

UNIVERSIDADE ESTADUAL DE GOIÁS  
Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo  
Curso de Matemática

**Uma introdução ao cálculo variacional e ao método de Rayleigh-Ritz com  
aplicações em *Python*.**

EDUARDO JOSÉ DE OLIVEIRA

Anápolis

2019



EDUARDO JOSÉ DE OLIVEIRA

**Uma introdução ao cálculo variacional e ao método de Rayleigh-Ritz com  
aplicações em *Python*.**

Trabalho de Curso (TC) apresentado a Coordenação Adjunta de TC, como parte dos requisitos para obtenção do título de Graduado no Curso de Matemática da Universidade Estadual de Goiás.

Orientador: Prof. Me. Tiago de Lima Bento Pereira

Anápolis

2019

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UEG  
com os dados fornecidos pelo(a) autor(a).

Ji

José de Oliveira, Eduardo

Uma introdução ao cálculo variacional e ao método de Rayleigh-Ritz com aplicações em Python / Eduardo José de Oliveira; orientador Tiago de Lima Bento Pereira. -- Anápolis, 2019.

81 p.

Graduação - Matemática -- Câmpus-Anápolis CET, Universidade Estadual de Goiás, 2019.

1. Métodos Variacionais. 2. Equação de Euler-Lagrange. 3. Método de Rayleigh-Ritz. 4. Programação. 5. Python. I. de Lima Bento Pereira, Tiago, orient. II. Título.

**UMA INTRODUÇÃO AO CÁLCULO VARIACIONAL E AO MÉTODO DE  
RAYLEIGH-RITZ COM APLICAÇÕES EM PYTHON**

**EDUARDO JOSÉ DE OLIVEIRA**

Trabalho de Curso de Matemática apresentado à Banca Examinadora como parte dos requisitos para a obtenção do grau de graduado em Licenciatura em Matemática.

Banca Examinadora do Trabalho de Curso de Matemática do Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo da Universidade Estadual de Goiás, Anápolis, terça-feira, 26 de novembro de 2019.



---

**M.e Tiago de Lima Bento Pereira**  
Presidente da Banca Examinadora



---

**Dr. Fabiano Boaventura de Miranda**  
1º Membro da Banca Examinadora



---

**M.e Cleber Giuglioli Carrasco**  
2º Membro da Banca Examinadora



*Este trabalho é dedicado à todos aqueles que,  
de alguma forma, mesmo sem consciência de tal feito,  
possibilitaram a minha caminhada até aqui.*





# AGRADECIMENTOS

Agradeço a minha mãe, Elza Oliveira, e meu irmão, Danilo Oliveira, por me acompanharem até aqui e possibilitarem que eu concluísse a graduação. Quero agradecer, de forma póstuma, meu pai, Lourenço Bernardes, por, mesmo em sua partida, promoveu, em mim, tamanho crescimento emocional e intelectual. Por último, gostaria de agradecer a toda a minha família.

Agradeço ao professor Tiago de Lima Bento Pereira por ter aceito me orientar nos estudos e na elaboração desse trabalho e por ter o feito com tamanha dedicação e empenho.

Agradeço aos professores que se dispuseram a fazer parte da banca e a ler, avaliar e propor melhorias a esse trabalho. Agradeço, também, a todos os demais professores que tiveram um papel importante ao longo da minha formação.

Agradeço aos meus amigos e colegas que me apoiaram e estiveram ao meu lado, pessoalmente ou por meio dos incríveis mensageiros instantâneos, durante essa jornada.

Agradeço, ainda, a todos aqueles que, mesmo desconhecidos, acalentaram meu coração, com pequenos gestos e sorrisos, ao longo dessa jornada.

Por último, gostaria de agradecer a todos os integrantes do Grupo Alegria, especialmente do Grupo Alegria Anápolis, por ensinarem que cada um de nós é capaz de realizar muito mais do que imagina, que o verdadeiro amor está nas pequenas coisas e, principalmente, que através do amor é possível mudar toda a história da humanidade.



*"Durante toda a sua vida, estudara o universo,  
mas desprezara sua mais clara mensagem:  
para criaturas pequenas como nós,  
a vastidão só é suportável através do amor."  
(Carl Sagan)*



# RESUMO

Apresenta-se neste trabalho uma introdução ao cálculo variacional e ao método de Rayleigh-Ritz. Um dos principais problemas pertinentes ao cálculo variacional é o de encontrar uma função exata que torna um determinado funcional estacionário. Mediante uma revisão bibliográfica buscou-se apresentar ferramentas para se encontrar a função exata que torna o funcional estacionário, dentre as quais, a primeira dessas ferramentas, apresentadas no trabalho, é a equação de Euler-Lagrange. Outra ferramenta apresentada é o método de Rayleigh-Ritz, que visa aproximar a função exata desejada. Complementando os estudos do método de Rayleigh-Ritz, são apresentadas, também, aplicações com a linguagem de programação *Python*.

**Palavras-chave:** Métodos Variacionais. Equação de Euler-Lagrange. Método de Rayleigh-Ritz. Programação. *Python*.



# LISTA DE ILUSTRAÇÕES

Figura 1.1 – Modelo de uma viga consola vibrando. . . . .	26
Figura 1.2 – Solução da equação de Poisson utilizando o <i>deep Ritz method</i> e o método das diferenças finitas . . . . .	26
Figura 2.1 – Representação gráfica das funções admissíveis. . . . .	31
Figura 3.1 – Viga prismática. . . . .	42
Figura 3.2 – Função triangular $\phi_i(x)$ . . . . .	49
Figura 4.1 – Resultado apresentado pelo Código 4.1. . . . .	55
Figura 4.2 – Resultado apresentado pelos Códigos 4.1, 4.2 e 4.3. . . . .	56
Figura 4.3 – Resultado do método de Rayleigh-Ritz utilizando funções de forma triangulares com $n = 5$ . . . . .	62
Figura 4.4 – Resultado do método de Rayleigh-Ritz utilizando funções de forma triangulares com $n = 10$ . . . . .	62
Figura C.1 – Página de <i>download</i> da linguagem <i>Python</i> . . . . .	75
Figura C.2 – Instalador da linguagem <i>Python</i> no <i>Windows</i> . . . . .	76
Figura C.3 – Barra de busca com o <i>prompt</i> de comandos do <i>Windows</i> . . . . .	76
Figura C.4 – <i>Prompt</i> de comandos após a instalação do pacote <i>numpy</i> . . . . .	77
Figura C.5 – Janela do IDLE com o menu <i>Run</i> aberto. . . . .	77
Figura C.6 – Execução do código para a resolução do sistema linear. . . . .	80





# LISTA DE CÓDIGOS

Código 4.1 – Código para a resolução do Exemplo 3.1 . . . . .	54
Código 4.2 – Código para a determinação do coeficiente $a_1$ do Exemplo 3.1 . . . . .	55
Código 4.3 – Código para a definição da função $v_3(x)$ do Exemplo 3.1 . . . . .	55
Código 4.4 – Importação dos pacotes utilizados para a resolução . . . . .	58
Código 4.5 – Criação dos intervalos numéricos . . . . .	58
Código 4.6 – Função de forma $\phi$ e sua derivada . . . . .	58
Código 4.7 – Algoritmo para integração numérica usando a regra dos trapézios . . . . .	59
Código 4.8 – Funções utilizadas para criação das matrizes do sistema linear . . . . .	59
Código 4.9 – Definição e resolução do sistema linear . . . . .	60
Código 4.10–Plotagem das funções exata e aproximada . . . . .	61
Código C.1 – Resolução de sistemas lineares em <i>Python</i> . . . . .	80



# LISTA DE TABELAS

Tabela 1 – Principais operações na linguagem <i>Python</i> . . . . .	78
--	----



# SUMÁRIO

	<b>INTRODUÇÃO</b> . . . . .	<b>21</b>
<b>1</b>	<b>CONTEXTO HISTÓRICO</b> . . . . .	<b>23</b>
1.1	Máximos e mínimos . . . . .	23
1.2	O cálculo variacional . . . . .	24
1.3	O método de Rayleigh-Ritz . . . . .	24
1.3.1	Aplicações do método de Rayleigh-Ritz . . . . .	25
1.4	A linguagem <i>Python</i> . . . . .	27
<b>2</b>	<b>CÁLCULO VARIACIONAL</b> . . . . .	<b>29</b>
2.1	Equação de Euler-Lagrange . . . . .	29
2.2	Condições de contorno naturais e essenciais . . . . .	32
2.3	Equação de Euler-Lagrange para funcionais que dependem das derivadas de ordem superior . . . . .	34
2.3.1	Processo para funcionais que dependem de derivadas de segunda ordem . . . . .	35
2.3.2	Equação geral . . . . .	36
2.4	Exemplos . . . . .	37
<b>3</b>	<b>MÉTODO DE RAYLEIGH-RITZ</b> . . . . .	<b>41</b>
3.1	O uso de funções triangulares no método de Rayleigh-Ritz . . . . .	49
<b>4</b>	<b>UTILIZANDO O MÉTODO DE RAYLEIGH-RITZ NA LINGUAGEM <i>PYTHON</i></b> . . . . .	<b>53</b>
4.1	Resolvendo problemas com o método de Rayleigh-Ritz utilizando funções triangulares em <i>Python</i> . . . . .	57
	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>63</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>65</b>
	<b>APÊNDICES</b> . . . . .	<b>67</b>
	<b>APÊNDICE A – CONCEITOS DE ANÁLISE NO <math>\mathbb{R}^n</math></b> . . . . .	<b>69</b>
	<b>APÊNDICE B – ENCONTRANDO A MÁXIMA DEFLEXÃO DA VIGA ESTUDADA</b> . . . . .	<b>73</b>

	<b>APÊNDICE C – BREVE INTRODUÇÃO À LINGUAGEM <i>PYTHON</i> .</b>	<b>75</b>
<b>C.1</b>	<b>Instalando . . . . .</b>	<b>75</b>
<b>C.2</b>	<b>Executando programas . . . . .</b>	<b>77</b>
<b>C.3</b>	<b>Principais operações . . . . .</b>	<b>78</b>
<b>C.4</b>	<b>Resolução de sistemas lineares . . . . .</b>	<b>78</b>

# INTRODUÇÃO

O cálculo variacional engloba diversos problemas, sendo um dos mais tratados, o de minimizar ou maximizar funcionais, que são, fundamentalmente, funções que dependem de outras funções (ASSAN, 2003).

Desde o enunciado do problema da braquistócrona, em 1696, que visa encontrar a curva em que uma partícula precisa descrever para sair de um ponto A e chegar a um ponto B, ambos os pontos em um plano vertical, no menor tempo possível estando sobre ação apenas da força gravitacional (LIMA, 2004), diversas técnicas surgiram de modo a descrever esse fenômeno.

Algumas das principais contribuições para a área do cálculo variacional se devem aos matemáticos Euler e Lagrange, criando generalizações para métodos específicos utilizados, até então, na resolução de cada problema proposto (COURANT; ROBBINS, 1996).

Além dos métodos que permitem a resolução analítica dos problemas, surgem métodos que fornecem resoluções aproximadas e que são utilizados, principalmente, em aplicações à engenharia. O método de Rayleigh-Ritz, por exemplo, prevê o uso de uma função aproximadora no lugar da função exata, permitindo, assim, resoluções aproximadas para o problema do funcional (ASSAN, 2003).

A justificativa para o trabalho é a importância acadêmica dos temas, tendo em vista a grande gama de aplicações do cálculo variacional e do método de Rayleigh-Ritz, desde os problemas clássicos, como a braquistócrona, a problemas não convencionais, como, por exemplo, a solução de Schwarzschild para as Equações de Einstein, tratada por Campos (2017), e a determinação da forma que um cabo longo, inestensível, assume quando suspenso por suas extremidades, em alturas iguais, o chamado problema da catenária, mostrado por Lima (2004).

O método de Rayleigh-Ritz, por sua vez, tem diversas aplicações, principalmente na área das engenharias, por exemplo, na determinação das frequências naturais, como tratado por Ilanko, Monterrubio e Mochida (2014), e no estudo da deflexão de vigas, de acordo com Rodrigues, Rodrigues e Hauser (2014). O método tem sido aplicado, atualmente, junto a tecnologias como as técnicas de aprendizado de máquinas, conforme E e Yu (2018), objetivando abordagens modernas da área de inteligência artificial para os problemas do cálculo variacional.

O primeiro objetivo do trabalho é o de introduzir, de forma clara e concisa, o cálculo variacional por meio do estudo das Equações de Euler-Lagrange para funcionais que dependem de uma única variável, da função e das suas derivadas, até a ordem  $n$  desejada. O segundo objetivo é explicar o método de Rayleigh-Ritz, de forma simples e introdutória, desenvolvendo exemplos básicos. O terceiro, e último, objetivo do trabalho consiste em apresentar formas de se

utilizar a computação para os cálculos do método de Rayleigh-Ritz por meio da linguagem de programação *Python*.

O trabalho foi realizado por meio de um levantamento bibliográfico em artigos, dissertações, livros, artigos e outros documentos acadêmicos.

A monografia foi dividida em quatro capítulos, tendo, ainda, três apêndices. O primeiro capítulo, intitulado de Contexto Histórico, apresenta, de forma breve e sucinta, a visão geral e o contexto histórico sobre os máximos e mínimos, o cálculo variacional, o método de Rayleigh-Ritz e a linguagem de programação *Python*.

O segundo capítulo, Cálculo Variacional, expõe um dos problemas clássicos do cálculo variacional e, partindo desse problema, são deduzidas as equações de Euler-Lagrange para funcionais que dependem de uma variável, da função e de derivadas da função até a ordem  $n$ , além de serem estudadas as Condições de Contorno Naturais e Essenciais. Além disso, são apresentados, no segundo capítulo, exemplos de uso da equação de Euler-Lagrange e das Condições de Contorno.

O terceiro capítulo, Método de Rayleigh-Ritz, define o método de Rayleigh-Ritz e o trabalha de forma analítica, através de exemplo, além de iniciar o estudo das funções triangulares como funções de forma.

O quarto capítulo, Utilizando o Método de Rayleigh-Ritz na Linguagem *Python*, aborda o uso da linguagem de programação *Python* para a resolução de problemas através do método de Rayleigh-Ritz, primeiramente de forma analítica e, logo em seguida, por meio das funções triangulares, cujo estudo foi iniciado no terceiro capítulo, que discretizam a curva exata.

No Apêndice A, Conceitos de Análise no  $\mathbb{R}^n$ , são apresentados alguns conceitos da área, tais como a definição de conjuntos compactos e a Regra de Leibniz para a derivação sob o sinal da integral, utilizados no decorrer do trabalho, por exemplo, ao deduzir a equação de Euler-Lagrange. No Apêndice B, Encontrando a Máxima Deflexão da Viga Estudada, é estudado a máxima deflexão por meio da solução analítica do primeiro exemplo estudado pelo método de Rayleigh-Ritz a fim de se ter comparações durante o estudo do método. Já o Apêndice C, intitulado Breve Introdução à Linguagem *Python*, traz uma breve introdução sobre a linguagem, explicando como instalar a mesma, executar programas, as principais operações e, ainda, sobre como resolver sistemas lineares por meio dessa linguagem.



# 1 CONTEXTO HISTÓRICO

Para elucidar a história do cálculo variacional é importante mostrar um pouco da história dos máximos e mínimos de funções, problema pertinente ao cálculo diferencial, para, então, adentrar aos problemas de máximos e mínimos de funcionais, problema atribuído a área do cálculo variacional.

## 1.1 Máximos e mínimos

Os problemas de máximo e mínimo são corriqueiros na vida cotidiana, por exemplo, quando se quer encontrar o caminho com menor distância entre dois lugares para se caminhar uma menor distância, dentre vários outros problemas mais elaborados. Para esse exemplo específico não é necessário o uso de matemática avançada, porém, quanto mais complexidades são adicionadas aos problemas, mais ferramentas matemáticas são necessárias para a resolução, exata ou aproximada. Para simplificar estes processos, surgem os métodos para o cálculo de máximos e mínimos das funções.

Uma das primeiras formulações matemáticas próxima das atuais para os problemas de máximos e mínimos foi feita por Pierre de Fermat<sup>1</sup> em 1629 considerando curvas  $y = f(x)$ . Ele fez comparações de  $f(x)$  e  $f(x + E)$ , onde  $E$  é um número real positivo próximo de zero, ou seja, ele fez comparações do valor da função para pontos próximos. Esses valores geralmente são diferentes, porém, próximo de máximos ou mínimos a diferença se torna pequena. Deste modo, para achar os pontos de máximo ou mínimo, Fermat fazia

$$\frac{f(x + E) - f(x)}{E}$$

e, após realizar a divisão, considerava  $E = 0$ . Após considerar o valor de  $E$  igual a 0, Fermat igualava a expressão obtida a 0, de onde conseguia extrair os valores das abscissas dos pontos de máximos e mínimos da função (BOYER, 1996).

O que Fermat fez, de fato, foi igualar a primeira derivada de uma função a 0. É importante ressaltar que esse método utilizado por Fermat veio antes mesmo da invenção do cálculo diferencial por Isaac Newton<sup>2</sup> em 1665-1666 e Gottfried Wilhelm Leibniz<sup>3</sup> em 1676, de forma independente (BOYER, 1996).

---

<sup>1</sup> ★ 1601 - † 1665

<sup>2</sup> ★ 1642 - † 1727

<sup>3</sup> ★ 1646 - † 1716

## 1.2 O cálculo variacional

O ponto de partida do cálculo variacional se deu com Johann Bernoulli<sup>4</sup>, em 1696, com a publicação do problema da braquistócrona no jornal científico *Acta Eruditorum* (COURANT; ROBBINS, 1996).

O problema pode ser enunciado como:

Sejam  $A$  e  $B$  dois pontos dados em um plano vertical. O problema da braquistócrona consiste em encontrar a curva que uma partícula  $M$  precisa descrever para sair de  $A$  e chegar em  $B$  no menor tempo possível, somente sob a ação da força da gravidade (LIMA, 2004, p. 3).

O próprio Johann Bernoulli foi um dos matemáticos que solucionou o problema da braquistócrona. Ele retardou a publicação da sua solução para estimular os matemáticos do seu tempo a testarem suas habilidades nesse novo tipo de problema matemático (COURANT; ROBBINS, 1996). Além de Johann Bernoulli, soluções independentes foram encontradas por diversos matemáticos, como Jakob Bernoulli<sup>5</sup> (1697), L'Hospital<sup>6</sup> (1697), Leibniz (1697), e Newton (1697), (STILLWELL, 2010).

A solução de Jakob Bernoulli considerava o aspecto da curva variável, sendo considerado o primeiro grande passo para o desenvolvimento do cálculo variacional (STILLWELL, 2010). Nesse problema, a quantidade a ser minimizada depende de uma curva e não apenas de uma variável real (COURANT; ROBBINS, 1996), diferentemente dos problemas relacionados ao cálculo diferencial, o que torna necessária a construção de novas ferramentas matemáticas.

Os métodos para a resolução de problemas deste tipo eram específicos com adaptações para cada caso, sendo que os métodos gerais para a resolução só foram desenvolvidos com o envolvimento dos matemáticos Euler e Lagrange nos estudos desses problemas (COURANT; ROBBINS, 1996).

## 1.3 O método de Rayleigh-Ritz

O processo de minimização de funcionais envolve diretamente a resolução de equações diferenciais que, na maioria das vezes, pode não ter solução analítica. Uma forma de se contornar esse problema é o uso de soluções aproximadas e, com esse intuito, surge o método de Rayleigh-Ritz, fazendo o uso de funções admissíveis.

O método de Rayleigh-Ritz carrega o nome de dois pesquisadores, Lord Rayleigh<sup>7</sup> e Walter Ritz<sup>8</sup>, porém seu desenvolvimento se deve, principalmente, a Walter Ritz, enquanto a

---

<sup>4</sup> ★ 1667 - † 1748

<sup>5</sup> ★ 1654 - † 1705

<sup>6</sup> ★ 1661 - † 1704

<sup>7</sup> ★ 1842 - † 1919

<sup>8</sup> ★ 1878 - † 1909

solução apresentada por Lord Rayleigh é tratada como introdução, como será descrito abaixo.

Lord Rayleigh, em 1877, trabalhando com as energias potencial e cinética de sistemas, assumiu um padrão de vibração numa determinada frequência calculando a frequência de vibração livre por meio do equacionamento das energias potencial e cinética. Esse método ficou conhecido como método de Rayleigh e a sua precisão depende de quão próximo o padrão de vibração escolhido está do correto (LEISSA, 2005).

Em 1908 e 1909, Walter Ritz publicou dois artigos descrevendo procedimentos simples para a resolução de problemas de valor de contorno e de autovalores numericamente para qualquer nível de exatidão necessário usando funcionais (LEISSA, 2005). Esse procedimento foi chamado de método de Ritz.

Não se sabe exatamente quando o nome Rayleigh passou a ser agregado ao método de Ritz, transformando-se em "método de Rayleigh-Ritz". Uma das razões pelas quais isso ocorreu é o fato de que o método de Rayleigh pode ser considerado como um caso especial do método de Ritz quando uma única função admissível é utilizada para a aproximação (LEISSA, 2005).

O método de Rayleigh foi utilizado frequentemente, porém, nas últimas décadas o método de Ritz ou, como é chamado hoje, de Rayleigh-Ritz se mostrou aplicável a uma gama enorme de problemas devido, principalmente, a revolução tecnológica promovida pelos computadores, a partir da invenção dos transistores e microprocessadores (LEISSA, 2005).

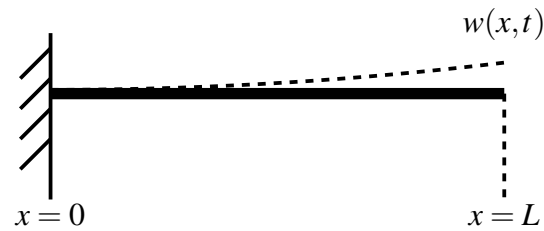
### 1.3.1 Aplicações do método de Rayleigh-Ritz

O método de Rayleigh-Ritz possui aplicações nas mais diversas áreas sendo, porém, a análise estrutural uma das áreas em que o método é mais utilizado. Seu uso é empregado, principalmente, com o objetivo de encontrar as chamadas frequências naturais. As frequências naturais "indicam a taxa de oscilação livre da estrutura, depois de cessada [a] força que provocou o seu movimento. Em palavras similares, representa o quanto [a] estrutura vibra quando não há força aplicada sobre ela." (BOLINA et al., 2015, p. 1).

Ilanko, Monterrubio e Mochida (2014) exemplificam essa aplicação ao procurar as frequências naturais em um tipo específico de viga, chamada de consola, que é uma viga apoiada apenas em uma de suas extremidades. A Figura 1.1 mostra uma representação de uma viga desse tipo, onde a curva  $w(x,t)$  representa o deslocamento dinâmico. Diversos outros trabalhos trazem essa mesma abordagem ao método, visando analisar frequências naturais de vibração, como, por exemplo, o trabalho de Mazanoglu (2015), que estuda a frequência natural de vigas seguindo o modelo de Timoshenko e de Euler usando o método de Rayleigh-Ritz. Grossi e Albarracin (2001) mostra a aplicação do método à problemas envolvendo placas, ao invés de vigas.

Além da abordagem convencional do método, existem trabalhos que visam aliar o método à técnicas de inteligência artificial, como o aprendizado de máquina. E e Yu (2018) afirmam que modelos baseados em redes neurais profundas podem ser utilizados em tarefas como

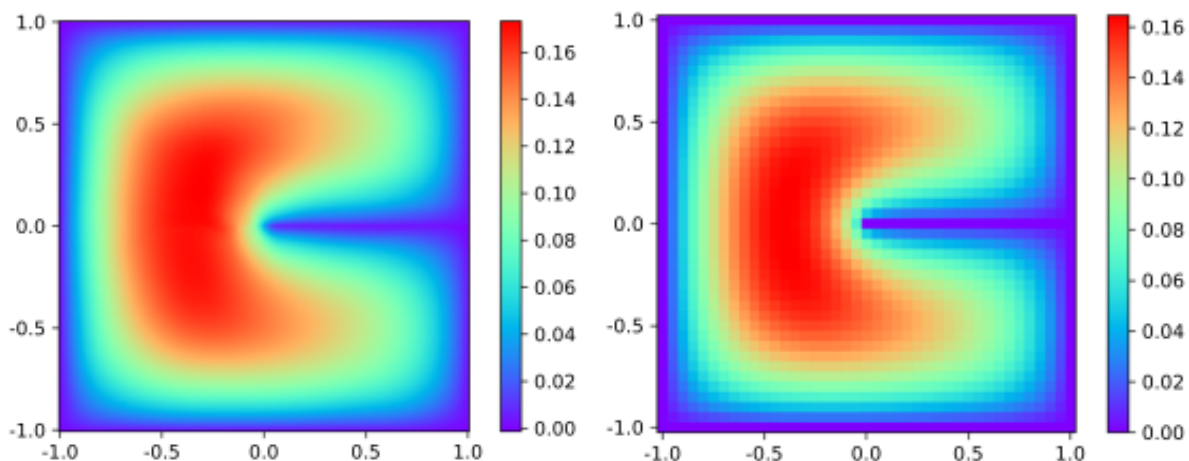
Figura 1.1 – Modelo de uma viga consola vibrando.



Fonte: Adaptada de Ilanko, Monterrubio e Mochida (2014, p. 23).

a resolução de equações diferenciais parciais ou a modelagem molecular. Além disso, o trabalho propõe um método chamado de *deep Ritz method* (método de Ritz profundo, em tradução literal) para a resolução de problemas variacionais. Esse método utiliza-se de representações de funções de redes neurais artificiais no contexto do método de Ritz (E; YU, 2018).

Um dos exemplos mostrados por E e Yu (2018) para ilustrar a eficácia da abordagem é a solução para a equação de Poisson em duas dimensões. A Figura 1.2 apresenta a solução utilizando dois métodos diferentes, sendo a Figura 1.2a a solução pelo *deep Ritz method* com 811 parâmetros e a Figura 1.2b a solução pelo método das diferenças finitas com 1681 parâmetros.

Figura 1.2 – Solução da equação de Poisson utilizando o *deep Ritz method* e o método das diferenças finitas(a) Solução pelo *Deep Ritz Method*.

(b) Solução pelo método das diferenças finitas.

Fonte: E e Yu (2018, p. 5).

Os exemplos acima citados mostram tanto aplicações clássicas quanto uma aplicação moderna do método de Rayleigh-Ritz e serve, inclusive, como motivação ao estudo do tema.

## 1.4 A linguagem Python

A linguagem de programação *Python* foi criada por volta de 1990 por Guido van Rossum no *Stichting Mathematisch Centrum* nos Países Baixos (ROSSUM, 2001). Guido é o seu principal criador, porém, com o desenvolvimento *open source*, a mesma apresenta colaborações de diversas pessoas — atualmente, há cerca de mil colaboradores no repositório da linguagem no GitHub<sup>9</sup>. Em 2001, foi criada a *Python Software Foundation*, organização sem fins lucrativos, responsável por possuir a propriedade intelectual relacionada a Linguagem *Python* (ROSSUM, 2001).

As principais motivações para o uso da linguagem *Python* neste trabalho é o seu funcionamento simples, a facilidade de uso, grande comunidade de usuários, além do fato de ser uma linguagem de programação gratuita e de código aberto tornando-a, deste modo, acessível.

Além dos diversos colaboradores diretos no desenvolvimento da linguagem, existem os colaboradores indiretos que criam, por exemplo, os pacotes com funções prontas para o uso. Dentre os pacotes científicos mais conhecidos estão alguns do ecossistema *SciPy*:

- *NumPy* é um pacote utilizado para computação científica contendo objetos para trabalhar com *arrays* (arranjos em português) multidimensionais, rotinas de operações em *arrays*, transformadas de Fourier, álgebra linear, estatística, dentre outras funcionalidades (NUMPY COMMUNITY, 2019).
- *SciPy* contém módulos para matemática, ciências e engenharia. As funções trazidas pela *SciPy* permitem o trabalho com álgebra linear, cálculo, interpolação, manipulação de arquivos, otimização, processamento de sinais, entre outros (SCIPY COMMUNITY, 2019).
- *Matplotlib* é uma biblioteca que permite plotar dados, tornando a visualização dos mesmos tão simples quanto possível (HUNTER et al., 2019). Segundo Hunter et al. (2019), a biblioteca possui um ambiente de estados de máquina similar ao utilizado no *software* comercial MATLAB<sup>10</sup>, trazendo uma experiência similar para usuários familiarizados com o MATLAB.
- *SymPy* é uma biblioteca que contém classes e funções que permitem a manipulação simbólica de elementos matemáticos, isto é, os elementos matemáticos são representados de forma exata e não aproximados (SYMPY DEVELOPMENT TEAM, 2019). Por exemplo, ao se utilizar uma calculadora científica para realizar algum cálculo com o número  $\pi$ , será apresentado no visor o cálculo utilizando uma aproximação para  $\pi$ . Já na matemática simbólica,  $\pi$  não seria aproximado e, portanto, o número continuaria sendo representado pela

<sup>9</sup> <<https://github.com/python/cpython/>>.

<sup>10</sup> O MATLAB, comercializado pela *MathWorks*, é uma ferramenta matemática e um ambiente de desenvolvimento integrado voltado para o desenvolvimento de programas de natureza técnica, como, por exemplo, programas de cálculo numérico.

letra grega. É importante perceber que, em determinados casos, a abordagem aproximada é mais eficiente e, em outros, a abordagem simbólica.

Existem outros pacotes científicos no ecossistema do *SciPy* e, também, fora do ecossistema, porém, serão citados, aqui, apenas esses devido ao fato de que os seus usos são voltados para a matemática e aplicações.

A linguagem e ambos os componentes citados são desenvolvidos em código aberto de forma colaborativa. Uma das vantagens dessa forma de desenvolvimento é o rápido aperfeiçoamento de suas funcionalidades, documentação e, também, a rápida correção de erros.

## 2 CÁLCULO VARIACIONAL

Um dos problemas pertinentes ao cálculo variacional é o de encontrar uma função  $y = y(x)$ , satisfazendo as condições de contorno  $y(x_1) = y_1$  e  $y(x_2) = y_2$ , que faça com que o funcional

$$I = \int_{x_1}^{x_2} f(x, y, y') dx, \quad (2.1)$$

onde  $f(x, y, y')$  é uma função de classe  $C^1$  (Ver Definição A.2 no Apêndice A), tenha valor mínimo ou máximo. Esse processo é chamado de minimizar ou maximizar o funcional. Minimizar ou maximizar um funcional é o mesmo que torná-lo estacionário. Assim, dizer que o funcional é estacionário é o mesmo que dizer que ele é mínimo ou máximo. Os valores  $x_1, x_2, y_1$  e  $y_2$  são números dados.

Para encontrar a função  $y = y(x)$  procurada, são necessárias algumas ferramentas, dentre as quais a abordada neste estudo é a equação de Euler-Lagrange. Os conceitos, definições e resultados apresentados neste capítulo foram elaboradas segundo Assan (2003), Lima (2004) e Campos (2017).

### 2.1 Equação de Euler-Lagrange

De início, é preciso demonstrar um Lema, conhecido como Lema de Lagrange, que servirá como base para a dedução da equação de Euler-Lagrange.

**Lema 2.1.** *Sejam  $x_1 < x_2$  fixos e  $G(x)$  uma função contínua particular para  $x_1 \leq x \leq x_2$ . Se*

$$\int_{x_1}^{x_2} \eta(x) G(x) dx = 0$$

*para cada função diferenciável  $\eta(x)$  tal que  $\eta(x_1) = \eta(x_2) = 0$ , concluímos que  $G(x) = 0$ , para todo  $x$  de modo que  $x_1 \leq x \leq x_2$ .*

*Demonstração.* Suponha que existe  $\bar{x}$  tal que  $x_1 < \bar{x} < x_2$  e  $G(\bar{x}) \neq 0$ . Podemos supor, sem perda de generalidade, que  $G(\bar{x}) > 0$ . Como  $G$  é contínua, existe uma vizinhança  $\bar{x}_1 \leq \bar{x} \leq \bar{x}_2$  onde  $G(x) > 0$  em toda a vizinhança.

É possível construir a seguinte função  $\eta(x)$ :

$$\eta(x) = \begin{cases} 0 & \text{para } x_1 \leq x < \bar{x}_1 \\ (x - \bar{x}_1)^2 (x - \bar{x}_2)^2 & \text{para } \bar{x}_1 \leq x \leq \bar{x}_2 \\ 0 & \text{para } \bar{x}_2 < x \leq x_2 \end{cases}$$

e então reescrever a integral do seguinte modo:

$$\int_{x_1}^{x_2} \eta(x)G(x)dx = \int_{\bar{x}_1}^{\bar{x}_2} (x - \bar{x}_1)^2(x - \bar{x}_2)^2G(x)dx.$$

Como  $G(x) > 0$  em  $\bar{x}_1 \leq x \leq \bar{x}_2$ , a integral do lado direito é estritamente positiva, contradizendo a hipótese. Portanto,  $G(\bar{x}) = 0$ . A demonstração considerando  $G(\bar{x}) < 0$  é análoga.  $\square$

Suponha  $x_1, x_2, y_1, y_2$  dados,  $f$  uma função de  $x, y$  e  $y'$  duas vezes diferenciável. É preciso construir uma família de funções admissíveis<sup>1</sup>,  $Y_\varepsilon(x)$ , definida por:

$$Y_\varepsilon(x) = y(x) + \varepsilon\eta(x), \quad (2.2)$$

onde  $\eta(x)$  é uma função diferenciável arbitrária para a qual  $\eta(x_1) = \eta(x_2) = 0$ . O número  $\varepsilon$  é o parâmetro da família. A representação gráfica das funções admissíveis pode ser vista na Figura 2.1. Assim, a derivada de  $Y$  em relação a  $x$ ,  $Y'(x)$ , é

$$Y'(x) = y'(x) + \varepsilon\eta'(x). \quad (2.3)$$

Reescrevendo a integral (2.1) utilizando as funções admissíveis definidas, tem-se

$$I(\varepsilon) = \int_{x_1}^{x_2} f(x, Y_\varepsilon, Y')dx. \quad (2.4)$$

Note que a função procurada  $y(x)$  é o membro da família  $Y_\varepsilon(x)$  quando  $\varepsilon = 0$ . Ou seja, se  $\varepsilon = 0$  pode-se substituir  $Y_\varepsilon$  e  $Y'$  por  $y$  e  $y'$ , respectivamente. Deste modo, a função que determina os extremos do funcional em (2.4) é a mesma que determina os extremos em (2.1) tomando  $\varepsilon = 0$ .

A condição necessária para que uma função de uma variável real tenha um extremo em algum ponto é que sua primeira derivada se anule nesse ponto. Então, é necessário que

$$I'(0) = 0. \quad (2.5)$$

Com o intuito de que não haja confusão com as notações, é necessário deixar claro que  $I'(\varepsilon)$  é a derivada de  $I$  em relação a  $\varepsilon$ , enquanto que  $y', Y', \eta'$  são derivadas em relação a  $x$ .

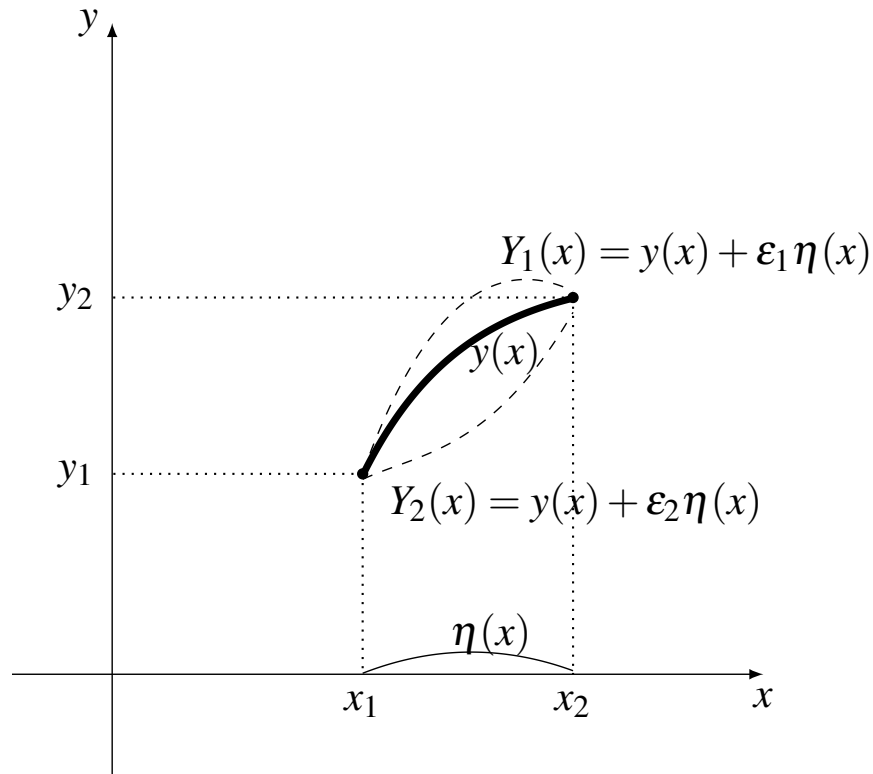
Utilizando a Regra de Leibniz (Teorema A.2 do Apêndice A), a derivada de (2.4) pode ser escrita como

$$I'(\varepsilon) = \int_{x_1}^{x_2} \frac{\partial f}{\partial \varepsilon}(x, Y_\varepsilon, Y')dx,$$

<sup>1</sup> Em outras literaturas, essas funções admissíveis podem ser chamadas, também, de funções aproximadoras, porém, devido ao uso de nomes parecidos em outros capítulos deste trabalho, foi preferível o emprego do nome funções admissíveis.



Figura 2.1 – Representação gráfica das funções admissíveis.



Fonte: Elaborada pelo autor, 2019.

e, aplicando regra da cadeia, obtêm-se

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial \varepsilon} + \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial \varepsilon} + \frac{\partial f}{\partial Y'} \frac{\partial Y'}{\partial \varepsilon} \right) dx,$$

onde a primeira parcela do integrando é nulo, dado ao fato de que  $x$  independe de  $\varepsilon$ , portanto  $\frac{\partial x}{\partial \varepsilon} = 0$ , ou seja,

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial \varepsilon} + \frac{\partial f}{\partial Y'} \frac{\partial Y'}{\partial \varepsilon} \right) dx. \quad (2.6)$$

Derivando (2.2) em função de  $\varepsilon$ , tem-se  $\frac{\partial Y}{\partial \varepsilon} = \frac{\partial y}{\partial \varepsilon} + \frac{\partial}{\partial \varepsilon}(\varepsilon\eta)$ , de onde conclui-se que  $\frac{\partial Y}{\partial \varepsilon} = \eta$ , pois  $y$  e  $\eta$  independem de  $\varepsilon$ . O mesmo acontece com (2.3), donde verifica-se que  $\frac{\partial Y'}{\partial \varepsilon} = \eta'$ . Deste modo, de (2.6), obtêm-se a integral

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial Y} \eta + \frac{\partial f}{\partial Y'} \eta' \right) dx.$$

Para calcular  $I'(0)$ , tem-se que  $\varepsilon = 0$  e, então, podemos trocar  $Y$  e  $Y'$  por  $y$  e  $y'$ , respectivamente, obtendo

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} \eta + \frac{\partial f}{\partial y'} \eta' \right) dx$$

$$I'(0) = \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx + \int_{x_1}^{x_2} \frac{\partial f}{\partial y'} \eta' dx.$$

Integrando a segunda parcela, por partes, tomando  $u = \frac{\partial f}{\partial y'}$  e  $dv = \eta' dx$  de onde obtêm-se  $du = \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) dx$  e  $v = \eta$ , segue que,

$$\begin{aligned} I'(0) &= \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx + \left( uv \Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} v du \right) \\ I'(0) &= \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx + \left( \frac{\partial f}{\partial y'} \eta \Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta dx \right) \end{aligned} \quad (2.7)$$

Sabe-se que  $\frac{\partial f}{\partial y'} \eta \Big|_{x_1}^{x_2} = 0$ , devido ao fato de que  $\eta(x_1) = \eta(x_2) = 0$ , portanto, (2.7) pode ser reescrita como

$$I'(0) = \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta dx \quad (2.8)$$

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \right) \eta dx$$

que, a partir da condição necessária (2.5), deve ser igualada a 0:

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \right) \eta dx = 0.$$

permitindo, pelo Lema 2.1, obter a seguinte equação:

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) = 0. \quad (2.9)$$

A equação diferencial (2.9) é chamada de equação de Euler-Lagrange e permite encontrar uma função  $f$  que maximiza ou minimiza a integral (2.1).

## 2.2 Condições de contorno naturais e essenciais

Durante o processo para a obtenção da função  $y(x)$  que torna o funcional estacionário aparecem algumas condições de contorno que devem ser tratadas cuidadosamente. Por exemplo, tomando o funcional

$$I = \int_{x_1}^{x_2} f(x, y, y') dx,$$

tem-se, da Seção 2.1, que sua condição de estacionariedade é dada por

$$\int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \right) \eta dx + \frac{\partial f}{\partial y'} \eta \Big|_{x_1}^{x_2} = 0, \quad (2.10)$$

onde, como já tratado,  $\eta$  representa uma variação da função exata conhecida.

Admitiu-se, em (2.8), que  $\eta(x_1) = \eta(x_2) = 0$ , porém isso nem sempre acontece. Existem casos em que, para que a condição de estacionariedade (2.10) seja válida é preciso, primeiro, que a primeira parcela de (2.10) seja nula, donde se resulta a equação de Euler-Lagrange

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) = 0,$$

e, ainda, que

$$\frac{\partial f}{\partial y'} \eta \Big|_{x_1}^{x_2} = 0,$$

de onde, geralmente, surgem as duas igualdades

$$\frac{\partial f}{\partial y'} \Big|_{x=x_1} = 0, \quad (2.11)$$

$$\frac{\partial f}{\partial y'} \Big|_{x=x_2} = 0. \quad (2.12)$$

É importante deixar claro que pode acontecer, por exemplo, de  $\eta(x_1) = 0$  e  $\eta(x_2) \neq 0$ , ou vice-versa, e, conseqüentemente, reduz-se uma dessas condições de contorno.

Essas condições de contorno são denominadas condições de contorno naturais do problema ou, ainda, problema de valor de contorno de Neumann. Elas tendem a aparecer normalmente durante o processo de encontrar  $y(x)$ , tornando o funcional estacionário.

Para o caso de funcionais que dependem de derivadas de ordem superior, da forma

$$I = \int_{x_1}^{x_2} f(x, y, y', \dots, y^{(n)}) dx,$$

aparecem diversas condições de contorno parecidas com (2.11) e (2.12). Um destes exemplos aparece em (2.22).

Outro tipo de condições de contorno que aparecem com problemas do cálculo variacional são condições de contorno geométricas. As condições de contorno geométricas precisam, necessariamente serem satisfeitas e são denominadas de condições de contorno essenciais ou, ainda, problemas de valor de contorno de Dirichlet.

A equação de Euler-Lagrange é uma equação diferencial e, quando resolvida, fornece uma solução geral, em função de suas constantes. Para determinar essas constantes e, deste modo, a solução particular para a equação de Euler-Lagrange trabalhada é necessário o uso tanto das condições de contorno naturais quanto das condições de contorno essenciais, como poderá ser visto no Exemplo 2.2 da Seção 2.4.

## 2.3 Equação de Euler-Lagrange para funcionais que dependem das derivadas de ordem superior

Existem também, funcionais que dependem de mais funções, funções de mais de uma variável ou, ainda, que dependem de derivadas de ordem superior. O último caso será mostrado aqui, tendo em vista seu uso nos próximos capítulos. Considere o funcional

$$I = \int_{x_1}^{x_2} f(x, y, y', y'', \dots, y^{(n)}) dx, \quad (2.13)$$

onde  $y^{(n)}$  representa a  $n$ -ésima derivada de  $y$ . Pode-se aproximar  $y(x)$  através de funções admissíveis  $Y_\varepsilon(x) = y(x) + \varepsilon \eta(x)$ , onde  $\eta(x_1) = \eta(x_2) = 0$  e  $\eta(x)$  deve ser uma função de classe  $C^n$  (Ver Definição A.2 no Apêndice A). Desse modo,

$$\begin{aligned} Y'(x) &= y'(x) + \varepsilon \eta'(x), \\ Y''(x) &= y''(x) + \varepsilon \eta''(x), \\ &\vdots \\ Y^{(n)}(x) &= y^{(n)}(x) + \varepsilon \eta^{(n)}(x). \end{aligned}$$

Substituindo  $y, y', y'', \dots, y^{(n)}$  por suas funções admissíveis  $Y, Y', Y'', \dots, Y^{(n)}$  em (2.13), o funcional pode ser escrito como

$$I(\varepsilon) = \int_{x_1}^{x_2} f(x, Y_\varepsilon, Y'_\varepsilon, Y''_\varepsilon, \dots, Y^{(n)}_\varepsilon) dx,$$

donde, derivando em função de  $\varepsilon$ , por meio da Regra de Leibniz (Teorema A.2 do Apêndice A), tem-se

$$I'(\varepsilon) = \int_{x_1}^{x_2} \frac{\partial f}{\partial \varepsilon}(x, Y_\varepsilon, Y'_\varepsilon, Y''_\varepsilon, \dots, Y^{(n)}_\varepsilon) dx. \quad (2.14)$$

Aplicando a Regra da Cadeia em (2.14),  $I'(\varepsilon)$  pode ser escrita como

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial x} \frac{\partial x}{\partial \varepsilon} + \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial \varepsilon} + \frac{\partial f}{\partial Y'} \frac{\partial Y'}{\partial \varepsilon} + \frac{\partial f}{\partial Y''} \frac{\partial Y''}{\partial \varepsilon} + \dots + \frac{\partial f}{\partial Y^{(n)}} \frac{\partial Y^{(n)}}{\partial \varepsilon} \right) dx,$$

onde  $\frac{\partial x}{\partial \varepsilon} = 0$ , pois  $x$  não depende de  $\varepsilon$ , assim,

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial Y} \frac{\partial Y}{\partial \varepsilon} + \frac{\partial f}{\partial Y'} \frac{\partial Y'}{\partial \varepsilon} + \frac{\partial f}{\partial Y''} \frac{\partial Y''}{\partial \varepsilon} + \dots + \frac{\partial f}{\partial Y^{(n)}} \frac{\partial Y^{(n)}}{\partial \varepsilon} \right) dx. \quad (2.15)$$

Derivando as funções admissíveis  $Y, Y', Y'', \dots, Y^{(n)}$  em relação a  $\varepsilon$ , como já feito anteriormente, é possível substituir em (2.15), os valores de  $\frac{\partial Y}{\partial \varepsilon}$  por  $\eta(x)$  e  $\frac{\partial Y^{(i)}}{\partial \varepsilon}$  por  $\eta^{(i)}(x)$  onde  $i = 1, 2, \dots, n$ . Assim, (2.15) pode ser reescrito como

$$I'(\varepsilon) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial Y} \eta + \frac{\partial f}{\partial Y'} \eta' + \frac{\partial f}{\partial Y''} \eta'' + \dots + \frac{\partial f}{\partial Y^{(n)}} \eta^{(n)} \right) dx. \quad (2.16)$$

Fazendo  $\varepsilon = 0$ , pode-se substituir  $Y, Y', Y'', \dots, Y^{(n)}$  por  $y, y', y'', \dots, y^{(n)}$ , respectivamente, então

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} \eta + \frac{\partial f}{\partial y'} \eta' + \frac{\partial f}{\partial y''} \eta'' + \dots + \frac{\partial f}{\partial y^{(n)}} \eta^{(n)} \right) dx$$

$$I'(0) = \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx + \int_{x_1}^{x_2} \frac{\partial f}{\partial y'} \eta' dx + \int_{x_1}^{x_2} \frac{\partial f}{\partial y''} \eta'' dx + \dots + \int_{x_1}^{x_2} \frac{\partial f}{\partial y^{(n)}} \eta^{(n)} dx. \quad (2.17)$$

As integrações para o caso geral serão omitidas aqui. Nas duas subseções próximas, será mostrado o processo para o caso em que  $n = 2$  e a equação geral para derivadas até a ordem  $n$ .

### 2.3.1 Processo para funcionais que dependem de derivadas de segunda ordem

Fazendo uso de (2.13) e (2.17) pode-se considerar o caso em que  $n = 2$ , onde o funcional é dado por

$$I = \int_{x_1}^{x_2} f(x, y, y', y'') dx,$$

e  $I'(0)$  é dado por

$$I'(0) = \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx + \int_{x_1}^{x_2} \frac{\partial f}{\partial y'} \eta' dx + \int_{x_1}^{x_2} \frac{\partial f}{\partial y''} \eta'' dx. \quad (2.18)$$

A segunda parcela pode ser integrada, por partes, com  $u = \frac{\partial f}{\partial y'}$  e  $dv = \eta' dx$ , donde  $du = \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) dx$  e  $v = \eta$ , logo

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial y'} \eta' dx = \frac{\partial f}{\partial y'} \eta \Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta dx,$$

mas  $\eta(x_1) = \eta(x_2) = 0$ , ou seja, a integral pode ser reescrita como

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial y'} \eta' dx = - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta dx. \quad (2.19)$$

A terceira parcela deve ser integrada duas vezes por partes. Na primeira integração, seja  $u = \frac{\partial f}{\partial y''}$  e  $dv = \eta'' dx$ , onde  $du = \frac{d}{dx} \left( \frac{\partial f}{\partial y''} \right) dx$  e  $v = \eta'$ . Assim,

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial y''} \eta'' dx = \frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y''} \right) \eta' dx. \quad (2.20)$$

Integrando, por partes, a segunda parcela de (2.20), tomando  $u = \frac{d}{dx} \left( \frac{\partial f}{\partial y''} \right)$  e  $dv = \eta' dx$ , de onde  $du = \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) dx$  e  $v = \eta$ . Portanto,

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial y''} \eta'' dx = \frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2} - \left[ \frac{d}{dx} \left( \frac{\partial f}{\partial y''} \right) \eta \Big|_{x_1}^{x_2} - \int_{x_1}^{x_2} \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) \eta dx \right],$$

e, como  $\eta(x_1) = \eta(x_2) = 0$ ,

$$\int_{x_1}^{x_2} \frac{\partial f}{\partial y''} \eta'' dx = \frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2} + \int_{x_1}^{x_2} \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) \eta dx. \quad (2.21)$$

Voltando a (2.18), e substituindo os valores de (2.19) e (2.21), tem-se

$$I'(0) = \int_{x_1}^{x_2} \frac{\partial f}{\partial y} \eta dx - \int_{x_1}^{x_2} \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) \eta dx + \frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2} + \int_{x_1}^{x_2} \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) \eta dx,$$

de onde, reorganizando os termos e colocando  $\eta$  em evidência, e, finalmente, fazendo  $I'(0) = 0$ , obtem-se

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) + \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) \right) \eta dx + \frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2} = 0. \quad (2.22)$$

A segunda parcela de (2.22),  $\frac{\partial f}{\partial y''} \eta' \Big|_{x_1}^{x_2}$ , é uma das chamadas Condições de Contorno Naturais e deve-se anular para que o funcional se torne estacionário. Porém, o foco nessa parte é tratar a equação de Euler-Lagrange, que é uma condição necessária, mas não suficiente, para que o funcional se torne estacionário. As Condições de Contorno Naturais são tratadas na Seção 2.2. Devido, então, ao fato de que a mesma deve-se anular, resta que

$$I'(0) = \int_{x_1}^{x_2} \left( \frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) + \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) \right) \eta dx, \quad (2.23)$$

onde, utilizando o Lema 2.1 em (2.23), a equação de Euler-Lagrange obtida é

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) + \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) = 0. \quad (2.24)$$

É importante ressaltar, novamente, que a equação de Euler-Lagrange (2.24) obtida é uma condição necessária, porém não é condição suficiente para a estacionariedade do funcional.

### 2.3.2 Equação geral

Para a obtenção da Equação Geral, para funcionais que dependem de derivadas até a ordem  $n$ , o processo para a obtenção da equação de Euler-Lagrange é análogo. Em (2.17) deve-se integrar a segunda parcela por partes uma vez, a terceira parcela duas vezes, a quarta parcela três vezes e assim por diante, até que a  $n$ -ésima parcela seja integrada  $n - 1$  vezes, também por partes. Ao fim do processo, a equação de Euler-Lagrange será obtida fazendo  $I'(0) = 0$ , isto é, aplicando a condição de estacionariedade e, por último, utilizando o Lema 2.1. Deste modo, a equação encontrada é

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) + \frac{d^2}{dx^2} \left( \frac{\partial f}{\partial y''} \right) - \frac{d^3}{dx^3} \left( \frac{\partial f}{\partial y'''} \right) + \dots + (-1)^n \frac{d^n}{dx^n} \left( \frac{\partial f}{\partial y^{(n)}} \right) = 0. \quad (2.25)$$

A equação (2.25) é escrita, também, como

$$\sum_{i=0}^n (-1)^i \frac{d^i}{dx^i} \left( \frac{\partial f}{\partial y^{(i)}} \right) = 0,$$

onde a primeira parcela,  $\frac{d^0}{dx^0} \left( \frac{\partial f}{\partial y^0} \right)$ , representa  $\frac{\partial f}{\partial y}$ .

## 2.4 Exemplos

**Exemplo 2.1** (Curva de Menor Comprimento entre Dois Pontos). *É sabido, do cálculo diferencial e integral, que o comprimento,  $I$ , de uma curva  $y(x)$  é determinado por*

$$I = \int_{x_1}^{x_2} \sqrt{1 + (y'(x))^2} dx, \quad (2.26)$$

que é um funcional que depende de  $y'(x)$ . Assim, pode-se determinar a curva  $y(x)$  que minimiza o funcional, ou seja, encontrar a menor curva entre dois pontos dados onde  $x_1$  e  $x_2$  são as abscissas do primeiro e do segundo ponto, respectivamente.

*Solução.* Como o funcional de (2.26) depende apenas de  $y'(x)$ , sua equação de Euler-Lagrange será da forma

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \left( \frac{\partial f}{\partial y'} \right) = 0, \quad (2.27)$$

onde  $f(x, y, y') = \sqrt{1 + (y'(x))^2}$ .

De (2.27), fazendo as devidas derivações, tem-se a equação

$$-\frac{d}{dx} \left( \frac{y'(x)}{\sqrt{1 + (y'(x))^2}} \right) = 0, \quad (2.28)$$

de onde tem-se

$$\frac{y''(x) \sqrt{1 + (y'(x))^2} - y'(x) \left( \frac{y'(x)y''(x)}{\sqrt{1 + (y'(x))^2}} \right)}{1 + (y'(x))^2} = 0,$$

e, devido ao fato de que o denominador  $1 + (y'(x))^2 \geq 1$ , pode-se analisar apenas o numerador, ou seja,

$$\begin{aligned} y''(x) \sqrt{1 + (y'(x))^2} - \frac{(y'(x))^2 y''(x)}{\sqrt{1 + (y'(x))^2}} &= 0 \\ \frac{y''(x)(1 + (y'(x))^2) - (y'(x))^2 y''(x)}{\sqrt{1 + (y'(x))^2}} &= 0, \end{aligned}$$

resultando, após manipulações no numerador, em

$$\frac{y''(x)}{\sqrt{1 + (y'(x))^2}} = 0,$$

e, finalmente,  $y''(x) = 0$ .

A equação diferencial  $y''(x) = 0$  pode ser resolvida por meio de integrações sucessivas

$$\int y''(x)dx = \int 0dx \implies y'(x) = C_1$$

$$\int y'(x)dx = \int C_1dx \implies y(x) = C_1x + C_2,$$

ou seja,  $y(x) = C_1x + C_2$ . Deste modo, a curva de menor comprimento entre os dois pontos é um segmento de reta.  $\square$

**Exemplo 2.2.** (ASSAN, 2003) Considere o funcional de energia potencial de uma viga dado por

$$\Pi = \frac{1}{2} \int_0^l EI \left( \frac{d^2v}{dx^2} \right)^2 dx - \int_0^l qvdx, \quad (2.29)$$

onde  $E$  é o módulo de Young do material,  $I$  é o momento de inércia da seção transversal da viga em relação ao eixo baricêntrico,  $l$  é o comprimento da viga e  $q$  é a carga distribuída de forma uniforme sobre a viga. Encontre  $v(x)$  de modo que  $\Pi$  seja estacionário, com as condições de contorno essenciais,  $v(0) = 0$  e  $v(l) = 0$ , e as condições de contorno naturais  $\left. \frac{d^2v}{dx^2} \right|_{x=0}$  e  $\left. \frac{d^2v}{dx^2} \right|_{x=l}$ .

*Solução.* A determinação de  $v(x)$  pode ser feita diretamente utilizando a equação de Euler-Lagrange (Ver Seção 2.1), porém, aqui será repetido o processo, mostrando que as condições de contorno naturais realmente surgem durante o processo de obtenção da equação. Variando  $v$  por  $\bar{v}_\varepsilon = v + \varepsilon\eta$  em (2.29), tem-se

$$\Pi(\varepsilon) = \frac{1}{2} \int_0^l EI \left( \frac{d^2\bar{v}}{dx^2} \right)^2 dx - \int_0^l q\bar{v}dx,$$

de onde, derivando  $\Pi(\varepsilon)$  em relação a  $\varepsilon$ , resulta em

$$\frac{d\Pi}{d\varepsilon} = \int_0^l EI \left( \frac{d^2\bar{v}}{dx^2} \right) \cdot \frac{d}{d\varepsilon} \left( \frac{d^2\bar{v}}{dx^2} \right) dx - \int_0^l q \frac{d\bar{v}}{d\varepsilon} dx. \quad (2.30)$$

Manipulando (2.30), e calculando  $\left. \frac{d\Pi}{d\varepsilon} \right|_{\varepsilon=0}$ , pois  $\bar{v}$  pode ser substituído por  $v$ , naturalmente, chega-se a

$$\left. \frac{d\Pi}{d\varepsilon} \right|_{\varepsilon=0} = \int_0^l EI \left( \frac{d^2v}{dx^2} \right) \left( \frac{d^2\eta}{dx^2} \right) dx - \int_0^l q\eta dx,$$

onde a primeira parcela deve ser integrada, por partes, duas vezes, resultando em

$$\left. \frac{d\Pi}{d\varepsilon} \right|_{\varepsilon=0} = \int_0^l \left( EI \frac{d^4v}{dx^4} - q \right) \eta dx + EI \left. \frac{d^2v}{dx^2} \eta' \right|_0^l - EI \left. \frac{d^3v}{dx^3} \eta \right|_0^l. \quad (2.31)$$

Aplicando a condição de estacionariedade,  $\left. \frac{d\Pi}{d\varepsilon} \right|_{\varepsilon=0} = 0$ , em (2.31),

$$\left. \frac{d\Pi}{d\varepsilon} \right|_{\varepsilon=0} = \int_0^l \left( EI \frac{d^4v}{dx^4} - q \right) \eta dx + EI \left. \frac{d^2v}{dx^2} \eta' \right|_0^l - EI \left. \frac{d^3v}{dx^3} \eta \right|_0^l = 0. \quad (2.32)$$



É preciso, então, que os termos de (2.32) se anulem. A primeira parcela de (2.32) resulta na equação de Euler-Lagrange

$$EI \frac{d^4 v}{dx^4} - q = 0, \quad (2.33)$$

enquanto que as duas outras parcelas são as condições de contorno naturais do problema

$$\frac{d^2 v}{dx^2} \Big|_{x=0} = 0, \quad (2.34)$$

$$\frac{d^2 v}{dx^2} \Big|_{x=l} = 0, \quad (2.35)$$

$$\frac{d^3 v}{dx^3} \Big|_{x=0} = 0, \quad (2.36)$$

$$\frac{d^3 v}{dx^3} \Big|_{x=l} = 0. \quad (2.37)$$

A equação de Euler-Lagrange (2.33) pode ser facilmente resolvida, isolando  $\frac{d^4 v}{dx^4}$  e integrando ambos os lados da equação quatro vezes seguidas. Ao final desse processo, a função  $v$  encontrada será

$$v(x) = \frac{q}{24EI} x^4 + \frac{C_1}{6} x^3 + \frac{C_2}{2} x^2 + C_3 x + C_4, \quad (2.38)$$

onde os coeficientes  $C_1$ ,  $C_2$ ,  $C_3$  e  $C_4$  podem ser determinados através do uso das condições de contorno naturais e essenciais.

Da condição de contorno essencial,  $v(0) = 0$ , resulta que o coeficiente  $C_4 = 0$ . Aplicando, então as condições de contorno naturais (2.34), tem-se

$$\left( \frac{q}{2EI} x^2 + C_1 x + C_2 \right) \Big|_{x=0} = 0,$$

resultando no valor do coeficiente  $C_2 = 0$ . Utilizando o outro extremo de (2.34), tem-se

$$\left( \frac{q}{2EI} x^2 + C_1 x + C_2 \right) \Big|_{x=l} = 0,$$

onde

$$\frac{ql^2}{2EI} + C_1 l = 0,$$

resultando em  $C_1 = -\frac{ql}{2EI}$ .

Para o último coeficiente,  $C_3$ , será utilizado a outra condição de contorno essencial,  $v(l) = 0$ ,

$$\frac{q}{24EI} l^4 - \frac{ql^4}{12EI} + C_3 l = 0,$$

tendo, assim, como resultado,  $C_3 = \frac{ql^3}{24EI}$ .

Substituindo, então, os coeficientes  $C_1$ ,  $C_2$ ,  $C_3$  e  $C_4$  determinados na função  $v(x)$ , encontrada em (2.38),

$$v(x) = \frac{q}{24EI}x^4 - \frac{ql}{12EI}x^3 + \frac{ql^3}{24EI}x. \quad (2.39)$$

□

### 3 MÉTODO DE RAYLEIGH-RITZ

Um dos objetivos do cálculo variacional é encontrar uma função  $y(x)$  que, dentre as funções admissíveis torna o funcional estacionário. Uma das formas para se encontrar a função  $y(x)$  é encontrando a Equação de Euler-Lagrange, ver Seção 2.1 e 2.3 do Capítulo 2, associada ao problema e, então, resolvê-la.

A resolução de equações diferenciais e a determinação da solução exata por meio das condições de contorno do problema é, entretanto, uma tarefa trabalhosa e, em muitos casos, até não factível. Para se esquivar de problemas como esses, surgem alguns métodos, dentre os quais está o método de Rayleigh-Ritz.

As explicações e os conceitos aqui utilizados foram elaboradas tomando como referências os autores Assan (2003) e Rodrigues, Rodrigues e Hauser (2014).

O método de Rayleigh-Ritz propõe que a função  $y(x)$ , exata, seja substituída por uma função aproximada  $v(x)$  que é formada pela combinação linear de funções  $\phi_i(x)$ . Uma das vantagens do uso do método é que ele não requer que as condições de contorno naturais sejam satisfeitas (GROSSI; ALBARRACIN, 2001). Deste modo, o método é empregado principalmente em problemas onde as condições de contorno naturais são de difícil resolução.

Considerando o funcional

$$I = \int_{x_1}^{x_2} f(x, y, y', \dots, y^{(m)}) dx,$$

com as condições de contorno  $y(x_1) = y(x_2) = 0$ , então a função  $y(x)$  é aproximada por meio da função  $v(x)$  definida como

$$v(x) = \sum_{i=1}^n a_i \phi_i(x),$$

onde os coeficientes  $a_i \in \mathbb{R}$  são chamados de parâmetros de deslocamento, as funções  $\phi_i(x)$  são chamadas de funções de forma e devem ser linearmente independentes. As funções  $\phi_i(x)$  devem ser, também, diferenciáveis até a ordem  $m$ , onde  $m$  é a ordem da maior derivada do funcional. A função  $v(x)$  é chamada de função aproximadora.

Substituindo  $y$  por  $v$  no funcional,  $I$  passa a ser uma função de  $a_1, a_2, \dots, a_n$ , isto é,  $I = I(a_1, a_2, \dots, a_n)$ . A condição de máximo ou mínimo para o funcional  $I$  é, então, que as suas derivadas parciais em relação a  $a_1, a_2, \dots, a_n$  se anulem, ou seja

$$\frac{\partial I}{\partial a_i} = 0,$$

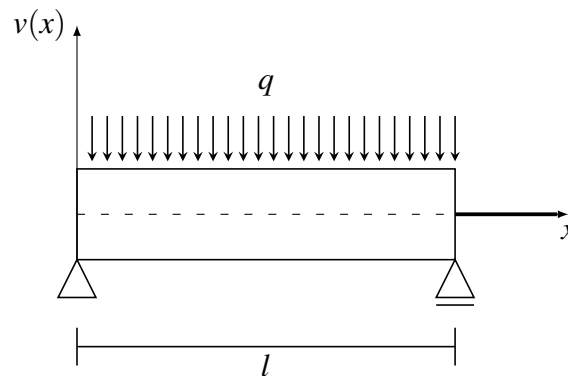
com  $i = 1, 2, 3, \dots, n$ .

**Exemplo 3.1.** (ASSAN, 2003) A viga prismática da Figura 3.1 possui o funcional de energia potencial total associado

$$\Pi = \frac{1}{2} \int_0^l EI \left[ \frac{d^2 v}{dx^2} \right]^2 dx - \int_0^l q v dx, \quad (3.1)$$

onde  $E$  é o módulo de Young do material,  $I$  é o momento de inércia da seção transversal da viga em relação ao eixo baricêntrico,  $l$  é o comprimento da viga e  $q$  é a carga distribuída de forma uniforme sobre a viga.

Figura 3.1 – Viga prismática.



Fonte: Assan (2003, p. 25).

Utilize o método de Rayleigh-Ritz para aproximar a função exata  $v(x)$  através de funções polinomiais de grau 2, 3 e 4 de modo que o funcional (3.1) se torne estacionário.

*Solução.* É possível encontrar uma função que aproxime a função exata  $v(x)$  tornando o funcional (3.1) estacionário utilizando o método de Rayleigh-Ritz. Esse processo será feito utilizando três funções aproximadoras:

$$v_1(x) = a_0 + a_1 x + a_2 x^2, \quad (3.2)$$

$$v_2(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \quad (3.3)$$

$$v_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4. \quad (3.4)$$

Para parâmetro de comparação com os resultados apresentados pelas funções  $v_1(x)$ ,  $v_2(x)$  e  $v_3(x)$  definidas em (3.2), (3.3) e (3.4), será utilizado o valor de máxima deflexão da Viga da Figura 3.1. Esse valor foi determinado no Apêndice B, em (B.4), como

$$v\left(\frac{l}{2}\right) = \frac{5ql^4}{384EI}, \quad (3.5)$$

Como primeira aproximação, pode-se utilizar a função  $v_1(x) = a_0 + a_1 x + a_2 x^2$ , de onde, considerando as condições de contorno essenciais, tem-se que  $v_1(0) = 0$ , levando a  $a_0 = 0$ ,

e de  $v_1(l) = 0$ , concluindo que  $a_1 = -a_2l$ . Assim, a função é escrita como

$$v_1(x) = a_2(x^2 - lx).$$

Sua derivada é  $v_1'(x) = a_2(2x - l)$  e, a derivada de segunda ordem é  $v_1''(x) = 2a_2$ . Substituindo esses valores no funcional (3.1), tem-se que

$$\Pi = \frac{1}{2} \int_0^l 4EIa_2^2 dx - \int_0^l qa_2(x^2 - lx) dx.$$

Da condição de estacionariedade tem-se, que  $\frac{\partial \Pi}{\partial a_2} = 0$  e, conseqüentemente

$$\frac{\partial \Pi}{\partial a_2} = \frac{\partial}{\partial a_2} \left[ \int_0^l (2EIa_2^2 - a_2q(x^2 - lx)) dx \right] = 0$$

$$\frac{\partial \Pi}{\partial a_2} = \int_0^l \frac{\partial}{\partial a_2} [2EIa_2^2 - a_2q(x^2 - lx)] dx = 0$$

$$\frac{\partial \Pi}{\partial a_2} = \int_0^l [4EIa_2 - q(x^2 - lx)] dx = 0$$

$$\left[ 4EIa_2x \Big|_0^l - q \left( \frac{x^3}{3} - \frac{lx^2}{2} \right) \Big|_0^l \right] = 0$$

$$4EIa_2l - q \left( \frac{l^3}{3} - \frac{l^3}{2} \right) = 0$$

$$4EIa_2l + \frac{ql^3}{6} = 0$$

$$a_2 = \frac{-ql^2}{24EI}.$$

Com o valor de  $a_2$  conhecido, é possível determinar o valor de  $a_1 = -a_2l$ , resultando em

$$a_1 = \frac{ql^3}{24EI}.$$

Assim, a função aproximadora  $v_1$  pode ser escrita como

$$v_1(x) = \frac{ql^3}{24EI}x - \frac{ql^2}{24EI}x^2.$$

Calculando  $v_1\left(\frac{l}{2}\right)$  a título de comparação com o valor exato de máxima deflexão da viga, dado por (3.5),

$$v_1\left(\frac{l}{2}\right) = \frac{ql^4}{96EI}. \quad (3.6)$$

Note que há uma grande diferença entre o valor encontrado pela função  $v_1(x)$ , (3.6), e o valor exato, (3.5). Visando uma melhoria nessa aproximação, faz-se necessário o uso da função

definida  $v_2(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ . Das condições de contorno essenciais,  $v_2(0) = 0$ , de onde  $a_0 = 0$ , e  $v_2(l) = 0$ , isolando  $a_1 = -a_2l - a_3l^2$ . Assim, a função  $v_2(x)$  é

$$v_2(x) = a_2(x^2 - lx) + a_3(x^3 - l^2x). \quad (3.7)$$

Derivando a função  $v_2$  em função de  $x$ , duas vezes, pode-se obter

$$\begin{aligned} v_2'(x) &= a_2(2x - l) + a_3(3x^2 - l^2) \\ v_2''(x) &= 2a_2 + 6a_3x. \end{aligned} \quad (3.8)$$

Substituindo (3.7) e (3.8) em  $\Pi$ , tem-se

$$\Pi = \frac{1}{2} \int_0^l (EI(2a_2 + 6a_3x)^2) dx - \int_0^l q(a_2(x^2 - lx) + a_3(x^3 - l^2x)) dx.$$

Derivando  $\Pi$  em função de  $a_2$  é possível obter

$$\begin{aligned} \frac{\partial \Pi}{\partial a_2} &= \int_0^l [2EI(2a_2 + 6a_3x) - q(x^2 - lx)] dx = 0 \\ \frac{\partial \Pi}{\partial a_2} &= \int_0^l [EI(4a_2 + 12a_3x) - q(x^2 - lx)] dx = 0 \\ &\int_0^l [4EIa_2 + 12EIa_3x - qx^2 + qlx] dx = 0 \\ 4EIa_2x \Big|_0^l + \frac{12a_3EI}{2}x^2 \Big|_0^l - \frac{q}{3}x^3 \Big|_0^l + \frac{ql}{2}x^2 \Big|_0^l &= 0 \\ 4EIa_2l + 6a_3EI l^2 &= \frac{2ql^3}{6} - \frac{3ql^3}{6} \\ 4a_2 + 6a_3l &= \frac{-ql^2}{6EI} \\ 2a_2 + 3a_3l &= \frac{-ql^2}{12EI}. \end{aligned} \quad (3.9)$$

Derivando  $\Pi$  em relação a  $a_3$  tem-se que

$$\begin{aligned} \frac{\partial \Pi}{\partial a_3} &= \int_0^l [6EI(2a_2 + 6a_3x)x - q(x^3 - l^2x)] dx = 0 \\ &\int_0^l [EI(12a_2 + 36a_3x)x - q(x^3 - l^2x)] dx = 0 \\ &\int_0^l [EI(12a_2x + 36a_3x^2) - q(x^3 - l^2x)] dx = 0 = 0 \\ EI \left( \frac{12a_2}{2}x^2 \Big|_0^l + \frac{36a_3}{3}x^3 \Big|_0^l \right) - q \left( \frac{x^4}{4} \Big|_0^l - \frac{l^2x^2}{2} \Big|_0^l \right) &= 0 \end{aligned}$$

$$\begin{aligned}
EI(6a_2l^2 + 12a_3l^3) - q\left(\frac{l^4}{4}\right) + q\left(\frac{l^4}{2}\right) &= 0 \\
6a_2 + 12a_3l &= -\frac{ql^4}{4EI} \\
2a_2 + 4a_3l &= -\frac{ql^2}{12EI}.
\end{aligned} \tag{3.10}$$

Montando um sistema de equações com (3.9) e (3.10):

$$\begin{cases} 2a_2 + 3a_3l = -\frac{ql^2}{12EI} \\ 2a_2 + 4a_3l = -\frac{ql^2}{12EI} \end{cases} \tag{3.11}$$

é possível encontrar os valores de  $a_2$  e  $a_3$ .

Subtraindo as equações em (3.11), tem-se

$$\begin{aligned}
4a_3l - 3a_3l &= -\frac{ql^2}{12EI} + \frac{ql^2}{12EI} \\
a_3 &= 0.
\end{aligned}$$

Consequentemente,

$$a_2 = -\frac{ql^2}{24EI}.$$

Assim, a função  $v_2(x)$  é a mesma que  $v_1(x)$ , não permitindo uma melhoria na aproximação desejada, sendo, portanto, inútil para esse caso. Ainda, visando melhor aproximação, deve-se passar para a próxima função aproximadora escolhida,  $v_3(x)$ .

Utilizando a função  $v_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ , tem-se das condições de contorno essenciais que  $v_3(0) = 0$ , donde  $a_0 = 0$  e,  $v_3(l) = 0$ , isto é  $a_1l + a_2l^2 + a_3l^3 + a_4l^4 = 0$ , donde pode-se concluir que  $a_1 = -a_2l - a_3l^2 - a_4l^3$ . Substituindo  $a_0$  e  $a_1$  em  $v_3(x)$ , podemos encontrar a função

$$v_3(x) = a_2(x^2 - lx) + a_3(x^3 - l^2x) + a_4(x^4 - l^3x). \tag{3.12}$$

Derivando  $v_3$  em relação a  $x$ , duas vezes:

$$v_3'(x) = a_2(2x - l) + a_3(3x^2 - l^2) + a_4(4x^3 - l^3)$$

$$v_3''(x) = 2a_2 + 6a_3x + 12a_4x^2. \tag{3.13}$$

Substituindo (3.12) e (3.13) em  $\Pi$ ,

$$\Pi = \frac{1}{2} \int_0^l [EI(2a_2 + 6a_3x + 12a_4x^2)^2] dx - \int_0^l q [a_2(x^2 - lx) + a_3(x^3 - l^2x) + a_4(x^4 - l^3x)] dx,$$

de onde pode-se derivar  $\Pi$  em relação a  $a_2$ ,  $a_3$  e  $a_4$  para obter um sistema de equações permitindo a determinação de  $a_2$ ,  $a_3$  e  $a_4$ .

Fazendo  $\frac{\partial \Pi}{\partial a_2} = 0$ :

$$\frac{\partial \Pi}{\partial a_2} = \int_0^l [2EI(2a_2 + 6a_3x + 12a_4x^2) - q(x^2 - lx)] dx = 0 \quad (3.14)$$

$$\int_0^l [EI(4a_2 + 12a_3x + 24a_4x^2) - q(x^2 - lx)] dx = 0$$

$$EI \left( 4a_2x \Big|_0^l + \frac{12a_3}{2}x^2 \Big|_0^l + \frac{24a_4}{3}x^3 \Big|_0^l \right) - q \left( \frac{x^3}{3} \Big|_0^l - \frac{lx^2}{2} \Big|_0^l \right) = 0$$

$$EI(4a_2l + 6a_3l^2 + 8a_4l^3) = \frac{ql^3}{3} - \frac{ql^3}{2}$$

$$4a_2l + 6a_3l^2 + 8a_4l^3 = -\frac{ql^3}{6EI}$$

$$2a_2 + 3a_3l + 4a_4l^2 = -\frac{ql^2}{12EI}. \quad (3.15)$$

Fazendo  $\frac{\partial \Pi}{\partial a_3} = 0$ , tem-se

$$\frac{\partial \Pi}{\partial a_3} = \int_0^l [6xEI(2a_2 + 6a_3x + 12a_4x^2) - q(x^3 - l^2x)] dx = 0 \quad (3.16)$$

$$\frac{\partial \Pi}{\partial a_3} = \int_0^l [EI(12a_2x + 36a_3x^2 + 72a_4x^3) - q(x^3 - l^2x)] dx = 0$$

$$EI \left( \frac{12a_2}{2}x^2 \Big|_0^l + \frac{36a_3}{3}x^3 \Big|_0^l + \frac{72a_4}{4}x^4 \Big|_0^l \right) - q \left( \frac{x^4}{4} \Big|_0^l - \frac{l^2x^2}{2} \Big|_0^l \right) = 0$$

$$EI(6a_2l^2 + 12a_3l^3 + 18a_4l^4) = \frac{ql^4}{4} - \frac{ql^4}{2}$$

$$EI(6a_2l^2 + 12a_3l^3 + 18a_4l^4) = -\frac{ql^4}{4}$$

$$6a_2 + 12a_3l + 18a_4l^2 = -\frac{ql^2}{4EI}$$

$$a_2 + 2a_3l + 3a_4l^2 = -\frac{ql^2}{24EI}. \quad (3.17)$$



Por último, fazendo  $\frac{\partial \Pi}{\partial a_4} = 0$ , tem-se

$$\frac{\partial \Pi}{\partial a_4} = \int_0^l [12x^2 EI(2a_2 + 6a_3x + 12a_4x^2) - q(x^4 - l^3x)] dx = 0 \quad (3.18)$$

$$\int_0^l [EI(24a_2x^2 + 72a_3x^3 + 144a_4x^4) - q(x^4 - l^3x)] dx = 0$$

$$EI \left( \frac{24a_2}{3} x^3 \Big|_0^l + \frac{72a_3}{4} x^4 \Big|_0^l + \frac{144a_4}{5} x^5 \Big|_0^l \right) - q \left( \frac{x^5}{5} \Big|_0^l - \frac{l^3x^2}{2} \Big|_0^l \right) = 0$$

$$EI \left( 8a_2l^3 + 18a_3l^4 + \frac{144}{5}a_4l^5 \right) = \frac{ql^5}{5} - \frac{ql^5}{2}$$

$$EI \left( 8a_2l^3 + 18a_3l^4 + \frac{144}{5}a_4l^5 \right) = -\frac{3ql^5}{10}$$

$$8a_2l^3 + 18a_3l^4 + \frac{144}{5}a_4l^5 = -\frac{3ql^5}{10EI}$$

$$8a_2 + 18a_3l + \frac{144}{5}a_4l^2 = -\frac{3ql^2}{10EI}. \quad (3.19)$$

É obtido, assim, um sistema linear de 3 equações em 3 incógnitas, com (3.15), (3.17) e (3.19):

$$\begin{cases} 2a_2 + 3a_3l + 4a_4l^2 = -\frac{ql^2}{12EI} \\ a_2 + 2a_3l + 3a_4l^2 = -\frac{ql^2}{24EI} \\ 8a_2 + 18a_3l + \frac{144}{5}a_4l^2 = -\frac{3ql^2}{10EI} \end{cases} \quad (3.20)$$

Chamando  $C = -\frac{ql^2}{12EI}$  e  $D = -\frac{3ql^2}{10EI}$ , tem-se a matriz aumentada do sistema linear (3.20).

$$\left( \begin{array}{ccc|c} 2 & 3l & 4l^2 & C \\ 1 & 2l & 3l^2 & \frac{1}{2}C \\ 8 & 18l & \frac{144}{5}l^2 & D \end{array} \right), \quad (3.21)$$

donde, resolvendo o sistema (Veja a Seção C.4 do Apêndice C), obtém-se os valores

$$a_2 = 0, \quad (3.22)$$

$$a_3 = -\frac{ql}{12EI}, \quad (3.23)$$

$$a_4 = \frac{q}{24EI}. \quad (3.24)$$

Deste modo, substituindo (3.22), (3.23) e (3.24) em (3.12), a função  $v_3(x)$  é escrita como

$$v_3(x) = \frac{q}{24EI}(x^4 - l^3x) - \frac{ql}{24EI}(x^3 - l^2x). \quad (3.25)$$

Utilizando os três valores, (3.22), (3.23) e (3.24), da solução do sistema linear com matriz aumentada (3.21), é possível determinar o coeficiente  $a_1 = -a_2l - a_3l^2 - a_4l^3$ , que é

$$a_1 = \frac{ql^3}{24EI}. \quad (3.26)$$

Não há, de fato, necessidade para a obtenção do valor do coeficiente  $a_1$ , pois  $v_3(x)$  foi escrita de forma independente de  $a_1$  em (3.12). Este apenas foi feito para que a função  $v_3(x)$  possa ser escrita, a partir de (3.4), como

$$v_3(x) = \frac{ql^3}{24EI}x - \frac{ql}{12EI}x^3 + \frac{q}{24EI}x^4. \quad (3.27)$$

O valor de máxima deflexão,  $v_3(\frac{l}{2})$  é dado por

$$v_3(\frac{l}{2}) = \frac{5ql^4}{384EI},$$

que é o valor exato de  $v(\frac{l}{2})$  indicado em (3.5). Não apenas isto, mas a função  $v_3(x)$ , indicada em (3.25) ou em (3.27), é a função  $v(x)$  exata.

□

A função  $v_3(x)$ , apresentada em (3.25), no Exemplo 3.1, é a função exata procurada. A escolha correta das funções de forma para a construção da função aproximadora é de suma importância para um bom resultado no método de Rayleigh-Ritz. Existem alguns critérios de convergência, que não serão tratados aqui devido sua complexidade, porém, ainda com os critérios de convergência a escolha das funções aproximadoras é uma tarefa que envolve tentativas e erros até que o resultado seja satisfatório.

Por exemplo, o funcional (3.1) do Exemplo 3.1 tem sua função  $v(x)$  exata conhecida, do Exemplo 2.2, facilitando, desta forma, a escolha das funções de forma. Como a função exata  $v(x)$  é uma função polinomial de grau 4, a escolha da função aproximadora como uma função polinomial foi bem prática.

Devido a função exata  $v(x)$  ser uma função polinomial de grau 4, caso a função aproximadora escolhida, no Exemplo 3.1, não fosse uma função polinomial de grau 4 ou superior, dificilmente a mesma se tornaria a função exata no método de Rayleigh-Ritz.

### 3.1 O uso de funções triangulares no método de Rayleigh-Ritz

Rodrigues, Rodrigues e Hauser (2014) apresentam o uso de funções triangulares como funções de forma, visando a discretização da função  $y(x)$ . Considere o funcional

$$I = \int_0^1 \left[ p(x) \left( \frac{dy}{dx} \right)^2 + q(x)(y(x))^2 - 2f(x)y(x) \right] dx, \quad (3.28)$$

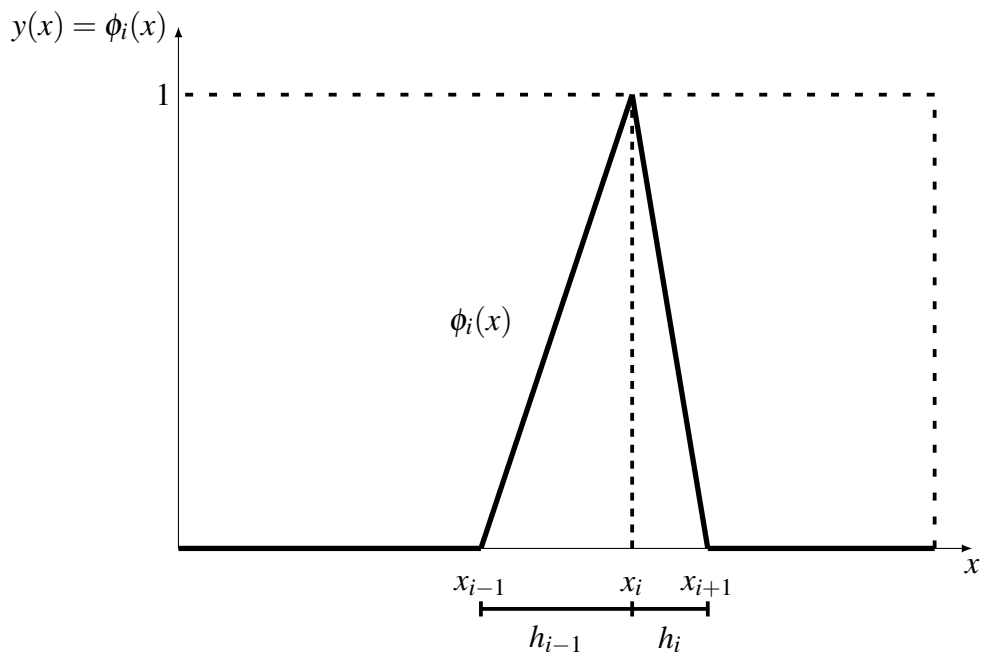
associado ao Problema de Valor de Contorno que modela a deflexão de uma viga apoiada sobre seus extremos. A função triangular trabalhada é definida como

$$\phi_i(x) = \begin{cases} 0 & \text{se } 0 \leq x \leq x_{i-1} \\ \frac{x - x_{i-1}}{h_{i-1}} & \text{se } x_{i-1} < x \leq x_i \\ \frac{x_{i+1} - x}{h_i} & \text{se } x_i < x \leq x_{i+1} \\ 0 & \text{se } x_{i+1} < x \leq 1 \end{cases}, \quad (3.29)$$

onde  $h_i = x_{i+1} - x_i$  e  $h_{i-1} = x_i - x_{i-1}$ . A Figura 3.2 apresenta uma ilustração da função  $\phi_i(x)$  definida. A função exata,  $y(x)$ , que torna um funcional estacionário, é aproximado, então, por meio da função aproximadora

$$v(x) = \sum_{i=1}^n a_i \phi_i(x). \quad (3.30)$$

Figura 3.2 – Função triangular  $\phi_i(x)$ .



Fonte: Elaborada pelo autor, 2019.

Observe que, de acordo com a definição das funções de forma  $\phi_i(x)$ , a função  $v(x)$ , definida como combinação linear das  $\phi_i(x)$ , funcionará apenas para funcionais que dependem

apenas de derivadas de primeira ordem, visto que a derivada de segunda ordem de qualquer  $\phi_i(x)$  é nula.

A primeira derivada das funções  $\phi_i(x)$  em relação a  $x$  é escrita como

$$\phi_i'(x) = \begin{cases} 0 & \text{se } 0 \leq x < x_{i-1} \\ \frac{1}{h_{i-1}} & \text{se } x_{i-1} < x < x_i \\ \frac{-1}{h_i} & \text{se } x_i < x < x_{i+1} \\ 0 & \text{se } x_{i+1} < x \leq 1 \end{cases}, \quad (3.31)$$

de onde a derivada de  $v(x)$  em relação a  $x$  é

$$v'(x) = \sum_{i=1}^n a_i \phi_i'(x). \quad (3.32)$$

Aproximando  $y(x)$  por meio da função  $v(x)$  definida em (3.30), o funcional (3.28) é escrito como

$$I = \int_0^1 \left[ p(x) \left( \sum_{i=1}^n a_i \phi_i'(x) \right)^2 + q(x) \left( \sum_{i=1}^n a_i \phi_i(x) \right)^2 - 2f(x) \left( \sum_{i=1}^n a_i \phi_i(x) \right) \right] dx. \quad (3.33)$$

Para determinar os coeficientes  $a_j$ , as derivadas parciais de (3.33) em relação aos  $a_j$  devem ser todas nulas, ou seja,

$$\frac{\partial I}{\partial a_j} = 0,$$

onde  $j = 1, 2, \dots, n$ . Utilizando A Regra de Leibniz (Teorema A.2 do Apêndice A),

$$\frac{\partial I}{\partial a_j} = \int_0^1 \left[ 2p(x) \left( \sum_{i=1}^n a_i \phi_i'(x) \right) \phi_j'(x) + 2q(x) \left( \sum_{i=1}^n a_i \phi_i(x) \right) \phi_j(x) - 2f(x) \phi_j(x) \right] dx. \quad (3.34)$$

Em (3.34) foi utilizado o fato de que a derivada de  $\sum_{i=1}^n a_i \phi_i(x)$  em relação a algum  $a_j$ , onde  $1 \leq j \leq n$ , resulta em  $\phi_j(x)$ . O mesmo vale para a derivada de  $\sum_{i=1}^n a_i \phi_i'(x)$  em relação a algum  $a_j$ ,  $1 \leq j \leq n$ , que resulta em  $\phi_j'(x)$ .

Aplicando a distributividade na primeira e segunda parcela do integrando em (3.34), tem-se

$$\begin{aligned} \frac{\partial I}{\partial a_j} &= \int_0^1 \left[ \sum_{i=1}^n (2p(x) a_i \phi_i'(x) \phi_j'(x)) + \sum_{i=1}^n (2q(x) a_i \phi_i(x) \phi_j(x)) - 2f(x) \phi_j(x) \right] dx \\ \frac{\partial I}{\partial a_j} &= \int_0^1 \left[ \sum_{i=1}^n (2p(x) a_i \phi_i'(x) \phi_j'(x) + 2q(x) a_i \phi_i(x) \phi_j(x)) \right] dx - \int_0^1 2f(x) \phi_j(x) dx, \end{aligned}$$

de onde, alterando a ordem de integração e soma, resulta em

$$\frac{\partial I}{\partial a_j} = \sum_{i=1}^n \left( \int_0^1 [2p(x)a_i\phi_i'(x)\phi_j'(x) + 2q(x)a_i\phi_i(x)\phi_j(x)] dx \right) - \int_0^1 2f(x)\phi_j(x)dx. \quad (3.35)$$

De (3.35), como  $a_i$  é constante na integração em relação a  $x$ , tem-se

$$\frac{\partial I}{\partial a_j} = \sum_{i=1}^n \left( a_i \int_0^1 [2p(x)\phi_i'(x)\phi_j'(x) + 2q(x)\phi_i(x)\phi_j(x)] \right) - \int_0^1 2f(x)\phi_j(x)dx,$$

que, aplicando a condição  $\frac{\partial I}{\partial a_j} = 0$ , obtêm-se

$$\sum_{i=1}^n \left( a_i \int_0^1 [2p(x)\phi_i'(x)\phi_j'(x) + 2q(x)\phi_i(x)\phi_j(x)] \right) - \int_0^1 2f(x)\phi_j(x)dx = 0,$$

resultando em um sistema de  $n$  equações com  $n$  incógnitas da forma

$$\sum_{i=1}^n \left( a_i \int_0^1 [2p(x)\phi_i'(x)\phi_j'(x) + 2q(x)\phi_i(x)\phi_j(x)] dx \right) = \int_0^1 2f(x)\phi_j(x)dx, \quad (3.36)$$

onde  $j = 1, 2, \dots, n$ .

Chamando

$$m_{j,i} = \int_0^1 [2p(x)\phi_i'(x)\phi_j'(x) + 2q(x)\phi_i(x)\phi_j(x)] dx, \quad (3.37)$$

e

$$b_j = \int_0^1 2f(x)\phi_j(x)dx, \quad (3.38)$$

o sistema (3.36) pode ser escrito na forma

$$\sum_{i=1}^n a_i \cdot m_{j,i} = b_j, \quad (3.39)$$

onde  $j = 1, 2, \dots, n$ . Seja as matrizes  $M = [m_{j,i}]$  de ordem  $n \times n$ ,  $B = [b_j]$  de ordem  $n \times 1$  e  $A = [a_j]$  de ordem  $n \times 1$ , então o sistema (3.39) é escrito como  $M \cdot A = B$ . A matriz  $M$  é a matriz de coeficientes, a matriz  $A$  é a matriz coluna de incógnitas e  $B$  é a matriz coluna contendo os termos independentes.

A partir de (3.29) e (3.31), os produtos  $\phi_i(x)\phi_j(x)$  e  $\phi_i'(x)\phi_j'(x)$  são não nulos apenas quando  $i = j - 1$ ,  $i = j$  e  $i = j + 1$ . Consequentemente,  $m_{j,i}$  é não nulo apenas para  $i = j - 1$ ,  $i = j$  e  $i = j + 1$ .

A construção do sistema  $M \cdot A = B$  e sua resolução, pelo método de Gauss-Jordan, serão implementado na linguagem de programação *Python* na Seção 4.1 do Capítulo 4.



# 4 UTILIZANDO O MÉTODO DE RAYLEIGH-RITZ NA LINGUAGEM PYTHON

Em se tratando de programação, existem diversas formas de se trabalhar o método de Rayleigh-Ritz em diversas linguagens. Para este trabalho foi escolhida a linguagem *Python* por ser uma linguagem gratuita, de código aberto, e com forte apelo a computação científica. Ainda assim, existem diversas formas de se implementar algoritmos para o uso deste método. O objetivo deste capítulo é de trabalhar e exemplificar como o método pode ser implementado utilizando a linguagem de programação *Python*.

Voltando ao funcional  $\Pi$ , definido em (3.1) no Exemplo 3.1 do Capítulo 3, tem-se

$$\Pi = \frac{1}{2} \int_0^l EI \left[ \frac{d^2 v}{dx^2} \right]^2 dx - \int_0^l qv dx$$

e tomando, ainda, a função aproximadora  $v_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ , donde aplicando as condições de contorno essenciais,  $v_3(0) = 0$ , obtendo  $a_0 = 0$ , e  $v_3(l) = 0$ , encontrando  $a_1 = -a_2l - a_3l^2 - a_4l^3$ . Substituindo  $a_0$  e  $a_1$  em  $v_3(x)$ , tem-se a função

$$v_3(x) = a_2(x^2 - lx) + a_3(x^3 - l^2x) + a_4(x^4 - l^3x).$$

Derivando  $v_3$  em relação a  $x$  duas vezes,

$$v_3''(x) = 2a_2 + 6a_3x + 12a_4x^2.$$

Para a determinação dos parâmetros  $a_2$ ,  $a_3$  e  $a_4$ , deve-se substituir  $v_3(x)$  no funcional  $\Pi$ , e resolver o sistema

$$\begin{cases} \frac{\partial \Pi}{\partial a_2} = 0 \\ \frac{\partial \Pi}{\partial a_3} = 0 \\ \frac{\partial \Pi}{\partial a_4} = 0 \end{cases} .$$

Tanto a derivação quanto a integração podem ser feitas manualmente ou utilizando o pacote *SymPy* em *Python*, como feito no Código 4.1.

Código 4.1 – Código para a resolução do Exemplo 3.1

```

1 from sympy import symbols, integrate, diff, Matrix
2 from sympy.solvers import solve
3
4 l, E, I, x, q = symbols('l E I x q')
5 a2, a3, a4 = symbols("a2 a3 a4")
6
7 v = a2 * (x**2 - l*x)
8 v = v + a3 * (x**3 - l**2 * x)
9 v = v + a4 * (x**4 - l**3 * x)
10
11 dv = diff(v, x)
12 dv2 = diff(dv, x)
13
14 f = (E*I*((dv2)**2)) / 2 - q * v
15
16 df_a2 = diff(f, a2)
17 df_a3 = diff(f, a3)
18 df_a4 = diff(f, a4)
19
20 eq1 = integrate(df_a2, (x, 0, l))
21 eq2 = integrate(df_a3, (x, 0, l))
22 eq3 = integrate(df_a4, (x, 0, l))
23
24 sol = solve([eq1, eq2, eq3], [a2, a3, a4])
25 print(sol)

```

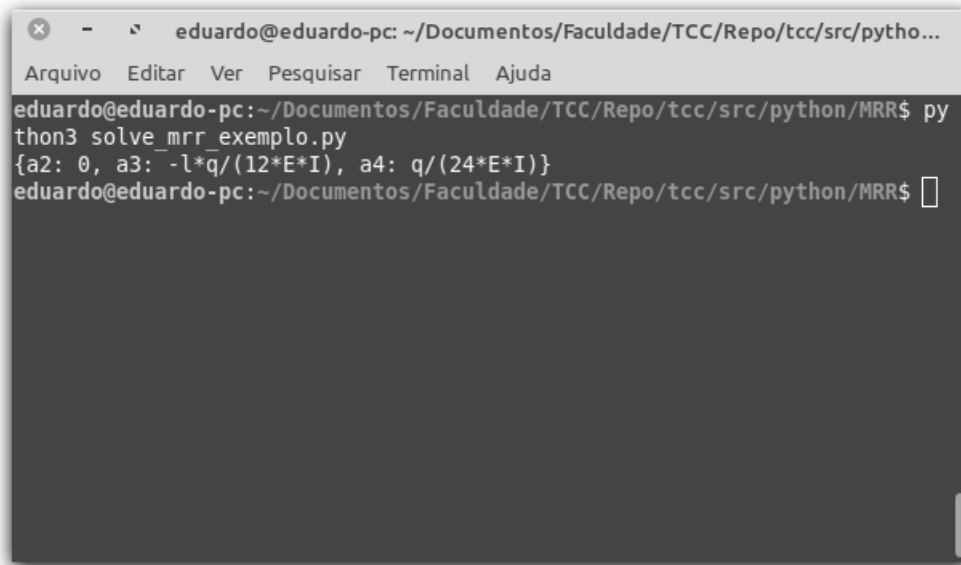
Fonte: Elaborado pelo autor, 2019.

O Código 4.1 é executado rapidamente e o resultado é mostrado na Figura 4.1, com  $a_2 = 0$ ,  $a_3 = -\frac{ql}{12EI}$  e  $a_4 = \frac{q}{24EI}$ . Esses resultados são exatamente os mesmos apresentados no Exemplo 3.1.

O valor de  $a_1 = -a_2l - a_3l^2 - a_4l^3$  é determinado substituindo os valores de  $a_2$ ,  $a_3$  e  $a_4$ . Esse processo pode ser realizado de modo manual ou, mais rapidamente, utilizando a própria linguagem, adicionando o Código 4.2 ao final do Código 4.1. Os comandos  $a1 = a1.subs(a2, sol[a2])$  indica que a variável  $a1$  receberá o valor de  $a1$  substituindo o símbolo  $a2$  pela solução do sistema encontrada no Código 4.1.



Figura 4.1 – Resultado apresentado pelo Código 4.1.



```

eduardo@eduardo-pc: ~/Documentos/Faculdade/TCC/Repo/tcc/src/pytho...
Arquivo Editar Ver Pesquisar Terminal Ajuda
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/MRR$ py
thon3 solve_mrr_exemplo.py
{a2: 0, a3: -l*q/(12*E*I), a4: q/(24*E*I)}
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/MRR$

```

Fonte: Elaborada pelo autor, 2019.

Código 4.2 – Código para a determinação do coeficiente  $a_1$  do Exemplo 3.1

```

26 a1 = -a2*1 - a3*1**2 - a4 * 1**3
27 a1 = a1.subs(a2, sol[a2])
28 a1 = a1.subs(a3, sol[a3])
29 a1 = a1.subs(a4, sol[a4])
30 print(a1)

```

Fonte: Elaborado pelo autor, 2019.

É possível, ainda, construir uma função  $v(x)$  utilizando os coeficientes  $a_1$ ,  $a_2$ ,  $a_3$  e  $a_4$  (O coeficiente  $a_0$  foi ignorado devido ao seu valor ser nulo) com o Código 4.3. Utilizando, novamente, o comando `subs` e a função definida no Código 4.3, pode-se calcular o valor de  $v_3(x)$  em algum ponto, como, por exemplo, em  $x = \frac{l}{2}$ . O valor  $v_3(\frac{l}{2})$  é o valor de máxima deflexão.

Código 4.3 – Código para a definição da função  $v_3(x)$  do Exemplo 3.1

```

31 v = a1*x + sol[a2]*x**2 + sol[a3]*x**3 + sol[a4]*x**4
32 print(v)
33 maxdef = v.subs(x, 1/2)
34 print(maxdef)

```


Fonte: Elaborado pelo autor, 2019.

É importante observar, no Código 4.3, que ao construir a variável  $v$  para representar a função, no primeiro coeficiente foi utilizado a variável  $a1$ , enquanto que nas demais, foi utilizado `sol[a2]`, `sol[a3]` e `sol[a4]`. Isso se deve ao fato de que os valores para os coeficientes  $a_2$ ,  $a_3$  e

$a_4$  foram encontrados pela resolução do sistema, sendo a variável  $sol$  a sua solução. Já,  $a_1$  foi definida como sendo a equação do termo  $a_1$  isolado em função de  $a_2$ ,  $a_3$  e  $a_4$ .

Os Códigos 4.1, 4.2 e 4.3 são executados em sequência e, por essa razão, estão com numeração de linhas sequenciais: O Código 4.1 tendo as linhas de 1 a 25, o Código 4.2 de 26 a 30 e o Código 4.3 de 31 a 34. A Figura 4.2 mostra o resultado da execução dos Códigos 4.1, 4.2 e 4.3 no terminal.

Figura 4.2 – Resultado apresentado pelos Códigos 4.1, 4.2 e 4.3.



```

eduardo@eduardo-pc: ~/Documentos/Faculdade/TCC/Repo/tcc/src/python...
Arquivo Editar Ver Pesquisar Terminal Ajuda
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/MRR$ py
thon3 solve_mrr_exemplo.py
{a2: 0, a3: -l*q/(12*E*I), a4: q/(24*E*I)}
l**3*q/(24*E*I)
l**3*q*x/(24*E*I) - l*q*x**3/(12*E*I) + q*x**4/(24*E*I)
5*l**4*q/(384*E*I)
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/MRR$ █

```

Fonte: Elaborada pelo autor, 2019.

Observando, no terminal da Figura 4.2, pela ordem dos comandos *print* executados no código, a primeira linha apresenta a solução do sistema, a segunda linha apresenta o valor do coeficiente  $a_1$ , a terceira linha apresenta a função  $v_3(x)$  e a quarta linha apresenta o resultado de  $v_3(\frac{l}{2})$  que é o valor de máxima deflexão da viga.

Ressaltando, novamente, que a escolha das funções de forma utilizadas na função aproximadora são de suma importância para o método de Rayleigh-Ritz. Para o uso de matemática simbólica, como nesse exemplo, deve-se tomar ainda mais cuidado com a escolha de tais funções pois, em alguns casos, a integração simbólica toma grande custo computacional. Em casos onde a integração simbólica se torna árdua, ou até mesmo impossível, o uso de algoritmos para integração numérica é necessário.

## 4.1 Resolvendo problemas com o método de Rayleigh-Ritz utilizando funções triangulares em *Python*

O sistema  $M \cdot A = B$ , encontrado na Seção 3.1 do Capítulo 3, pode ser resolvido, numericamente, utilizando a linguagem de programação *Python*. A título de comparação serão utilizadas as funções  $p(x) = e^x$ ,  $q(x) = e^x$  e  $f(x) = x + (2 - x)e^x$  que possui função exata  $y(x)$  conhecida dada por  $y(x) = (x - 1)(e^{-x} - 1)$ .

Viabilizando a implementação do algoritmo, faz-se necessário relembrar  $m_{j,i}$  e  $b_j$ , obtidos em (3.37) e (3.38), respectivamente:

$$m_{j,i} = \int_0^1 [2p(x)\phi_i'(x)\phi_j'(x) + 2q(x)\phi_i(x)\phi_j(x)] dx, \quad (4.1)$$

$$b_j = \int_0^1 2f(x)\phi_j(x)dx. \quad (4.2)$$

É útil, também, a definição da função de forma  $\phi_i(x)$ , feita em (3.29), e sua derivada,  $\phi_i'(x)$ , obtida em (3.31):

$$\phi_i(x) = \begin{cases} 0 & \text{se } 0 \leq x \leq x_{i-1} \\ \frac{x - x_{i-1}}{h_{i-1}} & \text{se } x_{i-1} < x \leq x_i \\ \frac{h_{i-1}}{x_{i+1} - x} & \text{se } x_i < x \leq x_{i+1} \\ 0 & \text{se } x_{i+1} < x \leq 1 \end{cases}, \quad (4.3)$$

$$\phi_i'(x) = \begin{cases} 0 & \text{se } 0 \leq x < x_{i-1} \\ \frac{1}{h_{i-1}} & \text{se } x_{i-1} < x < x_i \\ -\frac{1}{h_i} & \text{se } x_i < x < x_{i+1} \\ 0 & \text{se } x_{i+1} < x \leq 1 \end{cases}. \quad (4.4)$$

Para a resolução do problema, o primeiro passo é importar os componentes a serem utilizados. No Código 4.4 são importados os componentes *numpy*, utilizado para a criação de alguns intervalos numéricos espaçados. Do pacote *math* é importando a constante  $e$ . Do pacote *sympy* é importado o método *symbols* para a definição de símbolos matemáticos, de *sympy.matrices* é importado o método *zeros* para a criação de um matriz de tamanho determinado preenchida por números zeros e de *sympy.solvers* é importado o método *linsolve* utilizado para a resolução de sistemas lineares. Por último, é importado o pacote *matplotlib.pyplot* como *pyplot*<sup>1</sup> utilizado para plotar gráficos.

<sup>1</sup> Utilizar o comando `import matplotlib.pyplot as pyplot` faz com que, em vez de sempre escrever `matplotlib.pyplot`, se possa escrever apenas `pyplot` nos próximos comandos.

## Código 4.4 – Importação dos pacotes utilizados para a resolução

```

1 import numpy
2 from math import e
3 from sympy import symbols
4 from sympy.matrices import zeros
5 from sympy.solvers import linsolve
6 import matplotlib.pyplot as pyplot
7 import matplotlib.patches as patches

```

Fonte: Elaborado pelo autor, 2019.

Para o uso das funções de forma  $\phi_i(x)$  e  $\phi'_i(x)$ , definidas em (4.3) e (4.4), respectivamente, é preciso criar um conjunto numérico espaçado (aqui o conjunto será igualmente espaçado, mas nada impede o uso de conjuntos com espaçamentos diferentes). Para a criação desse conjunto foi utilizada a função *numpy.arange*<sup>2</sup>, em Código 4.5. No comando *arange*. O uso do extremo 1.0001 é apenas para garantir que o número 1 esteja no conjunto gerado. A variável *n* representa a quantidade de divisões a serem utilizadas no intervalo para a discretização da função exata.

## Código 4.5 – Criação dos intervalos numéricos

```

8 n = 5
9 space = 1/(n + 1)
10 intervals = numpy.arange(0, 1.0001, space)

```

Fonte: Elaborado pelo autor, 2019.

Código 4.6 – Função de forma  $\phi$  e sua derivada

```

11 def fi(i, x):
12     if x <= intervals[i - 1]:
13         return 0
14     elif x <= intervals[i]:
15         h = (intervals[i] - intervals[i - 1])
16         return ((x - intervals[i - 1]) / h)
17     elif x <= intervals[i + 1]:
18         h = (intervals[i + 1] - intervals[i])
19         return ((intervals[i + 1] - x) / h)
20     else:
21         return 0
22
23 def fi_diff(i, x):
24     if x <= intervals[i - 1]:

```

<sup>2</sup> A função *numpy.arange* cria valores espaçados uniformemente dentro de um determinado intervalo.

```

25         return 0
26     elif x <= intervals[i]:
27         return (1 / (intervals[i] - intervals[i - 1]))
28     elif x <= intervals[i + 1]:
29         return (1 / (intervals[i] - intervals[i + 1]))
30     else:
31         return 0

```

Fonte: Elaborado pelo autor, 2019.

Numericamente, as funções  $\phi_i(x)$  e  $\phi'_i(x)$  foram programadas utilizando condicionais, em Código 4.6.

No Código 4.6 o método utilizado para a integração numérica, que implementa a chamada regra do trapézio, é mostrado. Dentro do método, a variável *steps* define em quantos intervalos o domínio de integração será dividido. Os parâmetros opcionais *\*args*, do método *integrate*, são passados para a função a ser integrada, *func*, e serão utilizados para passar os índices *i* e *j* quando necessário.

Código 4.7 – Algoritmo para integração numérica usando a regra dos trapézios

```

32 def integrate(func, start, end, *args):
33     steps = 10000
34     h = ((end - start) / steps)
35     half = h / 2
36     value = 0
37     for i in range(0, steps):
38         dh = func(start + h * i + half, *args)
39         if i == 0 or i == (steps - 1):
40             value += dh
41         else:
42             value += (2 * dh)
43     return (value * (h / 2))

```

Fonte: Elaborado pelo autor, 2019.

Para a integração numérica por meio do algoritmo implementado em Código 4.7 é preciso definir funções com os integrandos de (4.1) e (4.2). Essas funções são chamadas, no código, de *m\_intern* e *b\_intern*, respectivamente. Na implementação do Código 4.8, os métodos *m(i, j)* e *b(i)* calculam os elementos das matrizes *M* e *B*, respectivamente. Nesses métodos, é somado 1 aos índices *i* e *j* pois, na programação os elementos são indexados a partir de 0 e, na forma como definimos as funções de forma  $\phi_i(x)$ , elas são indexadas a partir de 1.

Código 4.8 – Funções utilizadas para criação das matrizes do sistema linear

```

44 def p(x):

```

```

45     return e**x
46
47 def q(x):
48     return e**x
49
50 def f(x):
51     return (x+(2-x)*e**x)
52
53 def m_intern(x, i, j):
54     value = p(x) * fi_diff(i, x) * fi_diff(j, x)
55     value = value + q(x) * fi(i, x) * fi(j, x)
56     return value
57
58 def m(i, j):
59     i = i + 1
60     j = j + 1
61     return integrate(m_intern, 0, 1, i, j)
62
63 def b_intern(x, i):
64     return (f(x) * fi(i, x))
65
66 def b(i):
67     i = i + 1
68     return integrate(b_intern, 0, 1, i)

```

Fonte: Elaborado pelo autor, 2019.

O próximo passo, consiste em criar a matriz aumentada do sistema, os símbolos que representam as incógnitas e, então, resolvê-lo. Isso é feito no Código 4.9.

Código 4.9 – Definição e resolução do sistema linear

```

69 M = zeros(n, n+1)
70 for i in range(0, n):
71     M[i, n] = b(i)
72     M[i, i] = m(i, i)
73     if (i - 1) >= 0:
74         M[i, i - 1] = m(i, i - 1)
75     if (i + 1) < n:
76         M[i, i + 1] = m(i, i + 1)
77 coefficients = symbols('a_0:{}'.format(n))
78 solution, = linsolve(M, coefficients)

```

Fonte: Elaborado pelo autor, 2019.

Utilizando a solução do Código 4.9 e a função exata  $y(x) = (x - 1)(e^{-x} - 1)$ , é possível implementar métodos que representam a função exata e a função aproximada, plotando-as no mesmo gráfico, como feito no Código 4.10.

Código 4.10 – Plotagem das funções exata e aproximada

```
79 def function(solution, x):
80     value = 0
81     i = 0
82     for c in solution:
83         value += c * fi(i + 1, x)
84         i = i + 1
85     return value
86 def function_original(x):
87     return ((x - 1) * (e**(-x) - 1))
88
89 plot_interval = numpy.arange(0.0, 1.0, 0.01)
90 plot_original = function_original(plot_interval)
91 plot_aprox = []
92 for i in plot_interval:
93     plot_aprox.append(function(solution, i))
94 fig = pyplot.figure()
95 axes = fig.add_subplot(1, 1, 1)
96 color1='tab:blue'
97 color2='tab:orange'
98 axes.plot(plot_interval, plot_aprox, color=color1)
99 axes.plot(plot_interval, plot_original, color=color2)
100 axes.set_title('Rayleigh-Ritz With n=' + str(n))
101 blue_legend = patches.Patch(color=color1, label='Aproximada')
102 orange_legend = patches.Patch(color=color2, label='Exata')
103 pyplot.legend(handles=[blue_legend, orange_legend])
104 pyplot.show()
```

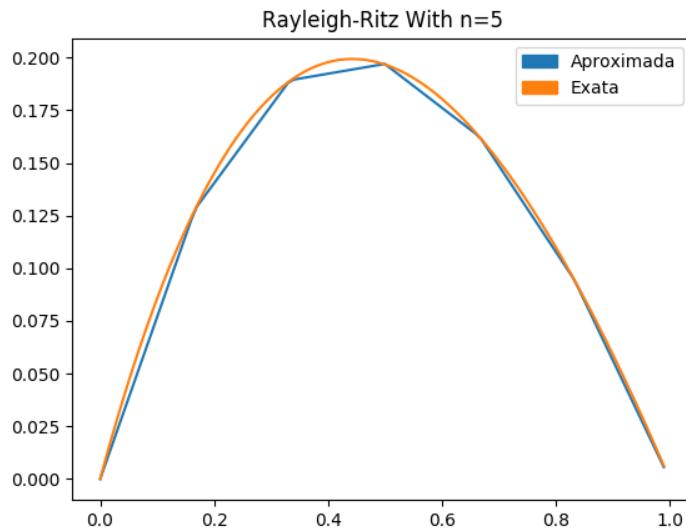
Fonte: Elaborado pelo autor, 2019.

Os Códigos 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 e 4.10 são sequenciais e devem ser executados dentro de um único arquivo, por isso suas numerações estão sequenciais. O resultado da execução do código para a resolução do sistema é mostrado na Figura 4.3, onde a curva de cor laranja é a solução exata e a curva de cor azul é a discretização. Para maior precisão do método, pode-se aumentar o valor da variável  $n$  do Código 4.5, por exemplo, para 10. O gráfico plotado para  $n = 10$  é apresentado na Figura 4.4.

Quanto maior o valor da variável  $n$  no Código 4.5 maior a precisão, porém, com maior custo computacional. Consequentemente, quanto maior o valor de  $n$ , mais tempo o computador

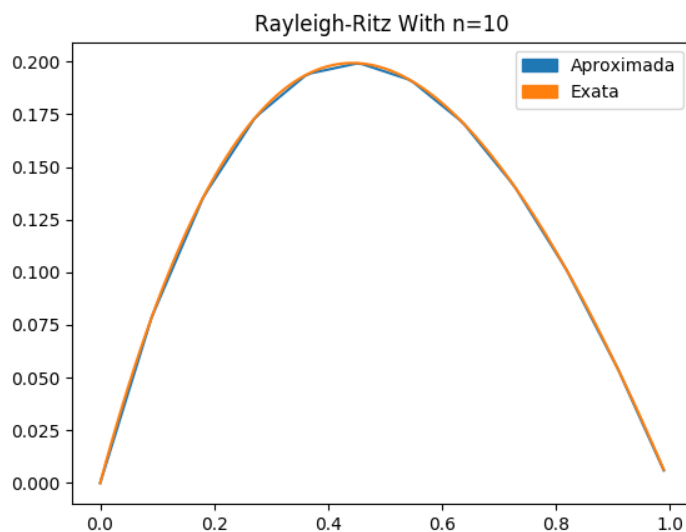
levará para a resolução do problema.

Figura 4.3 – Resultado do método de Rayleigh-Ritz utilizando funções de forma triangulares com  $n = 5$ .



Fonte: Elaborada pelo autor, 2019.

Figura 4.4 – Resultado do método de Rayleigh-Ritz utilizando funções de forma triangulares com  $n = 10$ .



Fonte: Elaborada pelo autor, 2019.



# CONSIDERAÇÕES FINAIS

Nesse trabalho, o cálculo variacional e o método de Rayleigh-Ritz foram expostos de forma introdutória, inclusive, desenvolvendo as demonstrações e cálculos, de modo que todo aluno com conhecimentos de cálculo em várias variáveis possa compreender as ideias trabalhadas. No decorrer dos estudos sobre tais temas foi observado a importância e a necessidade de materiais introdutórios sobre ambos os temas.

Para o cálculo variacional, o enfoque foi dado nas equações de Euler-Lagrange, condição necessária para a estacionariedade de um funcional, de funcionais que dependem de uma variável, uma função e de suas derivadas até uma ordem  $n$  qualquer. Esse enfoque dado se deve, primeiramente, ao enfoque introdutório do trabalho e, em segundo lugar, a grande gama de aplicações que podem ser desenvolvidas com as ferramentas estudadas, por exemplo, ao se provar que a curva de menor comprimento entre dois pontos é um segmento de reta.

Já no método de Rayleigh-Ritz, o foco foi dividido entre uma introdução ao tema, desenvolvendo todos os cálculos referentes ao exemplo, e o uso da Linguagem de Programação *Python* para agilizar o processo de minimização dos funcionais de forma rápida e simples, realizando as derivações, integrações e, ainda, resolvendo sistemas de equações.

Os estudos desenvolvidos com a linguagem *Python* mostraram, também, a utilidade da linguagem no meio científico, realizando tanto operações numéricas quanto manipulações simbólicas de forma eficiente. Isso evidencia a importância da linguagem por ser gratuita, de código aberto e multiplataforma, ou seja, uma linguagem acessível.

O estudo introdutório ao cálculo variacional e ao método de Rayleigh-Ritz, aqui apresentado, pode ser ampliado, por exemplo, para funções de mais variáveis, problemas de fronteira livres onde as condições de contorno são de outra forma, indo, até mesmo, aos métodos modernos que utilizam, por exemplo, inteligência artificial, como o método *Deep Ritz* citado no primeiro capítulo desse trabalho.



# REFERÊNCIAS

- ASSAN, A. E. *Método dos Elementos Finitos*. 2. ed. Campinas: Editora da Unicamp, 2003.
- BOLINA, C. et al. Vibrações: As frequências naturais estimada e experimental de uma estrutura. *Blucher Mathematical Proceedings*, v. 1, n. 1, p. 186 – 194, 2015. Disponível em: <[www.proceedings.blucher.com.br/article-details/vibraes-as-frequencias-naturais-estimada-e-experimental-de-uma-estrutura-11900](http://www.proceedings.blucher.com.br/article-details/vibraes-as-frequencias-naturais-estimada-e-experimental-de-uma-estrutura-11900)>. Acesso em: 5 out. 2019.
- BOYER, C. B. *História da Matemática*. 2. ed. São Paulo: Edgard Blucher, 1996.
- CAMPOS, C. A. de L. *Algumas Aplicações de Cálculo Variacional: da braquistócrona a desigualdade de hardy-sobolev*. Dissertação (Mestrado) — Universidade de Campinas, Campinas, jul. 2017.
- COURANT, R.; ROBBINS, H. *What is Mathematics?* 2. ed. New York: Oxford University Press, 1996.
- E, W.; YU, B. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, v. 6, n. 1, 2018.
- GROSSI, R. O.; ALBARRACIN, C. M. Some observations on the application of the rayleigh–ritz method. *Applied Acoustics*, v. 62, n. 10, p. 1171 – 1182, 2001.
- HUNTER, J. et al. *Matplotlib: Release 3.1.1*. [S.l.], 2019. Disponível em: <<https://matplotlib.org/Matplotlib.pdf>>. Acesso em: 15 out. 2019.
- ILANKO, S.; MONTEERRUBIO, L. E.; MOCHIDA, Y. The rayleigh–ritz method and simple applications. In: \_\_\_\_\_. *The Rayleigh–Ritz Method for Structural Analysis*. [S.l.]: John Wiley & Sons, Ltd, 2014. cap. 3, p. 21–31.
- LEISSA, A. The historical bases of the rayleigh and ritz methods. *Journal of Sound and Vibration*, v. 287, n. 4, p. 961 – 978, 2005.
- LIMA, E. L. *Curso de análise. Vol. 2*. 10. ed. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 2002. (Projeto Euclides).
- LIMA, G. L. de. *Cálculo Variacional: problemas clássicos, aspectos teóricos e desdobramentos*. Dissertação (Mestrado) — Universidade de Campinas, Campinas, jun. 2004.
- MAZANOGLU, K. Natural frequency analyses of segmented timoshenko–euler beams using the rayleigh–ritz method. *Journal of Vibration and Control*, v. 23, n. 13, p. 2135–2154, 2015.
- NUMPY COMMUNITY. *NumPy User Guide: Release 1.17.0rc1*. [S.l.], 2019. Disponível em: <<https://numpy.org/doc/1.17/numpy-user-1.17.0.pdf>>. Acesso em: 15 out. 2019.
- RODRIGUES, Y. E.; RODRIGUES, D. da S.; HAUSER, E. B. O método de rayleigh-ritz aplicado a um problema de deflexão de viga utilizando o matlab. *Proceeding Series of the Brazilian Society of Applied and Computational Mathematics*, v. 2, n. 1, 2014.

---

ROSSUM, G. van. *A.I History of the Software*. 2001. Disponível em: <<https://docs.python.org/2.0/ref/node92.html>>. Acesso em: 5 out. 2019.

SCIPY COMMUNITY. *SciPy Reference Guide: Release 1.3.0*. [S.l.], 2019. Disponível em: <<https://docs.scipy.org/doc/scipy-1.3.0/scipy-ref-1.3.0.pdf>>. Acesso em: 15 out. 2019.

STILLWELL, J. *Mathematics and Its History*. 3. ed. New York: Springer, 2010. (Undergraduate Texts In Mathematics).

SYMPY DEVELOPMENT TEAM. *SymPy Documentation: Release 1.4*. [S.l.], 2019. Disponível em: <<https://github.com/sympy/sympy/releases/tag/sympy-1.4>>. Acesso em: 15 out. 2019.

# Apêndices



# APÊNDICE A – CONCEITOS DE ANÁLISE NO $\mathbb{R}^n$

Durante o desenvolvimento dos resultados do texto, tais como a obtenção da equação de Euler-Lagrange, é necessário o uso de alguns conceitos e resultados de Análise no  $\mathbb{R}^n$ . Estes conceitos serão trabalhados neste apêndice, tomando como referência Lima (2002).

**Definição A.1.** Um conjunto  $K \subset \mathbb{R}^n$  é compacto se, e somente se, toda sequência de pontos em  $K$  possui uma subsequência que converge para um ponto de  $K$ .

**Teorema A.1.** Seja  $f : X \times K \rightarrow \mathbb{R}^n$  contínua, onde  $X \subset \mathbb{R}^m$ , para algum  $m$ , e  $K$  é compacto. Fixemos  $x_0 \in X$ . Para todo  $\varepsilon > 0$ , existe um  $\delta > 0$  tal que  $x \in X$ ,  $|x - x_0| < \delta \implies |f(x, \alpha) - f(x_0, \alpha)| < \varepsilon$ , seja qual for  $\alpha \in K$ .

*Demonstração.* Suponha que o teorema não seja válido, então existiriam  $\varepsilon > 0$  e sequências de pontos  $x_k \in X$ ,  $\alpha_k \in K$  tais que  $|x_k - x_0| < \frac{1}{k}$  e  $|f(x_k, \alpha_k) - f(x_0, \alpha_k)| \geq \varepsilon$ . Passando a uma subsequência, se necessário e admitindo que  $\lim \alpha_k = \alpha \in K$ , devido ao fato de que o conjunto  $K$  é compacto.

Como,  $|x_k - x_0| < \frac{1}{k}$ ,  $-\frac{1}{k} < x_k - x_0 < \frac{1}{k}$ , então,  $x_0 - \frac{1}{k} < x_k < x_0 + \frac{1}{k}$ , e, como  $\lim (x_0 - \frac{1}{k}) = x_0$  e  $\lim (x_0 + \frac{1}{k}) = x_0$ , então,  $\lim x_k = x_0$ . Devido a continuidade de  $f$  tem-se  $\varepsilon \leq \lim |f(x_k, \alpha_k) - f(x_0, \alpha_k)| = |f(x_0, \alpha) - f(x_0, \alpha)| = 0$ , uma contradição, pois da hipótese  $\varepsilon > 0$ .  $\square$

**Teorema A.2** (Derivação sob o sinal de integral ou Regra de Leibniz). Dado  $U \subset \mathbb{R}^n$ , aberto, seja  $f : U \times [a, b] \rightarrow \mathbb{R}$  uma função com as seguintes propriedades:

1. Para todo  $x \in U$ , a função  $x \mapsto f(x, t)$  é integrável em  $a \leq t \leq b$ .
2. A  $i$ -ésima derivada parcial  $\frac{\partial f}{\partial x_i}(x, t)$  existe para cada  $(x, t) \in U \times [a, b]$  e a função  $\frac{\partial f}{\partial x_i} : U \times [a, b] \rightarrow \mathbb{R}$  assim definida é contínua.

Então a função  $\varphi : U \rightarrow \mathbb{R}$  dada por

$$\varphi(x) = \int_a^b f(x, t) dt,$$

possui  $i$ -ésima derivada parcial em cada ponto  $x \in U$ , sendo

$$\frac{\partial \varphi}{\partial x_i}(x) = \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt.$$

*Demonstração.* Considere

$$\frac{\varphi(x + se_i) - \varphi(x)}{s} = \int_a^b \frac{f(x + se_i, t) - f(x, t)}{s} dt,$$

de onde subtraindo  $\int_a^b \frac{\partial f}{\partial x_i}(x, t) dt$  de ambos os lados, tem-se

$$\frac{\varphi(x + se_i) - \varphi(x)}{s} - \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt = \int_a^b \left[ \frac{f(x + se_i, t) - f(x, t)}{s} - \frac{\partial f}{\partial x_i}(x, t) \right] dt \quad (\text{A.1})$$

Pelo Teorema do Valor Médio para funções reais, existe  $\theta \in [0, 1]$  de modo que

$$\frac{f(x + se_i, t) - f(x, t)}{s} = \frac{\partial f}{\partial x_i}(x + \theta se_i, t),$$

onde  $\theta \in [0, 1]$  garante que  $\theta s$  esteja entre  $]0, s[$ , satisfazendo as condições do Teorema do Valor Médio. Assim, de (A.1) tem-se

$$\frac{\varphi(x + se_i) - \varphi(x)}{s} - \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt = \int_a^b \left[ \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right] dt. \quad (\text{A.2})$$

Como  $\frac{\partial f}{\partial x_i}$  é contínua e  $[a, b]$  é compacto, pelo Teorema A.1, para todo  $\varepsilon > 0$ , existe um  $\delta > 0$ , de modo que

$$|s| < \delta \implies \left| \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right| < \frac{\varepsilon}{b-a} \quad (\text{A.3})$$

seja qual for  $t \in [a, b]$ .

Usando o fato de que  $\left| \int_a^b f(x) dx \right| \leq \int_a^b |f(x)| dx$ , obtêm-se

$$\left| \int_a^b \left[ \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right] dt \right| \leq \int_a^b \left| \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right| dt,$$

e, utilizando (A.3), tem-se a inequação

$$\begin{aligned} \left| \int_a^b \left[ \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right] dt \right| &< \int_a^b \frac{\varepsilon}{b-a} dt \\ \left| \int_a^b \left[ \frac{\partial f}{\partial x_i}(x + \theta se_i, t) - \frac{\partial f}{\partial x_i}(x, t) \right] dt \right| &< \varepsilon. \end{aligned} \quad (\text{A.4})$$

De (A.2) e (A.4), verifica-se que

$$|s| < \delta \implies \left| \frac{\varphi(x + se_i) - \varphi(x)}{s} - \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt \right| < \varepsilon,$$

que é, a definição formal do limite

$$\lim_{s \rightarrow 0} \frac{\varphi(x + se_i) - \varphi(x)}{s} = \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt,$$

ou seja,

$$\frac{\partial \varphi}{\partial x_i}(x) = \int_a^b \frac{\partial f}{\partial x_i}(x, t) dt.$$

□



**Definição A.2.** *Seja o aberto  $U \subset \mathbb{R}^n$ . Uma função  $f : U \rightarrow \mathbb{R}$  é dita de classe  $C^0$  quando é contínua. Diz-se, também, que  $f : U \rightarrow \mathbb{R}$  é de classe  $C^1$  quando existem, em cada ponto  $x \in U$ , as derivadas parciais  $\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x)$  e, as  $n$  funções  $\frac{\partial f}{\partial x_i} : U \rightarrow \mathbb{R}$ , assim definidas, são contínuas. Mais geralmente, uma função  $f : U \rightarrow \mathbb{R}$  é dita de classe  $C^k$ , onde  $k > 0$  é um inteiro, quando ela possuir derivadas parciais em todos os pontos de  $U$  e as funções  $\frac{\partial f}{\partial x_i}, \dots, \frac{\partial f}{\partial x_n} : U \rightarrow \mathbb{R}$  forem de classe  $C^{k-1}$ .*

A Definição A.2 é equivalente a dizer que uma função  $f : U \rightarrow \mathbb{R}$ , onde  $U \subset \mathbb{R}^n$ , é dita de classe  $C^k$  se todas as suas derivadas parciais até a ordem  $k$  existirem e, ainda, forem todas contínuas. Completando,  $f$  será de classe  $C^0$  caso seja contínua.



# APÊNDICE B – ENCONTRANDO A MÁXIMA DEFLEXÃO DA VIGA ESTUDADA

Durante o estudo do método de Rayleigh-Ritz no Exemplo 3.1, no Capítulo 3, faz-se necessário ter valores para comparação. Com esse objetivo, pode-se determinar a máxima deflexão da viga a partir da solução analítica  $v(x)$  do problema, determinada no Exemplo 2.2 da Seção 2.4 do Capítulo 2, a partir de (2.39),

$$v(x) = \frac{q}{24EI}x^4 - \frac{ql}{12EI}x^3 + \frac{ql^3}{24EI}x. \quad (\text{B.1})$$

Derivando (B.1), tem-se

$$v'(x) = \frac{4q}{24EI}x^3 - \frac{3ql}{12EI}x^2 + \frac{ql^3}{24EI}. \quad (\text{B.2})$$

Fazendo  $v'(x) = 0$  é possível determinar os pontos de máximo e mínimo da função, donde

$$\begin{aligned} \frac{4q}{24EI}x^3 - \frac{6ql}{24EI}x^2 + \frac{ql^3}{24EI} &= 0 \\ 4x^3 - 6lx^2 + l^3 &= 0. \end{aligned} \quad (\text{B.3})$$

A equação (B.3) tem como uma de suas soluções o valor  $x_1 = \frac{l}{2}$ , pois

$$4\left(\frac{l}{2}\right)^3 - 6l\left(\frac{l}{2}\right)^2 + l^3 = 0,$$

de onde, realizando uma divisão do polinômio de (B.3) por  $x - \frac{l}{2}$  é possível determinar os outros extremos. Deste modo,

$$4x^3 - 6lx^2 + l^3 = \left(x - \frac{l}{2}\right)(4x^2 - 4lx - 2l^2) = 0,$$

e, portanto, para a determinação dos outros extremos deve-se resolver a equação de segundo grau  $4x^2 - 4lx - 2l^2 = 0$ , que resulta em  $x_2 = \frac{(1 + \sqrt{3})}{2}l$  e  $x_3 = \frac{(1 - \sqrt{3})}{2}l$ .

Fazendo o estudo de sinais de  $v'(x)$  resulta que  $x_1 = \frac{l}{2}$  é ponto de máximo, enquanto que os demais são pontos de mínimo, logo, a máxima deflexão da viga acontece quando  $x = \frac{l}{2}$ .

Calculando o valor da máxima deflexão,

$$v\left(\frac{l}{2}\right) = \frac{q}{24EI} \left(\frac{l}{2}\right)^4 - \frac{ql}{12EI} \left(\frac{l}{2}\right)^3 + \frac{ql^3}{24EI} \left(\frac{l}{2}\right)$$

$$v\left(\frac{l}{2}\right) = \frac{5ql^4}{384EI}. \quad (\text{B.4})$$

# APÊNDICE C – BREVE INTRODUÇÃO À LINGUAGEM *PYTHON*

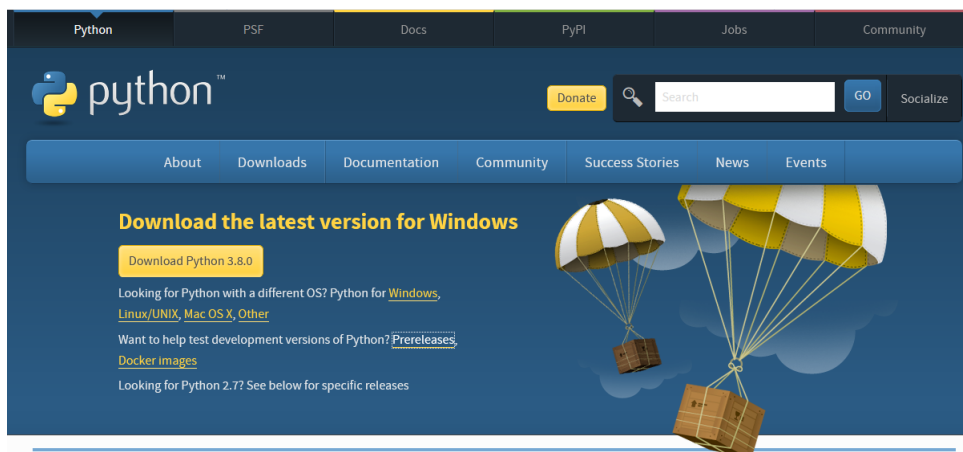
Durante o trabalho, são utilizados trechos de códigos na linguagem *Python*. Para auxiliar a introdução nesse universo, é apresentado aqui, uma breve explicação sobre sua instalação e execução de programas. Na última seção, é apresentado o código utilizado para a resolução de sistemas lineares na linguagem.

## C.1 Instalando

A linguagem de programação *Python* é distribuída gratuitamente e pode ser baixada no seu site<sup>1</sup>. É necessário ressaltar, também, que em distribuições *Linux*, como o *Linux Mint*, a linguagem geralmente já vem instalada por padrão. Devido a isso e, também, ao fato de que usuários de ambientes *Linux* costumam ter conhecimentos avançados sobre instalação de programas, será explicado nessa seção apenas a instalação no sistema operacional *Windows*.

O executável da linguagem *Python* pode ser baixado em seu site, na aba *Downloads*, como pode ser visto na Figura C.1.

Figura C.1 – Página de *download* da linguagem *Python*.



Fonte: Elaborada pelo autor, 2019.

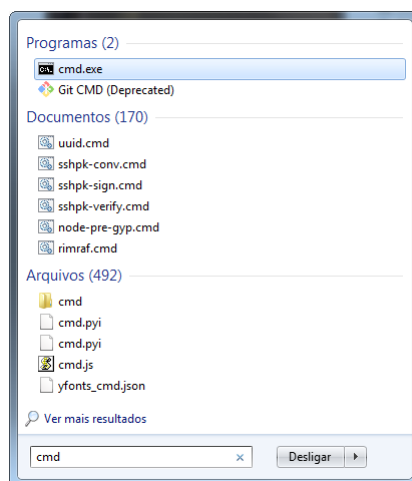
Sua instalação é extremamente simples e rápida. Basta executar o instalador baixado da página da Figura C.1, marcar a opção "*Add Python 3.8 to PATH*", como na Figura C.2 para facilitar a execução de códigos com a linguagem e clicar em *Install Now*, isto é, instalar agora e esperar até que a instalação termine.

<sup>1</sup> <<https://www.python.org/>>.

Figura C.2 – Instalador da linguagem *Python* no *Windows*.

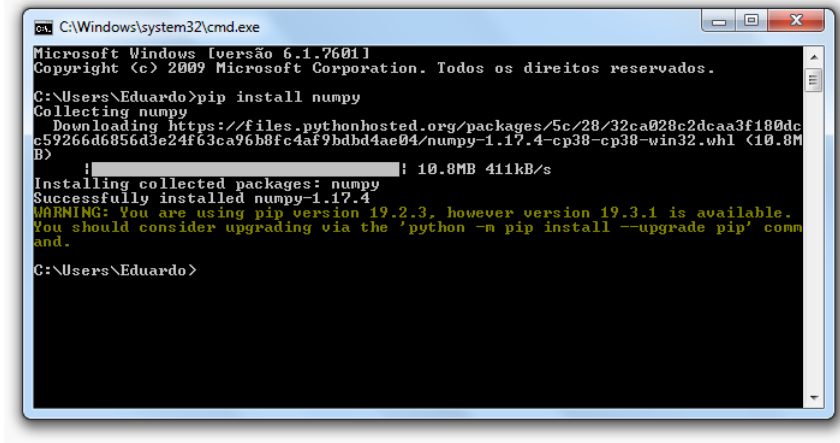
Fonte: Elaborada pelo autor, 2019.

Após a instalação da linguagem, é necessário instalar alguns pacotes utilizados aqui neste trabalho. Para tanto, é preciso executar o *prompt* de comandos do *Windows*, ou terminal em distribuições *Linux*. No *Windows*, basta digitar *cmd* na barra de pesquisa do menu iniciar, como na Figura C.3.

Figura C.3 – Barra de busca com o *prompt* de comandos do *Windows*.

Fonte: Elaborada pelo autor, 2019.

Com o *prompt* de comandos aberto, deve-se executar o código `pip install numpy` para instalar o pacote *numpy*. Após a instalação, uma janela parecida com a Figura C.4 deve ser vista. Esse mesmo processo deve ser repetido com os comandos `pip install sympy` para instalar o pacote *sympy* e `pip install matplotlib` para instalar o pacote *matplotlib*. Esses três pacotes são utilizados durante este trabalho.

Figura C.4 – *Prompt* de comandos após a instalação do pacote *numpy*.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

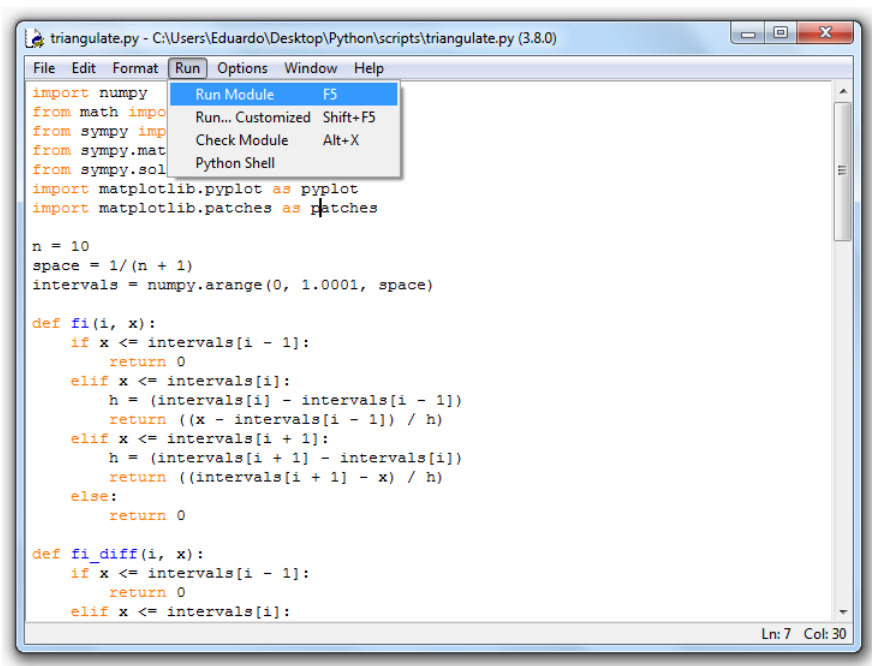
C:\Users\Eduardo>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/5c/28/32ca028c2dcaa3f180dc
e59266d6856d3e24f63ca96b8fc4af9bdd4ae04/numpy-1.17.4-cp38-cp38-win32.whl (10.8M
B)
    |#####|: 10.8MB 411kB/s
Installing collected packages: numpy
Successfully installed numpy-1.17.4
WARNING: You are using pip version 19.2.3, however version 19.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

C:\Users\Eduardo>
```

Fonte: Elaborada pelo autor, 2019.

## C.2 Executando programas

Para executar programas, ou *scripts*, na linguagem *Python*, existem diversas formas, sendo a mais convencional delas executar via o *prompt* de comandos do *Windows* ou no terminal do *Linux*. No *Windows* pode-se utilizar o ambiente de desenvolvimento que é instalado junto a linguagem, chamado de *IDLE*. Ao abri-lo, deve-se ir no menu *File* e, então, em *New File* para criar um novo arquivo. O código deve ser escrito nesse novo arquivo e salvo. Para executar este código basta, então, ir no menu *Run* e, depois, em *Run Module*, como na Figura C.5.

Figura C.5 – Janela do *IDLE* com o menu *Run* aberto.

```
triangulate.py - C:\Users\Eduardo\Desktop\Python\scripts\triangulate.py (3.8.0)
File Edit Format Run Options Window Help
import numpy
from math import pi
from sympy import sin, cos
from sympy.matrices import Matrix
from sympy.solvers import solve

import matplotlib.pyplot as pyplot
import matplotlib.patches as patches

n = 10
space = 1/(n + 1)
intervals = numpy.arange(0, 1.0001, space)

def fi(i, x):
    if x <= intervals[i - 1]:
        return 0
    elif x <= intervals[i]:
        h = (intervals[i] - intervals[i - 1])
        return ((x - intervals[i - 1]) / h)
    elif x <= intervals[i + 1]:
        h = (intervals[i + 1] - intervals[i])
        return ((intervals[i + 1] - x) / h)
    else:
        return 0

def fi_diff(i, x):
    if x <= intervals[i - 1]:
        return 0
    elif x <= intervals[i]:
```

Fonte: Elaborada pelo autor, 2019.

A segunda forma, indicada para usuários que detenham um nível de conhecimento mais alto sobre sistemas operacionais, é executar, no *prompt* de comandos ou no terminal a linha `python arquivo.py`, onde `arquivo.py` representa o caminho para o arquivo no sistema. Por exemplo, como os códigos deste trabalho foram desenvolvidos em um ambiente *Linux*, todos os códigos foram executados deste modo.

### C.3 Principais operações

Para facilitar o entendimento dos códigos ao longo do trabalho, é apresentada uma lista com algumas das principais operações aritméticas na linguagem *Python* na Tabela 1.

Tabela 1 – Principais operações na linguagem *Python*

Operação	Descrição
$x + y$	Soma de $x$ e $y$
$x - y$	Subtração de $x$ por $y$
$x * y$	Produto de $x$ por $y$
$x / y$	Divisão de $x$ por $y$
$x // y$	Parte inteira da divisão de $x$ por $y$
$x \% y$	O resto da divisão de $x$ por $y$
$x ** y$	$x$ elevado a potência $y$

Fonte: Elaborada pelo autor, 2019.

### C.4 Resolução de sistemas lineares

O sistema linear com matriz aumentada (3.21) pode ser facilmente resolvido utilizando métodos trabalhados em Álgebra Linear, como, por exemplo, o método de Gauss-Jordan, de forma manual. Outra forma de se resolver o sistema é fazendo o uso da linguagem de programação *Python*, através do comando `linsolve` que utiliza o método de Gauss-Jordan.

Deste modo, seja, o sistema com matriz aumentada (3.21)

$$\left( \begin{array}{ccc|c} 2 & 3l & 4l^2 & C \\ 1 & 2l & 3l^2 & \frac{1}{2}C \\ 8 & 18l & \frac{144}{5}l^2 & D \end{array} \right), \quad (\text{C.1})$$

onde

$$C = -\frac{ql^2}{12EI}, \quad (\text{C.2})$$

e

$$D = -\frac{3ql^2}{10EI}. \quad (\text{C.3})$$

Para a resolução do sistema usando *Python* é necessário, primeiro, importar alguns componentes da biblioteca SymPy:



```
from sympy import symbols, Matrix
from sympy.solvers import linsolve
```

Serão importados os componentes *symbols* para permitir a definição de símbolos matemáticos, *Matrix* para a manipulação de matrizes e, finalmente, *linsolve* para a resolução de sistemas lineares.

Para a definição do sistema, utilizaremos os mesmos símbolos, *l*, *C*, e *D*, que são definidos, dentro da linguagem, utilizando o comando *symbols* importado:

```
l, C, D = symbols('l C D')
```

A matriz aumentada é definida utilizando o componente *Matrix* importado. Cada uma de suas linhas é escrita como um Array contendo os elementos das colunas. A matriz é escrita, então, como um Array de linhas.

```
A = Matrix([
    [2, 3 * l, 4 * l**2, C],
    [1, 2 * l, 3 * l**2, C/2],
    [8, 18 * l, 144 / 5 * l**2, D]
])
```

A resolução do sistema é feita pelo método *linsolve* já importado. O método *linsolve* da linguagem *Python* utiliza o método de Gauss-Jordan para a resolução de sistemas lineares (SYMPY DEVELOPMENT TEAM, 2019). Primeiro, definimos três novos símbolos,  $a_1$ ,  $a_2$  e  $a_3$  que são as incógnitas do sistema e então, executamos o método passando como argumento a matriz *A* definida e os símbolos.

```
a2, a3, a4 = symbols("a2 a3 a4")
sol, = linsolve(A, [a2, a3, a4])
print(sol)
```

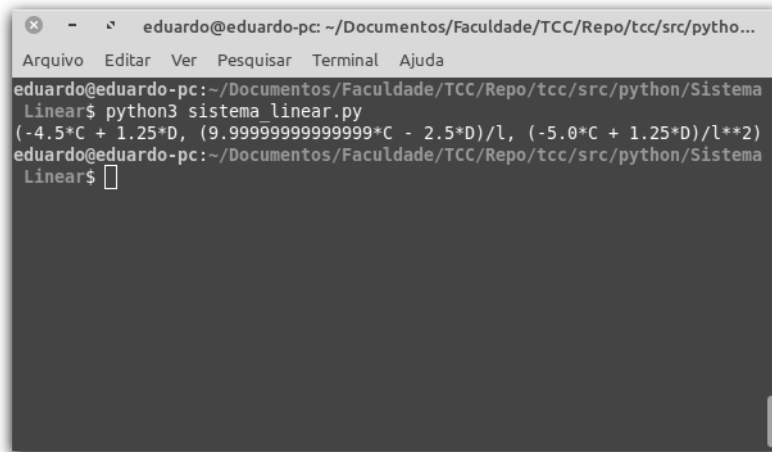
O comando *print(sol)* exibe a solução do sistema linear, como um Array (Veja Figura C.6). O primeiro valor do Array corresponde a incógnita  $a_2$ , o segundo valor a  $a_3$  e o último valor a  $a_4$ . Assim,  $a_2 = -4.5C + 1.25D$ ,  $a_3 = \frac{1}{l}(10C - 2.5D)$  e  $a_4 = \frac{1}{l^2}(-5C + 1.25D)$ . Pode-se converter os números decimais para frações obtendo

$$a_2 = \frac{5}{4}D - \frac{9}{2}C, \quad (\text{C.4})$$

$$a_3 = \frac{1}{l} \left( 10C - \frac{5}{2}D \right), \quad (\text{C.5})$$

$$a_4 = \frac{1}{l^2} \left( \frac{5}{4}D - 5C \right). \quad (\text{C.6})$$

Figura C.6 – Execução do código para a resolução do sistema linear.



```

eduardo@eduardo-pc: ~/Documentos/Faculdade/TCC/Repo/tcc/src/pytho...
Arquivo Editar Ver Pesquisar Terminal Ajuda
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/Sistema
Linear$ python3 sistema_linear.py
(-4.5*C + 1.25*D, (9.999999999999999*C - 2.5*D)/l, (-5.0*C + 1.25*D)/l**2)
eduardo@eduardo-pc:~/Documentos/Faculdade/TCC/Repo/tcc/src/python/Sistema
Linear$

```

Fonte: Elaborada pelo autor, 2019.

Substituindo  $C$  e  $D$ , de (C.2) e (C.3), em (C.4), (C.5) e (C.6) e simplificando os valores obtém-se a solução do sistema

$$\begin{aligned}
 a_2 &= 0, \\
 a_3 &= -\frac{ql}{12EI}, \\
 a_4 &= \frac{q}{24EI}.
 \end{aligned}$$

O código completo utilizado para a resolução do sistema linear com matriz aumentada (C.1) é apresentado em Código C.1.

#### Código C.1 – Resolução de sistemas lineares em *Python*

```

1 from sympy import symbols, Matrix
2 from sympy.solvers import linsolve
3
4 l, C, D = symbols('l C D')
5
6 A = Matrix([
7     [2, 3 * l, 4 * l**2, C],
8     [1, 2 * l, 3 * l**2, C/2],
9     [8, 18 * l, 144 / 5 * l**2, D]
10 ])
11
12 a2, a3, a4 = symbols("a2 a3 a4")
13 sol, = linsolve(A, [a2, a3, a4])
14
15 print(sol)

```

Fonte: Elaborado pelo autor, 2019.

# ÍNDICE REMISSIVO

## C

Cálculo Variacional, 24, 29  
Condições de Contorno de Dirichlet, 33  
Condições de Contorno de Neumann, 33  
Condições de Contorno Essenciais, 33  
Condições de Contorno Naturais, 33, 36  
Conjunto Compacto, 69

## D

Derivação Sob o Sinal de Integral, 69

## E

Equação de Euler-Lagrange, 32, 36

## F

Função Admissível, 30, 34  
Função Aproximadora, 41  
Função de Classe  $C^k$ , 71  
Função de Forma, 41  
Função Triangular, 49

## L

Lema de Lagrange, 29

## M

Método de Gauss-Jordan, 51, 78  
Método de Rayleigh, 25  
Método de Rayleigh-Ritz, 24, 41, 48, 53  
Método de Ritz, 25  
Matplotlib, 27

## N

NumPy, 27

## P

Parâmetros de Deslocamento, 41  
Problema da braquistócrona, 24  
Python, 27, 53, 57, 75

## R

Regra de Leibniz, 69

## S

SciPy, 27  
SymPy, 27