UNIVERSIDADE
BEIRA INTERIOR

Trabalho 01- Análise de Vigas em Flexão

José Miguel Correia Barros, M15321

Engenharia Aeronáutica
(2º ciclo de estudos)

Docente: Prof. Doutor Thiago Assis Dutra

5 de Outubro de 2025

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Descrição do Problema

Este relatório tem como finalidade apresentar a proposta de resolução do Trabalho 1 da disciplina de Placas e Cascas. O foco do estudo recai na investigação da resposta de vigas e placas quando submetidas a diversas formas de carregamento, um tema de grande relevância na Engenharia Aeronáutica. Compreender o comportamento dessas estruturas frente às cargas aplicadas é essencial para garantir tanto a segurança, quanto a estabilidade de componentes estruturais, especialmente em projetos aeronáuticos.

1.1 Tarefa 1

No primeiro exercício, considera-se uma viga de secção transversal retangular, com dimensões $b \times h$, constituída por um material isotrópico e sujeita a uma carga uniformemente distribuída q. Tal como ilustrado na Figura 1.1, a área da secção transversal A, o módulo de elasticidade E, o momento de inércia I e a rigidez da mola k mantêm-se constantes ao longo da viga.

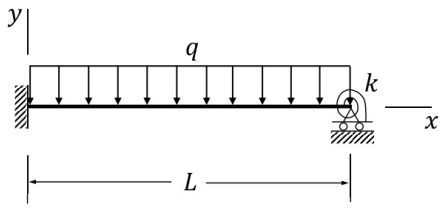


Figura 1.1: Viga de secção retangular sujeita a carga uniformemente distribuída.

Na alínea *(a)*, pretende-se determinar, recorrendo à teoria de Euler-Bernoulli, a equação da linha elástica da viga, o momento fletor no ponto médio e a força de reação vertical no apoio direito. Na alínea *(b)*, é solicitado calcular as deflexões e inclinações ao longo de toda a viga, desde $x = 0$ até $x = L$, e apresentar os resultados graficamente através de um software à escolha. Para a resolução do problema, optei por utilizar a linguagem *Python*. Os dados necessários para os cálculos encontram-se disponíveis na Tabela 1.1.

Tabela 1.1: Dados da Tarefa 1.

Caso	b [mm]	h [mm]	L [mm]	E [GPa]	q [N/m]	k [Nm/rad]
6	10	10	800	70	250	300

1.2 Tarefa 2

No segundo exercício, considera-se uma viga simplesmente apoiada em ambas as extremidades, submetida a um carregamento arbitrário que consiste numa carga distribuída q e em n cargas concentradas, conforme ilustrado na Figura 1.2.

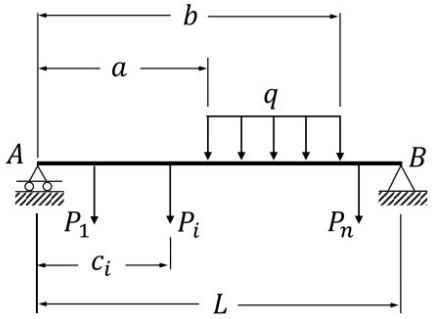


Figura 1.2: Viga simplesmente apoiada com carregamento distribuído e cargas concentradas.

Na alínea *(a)*, solicita-se o desenvolvimento de um programa informático que determine o diagrama do esforço transverso e do momento fletor ao longo da viga, em função da posição x. Este programa deve ser elaborado de forma flexível, permitindo o tratamento de um número indefinido de cargas concentradas, ou seja, não poderá estar limitado a um caso específico. Na alínea *(b)*, pretende-se que os resultados obtidos na primeira parte sejam representados graficamente, recorrendo a incrementos de discretização no eixo da viga de $\Delta x \leq \frac{L}{100}$. Para esse efeito, são disponibilizados os valores apresentados na Tabela 1.2.

Tabela 1.2: Dados da Tarefa 2.

Caso	a [m]	b [m]	L [m]	q [kN/m]	P_1 [kN]	c_1 [m]	P_2 [kN]	c_2 [m]	P_3 [kN]	c_3 [m]
6	4	8	8	25	12	1	26	4	37	5

Capítulo 2

Metodologia de Solução

2.1 Tarefa 1

A teoria de Euler–Bernoulli constitui um modelo clássico para a análise do comportamento estrutural de vigas e barras quando estas são solicitadas por diferentes tipos de carregamentos. No domínio da engenharia, esta formulação é amplamente empregue para estimar a deformação (deflexão) e as tensões que surgem ao longo da viga devido à ação das cargas aplicadas. O modelo assenta em hipóteses fundamentais que simplificam a realidade física, permitindo a obtenção de resultados práticos com boa aproximação:

- Assume-se que o comprimento da viga é muito superior às suas restantes dimensões;
- Os efeitos provocados pelo esforço transverso (corte) não são considerados;
- As secções transversais mantêm-se planas e ortogonais ao eixo neutro mesmo após a deformação;
- O material é suposto ser homogéneo e obedecer ao comportamento elástico linear descrito pela lei de Hooke.

Para determinar a equação da linha elástica, é necessário começar pela definição das condições de fronteira do problema. No ponto $x = 0$, a viga encontra-se encastrada, o que implica ausência de deslocamento vertical e de rotação. Já em $x = L$, está aplicado um apoio, que impede qualquer deslocamento vertical nesse ponto. Para que a estrutura se mantenha em equilíbrio, o somatório dos momentos em torno do apoio deve ser nulo, de modo que o momento gerado pela carga distribuída seja compensado pelo momento resultante da ação da mola. Desta maneira podemos entao defenir as nossas condições de fronteira como:

$$\begin{cases} v(0) = 0, \quad v'(0) = 0 \\ v(L) = 0, \quad M(L) = k \cdot v'(L) \end{cases}$$

A equação 2.1, que descreve o comportamento da viga, serve de base para a determinação da linha elástica.

$$EI \cdot v^{(iv)}(x) = -q(x) \quad (2.1)$$

A partir da equação (2.1), procede-se à sua integração sucessiva até alcançar a equação (2.5), que descreve a deflexão vertical e, por conseguinte, define a equação da linha elástica. No percurso, a equação (2.2), quando multiplicada pelas propriedades da viga (EI), representa o esforço transverso (corte). De forma análoga, a equação (2.3), também multiplicada por EI, traduz o momento fletor. Seguidamente, a equação (2.4) corresponde à inclinação da linha elástica. Assim, cada etapa conduz de modo encadeado até à formulação final da equação da linha elástica.

$$v'''(x) = \frac{-qx}{EI} + C_1 \quad \textbf{(2.2)}$$

$$v''(x) = \frac{-qx^2}{2EI} + C_1 x + C_2 \quad \textbf{(2.3)}$$

$$v'(x) = \frac{-qx^3}{6EI} + \frac{C_1 x^2}{2} + C_2 x + C_3 \quad \textbf{(2.4)}$$

$$v(x) = \frac{-qx^4}{24EI} + \frac{C_1 x^3}{6} + \frac{C_2 x^2}{2} + C_3 x + C_4 \quad \textbf{(2.5)}$$

De seguida ao substituir pelas condições de fronteira vamos obter os valores das constantes C_1, C_2, C_3 e C_4.

$$C_1 = \frac{L^2 q\ (-6EI + Lk)}{24EI\,(4EI - Lk)} \quad \textbf{(2.6)}$$

$$C_2 = \frac{L\,q\,(5EI - Lk)}{12EI(4EI - Lk)} \quad \textbf{(2.7)}$$

$$C_3 = 0 \quad \textbf{(2.8)}$$

$$C_4 = 0 \quad \textbf{(2.9)}$$

Então, a equação da linha elástica é defenida pela expressão 2.10, onde C_1 e C_2 sao 2.6 e 2.7, respetivamente.

$$v(x) = \frac{-qx^4}{24EI} + \frac{C_1 x^3}{6} + \frac{C_2 x^2}{2} \quad \textbf{(2.10)}$$

Para determinar o momento fletor no ponto médio da viga, isto é, quando $x = L/2$, utilizamos novamente as condições de fronteira. De recordar que a relação entre o momento e a derivada da deflexão é dada por $M(L) = k \cdot v'(L)$, sendo que, em particular, vale também $k \cdot v'(L) = -EI \cdot v''(x)$. Com base nesta formulação, chega-se então à expressão do

momento fletor no centro da viga, apresentada na equação (2.11).

$$M\left(\frac{L}{2}\right) = EI \cdot v''\left(\frac{L}{2}\right) \quad (2.11)$$

Finalmente, é necessário determinar a reação vertical no apoio situado à direita. Esse valor obtém-se recorrendo à expressão indicada na equação (2.12).

$$-EI \cdot v'''(L) = -EI\left(\frac{qL}{EI} + C_1\right) \quad (2.12)$$

2.2 Tarefa 2

Na segunda tarefa, considera-se uma viga simplesmente apoiada em ambas as extremidades, sujeita simultaneamente a uma carga distribuída q e a n cargas concentradas aplicadas em posições arbitrárias.

Partindo do princípio de que a estrutura se encontra em equilíbrio estático, impõe-se que $\sum F_x = \sum F_y = 0$ e $\sum M_A = 0$. Com base nestas condições fundamentais, foram determinadas as reações de apoio em A e em B, tal como apresentado nas equações (2.13) e (2.14).

$$R_A = -R_B + \sum_{i=1}^{n} P_i + q(b-a) \quad (2.13)$$

$$R_B = \frac{\sum_{i=1}^{n} P_i \cdot c_i + q(b-a)\left(a + \frac{b-a}{2}\right)}{L} \quad (2.14)$$

Após a determinação das reações, o passo seguinte consiste na construção dos diagramas de esforço transverso e de momento fletor. Estes são obtidos a partir do equilíbrio de cada secção da viga, sendo necessário contabilizar a contribuição das forças pontuais e do carregamento distribuído em cada intervalo definido pelas posições c_i.

De modo a garantir a generalidade da formulação, o método implementado permite a introdução de um número indefinido de cargas concentradas, não ficando limitado a um caso específico. Esta abordagem assegura a aplicabilidade do programa a diferentes cenários de carregamento.

Adicionalmente, para representar graficamente os resultados, o eixo da viga foi discretizado em incrementos de $\Delta x \leq \frac{L}{100}$, conforme indicado no enunciado, garantindo assim uma resolução adequada e a correta identificação de descontinuidades associadas às cargas pontuais.

Tal como na Tarefa 1, adotou-se o modelo clássico de Euler–Bernoulli para a viga. No en-

tanto, nesta etapa as reações foram obtidas apenas pelas condições de equilíbrio estático.

Para a implementação computacional foram ainda consideradas restrições de iteração e validação de dados, de forma a garantir que o algoritmo percorre corretamente todas as cargas aplicadas e apenas aceita valores admissíveis como variáveis de entrada. Estas condições asseguram que o programa não só respeita as exigências físicas do problema, como também mantém a robustez da introdução de dados e a generalidade do método de cálculo.

Por fim, avaliou-se a influência da posição das cargas concentradas ao longo do eixo x, uma vez que diferentes pontos de aplicação resultam em distribuições distintas do esforço transverso e do momento fletor, modificando assim a resposta global da viga.

Capítulo 3

Implementação do código

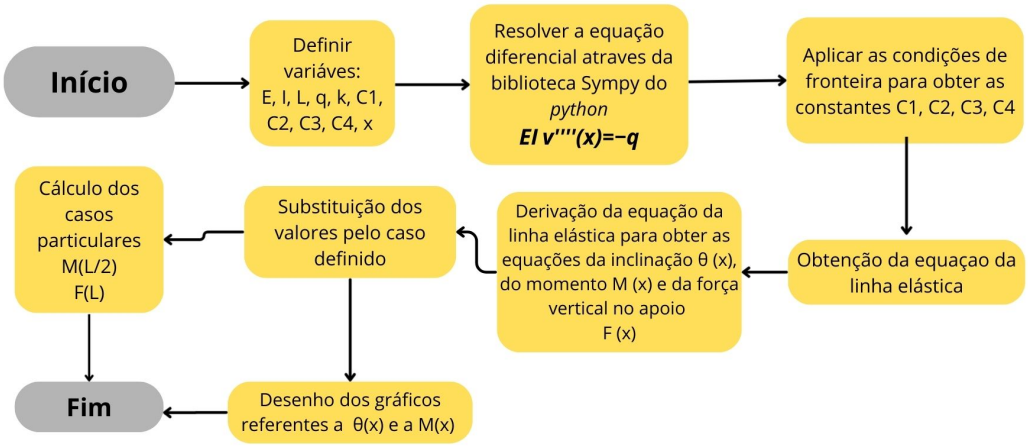


Figura 3.1: Fluxograma referente à Tarefa 1

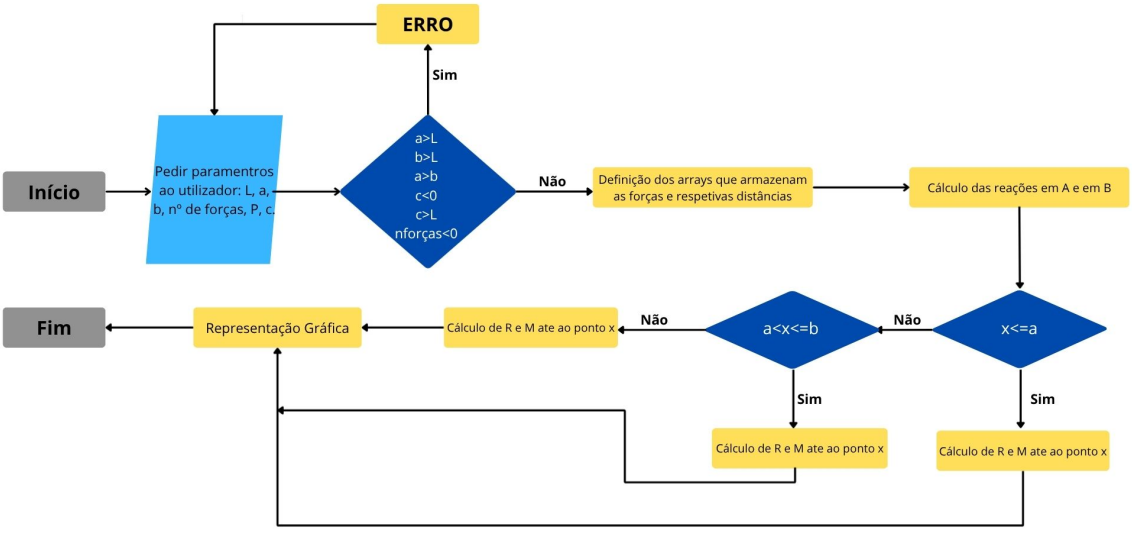


Figura 3.2: Fluxograma referente à Tarefa 2

Na Tarefa 1, começamos por recolher os dados essenciais à formulação do problema. De seguida, determinamos as constantes envolvidas, o que possibilita a construção da equação da linha elástica. A partir desta, foi possível calcular as reações e os momentos, permitindo obter tanto a expressão da deflexão em função da variável x, como a expressão da inclinação. Com estas relações estabelecidas, torna-se possível representar graficamente os resultados, conforme solicitado.

Já na Tarefa 2, o primeiro passo consistiu em solicitar ao utilizador a introdução das variáveis de entrada necessárias para a resolução. Posteriormente, implementou-se um ciclo que recorreu às instruções `for` e `while`, assegurando a aplicação correta das condições descritas no Capítulo 2. Com esta abordagem, foi possível calcular os somatórios das forças e dos momentos, considerando tanto a quantidade de cargas aplicadas como a posição onde atuam. Os valores obtidos foram organizados em vetores, os quais serviram de base à geração dos gráficos requeridos.

Capítulo 4

Resultados e Discussões

4.1 Tarefa 1

Após a implementação do código, verificou-se que, na alínea *(a)*, foi determinada a equação da linha elástica, conforme apresentada na equação (2.10).

Seguidamente, calculou-se o momento fletor no ponto $L/2$, obtendo-se

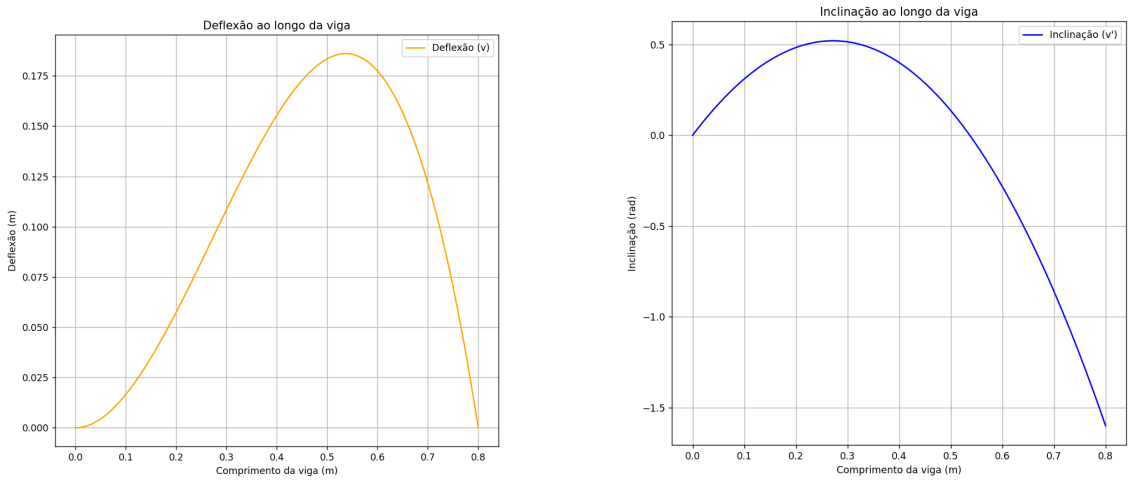
$$M(L/2) = 110.00\,\text{Nm}, \quad (4.1)$$

e, posteriormente, a força de reação vertical no apoio direito foi encontrada como

$$v'''(L) = 975.00\,\text{N}, \quad (4.2)$$

conforme indicado nas equações acima.

Utilizando a linguagem de programação *Python* e os parâmetros fornecidos para o caso 6, foram elaborados os gráficos apresentados na Figura 4.1, nos quais se representam as curvas de deflexão e de inclinação ao longo do comprimento da viga.



(a) Deflexões ao longo da viga.

(b) Inclinações ao longo da viga.

Figura 4.1: Gráficos referentes às deflexões e inclinações ao longo do comprimento da viga.

A observação do gráfico de deflexão permite concluir que, em conformidade com as condições

de fronteira previamente estabelecidas, a curva apresenta uma forma assimétrica. Nota-se que a viga atinge o valor máximo de deslocamento no ponto de mínimo da função, localizado em $x = 0.537\,\text{m}$, onde a deflexão é de aproximadamente $0.186\,\text{m}$. Já o gráfico correspondente às inclinações confirma a análise, pois, sendo obtido como derivada da curva de deflexão, apresenta uma raiz exatamente em $x = 0.537\,\text{m}$.

4.2 Tarefa 2

Posteriormente, com base nos resultados do Caso 6, foram obtidos os gráficos apresentados nas Figuras 4.2 e 4.3, correspondentes, respetivamente, às forças de reação e ao momento fletor em função de x, conforme requerido.

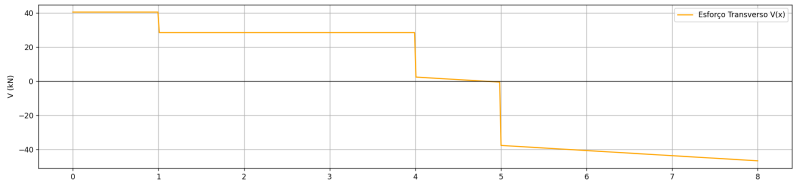


Figura 4.2: Diagrama das forças de reação em função da posição x.

O diagrama apresentado na Figura 4.2 mostra que, no troço inicial $[0, 1]\,\text{m}$, o esforço transverso mantém-se aproximadamente constante em $62.375\,\text{kN}$, devido à reacção de apoio R_A e à ausência de carregamentos localizados nessa zona. Em $x = 1\,\text{m}$ verifica-se um salto descendente provocado pela carga pontual P_1, reduzindo o esforço para cerca de $50.375\,\text{kN}$.

Entre 1 e $4\,\text{m}$, o esforço transverso permanece constante. Em $x = 4\,\text{m}$ ocorre um novo salto descendente devido à aplicação da carga P_2, resultando em $+24.17\,\text{kN}$. No intervalo $[4, 5]\,\text{m}$ o valor mantém-se inalterado até à aplicação de P_3 em $x = 5\,\text{m}$, onde se observa uma queda abrupta para cerca de $-37.675\,\text{kN}$.

A partir desse ponto, no intervalo $[5, 8]\,\text{m}$, o diagrama apresenta um decréscimo linear provocado pela acção da carga distribuída uniforme q, atingindo aproximadamente -112.63 kN junto ao apoio B. Em $x = 8\,\text{m}$, a reacção R_B equilibra o sistema, anulando o esforço transverso.

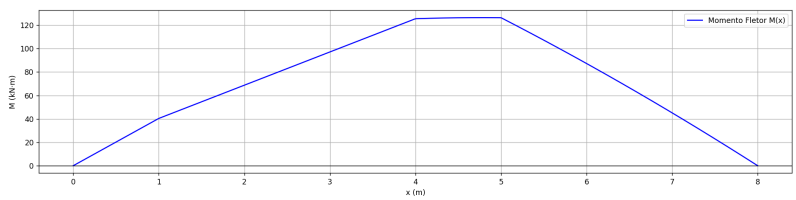


Figura 4.3: Diagrama do momento fletor ao longo da viga.

Relativamente à Figura 4.3, verifica-se que o momento flector anula-se nas extremidades da viga, em $x = 0$ e $x = 8\,\text{m}$, tal como seria de esperar para apoios simples. O momento cresce de forma praticamente linear até cerca de $213.4\,\text{kN·m}$ em $x = 4\,\text{m}$, atingindo o valor máximo em $x \approx 5,0\,\text{m}$, com $M_{\text{max}} \approx 225\,\text{kN·m}$.

Após esta posição, a acção da carga distribuída provoca uma redução progressiva do momento flector, que decresce até se anular novamente em $x = 8\,\text{m}$. Este comportamento confirma a coerência entre os diagramas de esforço transverso e de momento flector, uma vez que o ponto de máximo coincide com a região de inversão de sinal do esforço cortante.

Capítulo 5

Conclusão

No presente trabalho procedeu-se à análise de dois problemas estruturais envolvendo vigas, com base na teoria de Euler–Bernoulli e no recurso a métodos numéricos, com o objetivo de determinar reações de apoio, momentos fletores, deflexões e rotações ao longo do elemento.

Na primeira tarefa, a partir das condições de fronteira impostas e da equação diferencial governante do comportamento da viga, foi possível deduzir a expressão da linha elástica, calcular o momento fletor na secção central e a reação vertical no apoio direito. A implementação computacional em Python permitiu, adicionalmente, a representação gráfica da deflexão e da inclinação, destacando a localização do ponto correspondente à máxima flecha.

Na segunda tarefa, desenvolveu-se um programa com caráter genérico, capaz de calcular os esforços transversos e os momentos fletores resultantes da aplicação de um número arbitrário de cargas concentradas em diferentes posições ao longo da viga. A formulação proposta possibilitou a determinação precisa das reações e dos momentos em função da coordenada x, complementada por representações gráficas que descrevem a evolução dos esforços ao longo da viga, evidenciando as descontinuidades geradas nos pontos de aplicação das cargas.

Em síntese, o relatório apresentou uma análise detalhada do comportamento estrutural de vigas sujeitas a distintos esquemas de carregamento, confirmando a consistência dos resultados obtidos e reforçando a relevância da utilização integrada de modelos teóricos e ferramentas numéricas no estudo e dimensionamento de sistemas estruturais em engenharia.

Bibliografia

Gamboa, Pedro. Slides das aulas. 2025/2026. Placas e Cascas. Universidade da Beira Interior, Covilhã.

Apêndice A

Implementação do código

A.1 Tarefa 1

```
1  #Defenir todas as constantes que temos
2  #Resolver a equação diferencial EI*v''''(x) = -q
3  #Integrar 4 vezes e obter c1, c2, c3 e c4
4  #Defenir as condições de fronteira
5  #Substituir e resolver o sistema para calcular os valores das constantes c1, c2
       , c3 e c4 através das condições de fronteira
6  #Obter a equação da linha elástica v(x)
7  #Obter a equação do momento fletor no centro da viga (x=L/2)
8  #Obter a equação da força vertical no ponto mais a direita (x=L)
9  #Substituir pelos valores numerios as constantes defenidas na linha 1
10 #Calcular a deflexão (ja temos linha 6)
11 #calcular a inclinação= theta(x)= v'(x)
12 #Desenhar graficamente os dados da linha 10 e 11 ao longo de 0<x<L
13
14 #Bibliotecas necessárias para a resoluçao do exercicio
15 import sympy as sp #calculo diferencial
16 import numpy as np #transformar os dados e analisar todos os pontos para ser
       possível desenhar os gráficos
17 import matplotlib.pyplot as plt #Representar gráficamente as o declive e a
       deflexão ao longo da viga
18
19 #Defenir as variáveis
20 x = sp.symbols ('x')
21 E, I, L, q, k=sp.symbols('E I L q k')
22 C1, C2, C3, C4 = sp.symbols('C1 C2 C3 C4')
23 #Defenir a equaçao diferencial
24 v= sp.Function('v')(x)
25 EqDif= sp.Eq(E*I*sp.diff(v,(x,4)),-q)
26
27 #Resolver a EqDif
28 v_solv= sp.dsolve(EqDif)
29 V_geral= v_solv.rhs #Obter a expressão da linha elástica
30 print ("\n Equação da linha elástica (com constantes)")
31 sp.pprint(V_geral)
32
```

```
33 #Vamos defenir as condições de fronteira (4 equaçoes de fronteira, 2 para cada
       extermidade)
34 #Devido ao encastramento
35  #v(x)=0 --- vai anular c4- condição 1
36  #v'(x)=0---- vai anular c3- condição 2
37 #Devido ao apoio da mola
38 #v(L)=0--- vamos obter uma equação dependente de c1 e c2-condição 3
39 #Equação do momento
40 #Para a eq4 -> M_L+M_mola=0 <=> M_L= -k*v'(L) <=> EI*v''(L)= K*v'(L)<=> EI*v''(
       L)-K*v'(L)=0- condição 4
41 cond1 = sp.Eq(V_geral.subs(x,0), 0)
42 cond2 = sp.Eq(sp.diff(V_geral,x).subs(x,0), 0)
43 cond3 = sp.Eq(V_geral.subs(x,L), 0)
44 cond4 = sp.Eq(E*I*sp.diff(V_geral,(x,2)).subs(x,L) - k*sp.diff(V_geral,x).subs(
       x,L), 0)
45
46 #Resolver sistema para c1,c2,c3 e c4
47 sol_const = sp.solve([cond1, cond2, cond3, cond4], (C1, C2, C3, C4))
48 print("\nConstantes de integração encontradas:")
49 print(sol_const)
50
51 #Substituir na equação final da linha elástica
52 v_final = V_geral.subs(sol_const)
53
54 print("\nEquação da linha elástica v(x):")
55 sp.pprint(sp.simplify(v_final))
56
57 #Inclinação, momento e força vertical
58 theta= sp.diff(v_final,x)
59 M= E*I*sp.diff(v_final,(x,2))
60 F= -E*I*sp.diff (v_final,(x,3))  #equivalente à 3ª derivada da deflexao
61
62 print("\n Inclinaçao :")
63 sp.pprint(sp.simplify(theta))
64 print("\n Momento:")
65 sp.pprint(sp.simplify(M))
66 print("\n Força vertical:")
67 sp.pprint(sp.simplify(F))
68
69 #Introduzir os valores do caso 6
70 b_val= 0.01
71 h_val= 0.01
72 L_val= 0.8
73 E_val= 70e9
74 q_val= 250
75 k_val= 300
76 I_val= (b_val * h_val**3)/12
```

```
77
78 #Substituir os valores nas equações que tivemos a defenir até agora
79 subs_dict = {E: E_val, I: I_val, L: L_val, q: q_val, k: k_val}
80 v_val, theta_val, M_val, F_val = [
81     sp.simplify(expr.subs(subs_dict))
82     for expr in (v_final, theta, M, F)
83 ]
84
85 print("\n A linha elastica para o caso ....., v(x)")
86 sp.pprint(v_val)
87 print("\n A inclinaçao para o caso ....., theta(x)")
88 sp.pprint(theta_val)
89 print("\n A linha elastica para o caso ....., M(x)")
90 sp.pprint(M_val)
91 print("\n A linha elastica para o caso ....., F(x)")
92 sp.pprint(F_val)
93
94 #Transformar os valores de função simbólica para uma função numpy
95 def evaluate_expr(expr, x_vals):
96
97     f = sp.lambdify(x, expr, "numpy")
98     try:
99         y = f(x_vals)
100         y_arr = np.asarray(y)
101         if y_arr.dtype == object:
102             y_arr = np.array([float(sp.N(val)) for val in y_arr], dtype=float)
103         else:
104             y_arr = y_arr.astype(float)
105         return y_arr
106     except Exception:
107         return np.array([float(sp.N(expr.subs(x, float(t)))) for t in x_vals],
              dtype=float)
108
109 #De acordo com o caso que temos ,vamos calcular o momento a meio da viga e a
      força de apoio em L
110 M_meio = float(sp.N(M_val.subs(x, L_val/2)))
111 F_dir = float(sp.N(F_val.subs(x, L_val)))
112
113 #Apresentar os resultados da alínea a)
114
115 print("\nOs resultados da alínea a) são")
116 print(f"Momento a meio da viga M(L/2) [N·m]: {M_meio:.2f}")
117 print(f"Força no apoio direito F(L) [N]: {F_dir:.4f}")
118
119 #Representaçao da alínea b)
120 subs_dict = {E: E_val, I: I_val, L: L_val, q: q_val, k: k_val}
121 v_num = sp.simplify(v_final.subs(subs_dict))
```

```
122 theta_num = sp.simplify(theta.subs(subs_dict))
123
124 # Criar funções numéricas com lambdify
125 v_fun = sp.lambdify(x, v_num, "numpy")
126 theta_fun = sp.lambdify(x, theta_num, "numpy")
127
128 #Incremento dos valores (maior numero=maior precisão)
129 x_valores = np.linspace(0, float(L_val), 100000)
130
131 #Substituir pelos valores de x da função
132 v_valores = evaluate_expr(v_num, x_valores)
133 theta_valores = evaluate_expr(theta_num, x_valores)
134
135 plt.figure(figsize=(12,5))
136 plt.subplot(1, 2, 1)
137 plt.plot(x_valores, v_valores, label="Deflexão (v)", color="orange")
138 plt.title("Deflexão ao longo da viga")
139 plt.xlabel("Comprimento da viga (m)")
140 plt.ylabel("Deflexão (m)")
141 plt.grid(True)
142 plt.legend()
143
144 plt.subplot(1, 2, 2)
145 plt.plot(x_valores, theta_valores, label="Inclinação (v')", color="blue")
146 plt.title("Inclinação ao longo da viga")
147 plt.xlabel("Comprimento da viga (m)")
148 plt.ylabel("Inclinação (rad)")
149 plt.grid(True)
150 plt.legend()
151 plt.tight_layout()
152 plt.show()
```

Listagem A.1: Código em Python para o caso 6

A.2 Tarefa 2

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def resolver_viga():
5     print("=== Análise de Viga com Cargas Distribuídas e Pontuais ===")
6     L = input_positivo("Insira o comprimento da viga [m]: ")
7     a = input_a(L)
8     b = input_b(a, L)
9     q = input_positivo("Insira o valor da carga distribuída q [kN/m]: ")
```

```
    nforcas = input_numero_forcas()

    # Cargas pontuais
    cargas_pontuais = []
    for i in range(1, nforcas + 1):
        P = input_positivo(f"Insira a intensidade da carga P_{i} [kN]: ")
        c = input_distancia_forca(L, i)
        cargas_pontuais.append((P, c))

    # Carga distribuída equivalente
    Q = q * (b - a)
    xQ = a + (b - a) / 2

    # Reações nos apoios
    Reacao_A = (sum(P * (L - c) for P, c in cargas_pontuais) + Q * (L - xQ)) /
        L
    Reacao_B = (sum(P for P, _ in cargas_pontuais) + Q) - Reacao_A

    print(f"\nReações de apoio: RA = {Reacao_A:.2f} kN, RB = {Reacao_B:.2f} kN"
        )

    # Funções de esforço transverso e momento fletor
    def esforco_transverso(x):
        V = Reacao_A
        for P, c in cargas_pontuais:
            if x >= c:
                V -= P
        if a <= x <= b:
            V -= q * (x - a)
        elif x > b:
            V -= q * (b - a)
        return V

    def momento_fletor(x):
        M = Reacao_A * x
        for P, c in cargas_pontuais:
            if x >= c:
                M -= P * (x - c)
        if a <= x <= b:
            w = q * (x - a)
            centroide = (x - a) / 2
            M -= w * centroide
        elif x > b:
            w = q * (b - a)
            centroide = (b - a) / 2
            M -= w * (x - (a + centroide))
        return M
```

```
55
56      x_vals = np.linspace(0, L, 500)
57      V_vals = [esforco_transverso(x) for x in x_vals]
58      M_vals = [momento_fletor(x) for x in x_vals]
59
60      # Desenahr diagramas
61      plt.figure(figsize=(12, 6))
62      plt.subplot(2, 1, 1)
63      plt.plot(x_vals, V_vals, 'b', label='Esforço Transverso V(x)',color="orange
            ")
64      plt.axhline(0, color='k', linewidth=0.8)
65      plt.ylabel('V (kN)')
66      plt.grid(True)
67      plt.legend()
68      plt.subplot(2, 1, 2)
69      plt.plot(x_vals, M_vals, 'r', label='Momento Fletor M(x)', color="blue")
70      plt.axhline(0, color='k', linewidth=0.8)
71      plt.xlabel('x (m)')
72      plt.ylabel('M (kN·m)')
73      plt.grid(True)
74      plt.legend()
75      plt.tight_layout()
76      plt.show()
77
78  #Introduzir valores positivos
79  def input_positivo(mensagem):
80      valor = float(input(mensagem))
81      while valor <= 0:
82          print("Valor inválido! Insira um valor positivo.")
83          valor = float(input(mensagem))
84      return valor
85
86  #Introduzir um valor valido entre 0 e L
87  def input_a(L):
88      a = float(input("Insira o valor de a em metros (>= 0 e <= L): "))
89      while a < 0 or a > L:
90          print("Valor inválido! a deve ser >= 0 e <= L.")
91          a = float(input("Insira o valor de a em metros (>= 0 e <= L): "))
92      return a
93
94  #Introduzir o valor de b de maneira que seja maior que o valor anteriormente
        defenido (a) e menor que L
95  def input_b(a, L):
96      b = float(input(f"Insira o valor de b em metros (>= {a} e <= L): "))
97      while b < a or b > L:
98          print(f"Valor inválido! b deve ser >= {a} e <= {L}.")
99          b = float(input(f"Insira o valor de b em metros (>= {a} e <= L): "))
```

```
100     return b
101
102 #Introduzir um numero inteiro positivo de forças
103 def input_numero_forcas():
104     nforcas = input("Insira o número de forças: ")
105     while not nforcas.isdigit() or int(nforcas) <= 0:
106         print("Valor inválido! Deve ser um número inteiro positivo.")
107         nforcas = input("Insira o número de forças: ")
108     return int(nforcas)
109
110 #Introduzir o valor da distancia de cada força, com as devidas restriçoes
111 def input_distancia_forca(L, contador):
112     distancia = float(input(f"Qual a distância da carga P_{contador} ao ponto A
            (<= {L}): "))
113     while distancia > L or distancia < 0:
114         print(f"Distância inválida! Deve ser entre 0 e {L}.")
115         distancia = float(input(f"Qual a distância da carga P_{contador} ao
                ponto A (<= {L}): "))
116     return distancia
117
118 if __name__ == "__main__":
119     resolver_viga()
```

Listagem A.2: Código em Python para o caso 6