

#Plotagem dos gráficos da 3 questão, itens A, B, C e D do Trabalho de Mecânica do Sólidos

#Código da questao 3, item A:

#Plote um gráfico da curva de energia de deformação da viga AB com os valores de L, de 0 até 5 m, usando incrementos de 1 m, utilizando o alumínio como material da viga

```
import numpy as np
import matplotlib.pyplot as plt

#Código da questao 3, item A:
# dados
P = 95000 # Peso = 45+50 = 95kN
E_al = 69e9 #Aluminio
I = 1.13e-4 # poderia ser 113e-6 tambem, que foi o valor utilizado nos meus calculos

L = np.array([0, 1, 2, 3, 4, 5])
# formula da energia de deformação U = P²L³/(96EI)
U_al = (P**2 * L**3) / (96 * E_al * I)

print("Energia de Deformacao - Aluminio")
print("-" * 50)
for i in range(len(L)):
    print(f'L = {L[i]} m: U = {U_al[i]:.4f} J')

plt.figure(figsize=(10, 6))
plt.plot(L, U_al, 'bo-', linewidth=2, markersize=8, label='Alumínio')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Energia de Deformação U (J)', fontsize=12)
plt.title('Energia de Deformação vs Comprimento - Alumínio\nnP = 95 kN, E = 69 GPa', fontsize=14, fontweight='bold')
plt.legend(fontsize=11)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)
```

```
for i in range(len(L)):  
    plt.annotate(f'{U_al[i]:.2f}',  
                xy=(L[i], U_al[i]),  
                xytext=(5, 5),  
                textcoords='offset points',  
                fontsize=9)  
  
plt.tight_layout()  
plt.show()
```

```
#Código da questao 3, item B:
```

```
#Plote um gráfico da curva de energia de deformação da viga AB com os valores de L,
de 0 até 5 m, usando incrementos de 1 m, utilizando o aço como material da viga
```

```
import numpy as np
import matplotlib.pyplot as plt

# dados
P = 95000 # Peso = 45+50 = 95kN
E_aco = 200e9 #Aço
I = 1.13e-4 # poderia ser 113e-6 tambem, que foi o valor utilizado nos meus
calculos
L = np.array([0, 1, 2, 3, 4, 5])
# formula da energia de deformação  $U = P^2 L^3 / (96EI)$ 
U_aco = (P**2 * L**3) / (96 * E_aco * I)
print("Energia de Deformacao - Aco")
print("-" * 50)
for i in range(len(L)):
    print(f'L = {L[i]} m: U = {U_aco[i]:.4f} J')
plt.figure(figsize=(10, 6))
plt.plot(L, U_aco, 'ro-', linewidth=2, markersize=8, label='Aço')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Energia de Deformação U (J)', fontsize=12)
plt.title('Energia de Deformação vs Comprimento - Aço\nnP = 95 kN, E = 200 GPa',
          fontsize=14, fontweight='bold')
plt.legend(fontsize=11)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)
for i in range(len(L)):
    plt.annotate(f'{U_aco[i]:.2f} J',
                 xy=(L[i], U_aco[i]),
                 xytext=(5, 5),
                 textcoords='offset points',
                 fontsize=9)
plt.tight_layout()
```

```
plt.show()
```

```

#Código da questão 3, item C:

#Plote um gráfico da curva de deslocamento da viga AB com os valores de L, de 0 até
5m, usando incrementos de 1 m, utilizando o alumínio como material da viga

import numpy as np
import matplotlib.pyplot as plt


# dados
P = 95000 # Peso = 45+50 = 95kN
E_al = 69e9 # Aço
I = 1.13e-4 # poderia ser 113e-6 também, que foi o valor utilizado nos meus
calculos

L = np.array([0, 1, 2, 3, 4, 5])

# formula do deslocamento no ponto C: Xc = PL³/(48EI)
X_c_al = (P * L**3) / (48 * E_al * I)

print("Deslocamento no Ponto C - Alumínio (em METROS)")
print("-" * 50)
for i in range(len(L)):
    print(f'L = {L[i]} m: delta = {X_c_al[i]:.8f} m')

plt.figure(figsize=(10, 6))
plt.plot(L, X_c_al, 'bo-', linewidth=2, markersize=8, label='Alumínio')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Deslocamento Xc (m)', fontsize=12)
plt.title('Curva de Deslocamento da Viga AB - Alumínio\nP = 95 kN, E = 69 GPa',
          fontsize=14, fontweight='bold')
plt.legend(fontsize=11)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)

for i in range(len(L)):
    if L[i] > 0:

```

```
plt.annotate(f'{X_c_al[i]:.8f} m',
            xy=(L[i], X_c_al[i]),
            xytext=(5, 5),
            textcoords='offset points',
            fontsize=9)

plt.tight_layout()
plt.show()
```

```
#Código da questão 3, item D:
```

```
import numpy as np
import matplotlib.pyplot as plt

#Plote um gráfico da curva de deslocamento da viga AB com os valores de L, de 0 até 5 m, usando incrementos de 1 m, utilizando o aço como material da viga

# dados
P = 95000 # Peso = 45+50 = 95kN
E_aco = 200e9 # Aço
I = 1.13e-4 # poderia ser 113e-6 também, que foi o valor utilizado nos meus cálculos
L = np.array([0, 1, 2, 3, 4, 5])

# fórmula do deslocamento no ponto C: Xc = PL³/(48EI)
X_c_aco = (P * L**3) / (48 * E_aco * I)

print("Deslocamento no Ponto C - Aço")
print("-" * 50)
for i in range(len(L)):
    print(f'L = {L[i]} m: deltarac = {X_c_aco[i]:.8f} m')

plt.figure(figsize=(10, 6))
plt.plot(L, X_c_aco, 'ro-', linewidth=2, markersize=8, label='Aço')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Deslocamento Xc (m)', fontsize=12)
plt.title('Curva de Deslocamento da Viga AB - Aço\nP = 95 kN, E = 200 GPa', fontsize=14, fontweight='bold')
plt.legend(fontsize=11)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)
for i in range(len(L)):
    if L[i] > 0:
        plt.annotate(f'{X_c_aco[i]:.8f} m',
```

```
    xy=(L[i], X_c_aco[i]),
    xytext=(5, 5),
    textcoords='offset points',
    fontsize=9)

plt.tight_layout()
plt.show()
```

#SEGUNDA PARTE DO EXERCÍCIO TRÊS:

#Se o material A possui uma maior energia de deformação que o material B, altere o perfil da viga do material A para que ele tenha uma energia de deformação menor que o material B, e repita as letras (a) e (b) com o novo perfil para o material A.

#Código da questao 3, item A com o perfil do Alumínio alterado para 2.9 vezes o de aço:

#Plote um gráfico da curva de energia de deformação da viga AB com os valores de L, de 0 até 5 m, usando incrementos de 1 m, utilizando o alumínio com novo perfil como material da viga

```
import numpy as np
import matplotlib.pyplot as plt

#Código da questao 3, item A:

# dados
P = 95000 # Peso = 45+50 = 95kN
E_al = 69e9 #Aluminio
I = 327.7e-6 # novo perfil, 2.9 x o I do aço

L = np.array([0, 1, 2, 3, 4, 5])
# formula da energia de deformação U = P^2L^3/(96EI)
U_al = (P**2 * L**3) / (96 * E_al * I)

print("Energia de Deformacao - Aluminio Novo Perfil")
print("-" * 50)
for i in range(len(L)):
    print(f'L = {L[i]} m: U = {U_al[i]:.4f} J')

plt.figure(figsize=(10, 6))
plt.plot(L, U_al, 'bo-', linewidth=2, markersize=8, label='Alumínio')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Energia de Deformacao U (J)', fontsize=12)
plt.title('Energia de Deformação vs Comprimento - Alumínio Novo Perfil\nP = 95 kN, E = 69 GPa', fontsize=14, fontweight='bold')
```

```

plt.legend(fontsize=11)

plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)

for i in range(len(L)):

    plt.annotate(f'{U_aco[i]:.4f} J',
                 xy=(L[i], U_aco[i]),
                 xytext=(5, 5),
                 textcoords='offset points',
                 fontsize=9)

plt.tight_layout()
plt.show()

#A letra b permanece o mesmo código [#Código da questão 3, item B]:
import numpy as np
import matplotlib.pyplot as plt

# dados
P = 95000 # Peso = 45+50 = 95kN
E_aco = 200e9 #Aço
I = 1.13e-4 # poderia ser 113e-6 também, que foi o valor utilizado nos meus
calculos
L = np.array([0, 1, 2, 3, 4, 5])
# formula da energia de deformação  $U = P^2 L^3 / (96EI)$ 
U_aco = (P**2 * L**3) / (96 * E_aco * I)
print("Energia de Deformação - Aço")
print("-" * 50)
for i in range(len(L)):

    print(f'L = {L[i]} m: U = {U_aco[i]:.4f} J')

plt.figure(figsize=(10, 6))
plt.plot(L, U_aco, 'ro-', linewidth=2, markersize=8, label='Aço')
plt.grid(True, alpha=0.3)
plt.xlabel('Comprimento L (m)', fontsize=12)
plt.ylabel('Energia de Deformação U (J)', fontsize=12)

```

```

plt.title('Energia de Deformação vs Comprimento - Aço\nP = 95 kN, E = 200 GPa',
          fontsize=14, fontweight='bold')
plt.legend(fontsize=11)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)
for i in range(len(L)):
    plt.annotate(f'{U_aco[i]:.2f} J',
                 xy=(L[i], U_aco[i]),
                 xytext=(5, 5),
                 textcoords='offset points',
                 fontsize=9)
plt.tight_layout()
plt.show()

```

#C- Gráfico de comparação simultânea entre letras A e B

```

import numpy as np
import matplotlib.pyplot as plt

# Dados
P = 95000
E_al = 69e9
E_aco = 200e9
I_al = 327.7e-6 # (novo perfil)
I_aco = 113e-6 # (perfil original)

L_plotagem = np.linspace(0, 5, 100)
L_marcadores = np.array([0, 1, 2, 3, 4, 5])

# formula da Energia de deformação: U = P^2 L^3 / (96 E I)
U_al_plotagem = (P**2 * L_plotagem**3) / (96 * E_al * I_al)
U_aco_plotagem = (P**2 * L_plotagem**3) / (96 * E_aco * I_aco)

U_al_marcadores = (P**2 * L_marcadores**3) / (96 * E_al * I_al)
U_aco_marcadores = (P**2 * L_marcadores**3) / (96 * E_aco * I_aco)

```

```

print("*"*80)
print("COMPARACAO: ENERGIA DE DEFORMACAO - ALUMINIO (NOVO PERFIL) vs ACO")
print("*"*80)

print(f"ALUMINIO: E = {E_al/1e9:.0f} GPa | I = {I_al*1e6:.1f}x10^-6 m^4 (perfil otimizado)")

print(f"ACO:      E = {E_aco/1e9:.0f} GPa | I = {I_aco*1e6:.0f}x10^-6 m^4 (perfil original)")

print("-"*80)

print(f"{'L (m)':<8} {'U_Aluminio (J)':<18} {'U_Aco (J)':<18} {'Diferenca (%)':<15}")
print("-"*80)

for i in range(len(L_marcadores)):

    if L_marcadores[i] > 0:

        dif_perc = ((U_aco_marcadores[i] - U_al_marcadores[i]) / U_al_marcadores[i])
        * 100

        print(f"{L_marcadores[i]:<8.1f} {U_al_marcadores[i]:<18.4f}"
        f"{U_aco_marcadores[i]:<18.4f} {dif_perc:+14.2f}%")

    else:

        print(f"{L_marcadores[i]:<8.1f} {U_al_marcadores[i]:<18.4f}"
        f"{U_aco_marcadores[i]:<18.4f} {'---':>15}")

print("*"*80)

print(f"O Aco agora possui energia {U_aco_marcadores[5]/U_al_marcadores[5]:.2f} vezes maior que o Aluminio!")
print("*"*80)

plt.figure(figsize=(12, 8))

plt.plot(L_plotagem, U_al_plotagem, 'b-', linewidth=4,
         label=f'Aluminio (I={I_al*1e6:.1f}x10^-6 m^4)'),
         alpha=0.7, zorder=2)

plt.plot(L_plotagem, U_aco_plotagem, 'r-', linewidth=4,
         label=f'Aco (I={I_aco*1e6:.0f}x10^-6 m^4)'),
         alpha=0.7, zorder=2)

plt.plot(L_marcadores, U_al_marcadores, 'bo', markersize=14,
         markerfacecolor='lightblue', markeredgewidth=3,

```

```

        markeredgecolor='darkblue', zorder=3)

plt.plot(L_marcadores, U_aco_marcadores, 'rs', markersize=14,
          markerfacecolor='lightcoral', markeredgewidth=3,
          markeredgecolor='darkred', zorder=3)

plt.fill_between(L_plotagem, U_al_plotagem, U_aco_plotagem,
                 where=(U_aco_plotagem >= U_al_plotagem),
                 color='yellow', alpha=0.3, zorder=1)

plt.grid(True, alpha=0.4, linestyle='--', linewidth=1.2)
plt.xlabel('Comprimento L (m)', fontsize=14, fontweight='bold')
plt.ylabel('Energia de Deformação U (J)', fontsize=14, fontweight='bold')
plt.title('Energia de Deformação: Alumínio (Novo Perfil) vs Aço\nP = 95 kN',
          fontsize=15, fontweight='bold')
plt.legend(fontsize=12, loc='upper left', framealpha=0.95)
plt.xlim(-0.2, 5.2)
plt.ylim(bottom=0)

for i in range(1, len(L_marcadores)):

    # Aluminio
    plt.annotate(f'{U_al_marcadores[i]:.4f}',
                xy=(L_marcadores[i], U_al_marcadores[i]),
                xytext=(-15, -30),
                textcoords='offset points',
                fontsize=10, fontweight='bold', color='darkblue',
                bbox=dict(boxstyle='round,pad=0.4', facecolor='lightblue',
                          edgecolor='darkblue', linewidth=2, alpha=0.9),
                arrowprops=dict(arrowstyle='->', color='darkblue', lw=1.5))

    # Aço
    plt.annotate(f'{U_aco_marcadores[i]:.4f}',
                xy=(L_marcadores[i], U_aco_marcadores[i]),
                xytext=(15, 25),
                textcoords='offset points',
                fontsize=10, fontweight='bold', color='darkred',

```

```
        bbox=dict(boxstyle='round',pad=0.4, facecolor='lightcoral',
                  edgecolor='darkred', linewidth=2, alpha=0.9),
                  arrowprops=dict(arrowstyle='->', color='darkred', lw=1.5))

plt.tight_layout()
plt.show()

print("\n" + "*80)
print("CONCLUSAO - DUALISMO ENERGETICO:")
print("*80)

print("→ As RETAS mostram claramente a diferenca entre os materiais")

print(f"→ O Aco possui entre {((U_aco_marcadores[1] - U_al_marcadores[1]) /
U_al_marcadores[1]) * 100:.1f}% e {((U_aco_marcadores[-1] - U_al_marcadores[-1]) /
U_al_marcadores[-1]) * 100:.1f}% MAIS energia")

print("→ A area amarela representa o DUALISMO: zona onde Aco > Aluminio")
print("→ Perfil do Aluminio aumentado em ~2.9x inverte a relacao energetica")
print("*80)
```