DS n° 03 – DS03

- Faire tous les exercices dans un même fichier NomPrenom.py à sauvegarder,
- mettre en commentaire l'exercice traité (ex : # Exercice 1),
- ne pas oublier pas de commenter ce qui est fait dans votre code (ex : # Je créé une fonction pour calculer la racine d'un nombre),
- il est possible de demander un déblocage pour une question, mais celle-ci sera notée 0,
- il faut vérifier avant de partir que le code peut s'exécuter et qu'il affiche les résultats que vous attendez.

Étude du dimensionnement d'une étagère

L'objectif de cette partie est d'étudier le dimensionnement d'une étagère sur laquelle sont installés des livres.
L'étagère est posée sur deux équerres qui sont représentées par les deux appuis en A et en B sur la figure 2. Le poids des livres est modélisé par une charge linéique répartie q ($\mathrm{N\,m^{-1}}$) sur toute la largeur de l'étagère.
Le cahier des charges impose une flèche (déformation) maximale de 1 cm ($fmax$ sur la figure suivante).

FIGURE 1 – Étagère

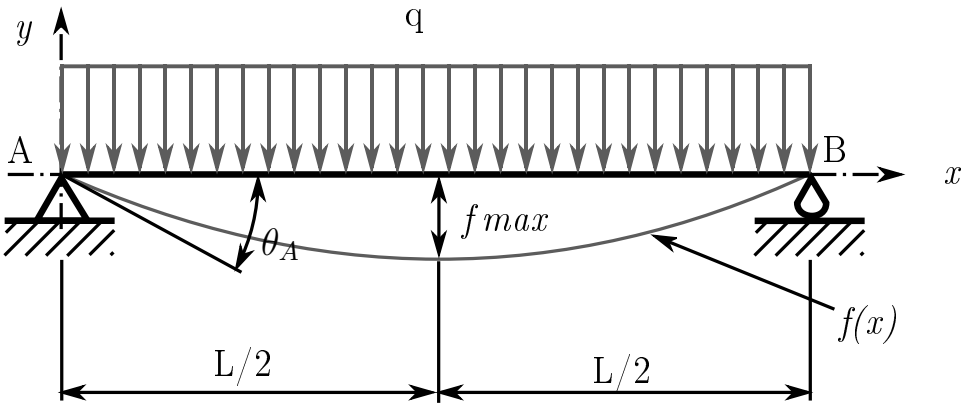

FIGURE 2 – Modèle poutre de l'étagère

Données :

- Masse volumique d'un livre (papier) : $\rho = 500\,\mathrm{kg\,m^{-3}}$,
- Profondeur moyenne des livres : $p = 10\,\mathrm{cm}$,
- Hauteur moyenne des livres : $h = 20\,\mathrm{cm}$,
- Module d'Young du bois de l'étagère : $E = 2{,}2\,\mathrm{GPa} = 2{,}2 \times 10^3\,\mathrm{N\,mm^{-2}}$,
- Longueur de l'étagère : $L = 2\,\mathrm{m}$,
- Épaisseur de l'étagère : $e = 18\,\mathrm{mm}$,
- Profondeur de l'étagère : $b = 30\,\mathrm{cm}$,
- Accélération de pesanteur : $g = 9{,}81\,\mathrm{m\,s^{-2}}$.

Mise en équations du problème

1. Déterminer l'expression de la charge linéique q. Rentrer les valeurs numériques de l'énoncé ainsi que l'expression de q dans votre script python et afficher la valeur numérique obtenue à l'aide d'un `print(q)`.

Solution 1.

```
rho=500.
g=9.81
p=0.1
h=0.2
L=2
E=2.2*10**9

q=rho*g*p*h

print(q)
```

La valeur affichée est `98.10000000000001`

Une étude statique a permis de déterminer le moment fléchissant tout au long de la poutre. Celui-ci représente une partie des contraintes (mais ce sont celles qui nous intéressent) à l'intérieur de la poutre. Ce sont ces contraintes qui sont à l'origine de sa déformation. Le moment fléchissant suivant est exprimé au point $M \in [AB]$ tel que $\overrightarrow{AM} = x.\overrightarrow{x}$.

$$Mfz(x) = \frac{x.q.L}{2} - \frac{q.x^2}{2}$$

On constate que si M est sur le point A $(x = 0)$ ou B $(x = L)$ alors le moment fléchissant est nul.

2. Créer une fonction `Mfz` qui prend en entrée x et qui retourne la valeur de $Mfz(x)$. Afficher la valeur de $Mfz(x)$ pour $x = 1{,}2\,\text{m}$.

   Solution 2.

```
def Mfz(x):
    return x*q*L/2-q*x**2/2

print(Mfz(1.2))
```

La valeur affichée est `47.087999999999994`

La théorie des poutres permet de déterminer la déformée d'une poutre (représentée par la fonction $f(x)$ sur la figure 2) à partir de ses caractéristiques géométriques (moment quadratique Ix), de son matériau (module d'Young E) et des contraintes qu'elle subit (moment fléchissant Mfz).
Ainsi, on obtient la formule suivante :

$$\frac{d^2f}{dx^2}(x) = \frac{Mfz(x)}{E.Ix}$$

Avec $Ix = \frac{e^3.b}{12}$ pour une poutre de section rectangulaire comme celle de l'étagère modélisée sur la figure 3.

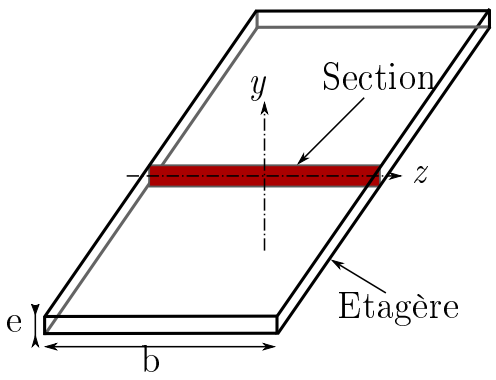

FIGURE 3 – Paramètres du moment quadratique de l'étagère

3. (a) Créer une fonction `Ix` qui prend en entrée le couple (e, b) et qui retourne la valeur de Ix. Afficher la valeur de Ix pour les dimensions de la planche.
Solution 3.

```
def Ix(e,l):
    return e**3*b/12
print(Ix(0.018,0.3))
```

La valeur affichée est`1.4579999999999997e-07`

(b) Créer une fonction `ddf` qui prend en entrée le x et qui retourne la valeur de $\frac{d^2f}{dx^2}(x)$. Afficher la valeur de $\frac{d^2f}{dx^2}(x)$ pour $x = 1{,}2\,\text{m}$.
Solution 4.

```
def ddf(x):
    return (Mfz(x))/(E*Ix(0.018,0.3))
print(ddf(1.2))
```

La valeur affichée est `0.14680134680134682`

Intégration numérique

La variable x sera modélisée à l'aide d'une fonction `linspace` contenant $nbx = 1000$ éléments.

4. (a) Créer la variable `x` dans votre code python et générer un tableau `numpy` de type array `ddfleche` qui contient l'image de x par $\frac{d^2f}{dx^2}(x)$. Afficher la 250$^{\text{e}}$ valeur de `ddfleche`
Solution 5.

```
nbx=1000
x = np.linspace(0, L, nbx)
ddfleche=f2(x)
print(ddfleche[250])
```

La valeur affichée est `0.11476504945282369`

(b) Par une méthode d'intégration, déterminer le tableau `numpy` de type array `dfleche` qui contient l'image de x par $\frac{df}{dx}(x)$. Dans un premier temps, la constante d'intégration sera considérée nulle. Afficher la valeur de $\frac{df}{dx}(\frac{L}{2})$. Tracer $\frac{df}{dx}(x)$ en fonction de x.
Solution 6.

```
nbi = nbx - 1
dfleche = np.linspace(1, 1, nbx)
dfleche[0]=0
for i in range(nbi):
    dfleche[i+1] = dfleche[i] + ddfleche[i]*(x[i+1]-x[i])
print(dfleche[500])
plt.plot(x[:len(dfleche)],dfleche)
```

La valeur affichée est `0.10194527757358089`

(c) A partir de la figure 2, déterminer la valeur de $\frac{df}{dx}(\frac{L}{2})$, en déduire la valeur de la constante d'intégration et modifier `dfleche` en conséquence. Tracer la nouvelle forme de $\frac{df}{dx}(x)$ en fonction de x.
Solution 7.
D'après la figure 2, $\frac{df}{dx}(\frac{L}{2}) = 0$, donc la constante d'intégration est $-\frac{df}{dx}(\frac{L}{2})$.

```
dfleche=dfleche-dfleche[500]
plt.plot(x[:len(dfleche)],dfleche)
```

(d) Par une méthode d'intégration déterminer le tableau `numpy` de type array `fleche` qui contient l'image de x par $f(x)$. Dans un premier temps, la constante d'intégration sera considérée nulle. Afficher la valeur de $f(0)$, en déduire la valeur de la constante d'intégration et corriger si nécessaire. Tracer $f(x)$ en fonction de x. Déterminer la valeur maximale de la flèche. Est-ce que le cahier des charges est respecté ?

Solution 8.

```
nbi = nbx - 1
fleche = np.linspace(1, 1, nbx)
fleche[0]=0
for i in range(nbi):
    fleche[i+1] = fleche[i] + dfleche[i]*(x[i+1]-x[i])
plt.plot(x[:len(fleche)],fleche)
print(fleche[500])
print("%f > 2cm, le cahier des charges n'est pas respecté." % (abs(fleche[500])*100
```

La valeur affichée est `-0.06391993143933793`, cela signifie que la poutre se déforme de 6,38 cm vers le bas.

Choix d'une solution

Cette dernière partie va permettre de choisir une solution pour cette étagère.
On propose la possibilité de fabriquer cette étagère dans deux autres matériaux. On met donc la liste des matériaux sous la forme de la liste de tuples suivante :
`liste_mat=[('médium',2.2),('contreplaqué',12.4),('verre',69)]`
Le premier élément du tuple contient le nom du matériau, le second, son module d'Young E en GPa. L'objectif de cette partie est de tester pour chaque matériau l'épaisseur de la planche qui permet de respecter le cahier des charges.

5. (a) Créer à partir des résultats précédents la fonction `calcul_fleche(E,e)` qui affiche la flèche maximale de l'étagère pour une configuration (E, e) donnée. Vérifier que cette fonction donne le même résultat que précédemment en affichant la flèche pour $E = 2\,\text{GPa}$ et $e = 18\,\text{mm}$.

Solution 9.

```
def calcul_fleche(E,e):
    def ddf(x):
        return (Mfz(x))/(E*Ix(e,0.3))
 nbx=1000
  x = np.linspace(0, L, nbx)
  ddfleche=ddf(x)

nbi = nbx - 1
  dfleche = np.linspace(1, 1, nbx)
    dfleche[0]=0
    for i in range(nbi):
        dfleche[i+1] = dfleche[i] + ddfleche[i]*(x[i+1]-x[i])
    dfleche=dfleche-dfleche[500]
    nbi = nbx - 1
    fleche = np.linspace(1, 1, nbx)
    fleche[0]=0
    for i in range(nbi):
                fleche[i+1] = fleche[i] + dfleche[i]*(x[i+1]-x[i])
    return fleche[500]
print(calcul_fleche(2.2*10**9,0.018))
```

On retrouve bien `-0.06391993143933793` comme précédemment.

(b) À l'aide d'un algorithme testant des épaisseurs de planche de 1 mm à 10 cm avec un incrément de 1 mm, déterminer pour chaque matériau l'épaisseur minimale de l'étagère pour respecter le cahier des charges. Afficher pour chaque matériau, le nom du matériau et l'épaisseur correspondante.

Solution 10.

```
fmax=0.02
liste_mat=[('médium',2.2),('contreplaqué',12.4),('verre',69)]
```

```
liste_ep=np.arange(0.001,0.1,0.001)
print('Pour respecter le cahier des charges:')
for (nom,E) in liste_mat:
    i=0
    while abs(calcul_fleche(E*10**9,liste_ep[i]))>fmax:
        i+=1
```

Le programme affiche :

```
Pour respecter le cahier des charges:
 - il faudrait une étagère en médium de 27 mm d'épaisseur
 - il faudrait une étagère en contreplaqué de 15 mm d'épaisseur
 - il faudrait une étagère en verre de 9 mm d'épaisseur
```