem;

E = módulo de elasticidade para o material;

I = menor momento de inércia para a área da seção transversal da coluna;

L = comprimento da coluna sem apoio, cujas extremidades estejam presas por pinos.

2.3 Fórmula da carga crítica com variação da seção transversal e/ou módulo de elasticidade

A expressão da carga crítica mostrada na seção anterior é usada para coluna cuja seção transversal e material não variam ao longo do comprimento. Casos em que há mudança brusca ou mesmo contínua do momento de inércia ou do módulo de elasticidade longitudinal do material não são contemplados pela expressão da carga crítica de Euler, como ilustra a Figura 2.5. (BASTOS *et al.*, 2017)

Desta forma, precisamos resolver uma EDO com coeficientes variáveis:

$$\frac{d}{dx} \left[E(x)I(x) \frac{dy(x)}{dx} \right] + Py(x) = 0 \quad (2.13)$$

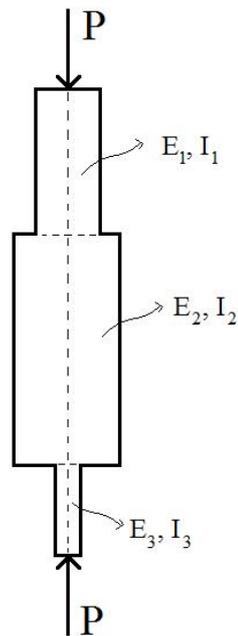


FIGURA 2.5 – Coluna com (E) e (I) variando ao longo do comprimento.

Fonte: Batos, *et al.* (2017)

Soluções para EDO's do tipo apresentada na Equação (2.13), muitas vezes, podem ser obtidas na forma de séries de potências, porém, pode ocorrer dificuldades em lidar com soluções dadas através de somas infinitas. Alternativamente, podemos resolver a equação acima utilizando-se um procedimento numérico a fim de se encontrar uma solução aproximada para tal EDO.

3 Método das Diferenças Finitas

Algumas equações diferenciais são tão complexas que a sua resolução analítica torna-se impossível, sendo necessário encontrar um valor aproximado para sua solução. Geralmente, são empregados dois procedimentos para tal resolução. Um desses procedimentos consiste em simplificar a equação diferencial de modo a permitir resolvê-la analiticamente. Pode-se então utilizar a solução da equação simplificada para aproximá-la da equação original. Já o segundo procedimento, e mais utilizado, consiste aproximar a solução do problema original por métodos numéricos computacionais. Portanto, neste capítulo é apresentado o Método das Diferenças Finitas e sua aplicação na resolução da EDO apresentada na Equação (2.13).

3.1 Aproximação da Derivada por Diferenças Finitas

O Método das Diferenças Finitas converte um problema de valor de contorno (PVC) em sistemas de equações algébricas substituindo todas as derivadas por aproximações baseadas em diferenças finitas. O objetivo desse método é transformar o problema composto por equações diferenciais em um problema formado por equações algébricas.

Seja $f'(x)$ a derivada de uma função $f(x)$ no ponto $x = a$, a derivada é definida como:

$$\left. \frac{df(x)}{dx} \right|_{x=a} = f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (3.1)$$

Graficamente, a definição é ilustrada na Figura 3.1. A derivada é o valor da inclinação da reta tangente à função em $x = a$. A derivada é obtida com a escolha de um ponto x próximo a $x = a$ e o cálculo da inclinação da reta que conecta os dois pontos.

A precisão do cálculo da derivada feito dessa forma aumenta à medida que o ponto x se aproxima do ponto a . No limite em que o ponto x tende ao ponto a , a derivada é a inclinação da reta tangente a $f(x)$ em $x = a$. Na aproximação de derivadas usando diferenças finitas, valores da função em diferentes pontos na vizinhança do ponto $x = a$ são usados na estimativa da inclinação. Existem várias fórmulas de aproximação por

diferenças finitas. Três dessas fórmulas, nas quais a derivada é calculada a partir dos valores de dois pontos, são apresentadas neste capítulo.



FIGURA 3.1 – Definição de derivada.

Fonte: Gilat; Subramaniam (2008)

3.1.1 Fórmulas de Diferença Progressiva, Regressiva e Central para a derivada primeira

As fórmulas de diferenças finitas progressiva, regressiva e central são as mais simples aproximações da derivada por diferenças finitas. Nessas aproximações, ilustradas nas Figuras 3.2, 3.3 e 3.4, a derivada no ponto (x_i) é calculada a partir do valor de dois pontos. A derivada é estimada como a inclinação da reta que conecta esses dois pontos.

- A diferença progressiva é a inclinação da reta que conecta os pontos $(x_i, f(x_i))$ e $(x_{i+1}, f(x_{i+1}))$:

$$\left. \frac{df(x)}{dx} \right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (3.2)$$

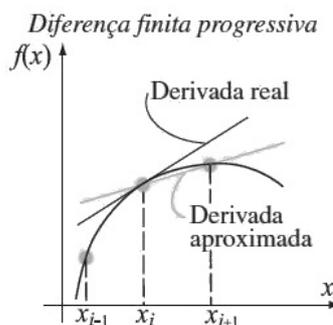


FIGURA 3.2 – Aproximação da derivada por diferenças finitas progressivas.

Fonte: Gilat; Subramaniam (2008)

- A diferença regressiva é a inclinação da reta que conecta os pontos $(x_{i-1}, f(x_{i-1}))$ e $(x_i, f(x_i))$:

$$\left. \frac{df(x)}{dx} \right|_{x=x_i} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (3.3)$$

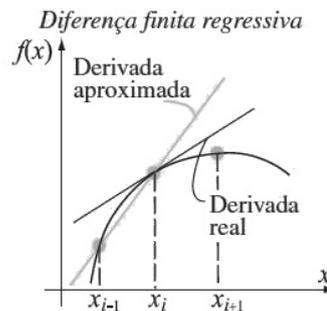


FIGURA 3.3 – Aproximação da derivada por diferenças finitas regressivas.

Fonte: Gilat; Subramaniam (2008)

- A diferença central é a inclinação da reta que conecta os pontos $(x_{i-1}, f(x_{i-1}))$ e $(x_{i+1}, f(x_{i+1}))$:

$$\left. \frac{df(x)}{dx} \right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} \quad (3.4)$$

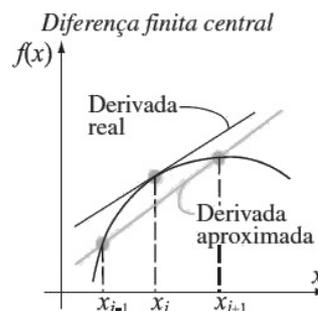


FIGURA 3.4 – Aproximação da derivada por diferenças finitas centradas.

Fonte: Gilat; Subramaniam (2008)

3.2 Fórmulas de Diferenças Finitas Usando a Expansão em Série de Talyor

Para (GILAT; SUBRAMANIAM, 2008.) as fórmulas de diferenças finitas progressiva, regressiva e central, bem como muitas outras fórmulas usadas para calcular derivadas de

forma aproximada, podem ser deduzidas a partir da expansão em série de Taylor. Essas fórmulas fornecem uma estimativa da derivada em um ponto usando valores de pontos em sua vizinhança. O número de pontos usados nos cálculos varia com a fórmula, e os pontos podem estar à frente, atrás ou em ambos os lados do ponto onde se calcula a derivada. Uma vantagem do uso da expansão em série de Taylor na dedução das fórmulas está no fato dela também fornecer uma estimativa do erro de truncamento presente na aproximação.

3.2.1 Fórmulas de Diferenças Finitas para a derivada primeira

Várias fórmulas usadas na aproximação da derivada primeira no ponto x_i com base nos valores de pontos próximos a x_i são deduzidas usando a expansão em série de Taylor. Todas as fórmulas deduzidas nesta seção se referem ao caso no qual os pontos estão igualmente espaçados.

- Fórmula de diferença finita progressiva com dois pontos para a derivada primeira:

O valor da função no ponto x_{i+1} pode ser aproximado usando a série de Taylor em termos do valor da função e de suas derivadas no ponto x_i :

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)h^2}{2!} + \frac{f'''(x_i)h^3}{3!} + \dots \quad (3.5)$$

onde $h = x_{i+1} - x_i$ é o espaçamento entre os pontos. Usando a expansão da série de Taylor com dois termos e um resíduo, a Equação (3.5) pode ser reescrita como:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(\zeta)}{2!}h^2 \quad (3.6)$$

onde ζ é um valor de x entre x_i e x_{i+1} .

Resolvendo a Equação (3.6) para $f'(x_i)$, obtém-se:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(\zeta)}{2!}h \quad (3.7)$$

Um valor aproximado pode agora ser calculado para a derivada $f'(x_i)$ se o segundo termo no lado direito da Equação (3.7) for ignorado, o que introduz um erro de truncamento (discretização). Como esse termo é proporcional a h , diz-se que o erro de truncamento é da ordem de h (escrito como $O(h)$):

$$\frac{f''(\zeta)}{2!}h = O(h) \quad (3.8)$$

Deve-se frisar aqui que a ordem de grandeza do erro de truncamento não é de fato conhecida, já que não se sabe o valor de $f''(\zeta)$. Entretanto, a Equação (3.8) é valiosa por sugerir que um menor h resulta em um erro menor. Além disso, conforme mostrado mais adiante neste capítulo, ela possibilita a comparação da ordem de grandeza do erro presente em diferentes fórmulas de diferenças finitas.

Usando a notação da Equação (3.8), o valor aproximado da derivada primeira é:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \quad (3.9)$$

A aproximação da Equação (3.9) é igual à fórmula de diferença progressiva apresentada na Equação (3.2).

- Fórmula de diferença finita regressiva com dois pontos para a derivada primeira

A fórmula de diferença finita regressiva também pode ser deduzida com a aplicação da expansão em série de Taylor. O valor da função no ponto x_{i-1} também pode ser aproximado usando a série de Taylor em termos do valor da função e de suas derivadas no ponto x_i :

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)h^2}{2!} - \frac{f'''(x_i)h^3}{3!} + \dots \quad (3.10)$$

onde $h = x_i - x_{i-1}$ é o espaçamento entre os pontos. Usando a expansão da série de Taylor com dois termos e um resíduo, a Equação (3.10) pode ser reescrita como:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(\zeta)h^2}{2!} \quad (3.11)$$

onde ζ é um valor de x entre x_{i-1} e x_i . Resolvendo a Equação (3.11) para $f'(x_i)$, obtém-se:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{f''(x_i)h}{2!} \quad (3.12)$$

Um valor aproximado pode agora ser calculado para a derivada $f'(x_i)$ se o segundo termo no lado direito da Equação (3.12) for ignorado. Isso resulta em:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h) \quad (3.13)$$

A aproximação da Equação (3.13) é igual à fórmula de diferença finita regressiva apresentada na Equação (3.3)

- Fórmula de diferença finita central com dois pontos para a derivada primeira

A fórmula de diferença finita central também pode ser deduzida usando três termos na série de Taylor e um resíduo. O valor da função no ponto x_{i+1} em termos do valor da função e de suas derivadas no ponto x_i é dado por:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\zeta_1)}{3!}h^3 \quad (3.14)$$

onde ζ_1 é um valor de x entre x_i e x_{i+1} . O valor da função no ponto x_{i-1} em termos do valor da função e de suas derivadas no ponto x_i é dado por:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(\zeta_2)}{3!}h^3 \quad (3.15)$$

onde ζ_2 é um valor de x entre x_{i-1} e x_i . Nas duas últimas equações, o espaçamento dos intervalos é igual, de forma que $h = x_{i+1} - x_i = x_i - x_{i-1}$. Subtraindo a Equação (3.15) da Equação (3.14), obtém-se:

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + \frac{f'''(\zeta_1)}{3!}h^3 + \frac{f'''(\zeta_2)}{3!}h^3 \quad (3.16)$$

Uma estimativa para a derivada primeira é obtida resolvendo-se a Equação (3.16) para $f'(x_i)$ sem considerar os resíduos, o que introduz um erro de truncamento da ordem de h^2 :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2) \quad (3.17)$$

A aproximação na Equação (3.17) é igual à fórmula de diferença central na Equação (3.4) para intervalos igualmente espaçados. Uma comparação entre as Equações (3.13), (3.13) e (3.17) mostra que, nas diferenças finitas progressiva e regressiva, o erro de truncamento é da ordem de h , enquanto na aproximação por diferença central o erro de truncamento é da ordem de h^2 . Isso indica que a aproximação por diferença central fornece uma aproximação mais precisa para a derivada. Isso pode ser observado esquematicamente nas Figuras 3.2, 3.3, 3.4, onde a inclinação da reta que representa a derivada na aproximação pela diferença central parece estar mais próxima da inclinação da reta tangente do que as retas referentes às aproximações progressiva e regressiva.

- Fórmulas de diferenças finitas progressiva e regressiva com três pontos para a derivada primeira

As fórmulas de diferenças finitas progressiva e regressiva dadas nas Equações (3.9) e (3.13) fornecem uma estimativa para a derivada primeira com um erro de truncamento de $O(h)$.

A fórmula de diferença progressiva avalia a derivada no ponto x_i com base nos valores nesse ponto e no ponto imediatamente à sua direita, x_{i+1} . A fórmula de diferença finita regressiva avalia a derivada no ponto x_i com base nos valores nesse ponto e naquele imediatamente à sua esquerda, x_{i+1} . Claramente, a fórmula de diferença finita progressiva pode ser usada na avaliação da derivada primeira do primeiro ponto x_1 e em todos os pontos internos, enquanto a fórmula de diferença finita regressiva pode ser usada na avaliação da derivada primeira no último ponto e em todos os pontos internos. A fórmula de diferença central, dada pela Equação (3.17), fornece uma estimativa para a derivada primeira com um erro de $O(h^2)$. A fórmula de diferença central avalia a derivada primeira em um dado ponto x_i usando os pontos x_{i-1} e x_{i+1} . Conseqüentemente, para uma função dada por um conjunto discreto de n pontos, a fórmula da diferença central pode ser usada apenas nos pontos internos e não nos pontos finais (x_1 ou x_n). Uma estimativa para a derivada primeira nos pontos finais, com erro de $O(h^2)$, pode ser calculada com fórmulas de diferenças finitas progressiva e regressiva com três pontos, deduzidas a seguir.

A fórmula de diferença finita progressiva com três pontos calcula a derivada no ponto x_i usando o valor da função nesse ponto e nos dois pontos seguintes, x_{i+1} e x_{i+2} . Assume-se que os pontos estejam igualmente espaçados, logo $h = x_{i+2} - x_{i+1} - x_i$ (o procedimento também pode ser aplicado em pontos não-uniformemente espaçados). A dedução da fórmula começa usando-se três termos da expansão em série de Taylor com um resíduo para escrever o valor da função nos pontos x_{i+1} e x_{i+2} em termos do valor da função e de suas derivadas no ponto x_i :

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\zeta_1)}{3!}h^3 \quad (3.18)$$

$$f(x_{i+2}) = f(x_i) + f'(x_i)2h + \frac{f''(x_i)}{2!}(2h)^2 + \frac{f'''(\zeta_2)}{3!}(2h)^3 \quad (3.19)$$

onde ζ_1 é um valor de x entre x_i e x_{i+1} , e ζ_2 é um valor de x entre x_i e x_{i+2} . As Equações (3.18) e (3.19) são em seguida combinadas de tal forma que os termos com a derivada segunda desapareçam. Isso é feito multiplicando a Equação (3.18) por 4 e subtraindo a Equação (3.19):

$$4f(x_{i+1}) - f(x_{i+2}) = 3f(x_i) + 2f'(x_i)h + \frac{4f'''(\zeta_1)}{3!}h^3 - \frac{f'''(\zeta_2)}{3!}(2h)^3 \quad (3.20)$$

Uma estimativa para a derivada primeira é obtida resolvendo a Equação (3.20) para

$f'(x_i)$ sem considerar os resíduos, o que introduz um erro de truncamento da ordem de h^2 :

$$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h} + O(h^2) \quad (3.21)$$

A Equação (3.21) é a fórmula de diferença finita progressiva com três pontos que estima a derivada primeira no ponto x_i usando o valor da função nesse ponto e nos dois pontos seguintes, x_{i+1} e x_{i+2} , com um erro de $O(h^2)$. Essa fórmula pode ser usada para calcular a derivada do primeiro ponto de uma função descrita por um conjunto discreto de n pontos. A fórmula de diferença finita regressiva com três pontos calcula a derivada no ponto x_i usando o valor desse ponto e dos dois pontos anteriores, x_{i-1} e x_{i-2} . A fórmula é deduzida da mesma forma que a Equação (3.21). A expansão em série de Taylor com três termos e um resíduo é escrita para o valor da função nos pontos x_{i-1} e x_{i-2} em termos do valor da função e de suas derivadas no ponto x_i . As equações são então manipuladas para se obter uma equação sem os termos das derivadas segundas, que é então resolvida para $f'(x_i)$. A fórmula obtida é:

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + O(h^2) \quad (3.22)$$

onde $h = x_i - x_{i-1} = x_{i-1} - x_{i-2}$ é a distância entre os pontos.

3.2.2 Fórmulas de Diferenças Finitas para a derivada segunda

As expressões baseadas em diferenças centrais, em diferenças progressivas unilaterais e em diferenças regressivas unilaterais são apresentadas para o cálculo da derivada segunda no ponto x_i .

- Fórmula de diferença central com três pontos para a derivada segunda

Fórmulas de diferença central podem ser desenvolvidas para a derivada segunda usando qualquer número de pontos em cada lado de x_i , no qual se deseja avaliar a derivada. As fórmulas são deduzidas escrevendo-se a expansão em série de Taylor com pontos em ambos os lados de x_i em termos do valor da função e de suas derivadas nesse ponto. Em seguida, as equações são combinadas de tal forma que os termos contendo as derivadas primeiras sejam eliminados. Por exemplo, para pontos x_{i+1} e x_{i-1} , a expansão em série de Taylor com um resíduo é:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^4(\zeta_1)}{4!}h^4 \quad (3.23)$$

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^4(\zeta_2)}{4!}h^4 \quad (3.24)$$

onde ζ_1 é um valor de x entre x_i e x_{i+1} e ζ_2 é um valor de x entre x_i e x_{i-1} . Somando as Equações. (3.23) e (3.24), obtém-se:

$$f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + 2\frac{f''(x_i)}{2!}h^2 + \frac{f^4(\zeta_1)}{4!}h^4 + \frac{f^4(\zeta_2)}{4!}h^4 \quad (3.25)$$

Pode-se obter uma estimativa para a derivada segunda resolvendo a Equação (3.25) para $f''(x_i)$ sem considerar os resíduos. Isso introduz um erro de truncamento da ordem de h^2 .

$$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1}))}{h^2} + O(h^2) \quad (3.26)$$

A Equação (3.23) é a fórmula de diferença central com três pontos que estima a derivada segunda no ponto x_i usando o valor da função nesse ponto, no ponto anterior, x_{i-1} , e no ponto seguinte, x_{i+1} , com um erro de truncamento de $O(h^2)$.

O mesmo procedimento pode ser usado para desenvolver uma fórmula de ordem superior (quarta ordem) mais precisa, envolvendo os pontos $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$:

$$f''(x_i) = \frac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2}))}{12h^2} + O(h^4) \quad (3.27)$$

- Fórmulas de diferenças progressiva e regressiva com três pontos para a derivada segunda

A fórmula de diferença progressiva com três pontos que estima a derivada segunda no ponto x_i usando o valor da função nesse ponto e nos dois pontos seguintes, x_{i+1} e x_{i+2} , é desenvolvida multiplicando-se a Equação (3.23) por 2 e subtraindo-a da Equação (3.24). A equação resultante é então resolvida para $f''(x_i)$:

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2}))}{h^2} + O(h) \quad (3.28)$$

A fórmula de diferença regressiva com três pontos que estima a derivada segunda no ponto x_i usando o valor da função nesse ponto e nos dois pontos anteriores, x_{i+1} e x_{i+2} , é deduzida de forma similar. Isto é feito escrevendo-se a expansão em série de Taylor com três termos e um resíduo para o valor da função nos pontos x_{i+1} e x_{i+2} em termos do

valor da função e de suas derivadas no ponto x_i . As equações são em seguida manipuladas para se obter uma equação sem os termos que incluem a derivada primeira, que é então resolvida para $f''(x_i)$. A fórmula resultante é:

$$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2} + O(h) \quad (3.29)$$

3.3 Resolução da EDO em estudo via Método das Diferenças Finitas

No final do capítulo 2, chegou-se na seguinte equação diferencial de segunda ordem não-homogênea com coeficientes variáveis:

$$\frac{d}{dx} \left[E(x)I(x) \frac{dy(x)}{dx} \right] + Py(x) = 0 \quad (3.30)$$

Passa-se, agora, a aplicação do método das diferenças finitas a determinação de valores aproximados de cargas críticas restringindo-se, aqui, ao caso de barras bi-rotuladas. Neste caso, pretendemos resolver a seguinte equação diferencial:

$$y''(x) + \frac{P}{E(x)I(x)}y(x) = 0 \quad (3.31)$$

que, com $k^2 = \frac{P}{E(x)I(x)}$, é equivalente a:

$$y''(x) + k^2y(x) = 0 \quad (3.32)$$

Para casos de colunas com inércia/material variável(is), escolheu-se a aplicar a aproximação de diferença central para a equação diferencial no ponto x_i , assim, escreve-se:

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{(x_{i+1} - x_i)^2} + \frac{Py(x_i)}{E(x_i)I(x_i)} = 0 \quad (3.33)$$

ou, de maneira mais reduzida:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + k_i^2y_i = 0 \quad (3.34)$$

Fazendo: $\alpha_i = -h^2k^2 + 2$, tem-se que: $y_{i-1} - \alpha_i y_i + y_{i+1} = 0$, com i variando de 1 até $n - 1$.

Para $i = 0$: $y_{-1} - \alpha y_0 + y_1 = 0 \Rightarrow$ Não serve, pois o ponto $i = -1$ está fora da coluna.

Para $i = 1$: $y_0 - \alpha y_1 + y_2 = 0$

Para $i = 2$: $y_1 - \alpha y_2 + y_3 = 0$

.

.

.

Para $i = n - 1$: $y_{n-2} - \alpha y_{n-1} + y_n = 0$

Para $i = n$: $y_{n-1} - \alpha y_n + y_{n+1} = 0 \Rightarrow$ Não serve, pois o ponto $i = n + 1$ está fora da coluna.

Assim:

$$\left\{ \begin{array}{l} y_0 - \alpha y_1 + y_2 = 0 \\ y_1 - \alpha y_2 + y_3 = 0 \\ \quad \quad \quad \cdot \\ \quad \quad \quad \cdot \\ \quad \quad \quad \cdot \\ y_{n-2} - \alpha y_{n-1} + y_n = 0 \end{array} \right.$$

Com:

$$\left\{ \begin{array}{l} y_0 = 0 \\ y_n = 0 \end{array} \right.$$

Assim, ficamos com o seguintes sistema:

$$\left\{ \begin{array}{l} -\alpha y_1 + y_2 = 0 \\ y_1 - \alpha y_2 + y_3 = 0 \\ \quad \quad \quad \cdot \\ \quad \quad \quad \cdot \\ \quad \quad \quad \cdot \\ y_{n-2} - \alpha y_{n-1} = 0 \end{array} \right.$$

Percebe-se que ao tentarmos determinar a carga critica, recaímos em sistemas de equações algébricas lineares. Estes sistemas são homogêneos e requerem soluções diferentes da trivial.

Para que isso ocorra, faz-se necessário que o determinante da matriz seja nulo o que então nos permite o cálculo dos valores aproximados da carga procurada. Além disso, quanto maior a quantidade de pontos em que dividimos a barra, o valor obtido se aproxima cada vez mais do valor exato do problema.

4 Simulações Numéricas

Este capítulo traz um breve histórico da linguagem de programação PYTHON e a descrição dos principais pacotes utilizados na implementação. Em seguida, é descrita a formulação numérica e por último, são apresentados as simulações numéricas realizadas com exemplos presentes na literatura. Estes requerem o valor da carga crítica em colunas cujo momento de inércia e/ou modelo de elasticidade são variáveis ao longo do comprimento, validando assim o algoritmo desenvolvido.

4.1 Linguagem de Programação PYTHON

A linguagem Python de programação foi lançada por Guido van Rossum em 1990 no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda. A princípio tinha foco em usuário físicos e engenheiros. Projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional, a linguagem Python possui uma sintaxe clara e concisa, que favorece a legibilidade do código fonte, tornando a linguagem mais produtiva. Nesse sentido, (ROSSUM, 2005) acredita que:

Python é uma linguagem de programação poderosa e de fácil aprendizado. Ela possui estruturas de dados de alto nível eficientes, bem como adota uma abordagem simples e efetiva para a programação orientada a objetos. Sua sintaxe elegante e tipagem dinâmica, em adição à sua natureza interpretada, tornam Python ideal para scripting e para o desenvolvimento rápido de aplicações em diversas áreas e na maioria das plataformas.

A linguagem inclui diversas estruturas de alto nível e uma vasta coleção de módulos prontos para uso. Também possui recursos encontrados em outras linguagens modernas, tais como: geradores, introspecção, persistência, metaclasses e unidades de teste. Mesmo os tipos básicos no Python são objetos. A linguagem é interpretada através de bytecode pela máquina virtual Python, tornando o código portátil. Sendo possível compilar aplicações em uma plataforma e rodar em outros sistemas ou executar direto do código fonte.

Para (BORGES, 2010) Python é um software de código aberto (com licença compatível com a General Public License (GPL), porém menos restritiva, permitindo que o Python seja inclusive incorporado em produtos proprietários). A especificação da linguagem é mantida pela Python Software Foundation (PSF).

Para construção do código utilizou-se, principalmente dois pacotes da linguagem Python: Numpy e Sympy. O primeiro permite trabalhar com arranjos, vetores e matrizes de N dimensões, com mais eficiência, e com toda a expressividade da linguagem. Provê diversas funções e operações sofisticadas, dentre elas: o objeto array para a implementação de arranjos multidimensionais, o objeto matrix para o cálculo com matrizes ferramentas para álgebra linear, transformadas de Fourier básicas e ferramentas sofisticadas para geração de números aleatórios. (PYSCIENCE, 2018)

Além disso tudo, as classes criadas podem ser facilmente herdadas, permitindo a customização do comportamento (por exemplo, dos operadores típicos de adição, subtração, multiplicação, etc.). O módulo é implementado em linguagem C, o que dá uma grande velocidade às operações realizadas.

O objetivo da biblioteca simbólica Sympy é tornar-se um sistema de álgebra computacional (CAS) completo, mantendo o código o mais simples possível, a fim de ser compreensível e facilmente extensível. Para tanto, o Sympy é escrito inteiramente em Python, depende apenas do mpmath, uma biblioteca Python pura para aritmética de ponto flutuante arbitrário, facilitando o uso e além disso pode ser usada como uma ferramenta interativa e ser incorporado em outros aplicativos e estendido com funções personalizadas.

4.2 Formulação numérica para utilizar o Método das Diferenças Finitas

A formulação numérica do Método das Diferenças Finitas (MDF) na solução de equações diferenciais governantes de fenômenos físicos segue uma ótica sistêmica, onde é possível notar uma ligação entre inúmeros conhecimentos, como de modelagem e técnicas matemáticas, implementação de algoritmos computacionais, formas otimizadas de processamento e solução, entre outras. (GILAT; SUBRAMANIAM, 2008.)

Pensando nisto, pode-se dividir a aplicação do MDF em três estágios: formulação matemática e escolha do método, solução do sistema linear e análises e interpretação de resultados. Para ilustrar esta ideia, o fluxograma apresentado na Figura 4.1 evidencia as principais etapas de aplicação do MDF. Estas etapas estão dispostas em três estágios, eles analisam desde o fenômeno físico até os resultados e interpretações dos dados.

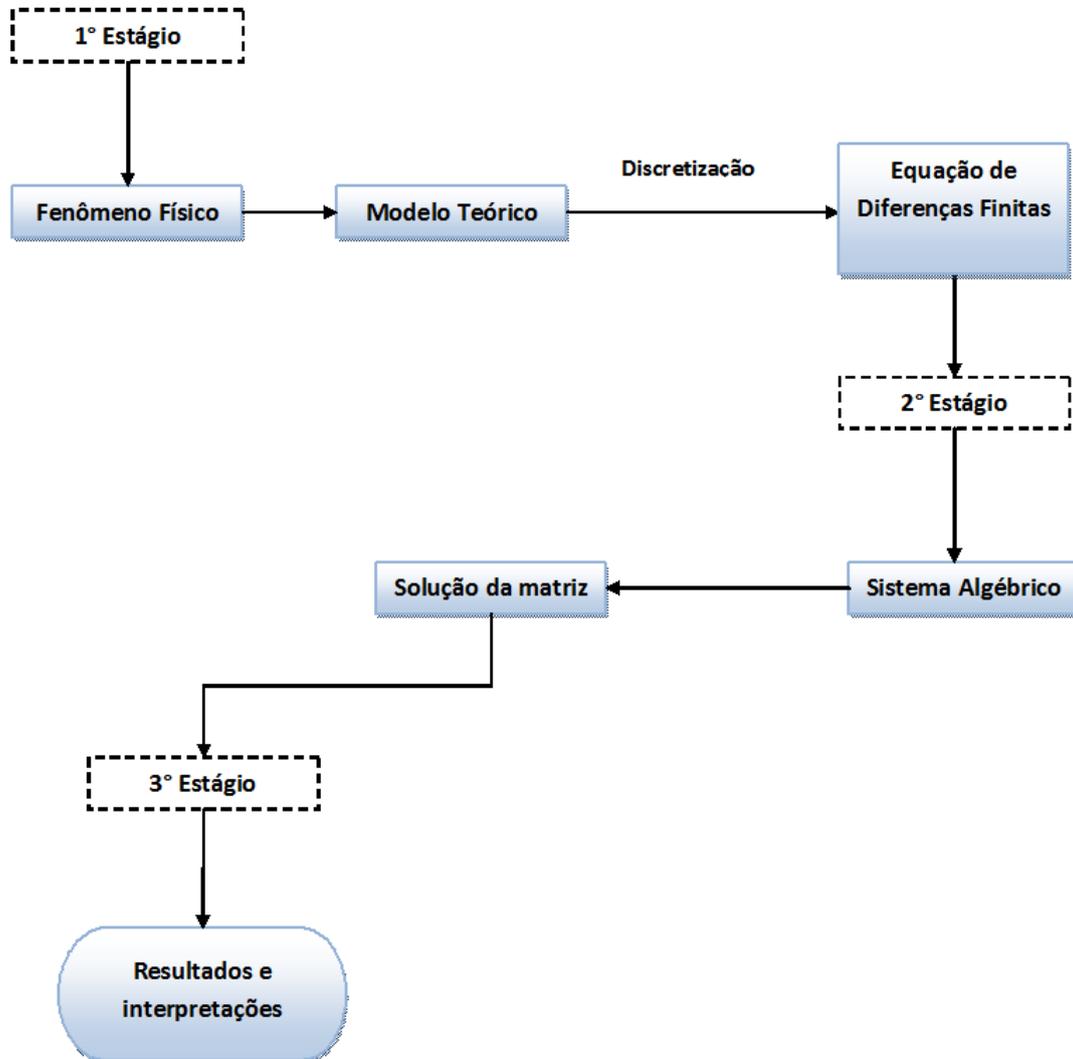


FIGURA 4.1 – Etapas e processos de aplicação do MDF.

Fonte: Neves, *et al.* (2008)

O primeiro estágio é destinado a todo processo de formulação e escolha apropriada do método ao problema estudado, assim, para iniciar o processo de aplicação numérica é necessário conhecer e compreender o modelo teórico diferencial, pois isto é crucial para realizar o entendimento correto dos resultados. Somente a partir disto, é realizada a aproximação da equação diferencial por expressões discretas.

O segundo estágio, por sua vez, aplica-se a EDF em cada ponto contido no domínio discreto, gerando assim um sistema algébrico matricial. A solução deste sistema pode ser feita com métodos diretos ou iterativos, como método de Gauss ou decomposição LU e Gauss-Seidel ou SOR. Esse estágio está ligado diretamente com o processo de otimização e desempenho computacional na resolução do sistema linear, acentuando ainda que, para problemas que possuem matrizes de alta ordem, a implementação de técnicas de otimização numérica são imprescindíveis. (FORTUNA, 2000)

Por fim, o terceiro estágio é o momento em que ocorre todo o processo de tratamento de resultados, isto é, uma análise de validação dos dados obtidos em comparação com valores de referência, como, soluções analíticas, valores experimentais ou até mesmo simulações numéricas já validadas. Nesta linha, todo o processo de validação e interpretação dos resultados é de competência de um profissional tecnicamente capacitado, a fim de distinguir e analisar os valores obtidos na simulação, para então, poder expor o fenômeno de forma correta. (CLÓVIS, 2013)

4.3 Exemplos numéricos

Para a obtenção do valor da carga crítica de flambagem dos exemplos a seguir, foi obedecido o seguinte roteiro:

- Inserir os dados de entrada;
- Dividir do comprimento da coluna ($0 \leq x \leq L$), em n partes segundo a sua configuração;
- Aplicar da fórmula da diferença central nos pontos da divisão, obtendo um sistema de equações algébricas linear que relaciona os valores de y nos pontos da divisão;
- Aplicar as condições de contorno do problema no sistema montado;
- Resolver o sistema de equações algébricas obtido, determinando o valor da carga crítica.

Para uma melhor aproximação do valor real da carga crítica, as colunas foram divididas em diversas partes, obtendo assim vários valores de cargas críticas. Por fim foi construído um gráfico de carga para análise do comportamento dos valores obtidos.

4.3.1 Coluna de seção circular de dois tramos com variação de módulo de elasticidade (E) e momento de inércia (I).

Deseja-se determinar o valor da carga crítica de flambagem da coluna abaixo.

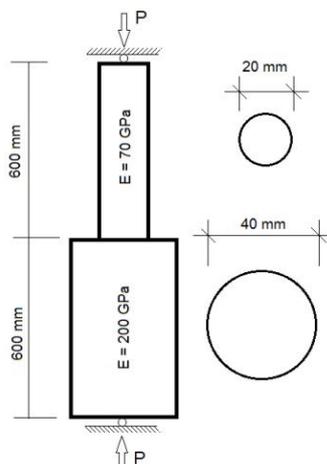


FIGURA 4.2 – Coluna bi-rotulada de seção circular.

Fonte: Bastos, *et al.* (2017)

Para obtenção da carga crítica de colunas do tipo acima, deve-se a priori dividir a coluna em subintervalos. Notamos que realizar apenas uma divisão torna-se inviável. Se optarmos por duas divisões estaríamos correndo o risco de dividirmos a coluna ao meio, o que causaria dualidade de valores no módulo de elasticidade (E). Já a divisão $n = 3$ torna-se possível pois podemos dividir o primeiro tramo correspondente ao $E = 70$ GPa ao meio e o segundo tramo com $E = 200$ GPa ao meio. Com isso, pode-se afirmar que o número mínimo de divisões é três e deste valor em diante pode-se utilizar números ímpares de divisões. Para $n = 3$, temos a seguinte divisão mostrada na Figura 4.3.

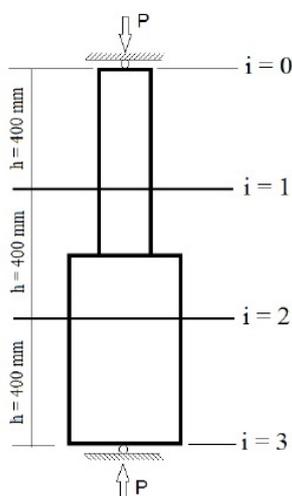


FIGURA 4.3 – Divisão da coluna bi-rotulada de seção circular para $n = 3$.

Fonte: Bastos, *et al.* (2017)

Os dados de entrada do simulador (ver Figura A.1) foram os seguintes:

- Módulo de elasticidade do primeiro tramo ($E1$) em Mpa;
- Módulo de elasticidade do segundo tramo ($E2$) em Mpa;
- Diâmetro do primeiro tramo ($d1$) em mm;
- Diâmetro do segundo tramo ($d2$) em mm;
- Altura da coluna (H) em mm;
- Número de partes ou subintervalos (n).

Após lançarmos os dados de entrada, o algoritmo calculou os momentos de inércia correspondentes a cada tramo e o passo (h), sendo o passo igual a $\frac{H}{n}$. Para a aplicação da equação da diferença centrada, foi necessário calcular os alfas correspondentes a cada tramo e a partir disso montar a matriz segundo a condição de contorno do problema. Após obtermos a matriz, foi calculado o determinante e igualado a zero, garantindo assim que o sistema possua solução diferente da trivial. Por último, obtivemos as raízes da função e a menor delas foi considerada a carga crítica de flambagem.

O resultado da carga crítica fornecido pelo algoritmo para $n = 3$ está mostrado na Figura 4.4 e foi condizente com o valor da carga crítica fornecido pela literatura apresentada por (BASTOS *et al.*, 2017).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\CLIENTE\Documents\TCC_Monaliza\Programas - Linguagem
PYTHON\Simulações TCC\Simulações\Simulação 1.py

Digite o valor do número de divisões (n): 3

Matriz A:
Matrix([[0.000291026181653752*P - 2, 1], [1, 6.36619772367581e-6*P - 2]])

Determinante da matriz A:
(6.36619772367581e-6*P - 2)*(0.000291026181653752*P - 2) - 1

Os valores das cargas em Newton são:
[5125.67946020356, 315905.819828504]

Portanto, o valor da carga crítica em kN é:
5.13
>>>
```

FIGURA 4.4 – Resultados fornecidos pelo simulador para coluna bi-rotulada de seção circular com $n = 3$.

Fonte: Autoria própria (2018)

4.3.2 Coluna com seção transversal cônica e módulo de elasticidade (E) constante.

Deseja-se determinar o valor da carga crítica de flambagem da coluna abaixo:

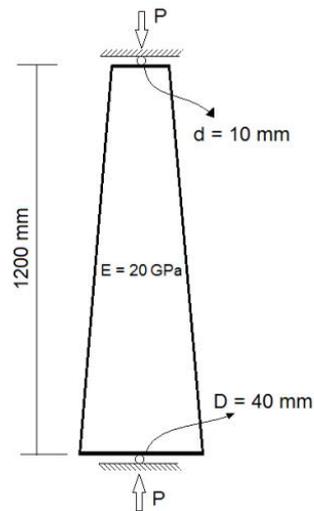


FIGURA 4.5 – Coluna de pinos fixos com seção transversal cônica.

Fonte: Bastos, *et al.*(2017)

Analisando a configuração da coluna acima pode-se afirmar que se realizarmos apenas uma divisão, esta torna-se sem sentido. Se optarmos por duas divisões a resolução do problema já pode ser considerada viável, sendo possível também dividir a coluna em mais subintervalos a partir de $n = 2$.

Para $n = 3$, temos a seguinte divisão mostrada na Figura 4.6 :

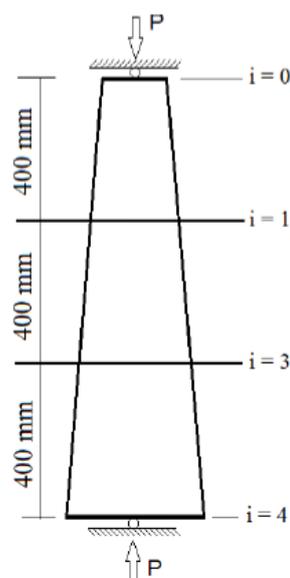


FIGURA 4.6 – Divisão da coluna de pinos fixos com seção transversal cônica para $n = 3$.

Fonte: Bastos, *et al.*(2017)

Nesse exemplo o simulador (ver Figura A.2), precisou dos seguintes dados de entrada:

- Módulo de elasticidade (E) em Mpa;
- Diâmetro da base (D) em mm;
- Diâmetro do topo (d) em mm;
- Altura da coluna (H) em mm;
- Número de partes ou subintervalos (n).

Em seguida o algoritmo calculou os dois passos h e $h2$, sendo o primeiro passo igual a $\frac{H}{n}$ e o segundo igual a $\frac{h}{2}$. Para a aplicação da equação da diferença centrada, foi necessário construir os vetores dos diâmetros, momentos de inércia e alfas intermediários.

Nesse momento foi montada a matriz segundo a condição de contorno do problema e após obtermos a matriz, foi calculado o determinante e igualado a zero. Por último, obtivemos as raízes da função e a menor delas foi considerada a carga crítica de flambagem.

O resultado da carga crítica fornecido pelo algoritmo para $n = 3$ está mostrado na Figura 4.7 e foi condizente com o valor da carga crítica fornecido pela literatura apresentada por (BASTOS *et al.*, 2017).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
  RESTART: C:\Users\CLIENTE\Documents\TCC_Monaliza\Programas - Linguagem
PYTHON\Simulações TCC\Simulações\Simulação 2.py

Digite o valor de n: 3

Matriz A:
Matrix([[0.00101859163578813*P - 2, 1], [1, 0.000201203286081606*P - 2]])

Determinante da matriz A:
(0.000201203286081606*P - 2)*(0.00101859163578813*P - 2) - 1

Os valores das cargas em Newton são:
[1392.64397339905, 10511.0469405935]

Portanto, o valor da carga em crítica em KN é:
1.393
>>>
```

FIGURA 4.7 – Resultados fornecido pelo simulador para coluna de seção transversal cônica com $n = 3$.

Fonte: Autoria própria (2018)

4.3.3 Coluna de seção quadrada de três tramos com variação do momento de inércia (I)

Deseja-se determinar o valor da carga crítica de flambagem da coluna abaixo.

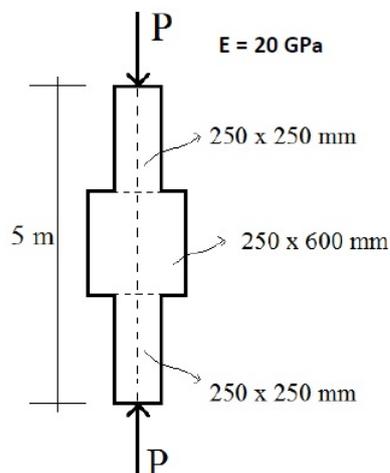


FIGURA 4.8 – Coluna bi-rotulada de seção quadrada.

Fonte: Bastos, *et al.*(2017)

Analisando a configuração da coluna acima, é preciso atentarmos que o valor de n não poderá ser múltiplo de 3, caso contrário ocorrerá uma dualidade no valor do módulo de elasticidade. Se começássemos com $n = 1$ esta divisão torna-se impossível, já para $n = 2$, a divisão torna-se inadequada. Sendo viável assim valores de n a partir de $n = 4$, sendo necessário evitar os múltiplos de 3. Para $n = 4$, temos a seguinte divisão mostrada na Figura 4.9 :

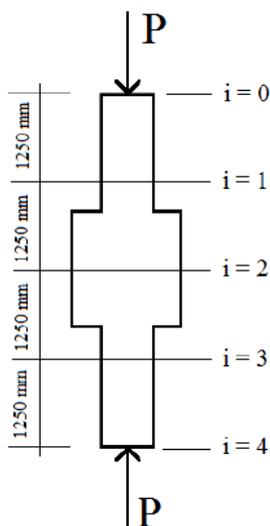


FIGURA 4.9 – Divisão da coluna bi-rotulada de seção quadrada de três tramos com $n = 4$.

Fonte: Bastos, *et al.* (2017)

Os dados de entrada do simulador (ver Figura A.3) foram:

- Módulo de elasticidade (E) em Mpa;
- Valor da base do primeiro tramo ($b1$) em mm;
- Valor da base do segundo tramo ($b2$) em mm;
- Valor da base do terceiro tramo ($b3$) em mm;
- Valor da altura do primeiro tramo ($h1$) em mm;
- Valor da altura do segundo tramo ($h2$) em mm;
- Valor da altura do terceiro tramo ($h3$) em mm;
- Altura da coluna (H) em mm;
- Número de partes ou subintervalos (n).

Em seguida o algoritmo calculou os momentos de inércias correspondentes a cada tramo e o passo (h), sendo que o passo igual a $\frac{H}{n}$. Para a aplicação da equação da diferença centrada, foi necessário calcular os alfas correspondentes a cada tramo e a partir daí montar a matriz segundo a condição de contorno do problema. Após obtermos a matriz, foi calculado o determinante e igualado a zero. Por último obtivemos as raízes da função e a menor delas foi considerada a carga crítica de flambagem.

O resultado da carga crítica fornecido pelo algoritmo para $n = 4$ está mostrado na Figura 4.10 e foi condizente com o valor da carga crítica fornecido pela literatura apresentada por (BASTOS *et al.*, 2017).

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\CLIENTE\Documents\TCC_Monaliza\Programas - Linguagem
PYTHON\Simulações TCC\Simulações\Simulação 3.py

Digite o valor do número de divisões (n): 4

Matriz A:
Matrix([[2.4e-7*P - 2, 1, 0], [1, 1.0e-7*P - 2, 1], [0, 1, 2.4e-7*P -
2]])

Determinante da matriz A:
-4.8e-7*P + (1.0e-7*P - 2)*(2.4e-7*P - 2)**2 + 4

Os valores das cargas em Newton são:
[3333333.33333333, 8333333.33333333, 25000000.0000000]

Portanto, o valor da carga crítica em MN é:
3.333
>>>
```

FIGURA 4.10 – Resultados fornecidos pelo simulador da coluna bi-rotulada de seção quadrada de três tramos com $n = 4$.

Fonte: Autoria própria (2018)

5 Resultados e Discussões

Quanto maior a quantidade de pontos em que dividirmos as colunas, melhor será o resultado da carga crítica. Nesse sentido, este capítulo apresenta o gráfico das cargas críticas correspondentes a cada simulação do capítulo anterior, bem como a fórmula de extrapolação.

5.1 Gráficos das cargas críticas

As simulações mostradas no capítulo anterior foram desenvolvidas utilizando-se discretizações com $n = 3$ e $n = 4$. Em se tratando de discretizações de um contínuo, cada valor que é retornado do simulador numérico, depende diretamente da malha utilizada. Quanto maior o valor de n e conseqüentemente menor o valor do passo h , mais próximo do valor exato estará a solução do problema. Para checar a convergência dos resultados pode-se calcular a carga crítica de cada simulação para diferentes n 's e plotar o gráfico (P_{cr}) pelo número de subintervalos (n). Os gráficos estão apresentados nas Figuras 5.1, 5.2 e 5.3.

É possível verificar através da Figura 5.1, para o caso da coluna de seção circular de dois tramos com variação de módulo de elasticidade (E) e momento de inércia (I), que a carga crítica converge para um valor de aproximadamente 6,20 kN. Um valor em torno de 17,12% maior do que o encontrado no caso da discretização com $n = 3$. Pode-se analisar ainda que a partir de $n = 17$ os valores das cargas críticas que o simulador retornou, foram bem próximas entre si.

Observando a Figura 5.2, para o caso da coluna com seção transversal cônica e módulo de elasticidade (E) constante, pode-se verificar que a carga crítica converge para um valor de aproximadamente 1,07 kN. Um valor em torno 50% menor do que o encontrado no caso da discretização com $n = 2$. Isso mostra que para uma coluna de seção cônica, a carga crítica de flambagem diminui à medida que o número de subintervalos cresce, sendo possível a sua estabilização até um certo número n . Neste caso para $n = 19$, o simulador começou a retornar valores próximos quase constantes.

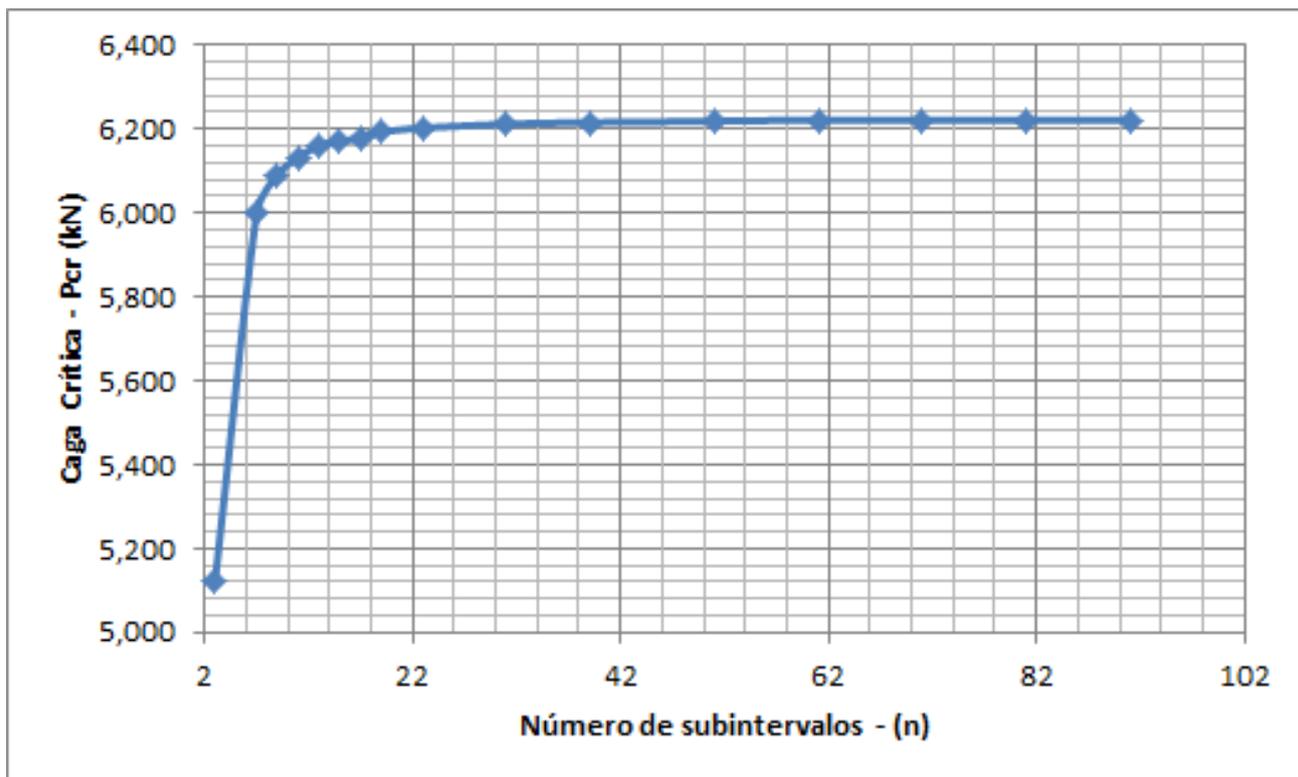


FIGURA 5.1 – Gráfico de cargas da coluna de seção circular dois tramos.

Fonte: Autoria própria (2018)

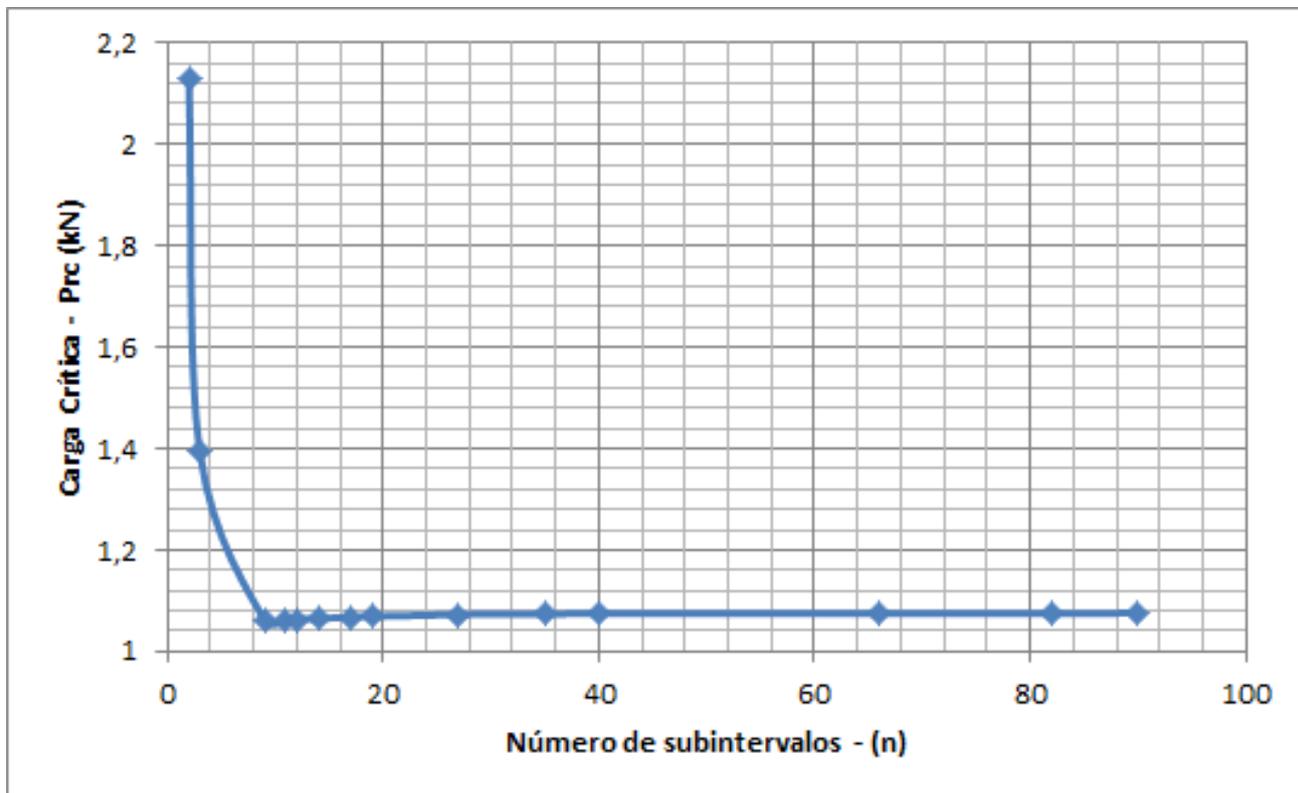


FIGURA 5.2 – Gráfico de cargas da coluna de seção transversal cônica.

Fonte: Autoria própria (2018)

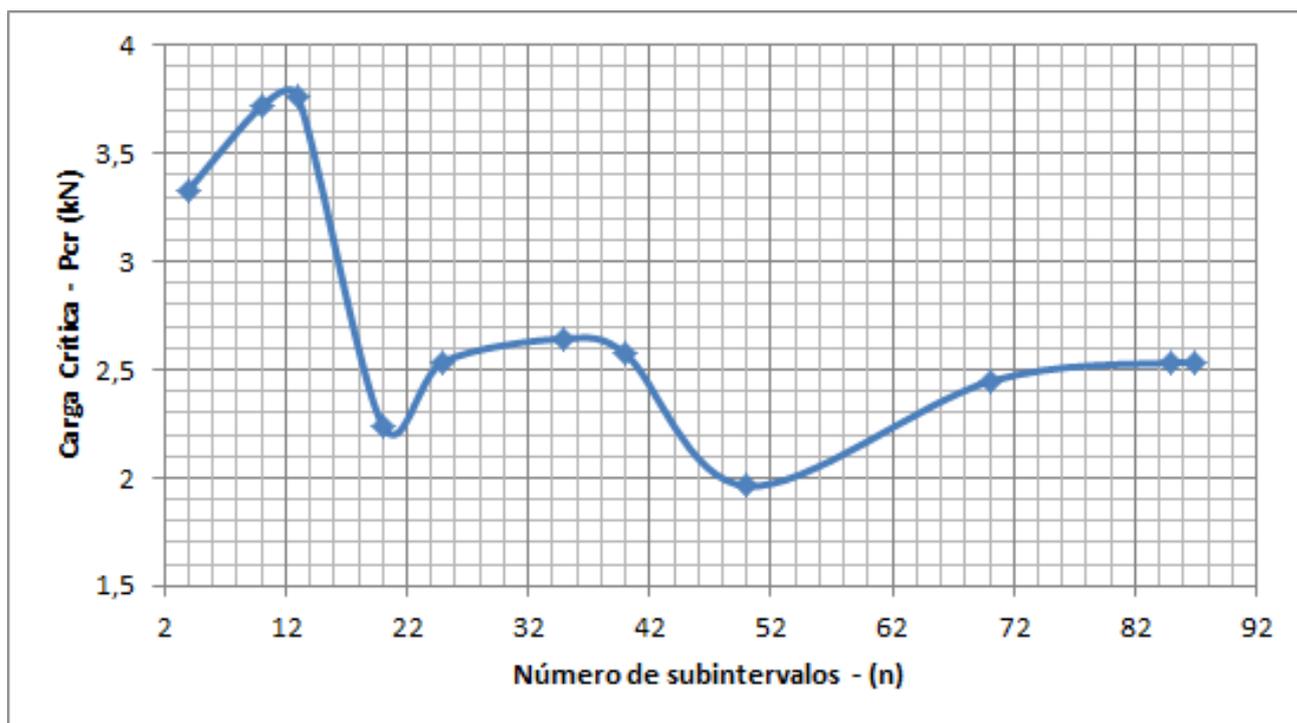


FIGURA 5.3 – Gráfico de cargas da coluna de seção quadrada de três tramos.

Fonte: Autoria própria (2018)

Analisando a Figura 5.3, para o caso da coluna de seção quadrada de três tramos com variação do momento de inércia (I), o valor da carga crítica oscilou para diferentes valores de n . Sendo que para $n = 10$ até $n = 13$ o simulador retornou valores próximos de 3,77 kN, em torno de 11,67%, assim o valor foi maior do que o encontrado no caso da discretização com $n = 4$.

Vale destacar que para o cálculo de matrizes de alta ordem foi utilizado o método de decomposição ou fatoração LU (ver Figuras A.4, A.5, A.6 e A.7) seguido do método da secante (ver Figura A.8) para encontrarmos as raízes da função, otimizando assim o desempenho computacional.

5.2 Fórmula de extrapolação

Outra forma de calcular a carga crítica de flambagem a partir dos dados fornecidos pelos simuladores é utilizando a fórmula de extrapolação (Equação 5.1). Sendo η o número de elementos da malha de DF e P_{cr1} e P_{cr2} valores de cargas obtidos através das discretizações.

$$P_{cr} \approx \frac{\eta_1^2 P_{cr1} - \eta_2^2 P_{cr2}}{\eta_1^2 - \eta_2^2} \quad (5.1)$$

Para o exemplo da figura 5.1, para os valores de cargas críticas P_{cr1} e P_{cr2} obtidos através de discretizações com $n = 3$ e $n = 7$, respectivamente, obtém-se, através da Equação (5.1), $P_{cr} \approx 6,20$ kN. Esse valor difere apenas 0,16 % da carga crítica obtida através do gráfico de convergência da Figura 5.1. Isso mostra que a Equação (5.1) pode ser útil na obtenção de uma melhor aproximação de P_{cr} .

Analisando o exemplo da figura 5.2, para os valores de cargas críticas P_{cr1} e P_{cr2} obtidos através de discretizações com $n = 3$ e $n = 9$, respectivamente, obtém-se, através da Equação (5.1), $P_{cr} \approx 1,02$ kN.

Para o exemplo da figura 5.3, para os valores de cargas críticas P_{cr1} e P_{cr2} obtidos através de discretizações com $n = 4$ e $n = 10$, respectivamente, obtém-se, através da Equação (5.1), $P_{cr} \approx 3,79$ kN. Já para valores de P_{cr1} e P_{cr2} obtidos através de discretizações com $n = 35$ e $n = 40$ tem-se $P_{cr} \approx 2,34$ kN, confirmando a oscilação de valores.

6 Conclusão

Os métodos numéricos computacionais são ferramentas de enorme importância na prática da engenharia civil. O MDF mostrou-se efetivo para a determinação das cargas de flambagem de pilares com variação do módulo de elasticidade e/ou momento de inércia.

Pode-se afirmar que todos os resultados foram condizentes com os valores esperados. Após a obtenção dos dados foi possível construir gráficos de cargas e além disso utilizar a fórmula de extrapolação para se aproximar P_{cr} . A partir dos resultados obtidos através das simulações, pode-se afirmar que o principal objetivo foi alcançado, pois em todos os casos o programa foi capaz de determinar a carga crítica das estruturas analisadas.

Diante desses resultados pode-se perceber que a aplicação da experimentação numérica via método das diferenças finitas, nos modelos diferenciais clássicos de pilares apresentou um excelente desempenho, com um baixo custo computacional associado e geração de resultados com erros diminutos no que tange à ordem de grandeza.

Logo, é imprescindível que os profissionais e estudantes de engenharia busquem um arcabouço conceitual-matemático consistente, que seja capaz de facultar-lhes a capacidade de interpretar e avaliar assertivamente os dados e comportamentos intrínsecos aos fenômenos físicos, modelos matemáticos correlatos e implementação de técnicas numéricas de solução.

Para trabalhos futuros, pretende-se tornar esta ferramenta criada mais robusta, aumentando a sua aplicabilidade, sobretudo para as demais configurações de apoios. Sendo necessário portanto, a discretização da equação governante e aplicação das condições de contorno.

Portanto, pode-se considerar o método de aproximações por diferenças finitas como uma técnica robusta e versátil na resolução de problemas na seara da engenharia estrutural.

Referências

ARBABI, B. F.; ASCE, M. Buckling of variable cross-section columns: Ntegral-equation approach. **Journal of Structural Enginering**, 1991.

BASTOS, F. de S.; CURY, A. A.; TOLEDO, E. M. **Apostila de Resistência dos Materiais II**. Juiz de Fora, 2017. 143 p.

BEER, F. P.; JOHNSTON, E. R.; DEWOLF, J. T.; MAZURED, D. F. **Mecânica dos Materiais**. 5. ed. Porto Alegre: AMGH Editora Ltda, 2011.

BORGES, L. E. **Pyhton para Desenvolvedores**. 2. ed. Rio de Janeiro: Edição do Autor, 2010.

BOYCE, W. E.; DIPRIMA, R. C. **Equações diferenciais e problemas de valores de contorno**. 10. ed. Rio de Janeiro: LTC, 2015.

BRACAL, T. P.; JUNIOR, R. B. Determinação da força crítica de flambagem de tubos cilíndricos de paredes finas. **CONGRESSO LATINOAMERICANO DA CONSTRUÇÃO METÁLICA**, 2012.

CLÓVIS, R. M. **Transferência de calor e mecânica dos fluidos computacional**. [S.l.]: LTC, 2013.

FORTUNA, A. **Técnicas computacionais para dinâmica dos fluidos**. [S.l.]: São Paulo, 2000.

GILAT, A.; SUBRAMANIAM, V. **Métodos numéricos para engenheiros e cientistas: uma introdução usando o MATLAB**. Porto Alegre: Bookman, 2008.

HIBBELER, R. C.; SILVA, J. P. N. da. **Resistência dos materiais**. 7. ed. São Paulo: Pearson Prentice Hall, 2010.

LONGO, H. I. **Pilares de Concreto Armado**. 3. ed. [S.l.], 2000. 32 p.

PYSCIENCE. **Numpy**. 2018. Disponível em:
<<http://pyscience-brasil.wikidot.com/module:numpy>>.

ROSSUM, G. van. **Tutorial Python**. [S.l.], 2005. 118 p.

ZILL, D. G.; CULLEN, M. R. **Equações diferenciais**. 3. ed. São Paulo: Pearson, 2012. Título original : Differential Equations with boundary-value problems.

Apêndice A - Códigos-fonte

```
#Código-fonte

#IFPB
#Trabalho de Conclusão de Curso
#Método das diferenças finitas
#Autora: Monaliza Araújo Parnaíba
#Cálculo da carga crítica de flambagem
#Coluna de seção circular de dois tramos com variação do Módulo de
Elasticidade e Momento de Inércia

import sympy as sym
from sympy import Matrix
from sympy import *
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve
import math
from sympy import Symbol, solve
from sympy.solvers import solve
from sympy.solvers.solvers import solve_linear
from sympy.plotting import plot

#Dados de entrada:

E1 = float(input("Digite o valor do Módulo de Elasticidade do primeiro
tramo em Mpa: "))
E2 = float(input("Digite o valor do Módulo de Elasticidade do segundo
tramo em Mpa: "))
d1 = float(input("Digite o valor do diâmetro do primeiro tramo em mm: "))
d2 = float(input("Digite o valor do diâmetro do segundo tramo em mm: "))
H = float(input("Digite o valor da altura da coluna em mm: "))
n = int(input("Digite o valor de subintervalos: "))

#Cálculo dos momentos de inércia e o passo h:

I1 = (math.pi*(d1)**4)/64
I2 = (math.pi*(d2)**4)/64
H = 1200
h = H/n

#Cálculo do alfa1 e alfa2:

P = sp.Symbol('P')
alfa1 = (2 - ((h**2)*P)/(E1*I1))
alfa2 = (2 - ((h**2)*P)/(E2*I2))

#Declaração da matriz A:

z = int(n-1)
A = zeros(z)

for i in range(0,int(z/2)):
    A[i,i] = -alfa1
for i in range(int(z/2),z):
    A[i,i] = -alfa2

for i in range(1,z):
    if (i < z):
        A[i-1,i] = 1
        if (i <= z-1):
            A[i,i-1] = 1

#Cálculo do determinante:

def f(P):
    return A.det()

#Resolução do determinante:

def g(P):
    return solve(f(P),P)

#Escolha da menor raiz da função:

def h(P):
    return min(g(P))

#Dados de saída:

print ("Matriz A:")
print (A)
print("Determinante da matriz A:")
print (A.det())
print ("O valores das cargas em Newton são:")
print (g(P))
print ("Portanto, o valor da carga critica em kN é:")
print (round(h(P)/1000,2))
```

FIGURA A.1 – Coluna de seção circular de dois tramos.

Fonte: Autoria própria (2018)

```

#Código-fonte

#IFPB
#Trabalho de Conclusão de Curso
#Método das diferenças finitas
#Autora: Monaliza Araújo Parnaíba
#Cálculo da carga crítica de flambagem
#Coluna com seção transversal cônica e Módulo de Elasticidade constante

import numpy as np
import matplotlib.pyplot as plt
import sympy as sym
from sympy import Matrix
from sympy import *
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve
import math
from sympy import Symbol, solve
from sympy.solvers import solve

#Dados de entrada:

E = float(input("Digite o valor do Módulo de Elasticidade em Mpa: "))
D = float(input("Digite o valor do diâmetro da base em mm: "))
d = float(input("Digite o valor do diâmetro do topo em mm: "))
H = float(input("Digite o valor da altura da coluna em mm: "))
n = int(input("Digite o valor de subintervalos: "))

#Cálculo dos passos:

h = H/n
h2 = h/D

#Cálculo dos diâmetros:

F = np.zeros ((n-1,1))
F[0,0] = d + h2
for i in np.arange(1,n-1):
    F[i] = F[0,0] + i*h2

#Cálculo dos momentos de inércia:

I = np.zeros ((n-1,1))
I[0,0] = (math.pi*(F[0,0])**4)/64

for i in np.arange(1,n-1):
    I[i] = (math.pi*(F[i])**4)/64

#Cálculo dos alfas:

P = sp.symbols('P')
alfa = zeros(n-1,1)
alfa[0,0] = (2 - ((h**2)*P)/(E*I[0,0]))

for i in range(1,n-1):
    alfa[i] = (2 - ((h**2)*P)/(E*I[i]))

#Declaração de matriz A:

z = n-1
A = zeros(z)
for i in range (0,n-1):
    A[i,i] = -alfa[i]
for i in range(1,z):
    if (i < z):
        A[i-1,i] = 1
        if (i <= z-1):
            A[i,i-1] = 1

#Cálculo do determinante:

def f(P):
    return A.det()

#Resolução do determinante:

def g(P):
    return solve(f(P),P)

#Escolha da menor raiz da função:

def h(P):
    return min(g(P))

#Dados de saída:

print ("Matriz A:")
print (A)
print("Determinante da matriz A:")
print (A.det())
print ("Os valores das cargas em Newton são:")
print (g(P))
print ("Portanto, o valor da carga em crítica em KN é:")
print (round(h(P)/1000,3))

```

FIGURA A.2 – Coluna com seção transversal cônica.

Fonte: Autoria própria (2018)

```

#Código-fonte

#IFPB
#Trabalho de Conclusão de Curso
#Método das diferenças finitas
#Autora: Monaliza Araújo Parnaíba
#Cálculo da carga crítica de flambagem
#Coluna de seção quadrada de três tramos com variação do Momento de
Inércia

import sympy as sym
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve

#Dados de entrada:

E = float(input("Digite o valor do Módulo de Elasticidade em Mpa: "))
b1 = float(input("Digite o valor da base do primeiro tramo em mm: "))
b2 = float(input("Digite o valor da base do segundo tramo em mm: "))
b3 = float(input("Digite o valor da base do terceiro tramo em mm: "))
h1 = float(input("Digite o valor da altura do primeiro tramo em mm: "))
h2 = float(input("Digite o valor da altura do segundo tramo em mm: "))
h3 = float(input("Digite o valor da altura do terceiro tramo em mm: "))
H = float(input("Digite o valor da altura da coluna em mm: "))
n = float(input("Digite o número de subintervalos: "))

#Cálculo dos momentos de inércia:

I1 = (b1**4)/12
I2 = (h2*(b2**3))/12
I3 = (b3**4)/12

#Cálculo dos passos:

h = H/n
n1 = H/3
n2 = 2*n1

#Cálculo dos alfas:

P = sp.symbols('P')
alfa1 = (2 - ((h**2)*P)/(E*I1))
alfa2 = (2 - ((h**2)*P)/(E*I2))
alfa3 = (2 - ((h**2)*P)/(E*I3))

#Declaração de matriz A:
z = int(n-1)
A = zeros(z)

for i in range(1,z):
    if (i < z):
        A[i-1,i] = 1
        if (i <= z-1):
            A[i,i-1] = 1

#Cálculo do determinante:

def f(P):
    return A.det()

#Resolução do determinante:

def g(P):
    return solve(f(P),P)

#Escolha da menor raiz da função:

def h(P):
    return min(g(P))

#Dados de saída:

print ("Matriz A:")
print (A)
print("Determinante da matriz A:")
print (A.det())
print ("Os valores das cargas em Newton são:")
print (g(P))
print ("Portanto, o valor da carga crítica em MN é:")
print (round ((h(P)/1000000),3))

```

FIGURA A.3 – Coluna de seção quadrada de três tramos.

Fonte: Autoria própria (2018)

```
#IFPB
#Trabalho de Conclusão de Curso
#Fatoração LU
#Autora: Monaliza Araújo Parnaíba

from numpy import zeros, mat, transpose, dot
import numpy as np
from numpy import linalg
import sympy as sp

n = int(input("Digite o valor do número de divisões (n): "))

def f(A):
    U = A
    L = sp.eye(n)
    for j in np.arange(n-1):
        for i in np.arange(j+1,n):
            L[i,j] = U[i,j]/U[j,j]
            for k in np.arange(j+1,n):
                U[i,k] = U[i,k] - L[i,j]*U[j,k]
            U[i,j] = 0
    return L, U
```

FIGURA A.4 – Fatoração LU.

Fonte: Autoria própria (2018)

```

#IFPB
#Trabalho de Conclusão de Curso
#Fatoração LU
#Coluna de seção circular de dois tramos com variação do Módulo de
Elasticidade e Momento de Inércia
#Autora: Monaliza Araújo Parnaíba

import matplotlib.pyplot as plt
import math
import sympy as sym
from sympy import Matrix
from sympy import *
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve
from sympy import Symbol, solve
from sympy.solvers import solve
from sympy.solvers.solvers import solve_linear
from sympy.plotting import plot
from numpy import array

import fatoraLU_S

#Dados de entrada:

Nu = int(input("Digite o valor do número de divisões (n): "))
E1 = 70000
E2 = 200000
d1 = 20
d2 = 40
I1 = (math.pi*(d1)**4)/64
I2 = (math.pi*(d2)**4)/64
H = 1200
h = H/Nu
P = sp.Symbol('P')

#Cálculo do alfa1 e alfa2:

alfa1 = (2 - ((h**2)*P)/(E1*I1))
alfa2 = (2 - ((h**2)*P)/(E2*I2))

#Declaração da matriz a:

z = int(Nu-1)
a = zeros(z)

for i in range(0,int(z/2)):
    a[i,i] = -alfa1
for i in range(int(z/2),z):
    a[i,i] = -alfa2
for i in range(1,z):
    if (i < z):
        a[i-1,i] = 1
        if (i <= z-1):
            a[i,i-1] = 1

L, U = fatoraLU_S.f(a)

#Declaração do vetor X:

X = zeros(1,z)
for i in range(0,z):
    X[0,i] = U[i,i]

y = 1
for i in range(0,z):
    D = y*X[0,i]
    y = D

print('A função y é',y)

```

FIGURA A.5 – Coluna de seção circular de dois tramos - Fatoração LU.

Fonte: Autoria própria (2018)

```

#IFPB
#Trabalho de Conclusão de Curso
#Fatoração LU
#Coluna com seção transversal cônica e Módulo de Elasticidade constante
#Autora: Monaliza Araújo Parnaíba

import numpy as np
import matplotlib.pyplot as plt
import sympy as sym
from sympy import Matrix
from sympy import *
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve
import math
from sympy import Symbol, solve
from sympy.solvers import solve
#from sympy.solvers.solvers import solve_linear

import fatoraLU_S

#Dados de entrada:

Nu = int(input("Digite o valor de subintervalos: "))
E = 20000
H = 1200
h = H/Nu
d = 10
D = 40
h2 = h/D

#Cálculo dos diâmetros:

F = np.zeros ((Nu-1,1))
F[0,0] = d + h2
for i in np.arange(1,Nu-1):
    F[i] = F[0,0] + i*h2

#Cálculo dos momentos de inércia:

I = np.zeros ((Nu-1,1))
I[0,0] = (math.pi*(F[0,0])**4)/64

for i in np.arange(1,Nu-1):
    I[i] = (math.pi*(F[i])**4)/64

P = sp.symbols('P')

#Cálculo dos alfas:

alfa = zeros (Nu-1,1)
alfa[0,0] = (2 - ((h**2)*P)/(E*I[0,0]))

for i in range(1,Nu-1):
    alfa[i] = (2 - ((h**2)*P)/(E*I[i]))

#Declaração de matriz a:

z = Nu-1
a = zeros(z)
for i in range (0,Nu-1):
    a[i,i] = -alfa[i]
for i in range(1,z):
    if (i < z):
        a[i-1,i] = 1
        if (i <= z-1):
            a[i,i-1] = 1

L, U = fatoraLU_S.f(a)

#Declaração do vetor X:

X = zeros(1,z)

for i in range(0,z):
    X[0,i] = U[i,i]

y = 1
for i in range(0,z):
    D = y*X[0,i]
    y = D

print('A função y é',y)

```

FIGURA A.6 – Coluna de seção transversal cônica - Fatoração LU.

Fonte: Autoria própria (2018)

```

#IFPB
#Trabalho de Conclusão de Curso
#Fatoração LU
#Coluna de seção quadrada de três tramos com variação do Momento de
Inércia
#Autora: Monaliza Araújo Parnaíba

import matplotlib.pyplot as plt
import math
import sympy as sym
from sympy import Matrix
from sympy import *
import sympy as sp
from sympy.matrices import *
import sympy as sym
import sympy as sp
from sympy.matrices import *
from sympy.solvers import solve

#Dados de entrada:

Nu = float(input("Digite o valor de subintervalos: "))
E = 20000
b1 = 250
b2 = 250
b3 = 250
h1 = 250
h2 = 600
h3 = 250
I1 = ((b1)**4)/12
I2 = (h2*((b2)**3))/12
I3 = ((b3)**4)/12
H = 5000
h = H/Nu
n1 = H/3
n2 = 2*n1
P = sp.symbols('P')

#Cálculo dos alfas:

alfa1 = (2 - ((h**2)*P)/(E*I1))
alfa2 = (2 - ((h**2)*P)/(E*I2))
alfa3 = (2 - ((h**2)*P)/(E*I3))

#Declaração de matriz a:

z = int(Nu-1)
a = zeros(z)

for i in range (0,int(n1/h)):
    a[i,i]= -alfa1
    for i in range (int(n1/h),int(n2/h)):
        a[i,i] = -alfa2
        for i in range (int(n2/h),z):
            a[i,i] = -alfa3
for i in range(1,z):
    if (i < z):

        a[i-1,i] = 1
        if (i <= z-1):
            a[i,i-1] = 1

L, U = fatoralu_s.f(a)

#Delaração do vetor X:

X = zeros(1,z)
for i in range(0,z):
    X[0,i] = U[i,i]

y = 1
for i in range(0,z):
    D = y*X[0,i]
    y = D

print('A função y é',y)

```

FIGURA A.7 – Coluna de seção quadrada de três tramos - Fatoração LU.

Fonte: Autoria própria (2018)

```
#IFPB
#Trabalho de Conclusão de Curso
#Método da Secante
#Autora: Monaliza Araújo Parnaíba

def f(P):
    return y

a = float(input('digite a aproximação inicial '))

tol = float(input('Informe o valor da tolerância '))
d = 0.001
N = int(input('Informe o número máximo de iterações '))

i = 1

while (i <= N):

    p = a - d*func.f(a)/(func.f(a+d)-func.f(a))

    if (abs(func.f(p)) < tol):

        print('O valor da raiz é igual a',p)
        break # encerra o while

    i = i + 1

    a = p

if (i >= N):
    print("Num. max. de iter. excedido!")
```

FIGURA A.8 – Método da Secante.

Fonte: Autoria própria (2018)


```

- 2 - 1/(4.71730289693582e-6*P - 2 - 1/(5.63526552140145e-6*P - 2 -
1/(6.78778266247816e-6*P - 2 - 1/(8.25059224988386e-6*P - 2 -
1/(1.01295628485157e-5*P - 2 - 1/(1.25752053801004e-5*P - 2 -
1/(1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (3.97887357729738e-6*P - 2 - 1/(4.71730289693582e-6*P -
2 - 1/(5.63526552140145e-6*P - 2 - 1/(6.78778266247816e-6*P - 2 -
1/(8.25059224988386e-6*P - 2 - 1/(1.01295628485157e-5*P - 2 -
1/(1.25752053801004e-5*P - 2 - 1/(1.5805543036858e-5*P - 2 -
1/(2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (4.71730289693582e-6*P - 2 -
1/(5.63526552140145e-6*P - 2 - 1/(6.78778266247816e-6*P - 2 -
1/(8.25059224988386e-6*P - 2 - 1/(1.01295628485157e-5*P - 2 -
1/(1.25752053801004e-5*P - 2 - 1/(1.5805543036858e-5*P - 2 -
1/(2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (5.63526552140145e-6*P - 2 -
1/(6.78778266247816e-6*P - 2 - 1/(8.25059224988386e-6*P - 2 -
1/(1.01295628485157e-5*P - 2 - 1/(1.25752053801004e-5*P - 2 -
1/(1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (6.78778266247816e-6*P - 2 - 1/(8.25059224988386e-6*P - 2 -
1/(1.01295628485157e-5*P - 2 - 1/(1.25752053801004e-5*P - 2 -
1/(1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (7.87778266247816e-6*P - 2 - 1/(1.01295628485157e-5*P - 2 -
1/(1.25752053801004e-5*P - 2 - 1/(1.5805543036858e-5*P - 2 -
1/(2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (8.25059224988386e-6*P - 2 -
1/(1.01295628485157e-5*P - 2 - 1/(1.25752053801004e-5*P - 2 -
1/(1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (9.01642483424231e-5*P - 2 - 1/(1.01295628485157e-5*P - 2 -
1/(1.25752053801004e-5*P - 2 - 1/(1.5805543036858e-5*P - 2 -
1/(2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (1.01295628485157e-5*P - 2 -
1/(1.25752053801004e-5*P - 2 - 1/(1.5805543036858e-5*P - 2 -
1/(2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (1.25752053801004e-5*P - 2 -
1/(1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (1.5805543036858e-5*P - 2 - 1/(2.0143047485068e-5*P - 2 -
1/(2.60759458761761e-5*P - 2 - 1/(3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (2.0143047485068e-5*P - 2 - 1/(2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (2.60759458761761e-5*P - 2 -
1/(3.43631497287957e-5*P - 2 - 1/(4.62201869675927e-5*P - 2 -
1/(6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (3.43631497287957e-5*P - 2 -
1/(4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (4.62201869675927e-5*P - 2 - 1/(6.36619772367582e-5*P - 2 -
1/(9.01642483424231e-5*P - 2 - 1/(0.000132009475998142*P -
2)))))))))))* (6.36619772367582e-5*P - 2 - 1/(9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2)))))))))))* (9.01642483424231e-5*P - 2 -
1/(0.000132009475998142*P - 2))))))))))
>>>
RESTART: C:\Users\CLIENTE\Dropbox\TCC_Monaliza\Programas - Linguagem
PYTHON\Simulações TCC\Método de Newton\func.py
>>>
RESTART: C:\Users\CLIENTE\Dropbox\TCC_Monaliza\Programas - Linguagem
PYTHON\Simulações TCC\Método de Newton\newton.py
O valor da raiz é igual a 1073.0193429486978
>>>

```

FIGURA A.16 – Coluna com seção transversal cônica - Carga Crítica com $n = 27$.
 Fonte: Autoria própria (2018)

