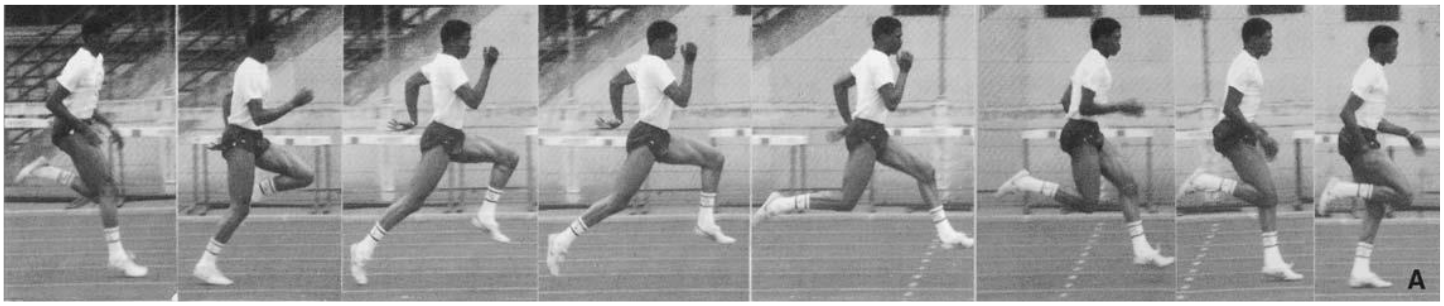
DM2 D'INFORMATIQUE
CORRECTION

ÉTUDE BIOMECANIQUE DE LA COURSE A PIED



L'étude biomécanique de la course à pied fait l'objet de nombreux travaux de recherche motivés par les outils techniques d'observation et de mesures de plus en plus perfectionnés. Ces plateaux techniques permettent d'évaluer le mouvement suivant les trois axes de l'espace et offrent les moyens d'accéder à des grandeurs non mesurables comme les efforts aux niveaux articulaires. Ils contribuent aussi à l'élaboration de matériels sportifs ainsi qu'à la mise au point des chaussures de course à pied qui est la thématique du sujet proposé.

Ce sujet comporte **trois parties indépendantes**, elles pourront donc être traitées séparément :

- la **première partie** est consacrée à l'étude expérimentale de la phase d'appui du pied du coureur sur le sol, en vue de caractériser les phénomènes dynamiques mis en jeu et proposer une mesure des performances associées à la chaussure de sport ;
- la **seconde partie** est dédiée à l'élaboration d'un modèle de connaissance visant à prévoir les phénomènes dynamiques mis en jeu. On s'intéresse plus particulièrement à l'identification de certains des paramètres du modèle à partir d'essais de caractérisation ;
- la **dernière partie** traite de la résolution numérique des équations issues du modèle, ouvrant ainsi la voie à la simulation de la phase d'appui du pied et à la prévision du comportement dynamique de différentes chaussures de sport .

✓ Dans tout le sujet, il sera supposé que les modules et bibliothèques sont déjà importés dans l'espace de travail.

✓ Une attention toute particulière sera portée sur l'initialisation des algorithmes, l'indentation, les commentaires ainsi que la lisibilité des solutions proposées.

✓ Des points dans le barème de notation seront accordés à la qualité de la présentation et de la rédaction

PARTIE I. MESURES ET TRAITEMENT DES DONNEES

Objectif : cette première partie est consacrée à l'étude expérimentale de la phase d'appui du pied du coureur sur le sol, en vue de caractériser les phénomènes dynamiques mis en jeu et proposer une mesure des performances associées à la chaussure de sport

Si les bienfaits psychiques et cognitifs du sport ne sont plus à démontrer, il n'en reste pas moins que la pratique répétée d'une activité intensive peut avoir des effets néfastes et entraîner divers traumatismes chez le sportif. C'est bien souvent le cas lors de la pratique de la course à pied, où l'effort engendré au cours de l'impact du pied sur le sol possède une intensité de l'ordre de 3 fois le poids du coureur et s'accompagne d'une onde de choc qui se propage dans le squelette jusqu'à l'occiput. Une pratique désormais courante chez les biomécaniciens consiste à équiper le tibia, le rachis et le crâne des coureurs de capteurs accéléromètriques, afin de caractériser les phénomènes dynamiques mis en jeu durant la course. On constate que les accélérations mesurées au cours de la phase d'appui sont importantes chez un coureur pied nu, elles diminuent sur un sol sportif et surtout lors du port de certaines chaussures limitant ainsi les traumatismes musculo-squelettiques tout en améliorant le confort.

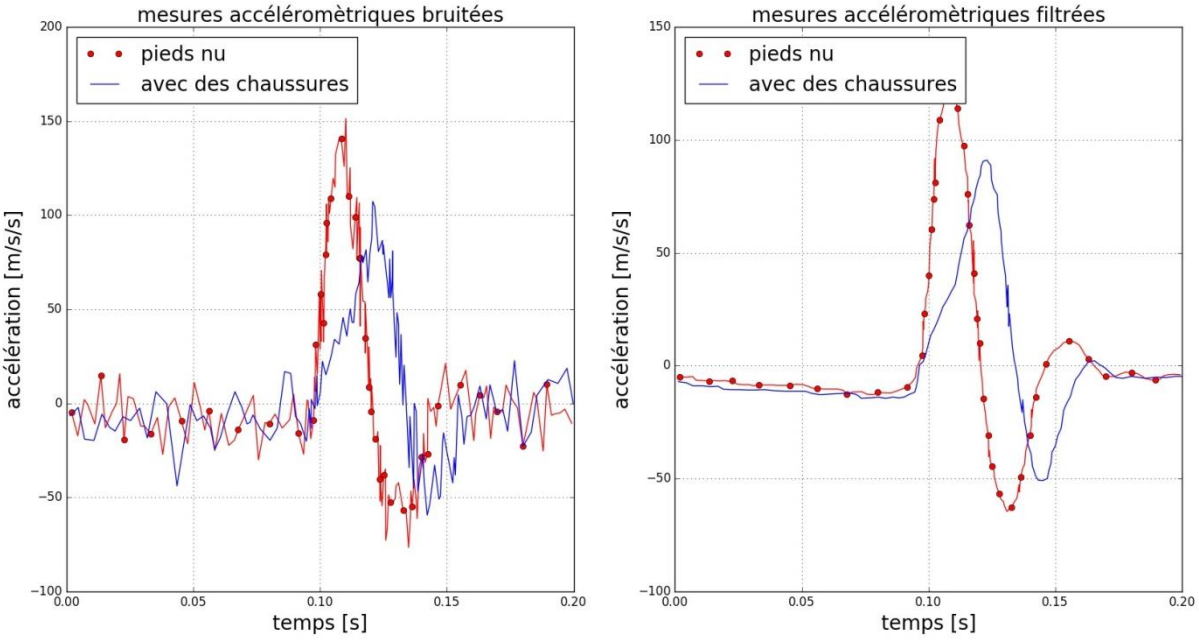


Figure 1 : mesures accéléromètriques bruitées (à gauche) et filtrées (à droite)

La figure 1 représente une mesure[1] relevée au niveau du tibia avec et sans chaussure que l'on notera respectivement mesure1 et mesure2. Le fichier est représenté partiellement ci-dessous. Les éléments sont de type « chaîne ». Les informations {temps1, mesure1, temps2, mesure2} ont pour séparateur « \t » et un retour à la ligne est renseigné par « \n ».

```
['0.00191\t-4.95\t0.00134\t-7.08\n', '0.00439\t-5.11\t0.0046\t-7.59\n', '0.00698\t-4.78\t0.0070
'0.01351\t-6.75\t0.01699\t-9.09\n', '0.01531\t-6.58\t0.01913\t-10.44\n', '0.01791\t-6.75\t0.022
7\n', '0.02264\t-6.58\t0.03185\t-10.77\n', '0.02421\t-8.06\t0.03523\t-10.6\n', '0.02635\t-8.39\
-11.26\n', '0.03322\t-8.55\t0.05133\t-11.26\n', '0.03581\t-8.55\t0.05403\t-12.27\n', '0.03795\t
6057\t-12.6\n', '0.04528\t-8.7\t0.06338\t-12.43\n', '0.04775\t-8.21\t0.06642\t-12.26\n', '0.050
1\t0.07228\t-13.26\n', '0.05609\t-10.01\t0.07351\t-14.44\n', '0.05845\t-10.34\t0.07644\t-14.27\
06623\t-10.83\t0.08601\t-13.93\n', '0.06769\t-12.48\t0.08838\t-14.26\n', '0.07017\t-11.65\t0.09
1.73\n', '0.07828\t-12.8\t0.09514\t-9.53\n', '0.07997\t-11.81\t0.09571\t-6.33\n', '0.08222\t-11
t1.94\n', '0.08954\t-10.31\t0.09877\t5.41\n', '0.09146\t-9.32\t0.09888\t4.97\n', '0.09349\t-7.3
6.45\n', '0.09631\t1.56\t0.10341\t19.32\n', '0.09744\t4.7\t0.10454\t22.52\n', '0.09778\t8.16\t0
\n', '0.09801\t15.91\t0.11053\t35.85\n', '0.09824\t23\t0.11133\t40.58\n', '0.09881\t25.97\t0.11
', '0.10006\t36.69\t0.11405\t56.43\n', '0.10017\t39.99\t0.1153\t59.64\n', '0.10029\t43.12\t0.11
', '0.10052\t52.52\t0.11882\t78.7\n', '0.1012\t60.44\t0.11973\t82.41\n', '0.10131\t59.03\t0.170
, '0.10166\t71.32\t0.12301\t91.02\n', '0.10212\t73.79\t0.12458\t93.18\n',
 '0.10257\t80.83\t0.12715\t75.67\n', '0.10354
```

[1] Afin d'améliorer la représentation, la mesure est filtrée pour réduire le bruit et améliorer le rapport signal sur bruit.

LECTURE DES FICHIERS DE MESURES ET AFFICHAGE

Q1. *En utilisant la documentation fournie en annexe, commenter la fonction* `lecture_fichier` *présentée sur le document réponse question* $Q1$.

Cette fonction retourne un objet nommé $MatDeMes$ qui est un tableau de flottants organisé de la façon suivante :

temps1	mesure1	temps2	mesure2
0.002	-4.950	0.001	-7.080
0.004	-5.110	0.005	-7.590
0.007	-4.780	0.007	-8.930
0.008	-6.420	0.011	-8.930
0.011	-6.590	0.014	-9.270
0.014	-6.750	0.017	-9.090
0.015	-6.580	0.019	-10.440
0.018	-6.750	0.022	-10.610
0.020	-6.580	0.025	-10.610
0.021	-6.910	0.029	
0.023	-6.580		

Q2. *Proposer une fonction* `affichage(MatDeMes)` *qui a pour argument d'entrée le tableau de flottants. Cette fonction trace mesure1 en fonction de temps1 et mesure2 en fonction de temps2 sur un même graphique. Renseigner les axes ainsi que le titre.*

```
def affichage(MatDeMes):
    temps1=MatDeMes[:,0]
    mesure1=MatDeMes[:,1]
    temps2=MatDeMes[:,2]
    mesure2=MatDeMes[:,3]
    plt.plot(temps1,mesure1,"r",label="Pieds nus",marker="o")
    plt.plot(temps2,mesure2,"b",label="Avec des chaussures")
    plt.title("Mesures accélérométriques bruitées");plt.xlabel("temps (s)");plt.ylabel("accélération (m/s/s")
    plt.legend()
    plt.grid(True)
    return
```

FILTRAGE DES MESURES ET ESTIMATION DU BRUIT

La mesure $\{m[n]\}_{n=0,\dots,N-1}$ est bruitée (*cf.* figure 1, graphique de gauche). Afin d'améliorer sa qualité[2], elle est filtrée par un filtre de type Butterworth d'ordre 3 qui a la particularité d'avoir un gain constant dans sa bande passante. Il a pour équation récurrente l'expression suivante :

$$s[n] = b[0]m[n] + b[1]m[n-1] + b[2]m[n-2] + b[3]m[n-3] - a[1]s[n-1] - a[2]s[n-2] - a[3]s[n-3]$$

[2] Réduire la puissance du bruit b par rapport à la puissance du signal m ; améliorer le rapport signal sur bruit (S/B).

Dans cette équation, $\{s[n]\}_{n=0,\dots,N-1}$ désigne la mesure filtrée, $a[.]$ et $b[.]$les coefficients du filtre donnés dans le tableau ci-dessous :

$indices$	[0]	[1]	[2]	[3]
$b[.]$	0.018	0.054	0.054	0.018
$a[.]$	1.000	−1.760	1.183	−0.278

Tableau des coefficients du filtre de Butterworth d'ordre 3

Q3. *La fonction* `filtrage(m)` *a pour arguments d'entrée le tableau de mesure* `m` *et renvoie la filtrée* `s`*, un tableau de même dimension que l'entrée* `m`*. Compléter la question* $Q3$ *du document réponse.*

La qualité de la mesure est définie classiquement par le rapport de la puissance du signal sur la puissance du bruit. Un rapport trop faible a souvent pour origine un capteur mal fixé. Il faut dans ce cas refaire la mesure. On se propose de calculer ce rapport noté $R_{S/B}$.

Q4. *Ecrire la fonction* `puissance(signal)` *qui a pour argument* `signal={sᵢ}, i=0,…,L-1` *un tableau de réels de* `L` *éléments et qui renvoie le flottant* `p`$=\frac{1}{L}\sum_{i=0}^{L-1} s_i^2$*, une estimation de la puissance de* `signal`*.*

```
def puissance(signal):

    l=len(signal)

    p=0

    for i in range(l):
       p=p+signal[i]*signal[i]

    p=p/l

    return p
```

Le bruit b est estimé par différence entre la mesure m et sa filtrée s, soit $b[n] = m[n] - s[n]$

Q5. *Ecrire la fonction* `bruit(s,m)` *qui renvoie un tableau* `b`*, de même dimension que* `s` *et* `m`*, qui est une estimation du bruit additif à la mesure* `m`*.*

```
def bruit(m,s):

    l=len(m)
    b=zeros(l,dtype(float))
    for i in range(l):
        b[i]=m[i]-s[i]

    return b
```

Q6. *A partir des codes précédents, proposer une fonction* `RSB(m,s)` *qui renvoie le rapport signal sur bruit* $R_{(S/B)}$ *que vous exprimerez en dB et qui a pour expression :* $R_{S/B}(dB) = 10 \times log_{10}\left(\frac{puissance\ du\ signal}{puissance\ du\ bruit}\right)$

```
def RSB(m,s):

    ps=puissance(s)
    pb=puissance(bruit(m,s))

    RSB=10*np.log10(ps/pb)
    return RSB
```

CARACTERISATION DES PHENOMENES DYNAMIQUES MIS EN JEU

L'onde de choc subie par le coureur lors d'une course est-elle nuisible ? A ce jour, il n'y a pas de preuve scientifique qui confirme le lien entre une pathologie et l'onde de choc transmise par le talon à la structure osseuse. Il y a néanmoins un faisceau de présomptions[3]. Toutefois, le rôle indispensable de l'onde de choc est démontré. Les recherches menées en impesanteur[4] comme l'étude de l'ostéoporose a prouvé le rôle primordial de l'onde de choc dans la minéralisation et la résistance de notre squelette. Courir sur des semelles amortissantes a donc des répercussions importantes sur notre corps qu'il convient d'étudier aussi bien dans le **domaine temporel** que **fréquentiel**.

- **Analyse Temporelle**

Une première analyse des mesures dans le domaine temporel consiste à en extraire les paramètres objectifs qui sont essentiellement le maximum, l'atténuation qui est le rapport des maximums pour les deux situations avec et sans chaussures ainsi que la vitesse qui précède le maximum.

Q7. *Ecrire une fonction* `maximum(temps, mesure)` *qui a pour arguments d'entrée le temps et la mesure de type « tableau » et qui renvoie le flottant* $(mes)_{max}$ *tel que* $mes_{max} = max\{mesure\}$ *avec le temps correspondant.*

```
def maximum(temps,mesure):
    temps_max=0
    mesure_max=0

    l=len(temps)

    for i in range(l):
        if mesure_max<abs(mesure[i]):
            mesure_max=mesure[i]
            temps_max=temps[i]
    return [temps_max,mesure_max]
```

[3] La première étude date de 1982 et a révélé des lésions cartilagineuses du genou chez des moutons que l'on a obligé à marcher pendant plusieurs jours sur du macadam au lieu du pâturage habituel.
[4] L'impesanteur ou apesanteur est l'absence de sensation de poids.

Q8. *Ecrire une fonction* `gain(mesure1,mesure2)` *qui a pour arguments d'entrée deux mesures de type « tableau » et qui retourne le rapport des maximums, c'est le « gain » apporté par les éléments absorbants de la semelle.*

```
def gain(mesure1,mesure2):
    mesure1_max,mesure2_max=0,0

    l1=len(mesure1)
    l2=len(mesure2)

    for i in range(l1):
        if mesure1_max<abs(mesure1[i]):
            mesure1_max=mesure1[i]
        if mesure2_max<abs(mesure2[i]):
            mesure2_max=mesure2[i]

    gain=mesure1_max/mesure2_max

    return gain
```

L'analyse de l'enregistrement représenté sur la figure 1 montre qu'une chaussure à semelle épaisse élastique et amortissante transforme profondément les contraintes transmises au squelette. Non seulement le maximum est retardé et sa valeur diminuée d'environ 20 % mais aussi la vitesse d'évolution de l'effort est nettement plus faible. Sa connaissance renseigne le médecin sportif des risques de traumatismes.

L'objectif est d'estimer la vitesse pendant la phase qui correspond à la compression de la semelle. La procédure consiste en un clic de souris qui renvoie les coordonnées (t_i, m_i) du point (qui appartient à la courbe) et qui précède le maximum. On assimilera la fonction entre ces deux points par la corde qui les réunit.

Q9. *Ecrire une fonction* `pente(tᵢ,mᵢ,mesure)` *qui a pour arguments d'entrée les coordonnées du point ainsi que la mesure et qui renvoie une estimation de la pente de la fonction entre ces deux points.*

```
def pente(temps,mesure):
    temps_max=0
    mesure_max=0

    l=len(temps)

    for i in range(l):
        if mesure_max<abs(mesure[i]):
            mesure_max=mesure[i]
            mi=mesure[i-1]
            temps_max=temps[i]
            ti=temps[i-1]
    pente=(mesure_max-mi)/(temps_max-ti)
    return pente
```

- **Analyse Fréquentielle**

La démarche scientifique intègre plus difficilement des critères tels que le ressenti ou le bien-être. Le critère CP (Cushioning Parameter) est proposé dans plusieurs publications et permet de quantifier ces données subjectives. Plus le critère CP est important et meilleur sera le ressenti. Il est calculé à partir de la connaissance du module de la FFT (Fast Fourier Transform) de la mesure et a pour expression :

$$CP = \sum_i \frac{d(f_i)}{f_i^3}$$

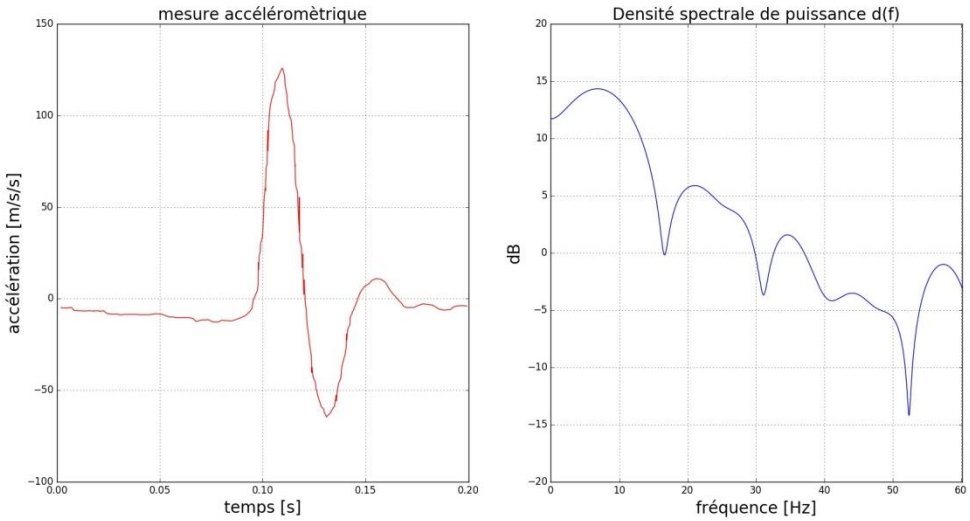


Figure 2 : représentation d'une mesure et de sa densité spectrale de puissance.

Soit $fréquence$ et $puissance$ deux tableaux de même dimension qui représentent pour un même index l'abscisse et l'ordonnée de la densité spectrale de puissance $d(f)$.

Q10. *Ecrire une fonction* `DSP(frequence, puissance)` *qui a pour arguments d'entrée ces deux tableaux et qui retourne le calcul du critère* CP.

```
def DSP(frequence,puissance):
    CP=0

    l=len(frequence)

    for i in range(l):
        CP=CP+puissance[i]/pow(frequence[i],3)

    return CP
```

Les matériaux qui entrent dans la composition des chaussures ont chacun leurs caractéristiques mécaniques, ils sont plus ou moins amortissants, plus ou moins durs, plus ou moins élastiques. L'amélioration du critère CP passe bien souvent par l'association de plusieurs matériaux. De nombreux fabricants d'équipements sportifs mettent en avant le confort et l'absorption nécessaire des chocs avec pour argument de vente l'insuffisance de notre capiton plantaire[5] pour la course de longue distance. Or, il est démontré que l'absence d'onde de choc fragilise les os, pour autant son intensité excessive détruit les tissus. Les seuils restent donc à préciser d'autant plus qu'ils dépendent de la fréquence. L'étude des effets des matériaux amortisseurs est réalisée dans le domaine de Fourier et repose dans le cadre d'une première approche, sur le calcul d'une fonction de coût en fonction de la fréquence.

Soit les deux mesures $m_1(t)$ et $m_2(t)$ présentées sur la figure 1 et leur transformées de Fourier respective $\mathcal{M}_1(f)$ et $\mathcal{M}_2(f)$, qui sont des grandeurs complexes de même dimension et calculées pour une même abscisse. Considérons la fonction de coût suivante :

$$\mathcal{C}(f_i) = \|\mathcal{M}_1(f_i) - \mathcal{M}_2(f_i)\|^2$$

[5] Le capiton plantaire est un tissu graisseux situé dans la partie profonde de la peau sous la plante des pieds. C'est la couche de peau la plus épaisse du corps. Il protège la structure osseuse du pied en contact avec le sol et a pour fonction d'amortir et d'absorber les chocs.

Dans cette expression, la notation $\|.\|^2$ désigne le module carré de la transformée de Fourier, c'est une estimation de la densité spectrale de puissance.

Q11. *Ecrire une fonction nommée* `coût(`$\mathcal{M}_1$, $\mathcal{M}_2$`)` *qui a pour arguments d'entrée les deux tableaux* $\mathcal{M}_1$ *et* $\mathcal{M}_2$ *de même dimension* L *et qui retourne le tableau* $\mathcal{C}$ *constitué des coûts élémentaires* $\mathcal{C}(f_i)$ *pour* $i = 0, \dots, L-1$.

```
def cout(M1,M2):

    C=[]

    l=len(M1)

    for i in range(l):
        C=np.append(C,pow(M1[i]-M2[i],2))

    return C
```

Q12. *Quelle est la complexité temporelle de la fonction de coût* $\mathcal{C}$ *dans le pire cas en fonction du nombre d'éléments* L *des deux tableaux ? (L'analyse devra être soignée, utiliser de préférence un tableau pour faire la synthèse du coût calculatoire de chaque ligne de code).*

Programme	Complexité
`def cout(M1,M2):`	
`C=[]`	1
`l=len(M1)`	1
`for i in range(l):` `  C=np.append(C,pow(M1[i]-M2[i],2))`	L
`return C`	*La complexité est donc de* $L + 2 \approx L$

PARTIE II. ELABORATION D'UN MODELE DE CONNAISSANCE

Objectif : cette seconde partie porte sur l'élaboration d'un modèle de connaissance visant à prévoir les performances de la chaussure de sport en terme d'amortissement des chocs. L'objectif est de mettre en place des outils numériques nécessaires à l'identification de certains paramètres du modèle à partir d'essais expérimentaux.

CONTEXTE

Afin de quantifier les performances de différents types de chaussures de sport en terme d'amortissement du choc, on élabore un modèle permettant de prévoir l'évolution de la force de réaction du sol lors de la phase d'appui du pied du coureur. Un modèle relativement simple mettant en œuvre des éléments de type masse-ressort-amortisseur est présenté sur la **Figure 3** :

.

- La partie inférieure du corps y est modélisée par une masse m_1 qui représente les parties rigides du corps (os) et m_2 pour ce que l'on appelle les masses molles (muscles, viscères, etc…).
- De même, la partie supérieure est modélisée par une masse m_3 pour les parties rigides et une masse m_4 pour les masses molles. Ces différents éléments sont couplés par des ressorts de raideur k_i et des amortisseurs de coefficients c_i.
- La chaussure est modélisée par une masse m_5 en liaison avec la partie inférieure du corps par un ressort non linéaire ayant pour paramètres caractéristiques 2 constantes (k_6, b_6) et un amortisseur c_6 : un modèle de type visco-élastique est donc adopté pour décrire le comportement de la chaussure.

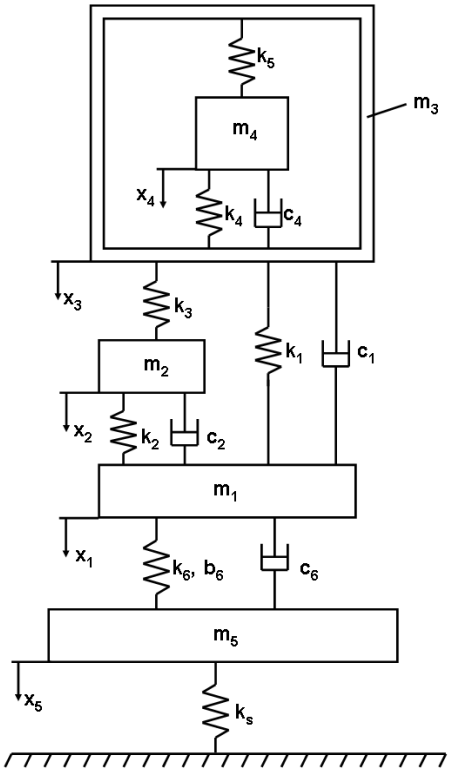


Figure 3: modèle masses-ressorts-amortisseurs.

Les paramètres caractéristiques (k_6, b_6 et c_6) associés à ce modèle sont identifiés à partir de relevés expérimentaux obtenus en conduisant des essais mécaniques d'impact (**Figure 4**). Selon la norme ASTM F1614-99, un pénétrateur sphérique impose un effort périodique sur la semelle de la chaussure qui se déforme sous l'effet de ce chargement. On relève durant l'essai, l'évolution de l'écrasement de la semelle (déplacement **δ**) en fonction de l'effort **F** appliqué au cours d'un cycle de charge/décharge. Un cycle caractéristique de charge/décharge est représenté sur la **Figure 5**.

Figure 4: essai mécanique d'impact

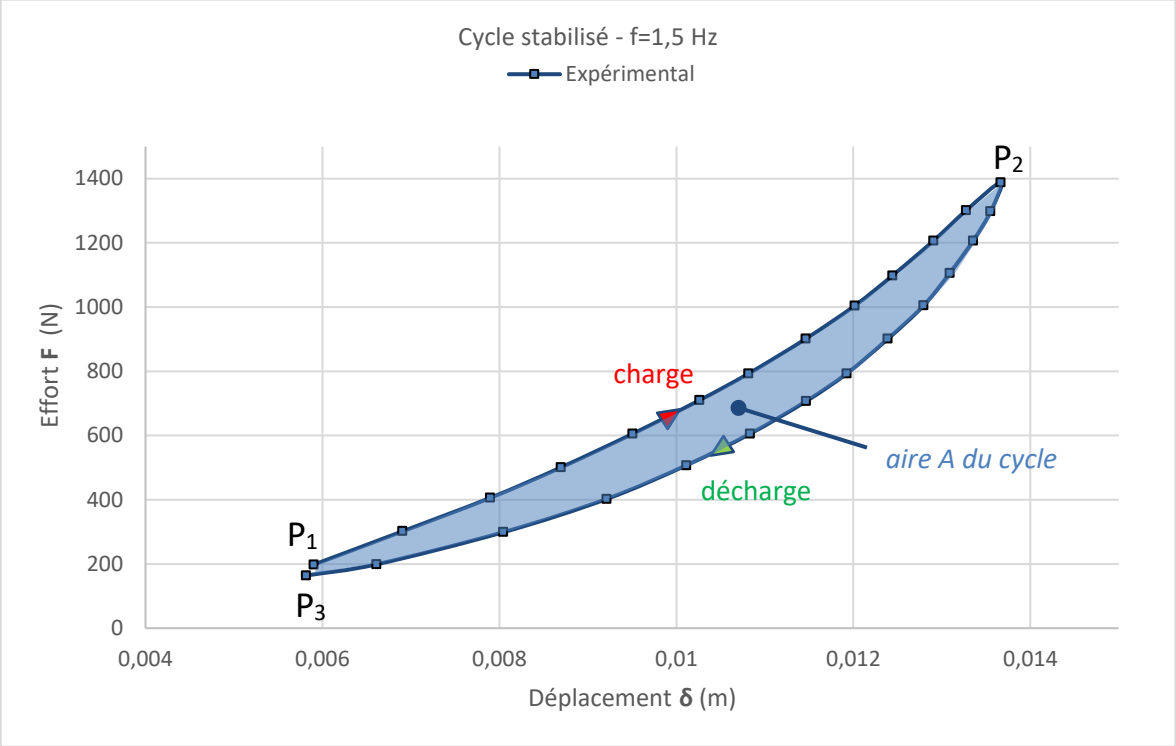


Figure 5: cycle de charge/décharge relevé au cours de l'essai

Les paramètres mécaniques k_6, b_6 et c_6 sont identifiés à partir de cette courbe caractéristique en adoptant la démarche suivante :

- une régression sur l'ensemble du cycle avec une loi en puissance du type $\hat{F}(\delta) = a.\delta^b$ permet de déterminer les paramètres élastiques k_6 et b_6 ,
- la détermination de l'aire A associée au cycle de charge-décharge par intégration numérique permet de déterminer l'énergie dissipée au cours d'un cycle à partir de laquelle on estime l'amortissement c_6.

Cette partie vise donc à mettre en place les outils numériques nécessaires à l'identification des paramètres mécaniques k_6, b_6 et c_6.

REGRESSION DU CYCLE PAR UNE LOI EN PUISSANCE

Au cours d'un cycle stabilisé de l'essai mécanique, on procède à l'acquisition d'un ensemble de N mesures $\{\delta_i, F_i\}, i = 1, \dots, N$ (celles qui ont permis le tracé de la **Figure 5**). On souhaite à présent associer un modèle mathématique à cet ensemble de données ; pour des raisons physiques, le choix est fait ici d'adopter une loi en puissance du type $\hat{F}(\delta) = a.\delta^b$. L'objectif de la régression est de **déterminer les paramètres du modèle** mathématique $\hat{F}(\delta)$ qui représente au mieux ces données (**Figure 6**).

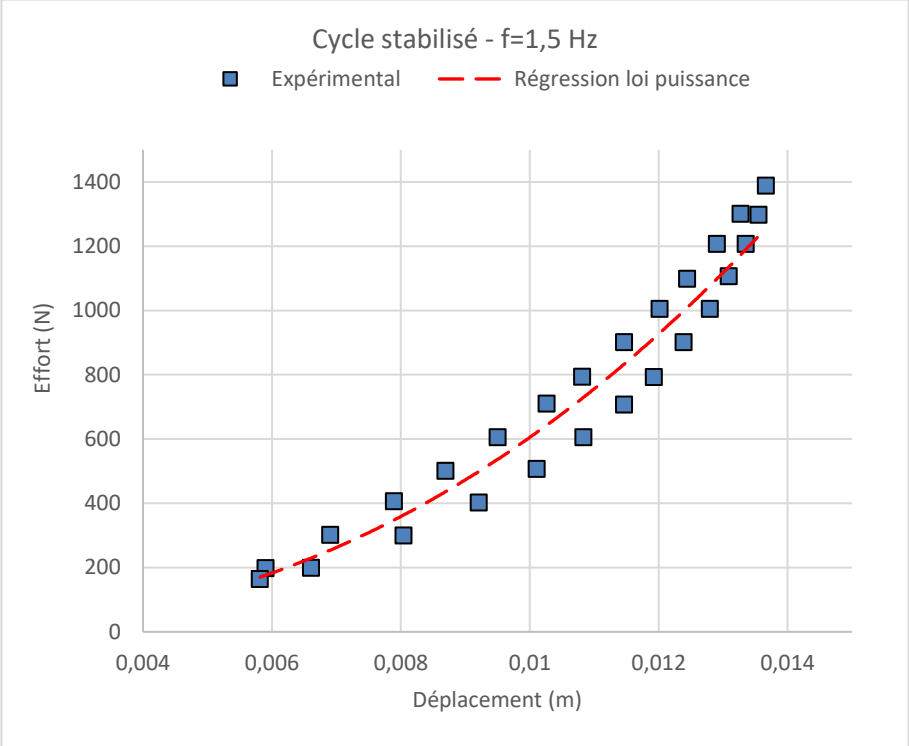


Figure 6 : régression par une loi en puissance

Le critère retenu pour la détermination des paramètres du modèle mathématique est le critère des moindres carrés qui vise à minimiser la somme quadratique des écarts W entre la mesure et le modèle mathématique :

$$W = \sum_{i=1}^{N} \varepsilon_i^2 = \sum_{i=1}^{N} \left(F_i - \hat{F}(\delta_i)\right)^2$$

La régression par une loi du type $\hat{F}(\delta) = a.\delta^b$ s'effectue ici en remarquant que :

$$\log\left(\hat{F}(\delta)\right) = \log(a) + b\ \log(\delta)$$

Ainsi, en effectuant le changement de variables $v = \log(F)$ et $u = \log(\delta)$, on se ramène à déterminer l'équation de la droite de régression de v en u. En posant $K_1 = \log(a)$ et $K_2 = b$ la méthode des moindres carrés conduit aux expressions suivantes :

$K_2 = \dfrac{\sum_{i=1}^{N} \log(\delta_i).\log(F_i) - \dfrac{\sum_{i=1}^{N}\log(\delta_i).\sum_{i=1}^{N}\log(F_i)}{N}}{\sum_{i=1}^{N}\left(\log(\delta_i)\right)^2 - \dfrac{\sum_{i=1}^{N}\left(\log(\delta_i)\right)^2}{N}}$	$K_1 = \dfrac{\sum_{i=1}^{N}\log(F_i)}{N} - K_2\dfrac{\sum_{i=1}^{N}\log(\delta_i)}{N}$

Q13. *Proposer le script d'une fonction* `regression_puissance(d,F)` *recevant en arguments d'entrée les listes contenant les valeurs expérimentales du déplacement* δ_i *et de l'effort* F_i *et renvoyant les variables* `k1` *et* `k2` *associées aux coefficients* K_1 *et* K_2 *de la régression avec une loi puissance.*

```
def regression_puissance(d,F):

    A,B,C,D=0,0,0,0

    l=len(F)

    for i in range(l):
        A=A+np.log10(d[i])
        B=B+np.log10(F[i])
        C=C+np.log10(d[i])*np.log10(F[i])
        D=D+pow(np.log10(d[i]),2)

    k2=(C-(A*B)/l)/(D-D/l)
    k1=(B/l)-k2*(A/l)

    return [k1,k2]
```

Q14. *Rédiger les lignes de code permettant d'obtenir la liste* `f_reg` *des valeurs* $\hat{F}(\delta_i)$ *de la régression (on fera appel à la fonction* `regression_puissance`*).*

```
def f_reg(d,F):

    l=len(F)
    k1,k2=regression_puissance(d,F)

    F_reg=[]

    for i in range(l):
        F_reg=np.append(F_reg,np.exp(k1)*pow(d[i],k2))

    return F_reg
```

Afin de caractériser la qualité de la régression on s'intéresse à certaines propriétés statistiques des résidus ε_i. Les méthodes qui doivent être mises en œuvre sont relativement complexes, nous nous contenterons ici de calculer la moyenne des résidus et la variance.

On rappelle que la moyenne est : $\bar{\varepsilon} = \frac{\sum_{i=1}^{N} \varepsilon_i}{N}$ et la variance : $\sigma_\varepsilon = \frac{\sum_{i=1}^{N} (\varepsilon_i - \bar{\varepsilon})^2}{N}$

Q15. *Proposer le script d'une fonction* `moyenne()` *dont on précisera les arguments, permettant de calculer la moyenne* $\bar{\varepsilon}$.

```
def moyenne(d,F):

    l=len(F)

    F_reg=f_reg(d,F)

    somme=0

    for i in range(l):
        somme=somme+(F[i]-F_reg[i])

    moyenne=somme/l

    return moyenne
```

Q16. *Proposer le script d'une fonction* `variance()` *dont on précisera les arguments, permettant de calculer la variance* σ_ε

```
def variance(d,F):

    l=len(F)

    F_reg=f_reg(d,F)

    somme=0
    moy=moyenne(d,F)

    for i in range(l):
        somme=somme+pow((F[i]-F_reg[i])-moy,2)

    variance=somme/l

    return variance
```

Cette première étape conduit à la détermination des paramètres élastiques du modèle.

DETERMINATION DE L'AIRE ASSOCIEE A UN CYCLE

On s'intéresse à présent l'estimation du paramètre de viscosité. Le cycle d'essai est constitué de deux phases (Figure 5) :

- une phase de charge, qui suit le trajet allant du point P_1 au point P_2, où la structure emmagasine une énergie de déformation W_{12},
- une phase de décharge, qui suit le trajet allant du point P_2 au point P_3, où la structure restitue une partie de l'énergie emmagasinée au cours de la phase de charge (W_{23}).

Ces différentes quantités énergétiques peuvent être estimées par l'intégration numérique des travaux élémentaires $F.d\delta$.

On rappelle que les mesures expérimentales sont contenues dans deux listes :

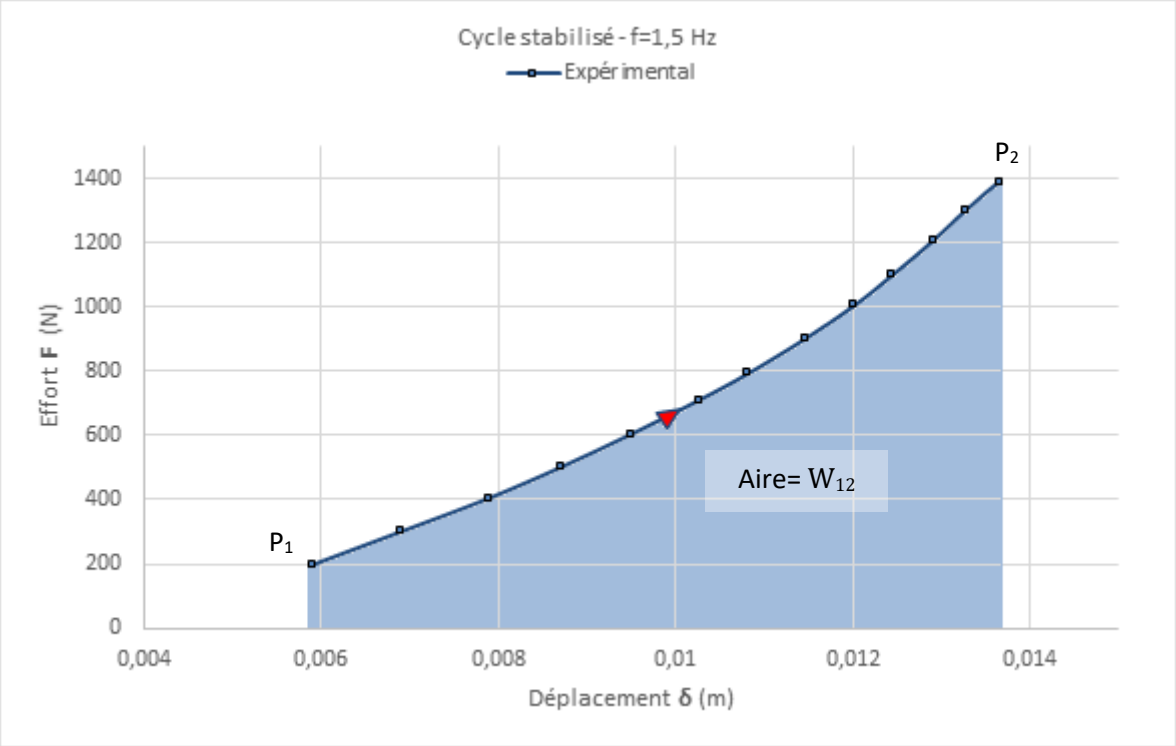
- La liste `d`, pour les valeurs des déplacements δ_i
- La liste `F`, pour les valeurs des efforts F_i

Q17. *Rédiger les lignes de code permettant d'obtenir* `i_max`*, entier associé à l'index (ou position) dans les listes* `d` *et* `F` *des mesures correspondant au point* P_2 *marquant la fin du cycle de charge. On pourra faire appel à la fonction* `maximum` *codée en partie* ***I****.*

```
def i_max(d,F):

    d_max,F_max=maximum(d,F)

    i=0

    while F[i]<F_max:
        i=i+1
        print(i)

    i_max=i

    return i_max
```

Q18. *Proposer une méthode permettant d'estimer numériquement l'énergie emmagasinée W_{12} et l'énergie restituée W_{23}. Illustrer vos explications par les schémas qui vous semblent nécessaires.*

Q19. *Proposer le script d'une fonction* `calcul_energie(d,F,i_max)` *prenant en arguments d'entrée les listes contenant les valeurs expérimentales du déplacement δ_i, de l'effort F_i, l'index* `i_max` *et renvoyant une liste* `(w12,w23)` *associée aux énergies emmagasinée et restituée.*



L'aire sous la courbe de P_1 à P_2 est donc l'énergie emmagasinée W_{12}. De même, l'aire sous la courbe de P_2 à P_3 est égale à W_{23}.

On peut utiliser plusieurs méthodes numériques pour trouver l'aire sous la courbe. Pour une série de données, on ne peut utiliser que la méthode des rectangles à gauche, des rectangles à droite ou celle des trapèzes. La méthode des trapèzes est pour un coût identique, celle qui s'approche le plus de la solution réelle.

```
def calcul_energie(d,F,i_max):

    l=len(F)

    W12_gauche,W12_droite,W12_trapeze=0,0,0
    W23_gauche,W23_droite,W23_trapeze=0,0,0

    for i in range(i_max):
        W12_gauche=W12_gauche+F[i]*(d[i+1]-d[i])
        W12_droite=W12_droite+F[i+1]*(d[i+1]-d[i])
        W12_trapeze=W12_trapeze+((F[i+1]+F[i])/2)*(d[i+1]-d[i])

    for i in range(i_max,l-1):
        W23_gauche=W23_gauche+F[i]*(d[i+1]-d[i])
        W23_droite=W23_droite+F[i+1]*(d[i+1]-d[i])
        W23_trapeze=W23_trapeze+((F[i+1]+F[i])/2)*(d[i+1]-d[i])

    return [W12_gauche,W12_droite,W12_trapeze,W23_gauche,W23_droite,W23_trapeze]
```

Il est ainsi possible de calculer l'énergie dissipée (W_{12}-W_{23}), à partir de laquelle est estimé le paramètre de viscosité c_6 du modèle…

COMPARAISON DES PERFORMANCES ENERGETIQUES DE DIFFERENTS MATERIAUX

On souhaite mener une étude comparative pour différents matériaux, basée sur un critère de performance énergétique.
Suite à une série d'essais sur 5 matériaux différents, on dispose de :

- 5 listes : `d1, d2, d3, d4, d5,` pour les valeurs des déplacements δ_i
- 5 listes : `F1, F2, F3, F4 ,F5 ,` pour les valeurs des efforts F_i relevées pour chaque matériau.

Q20. *Rédiger les lignes de code permettant de construire une liste de liste* `perf_energ` *contenant pour chaque matériau un « triplet » : [énergie emmagasinée, énergie dissipée, rapport énergie dissipée/énergie emmagasinée].*

```
def calcul_energie(d,F,i_max):

    l=len(F)

    W12,W23=0,0

    for i in range(i_max):
        W12=W12+((F[i+1]+F[i])/2)*(d[i+1]-d[i])

    for i in range(i_max,l-1):
        W23=W23+((F[i+1]+F[i])/2)*(d[i+1]-d[i])

    rapport=W23/W12
    item_energ=[W12,W23,rapport]
    return item_energ

perf_energ=[0,0,0,0,0]
perf_energ[0]=perf_energ,calcul_energie(d1,F1,i_max1)
perf_energ[1]=perf_energ,calcul_energie(d2,F2,i_max2)
perf_energ[2]=perf_energ,calcul_energie(d3,F3,i_max3)
perf_energ[3]=perf_energ,calcul_energie(d4,F4,i_max4)
perf_energ[4]=perf_energ,calcul_energie(d5,F5,i_max5)
```

Sur les figures ci-dessous, une idée ce à quoi l'on peut arriver avec tout ça…

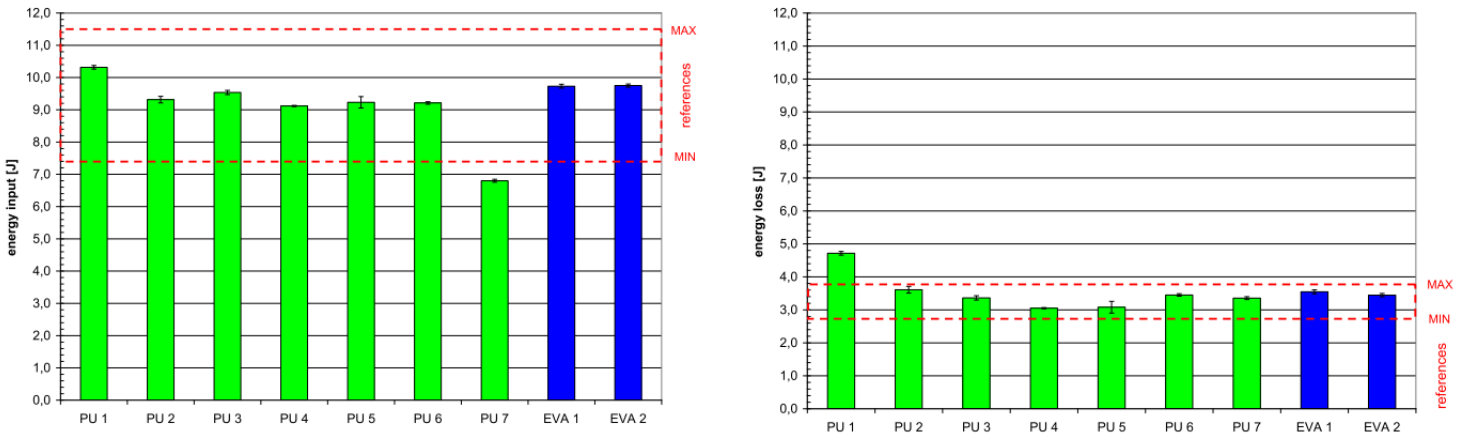


Figure 7 : énergie emmagasinée (à gauche) et énergie dissipée (à droite) pour différents matériaux
(extrait d'une étude sur les semelles à base de mousses polyuréthanes PU comparées au matériau traditionnel EVA éthylène-acétate de vinyle)

Document réponses

$Q1$

Script ***Python*** du programme lecture_fichier	Commentaires
12 def **lecture_fichier**(mesures): 13 14 mon_fichier = open("mesures.txt", "r") 15 16 MatDeMes=[] 17 18 for L in mon_fichier: 19 20 a=L.split() 21 22 i=0 23 24 for e in a: 25 26 a[i]=float(e) 27 28 i+=1 29 30 MatDeMes.append(a) 31 32 MatDeMes=np.array(MatDeMes) 33 34 mon_fichier.close() 35 36 return(MatDeMes)	L12 : Le mot clé « def » définit une fonction. Il est placé avant le nom de la fonction, ici « lecture_fichier ». Def est suivi d'une séquence d'instructions indentées. L14 : Permet de lire ("r" pour reading) le fichier mesures.txt et d'affecter à la variable mon_fichier les données de ce fichier. L16 : initialise une liste vide nommée MatDeMes. L18 : pour effectuer une itération sur les objets contenus dans mon_fichier L20 : La fonction split fracture la chaîne au niveau des espaces L22 : initialisation de l'index i à 0 L24 : idem L 18 L26 : la chaine de caractères est typée float puis remis en place dans la même liste a. L28 : incrémentation de l'index i d'une unité. L30 : concaténation de la liste de float dans la liste MatDeMes L32 : MatDeMes est typée tableau. L34 : Le fichier qui était ouvert de nom mon_fichier est fermé. L36 : retourne le tableau de flottants.

$Q3$

Script ***Python*** de la fonction $filtrage(m)$

```
def filtrage(m):

    l=len(m)

    s=zeros(l,dtype=float)                  # initialisation
    b=[0.018,0.054,0.054,0.018]             # coef. du filtre : numérateur
    a=[1,-1.760,1.183,-0.278]               # coef. du filtre : dénominateur
    for i in range(l):                      # opération de filtrage

        s[i]=b[0]*m[i]+b[1]*m[i-1]+b[2]*m[i-2]+b[3]*m[i-3]-a[1]*s[i-1]-a[2]*s[i-2]-a[3]*s[i-3]

    return s
```