Conditions d'optimisation des semelles

Sidibé Kassim Junior 11667

Positionnement thématique
Physique (Mécanique, Physique ondulatoire)
Informatique(Informatique pratique)

2023-2024

Sommaire

Sommaire

Sommaire

- Introduction
- Problématique
- Objectifs
- Étude théorique de la modélisation de la semelle en association de ressorts
- Étude théorique de modélisation de la semelle une poutre
- Étude expérimentale
- Conclusion

Sommaire

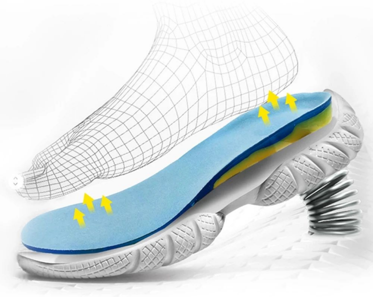
Sommaire

- Introduction
- Problématique
- Objectifs
- Étude théorique de la modélisation de la semelle en association de ressorts
- Étude théorique de modélisation de la semelle une poutre
- Étude expérimentale
- Conclusion

Sommaire

Sommaire

- Introduction
- Problématique
- Objectifs
- Étude théorique de la modélisation de la semelle en association de ressorts
- Étude théorique de modélisation de la semelle une poutre
- Étude expérimentale
- Conclusion



a)Une semelle

b) Une course de marathon

Mise en situation

Problématique

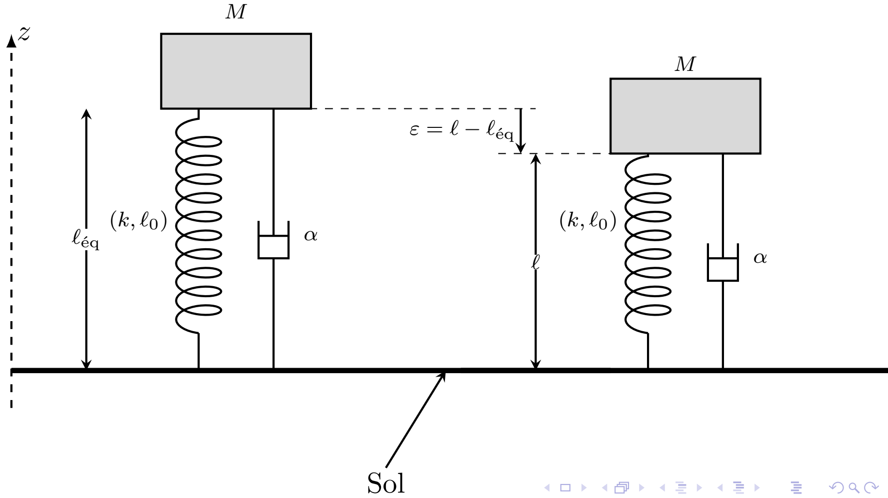
Comment faut-il agir sur les paramètres d'une semelle pour améliorer l'amortissement des chocs dûs aux contacts avec le sol tout en conservant la stabilité de la chaussure afin d'optimiser le confort des sportifs?

Objectifs

- Déterminer les conditions sur les paramètres sur lequels agir afin d'amortir le plus vite possible les chocs et en conservant la stabilité de la semelle.
- Déterminer les conditions sur les paramètres sur lequels agir afin d'augmenter de la vitesse de dissipation de l'énergie transmisse suite au choc avec le sol.

- Déterminer les conditions sur les paramètres sur lequels agir afin d'amortir le plus vite possible les chocs et en conservant la stabilité de la semelle.
- Déterminer les conditions sur les paramètres sur lequels agir afin d'augmenter de la vitesse de dissipation de l'énergie transmisse suite au choc avec le sol.

Modélisation d'une semelle par la résultante de l'association en parallèle de n ressorts identiques en parallèle



Bilan des actions mécaniques exterieures s'exerçant sur le ressort équivalent:

- Force de rappel: $\vec{T} = -k(\ell - \ell_0)\vec{e}_z, \quad \ell = z$
- Force de pesanteur: $\vec{P} = M\vec{g}$
- Force de frottement: $\vec{f} = -\alpha\dot{z}\vec{e}_z$
- Force d'impulsion: $\vec{F} = F_0\delta(t)\vec{e}_z$

$$avec\ \delta(t) = \begin{cases} 1 & \text{si} t \in [t_0, t_0 + \Delta t] \\ 0 & \text{sinon} \end{cases}$$

D'après le principe fondamental ,nous avons:

$$M\vec{a} = -k(z - \ell_0)\vec{e}_z - \alpha\dot{z}\vec{e}_z + F_0\delta(t)\vec{e}_z + M\vec{g}$$

L'équation differentielle du mouvement du ressort autour de sa position d'équilibre est donnée par:

$$M\ddot{\varepsilon} + \alpha\dot{\varepsilon} + k\varepsilon = F_0\delta(t) \qquad (1)$$

En utilisant la transformée de Laplace avec les conditions initiales nulles, nous obtenons:

$$\varepsilon(p) = \frac{\frac{F_0}{M}}{p^2 + 2\beta\omega_0 p + \omega^2} = \frac{\frac{F_0}{M}}{(p + \beta\omega_0)^2 + \omega_0^2(1 - \beta^2)} \qquad (2)$$

$$\text{avec } \omega_0 = \sqrt{\frac{k}{M}} \text{ et } \beta = \frac{\alpha}{2\sqrt{kM}}$$

En utilisant la transformée de Laplace inverse, nous avons finalement:

$$\varepsilon(t) = \frac{F_0}{M\omega_0\sqrt{1 - \beta^2}} \exp\left(-\beta\omega_0 t\right)\sin(\omega_0 t) \qquad (3)$$

$$E_m = E_c + E_{pp} + E_{pe}$$

- $E_c = \frac{1}{2} M \dot{z}^2$
- E_{pp}=Mgz
- $E_{pe} = \frac{1}{2} k(z - \ell_0)^2$

D'après le théorème de la puissance mécanique, nous obtenons:

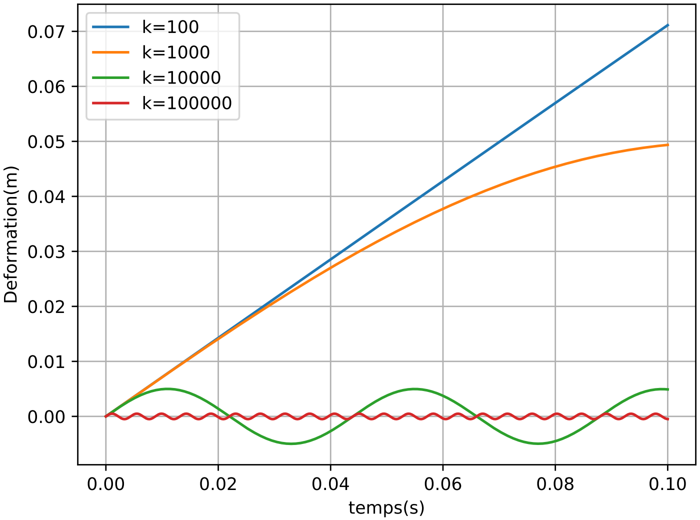
$$\dot{E_m} = -\alpha \dot{z}^2 + F_0 \delta(t) \dot{z}$$

- $\dot{E_m}$: Vitesse de variation de l'énergie mécanique
- $-\alpha \dot{z}^2$: Puissance due aux forces de frottement
- $F_0 \delta(t) \dot{z}$: Puissance de forçage causé par la force d'impulsion

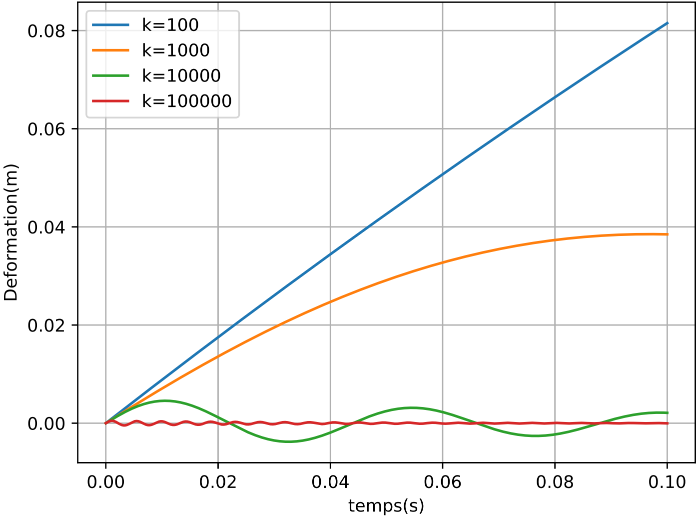
$$\dot{z} = \frac{F_0}{M\omega_0\sqrt{1-\beta^2}} \exp\left(-\beta\omega_0 t\right)(cos(\omega t) - \beta sin(\omega t)) \tag{4}$$

$$\dot{E_m} = -\alpha \dot{z}^2 + F_0 \dot{z} \tag{5}$$

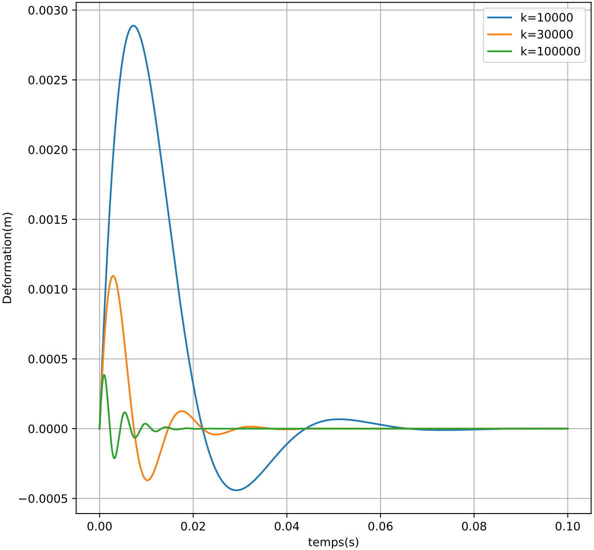
Représentation graphique: Deformation. Cas 1: $\alpha = 1 N.s.m{-1}$



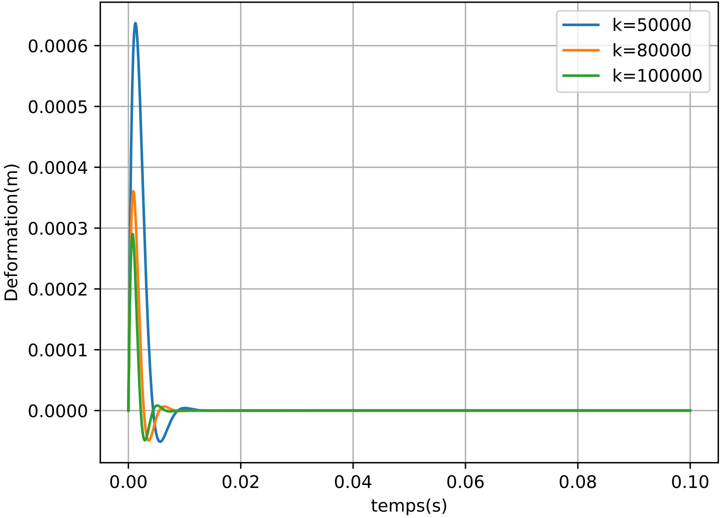
Cas 2: $\alpha = 100 N.s.m{-1}$



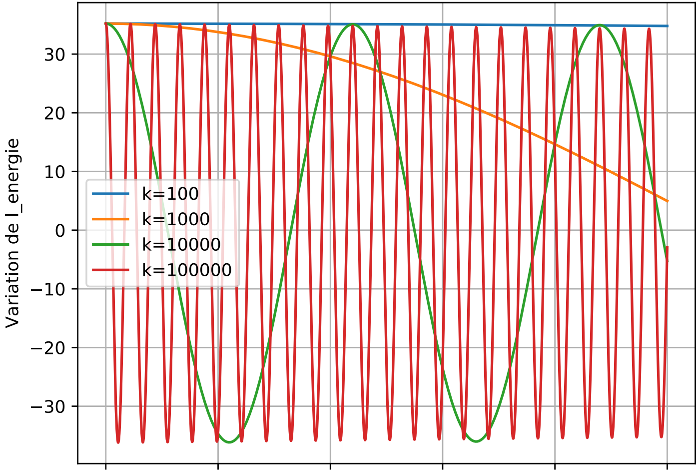
Cas 3: $\alpha = 1000 N.s.m-1$



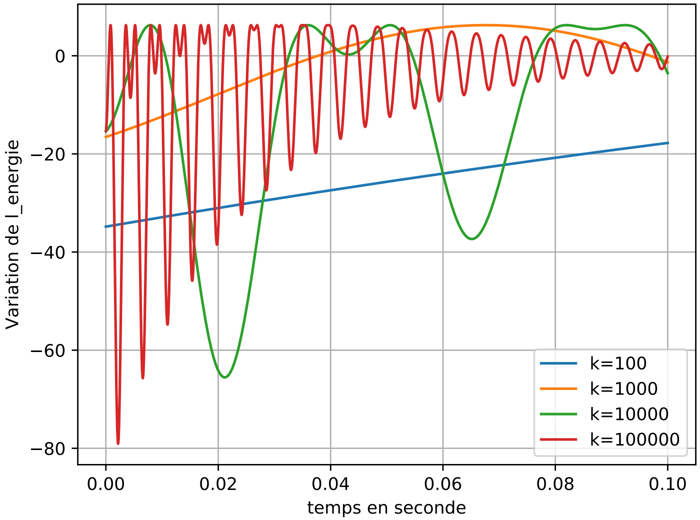
Cas 4: $\alpha = 3000 N.s.m{-1}$



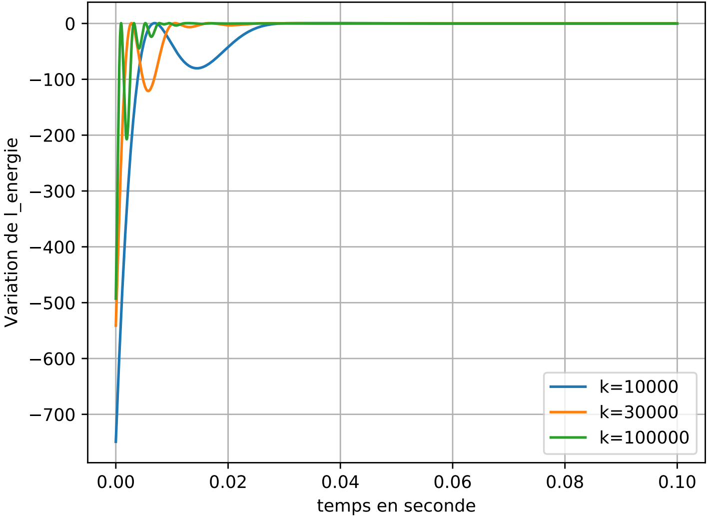
Représentation graphique de la dissipation d'energie.
Cas 1: $\alpha = 1 N.s.m{-}1$



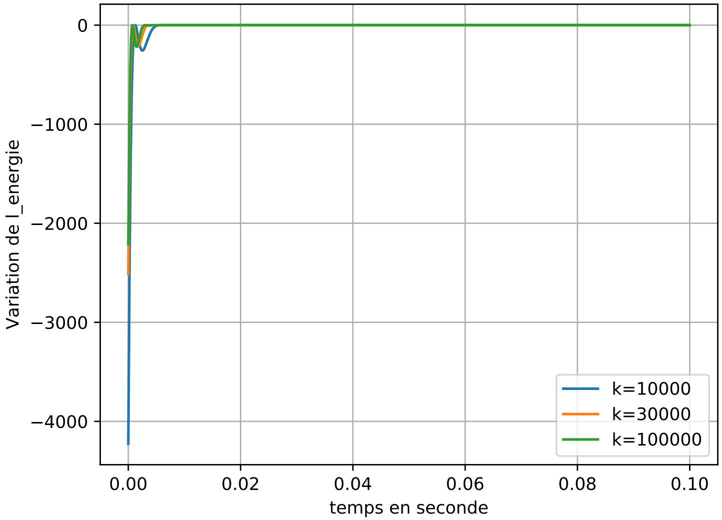
Cas 2: $\alpha = 100 N.s.m{-}1$



Cas 3: $\alpha = 100 N.s.m{-}1$



Cas 4: $\alpha = 3000 N.s.m{-1}$



Conclusion tirée de la prière approche théorique

- Contrainte: $\alpha < 2\sqrt{kM}$
- Plus k est grand, plus on a d'oscillations autour de la position d'équilibre et la vitesse de dissipation de l'energie transmisse prend du temps à s'annuler.
- Plus α est grand, plus les oscillations sont vite amorties et la vitesse de dissipation de l'energie transmisse augmente.
- Par conséquent afin d'atteindre notre objectif, il faut maximiser α et minimiser k le plus possible tout en s'assurant de respecter les contraintes de constructions et de ne pas rendre la semelle trop ridige.

On en déduit alors que le confort d'une chausure repose sur le choix du materiau utilisé pour faire la semelle.

Loi de Hooke

$F = ES\frac{\Delta l}{\ell_0}$

E: Module de Young

S: La section du solide

$\Delta\ell$: La déformation

ℓ_0: la longueur à vide du solide

On considère une tranche de la poutre entre z et $z + dz$.
Hypothèses:

La masse est repartie uniformement.

La masse volumique est uniforme.

Bilan des forces extérieures:

- $\vec{f} = -\alpha S\dot{\varepsilon}(z,t)\vec{e_z}$
- $\vec{F} = (F(z+dz,t) - F(z,t))\vec{e_z}$

D'après le principe fondamental de la dynamique , on a :

$$\delta m\vec{a} = \vec{F} + \vec{f} \tag{6}$$

$$\frac{\partial^2\varepsilon}{\partial t^2} + \frac{\alpha}{\mu}\frac{\partial\varepsilon}{\partial t} - \frac{E}{\mu}\frac{\partial\varepsilon}{\partial z} = 0 \tag{7}$$

On recherche une solution de la forme: $\varepsilon(z,t) = f(z)g(t)$
On considère $g(t) = g_0 \exp(jwt)$
Ainsi on a l'équation suivante verifiée par f :

$$\frac{df(z)}{dz} - \frac{w}{E}(j\alpha - \mu w)f(z) = 0 \tag{8}$$

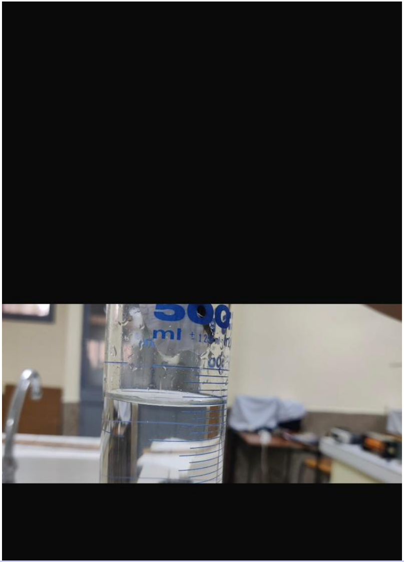
Comme solution nous avons

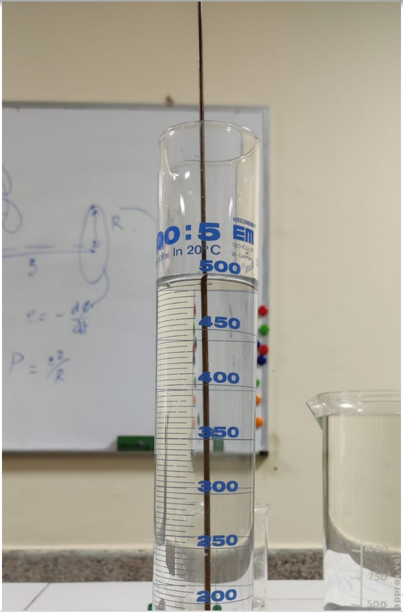
$$f(z) = f_0 \exp(\frac{-\mu w^2}{E}z)exp(j\frac{w\alpha}{E}z) \tag{9}$$

Solution générale

$$\epsilon(z,t) = \epsilon_0 \exp(-\frac{\mu w}{E} z) \cos(w(t + \frac{\alpha}{E} z)) \qquad (10)$$

Experience


ml


In 20°C
500
450
400
350
300
250
200

Données sur la barre de fer

$Diamètre = 2mm$

$Longueur\ \ plongée = 24,8cm \pm 0.1cm$

Données sur le morçeau de la semelle

$Volume(v) = 24,22ml \pm 1,25ml$

$m = 5,53g \pm 0.01g$

$v_{lim} =$

$\mu = 0.23.10^3 kg.m^{-3}$

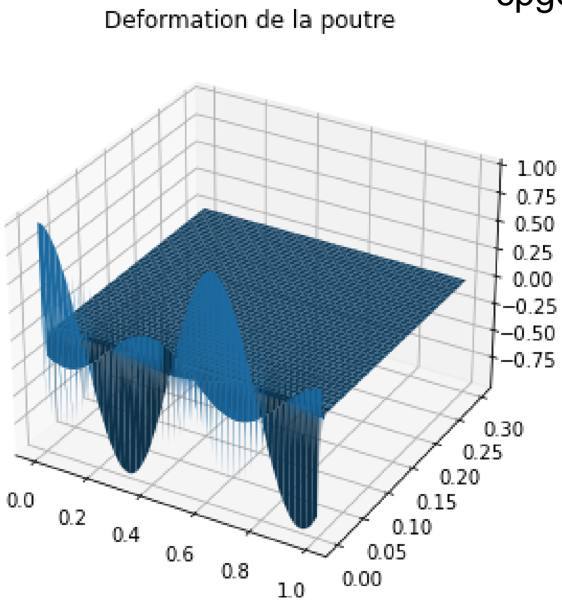
$\alpha = \frac{(\mu_{eau}v - m)g}{V_{lim}}$

$\alpha = 1408.94N.s.m^{-1}$

Donnée générale

- $\mu_{eau} = 1000g.l^{-1}$
- $g = 9.8N.kg^{-1}$

Test avec des données réelles



Conclusion

On constate que la deformation est fortement attenuée si :

- La masse volumique du materiau de base de la semelle est assez elevé.
- Le module de Young est très petit.
- Si la frequence de l'onde est tres grande

Ainsi on conclut, des deux modélisations,que pour assurer des attenuations rapides est deformations dues aux chocs avec le sol afin de conserver la stabilité d'une chaussure, il faut utiliser des materiaux adéquats.

MERCI POUR VOTRE ATTENTION.

Annexe : Code python de la representation python de la deformation


```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=1
k=100000
def deformation(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    return F0/(M*omega*sqrt(1-beta**2)) *np.exp(-beta*omega*t)*np.sin(omega*t)

t=np.linspace(0,10**(-1),10000000)
plt.xlabel('temps(s)')
plt.ylabel('Deformation(m)')
for k in (100,1000,10000,100000):
      y=deformation(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=100','k=1000','k=10000','k=100000'])
plt.show()
```

1

```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=100
def deformation(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    return F0/(M*omega*sqrt(1-beta**2)) *np.exp(-beta*omega*t)*np.sin(omega*t)

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps(s)')
plt.ylabel('Deformation(m)')
for k in (100,1000,10000,100000):
      y=deformation(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=100','k=1000','k=10000','k=100000'])
plt.show()
```

2

Annexe : Code python de la representation python de la deformation


```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=1000
def deformation(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    return F0/(M*omega*sqrt(1-beta**2)) *np.exp(-beta*omega*t)*np.sin(omega*t)

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps(s)')
plt.ylabel('Deformation(m)')
for k in (10000,30000,100000):
      y=deformation(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=10000','k=30000','k=100000'])
plt.show()
```

3

```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=3000
def deformation(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    return F0/(M*omega*sqrt(1-beta**2)) *np.exp(-beta*omega*t)*np.sin(omega*t)

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps(s)')
plt.ylabel('Deformation(m)')
for k in (50000,80000,100000):
      y=deformation(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=50000','k=80000','k=100000'])
plt.show()
```

4

Annexe : Code python de la representation python de la dissipation de l'energie mécanique


```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=1
def vitesse_de_dissipation_E(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    Zpoint=F0/(M*sqrt(1-beta**2))*np.exp(-beta*omega*t)*(np.cos(omega*t)-beta*np.sin(omega*t))
    return -alpha*Zpoint**2+F0*Zpoint

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps(s)')
plt.ylabel('Variation de l_energie')
for k in (100,1000,10000,100000):
      y=vitesse_de_dissipation_E(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=100','k=1000','k=10000','k=100000'])
plt.show()
```

5

```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=100
def vitesse_de_dissipation_E(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    Zpoint=F0/(M*sqrt(1-beta**2))*np.exp(-beta*omega*t)*(np.cos(omega*t)-beta*np.sin(omega*t))
    return -alpha*Zpoint**2+F0*Zpoint

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps en seconde')
plt.ylabel('Variation de l_energie')
for k in (100,1000,10000,100000):
      y=vitesse_de_dissipation_E(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=100','k=1000','k=10000','k=100000'])
plt.show()
```

6

Annexe : Code python de la representation python de la dissipation de l'energie mécanique


```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=1000
def vitesse_de_dissipation_E(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    Zpoint=F0/(M*sqrt(1-beta**2))*np.exp(-beta*omega*t)*(np.cos(omega*t)-beta*np.sin(omega*t))
    return -alpha*Zpoint**2+F0*Zpoint

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps en seconde')
plt.ylabel('Variation de L_energie')
for k in (10000,30000,100000):
      y=vitesse_de_dissipation_E(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=10000','k=30000','k=100000'])
plt.show()
```

7

```python
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt
F0=50
M=70
alpha=3000
def vitesse_de_dissipation_E(alpha,k,t):
    beta=alpha/(2*sqrt(k*M))
    omega=k/M
    Zpoint=F0/(M*sqrt(1-beta**2))*np.exp(-beta*omega*t)*(np.cos(omega*t)-beta*np.sin(omega*t))
    return -alpha*Zpoint**2+F0*Zpoint

t=np.linspace(0,10**(-1),10000)
plt.xlabel('temps en seconde')
plt.ylabel('Variation de L_energie')
for k in (50000,80000,100000):
      y=vitesse_de_dissipation_E(alpha,k, t)
      plt.plot(t,y)
plt.legend(['k=10000','k=30000','k=100000'])
plt.show()
```

8

Les forces qui s'exercent sur le morçeau de semelle plongée dans l'eau sont:

- Le poids: $\vec{P} = -mg\vec{u}_z$
- La poussée d'archimède $\vec{\pi} = \mu_{eau} v_{deplae} g\vec{u}_z$
- La force de frottement visqueux: $\vec{F}_f = -\alpha\vec{v}$

D'après le principe fondamental de la dynamique on a:

$$m\vec{a} = \vec{P} + \vec{\pi} + \vec{F}_f \tag{11}$$

Annexe : Determination de α

$$\dot{\vec{v}} + \frac{\alpha}{m}\vec{v} = (\frac{\mu_{eau} V}{m} - 1) g \vec{u}_z \quad (12)$$

Donc $\vec{v} = \vec{A}\exp(-\frac{\alpha}{m}t) + \frac{(\mu_{eau} V - m)g}{\alpha}\vec{u}_z$
avec $\vec{A}$ une constante vectorielle
Lorque t tend vers $+\infty$ on a : $V_{lim} = \frac{(\mu_{eau} V - m)g}{\alpha}$
Donc $\alpha = \frac{(\mu_{eau} V - m)g}{v_{lim}}$
Grace au logiciel *Latispro* , on obtient $v_{lim} = 0.13 m.s^{-1}$

Annexe : Code python de la representation en 3D de la deformation d'une poutre


```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
E=0.001
mu=0.23*10**(3)
alpha=1408.94
def epsilon(z,t,w):
    return np.exp(-mu*w*z/w)*np.cos(w*(t+alpha*z/E))
ax=Axes3D(plt.figure())
Z,T= np.meshgrid(np.arange(0,0.3,0.001), np.arange(0,1,0.001))
Y=epsilon(Z,T,10)
ax.plot_surface(T,Z,Y)
ax.set_title("Deformation de la poutre")
plt.show()
```