SESSION 2019 **PCMS006**

CONCOURS COMMUN INP

ÉPREUVE SPÉCIFIQUE - FILIÈRE PC

MODÉLISATION DE SYSTÈMES PHYSIQUES OU CHIMIQUES

Jeudi 2 mai : 8 h - 12 h

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

Les calculatrices sont autorisées

Le sujet est composé de trois parties partiellement indépendantes.

Téléchargé gratuitement sur Doc-Solus.fr .

Modélisation du mouvement d'une plateforme en mer

On s'intéresse à la résolution d'équations du mouvement dans une approche classique de la mécanique afin d'étudier le mouvement d'une plateforme en mer. Le modèle envisagé est un système à un degré de liberté considéré comme oscillateur harmonique : une masse est reliée à un ressort, avec ou sans amortissement, et peut être soumise à une excitation externe.

La résolution est tout d'abord abordée de façon analytique puis de façon numérique, avant enfin de comparer les résultats obtenus.

Les résolutions analytiques et numériques sont largement indépendantes.

Dans la suite de l'énoncé, toutes les grandeurs vectorielles sont indiquées en gras.

Un aide-mémoire sur `numpy` *est donné en Annexe.*

On considère le mouvement d'une plateforme en mer soumise à un courant marin. Sa partie supérieure de masse $m = 110$ tonnes est considérée comme rigide et le mouvement principal de la plateforme a lieu suivant x (**figure 1(a)**).

Afin d'étudier le mouvement de cette plateforme, on la représente par une masse m, liée à un ressort de constante de raideur k et à un amortisseur de constante d'amortissement γ, pouvant subir une excitation externe de force $\mathbf{F_{exc}}$, et se déplaçant sur un support (**figure 1(b)**). Le ressort représente la rigidité de l'ensemble du support de la plateforme. L'amortisseur permet de prendre en compte l'effet de l'eau environnante et la force d'excitation externe celui des vagues qui frappent périodiquement la plateforme. La masse est supposée se déplacer selon une seule direction parallèle à l'axe Ox en fonction du temps t.

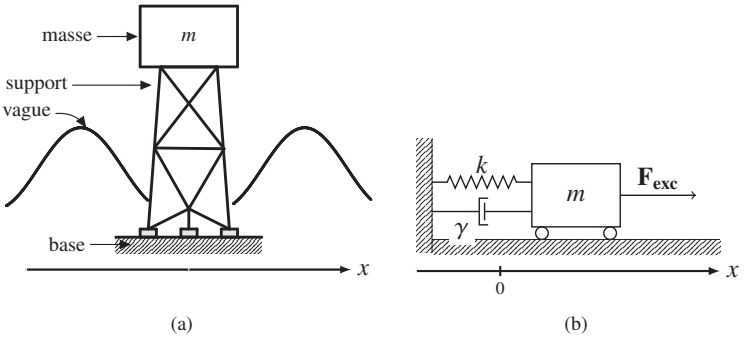

Figure 1 – (a) Plateforme en mer soumise aux vagues marines, (b) système masse (m), ressort (k), amortisseur (γ) et excitation externe ($\mathbf{F_{exc}}$)

Les projections sur l'axe Ox de la position, de la vitesse et de l'accélération de la masse en fonction du temps sont notées respectivement $x(t)$, $\dot{x}(t)$ et $\ddot{x}(t)$. La force totale $\mathbf{F_{tot}}$ agissant sur la masse correspond à la réaction normale $\mathbf{R_N}$ de la base horizontale, à la force de frottement $\mathbf{F_d}$, à la force de rappel $\mathbf{F_k}$ du ressort, au poids $\mathbf{P}$ de la masse et à la force $\mathbf{F_{exc}}$ d'excitation externe. La position d'équilibre de la masse sera choisie à $x = 0$. En l'absence d'action de l'amortisseur, la masse se déplace sur la base horizontale sans frottements.

Téléchargé gratuitement sur Doc-Solus.fr .

Partie I - Résolution analytique et détermination des paramètres pour la modélisation

Q1. En effectuant une projection sur l'axe Ox, montrer que $\mathbf{P}$ et $\mathbf{R_N}$ n'interviennent pas dans le bilan des forces.

I.1 - Ressort sans amortissement et sans excitation

Q2. Démontrer que l'équation du mouvement de la masse correspond à l'équation différentielle du second ordre suivante :

$$m\ddot{x} + kx = 0. \tag{1}$$

Q3. La solution de cette équation prend la forme générale suivante

$$x(t) = A_0 \sin(\omega_0 t) + B_0 \cos(\omega_0 t) \tag{2}$$

avec A_0 et B_0 deux coefficients réels. Exprimer ω_0 en fonction des grandeurs caractéristiques du système et donner sa signification physique. De plus, en remarquant qu'à $t = 0$: $x(t) = x_0$ et $\dot{x}(t) = \dot{x}_0$, déterminer les expressions de A_0 et de B_0 en fonction de x_0, $\dot{x}_0$ et de ω_0.

Q4. On cherche à reformuler l'équation précédente sous une forme plus compacte du type :

$$x(t) = R_0 \cos(\omega_0 t - \phi_0). \tag{3}$$

Donner les expressions de R_0 et de ϕ_0 en fonction de x_0, $\dot{x}_0$ et de ω_0.

Q5. Représenter qualitativement $x(t)$ en fonction de t et indiquer sur le tracé R_0, x_0 et $2\pi/\omega_0$.

Q6. En utilisant les expressions des énergies cinétique $K(t)$ et potentielle $U(t)$ du système, montrer que l'énergie totale $E(t)$ du système est alors :

$$E(t) = \frac{kR_0^2}{2}. \tag{4}$$

Justifier le résultat obtenu.

Q7. Représenter qualitativement $E(t)$, $K(t)$ et $U(t)$ en fonction de t.

I.2 - Ressort avec amortissement et sans excitation

Q8. La force de frottement que l'amortisseur exerce sur la masse est considérée comme linéaire, c'est-à-dire proportionnelle au vecteur vitesse $\mathbf{v}$ de celle-ci : $\mathbf{F_d} = -\gamma\,\mathbf{v}$, avec γ constante d'amortissement (> 0). En considérant une projection sur l'axe Ox, démontrer que la position de la masse en fonction du temps suit l'équation du mouvement ci-après

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0 \tag{5}$$

avec ω_0 défini en question **Q3** et ζ à exprimer en fonction de γ, k et m.

Téléchargé gratuitement sur Doc-Solus.fr .

Q9. Dans le cas où $\zeta < 1$, $x(t)$ prend la forme suivante :

$$x(t) = e^{-\zeta\omega_0 t}\left(A_\mathrm{d}\cos(\omega_\mathrm{d}t) + B_\mathrm{d}\sin(\omega_\mathrm{d}t)\right). \tag{6}$$

Déterminer les deux coefficients réels A_d et B_d en fonction de x_0, $\dot{x}_0$, ζ, ω_0 et $\omega_\mathrm{d} = \omega_0 \cdot \sqrt{1-\zeta^2}$.
On utilisera pour cela les mêmes conditions initiales que celles utilisées en question **Q3**.

Q10. Montrer alors que l'on peut obtenir une forme du type

$$x(t) = R_\mathrm{d} e^{-\zeta\omega_0 t}\cos(\omega_\mathrm{d}t - \phi_\mathrm{d}) \tag{7}$$

avec R_d et ϕ_d à préciser.

Q11. Représenter qualitativement $x(t)$ en fonction de t et indiquer sur le tracé $R_\mathrm{d}e^{-\zeta\omega_0 t}$, x_0 et $2\pi/\omega_\mathrm{d}$.

Q12. Donner l'expression de $E(t)$ et commenter les cas où $\zeta = 0$ et $\zeta = 1$.

Q13. Montrer de façon simple que E est une fonction décroissante de t. À quoi cela est-il dû ?

Q14. On envisage deux temps successifs t_1 et t_2 pour lesquels les déplacements sont x_1 et x_2, tels que $t_2 > t_1$ et $t_2 - t_1 = \tau_\mathrm{d}$, avec τ_d : période des oscillations amorties. En utilisant l'équation (7) et en considérant que $\zeta \ll 1$, montrer que :

$$\ln(x_1/x_2) \approx 2\pi\zeta. \tag{8}$$

Q15. Le relevé du déplacement horizontal de la plateforme en fonction du temps est représenté en **figure 2**.
En utilisant les deux points qui sont indiqués sur la figure, déterminer k, ζ et γ.
Comment ce tracé serait modifié en fonction de la valeur de ζ ?

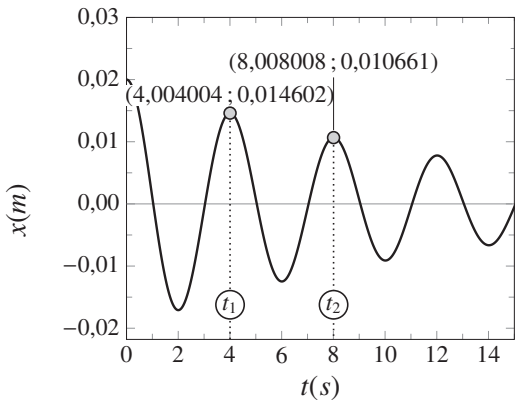

Figure 2 – Relevé du déplacement horizontal x (en m) de la plateforme de masse $m = 110$ tonnes en fonction du temps t (en s). Les deux temps t_1 et t_2 mentionnés en question **Q14** sont indiqués

Téléchargé gratuitement sur Doc-Solus.fr .

I.3 - Ressort avec amortissement et avec excitation

On envisage enfin le cas où le système est soumis à la fois aux effets d'amortissement et d'excitation.

On se limite ici à la réponse à une excitation harmonique sinusoïdale de fréquence ω produite par une force extérieure au système

$$\mathbf{F_{exc}}(t) = F_0 \cos(\omega t)\mathbf{e_x} \tag{9}$$

avec $\mathbf{e_x}$ vecteur unitaire sur l'axe Ox et on se place dans le cas traité précédemment pour l'étude de l'amortisseur, c'est-à-dire $\zeta < 1$ (**I.2**).

On admet de plus dans ce qui suit que la réponse du système dans le cas où amortisseur et excitation sont pris en compte peut s'écrire comme somme de la solution donnée par l'équation (6) et de la contribution due à l'excitation :

$$x_{exc}(t) = X \cos(\omega t - \phi). \tag{10}$$

Q16. Montrer que l'équation différentielle caractérisant le système devient alors :

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = \frac{F_0}{m}\cos(\omega t). \tag{11}$$

Q17. En utilisant l'équation (10) et en privilégiant une représentation complexe, vérifier que :

$$\begin{cases} X = \dfrac{F_0}{m} \cdot \dfrac{1}{\sqrt{(\omega_0^2 - \omega^2)^2 + (2\zeta\omega_0\omega)^2}} \\ \tan\phi = \dfrac{2\zeta\omega_0\omega}{\omega_0^2 - \omega^2} \end{cases} . \tag{12}$$

Q18. Exprimer la grandeur $M = \dfrac{X}{F_0/k}$ en fonction de $r = \omega/\omega_0$ et expliciter le sens physique de M.

Q19. Trouver la condition sur r puis sur ω pour laquelle M est maximale.

Q20. Si l'on considère une période moyenne des vagues en mer de 8 s, que peut-on conclure sur le mouvement de la plateforme ?

Téléchargé gratuitement sur Doc-Solus.fr .

Partie II - Modélisation : codage

On souhaite maintenant obtenir $x(t)$ et $E(t)$ de façon numérique et comparer les résultats obtenus à ceux fournis par les solutions analytiques précédentes pour $x(t)$. On rappelle que $x(t)$ et $E(t)$ représentent respectivement la position de la masse et l'énergie mécanique totale en fonction du temps.

Pour cela, le temps est discrétisé en N points $t = 0, \Delta t, 2\Delta t, \ldots, (N-1)\Delta t$ avec un pas de temps constant Δt. Les $N-1$ pas sont effectués pendant la simulation de durée totale t_{max}. On note respectivement x_{n}, v_{n}, a_{n}, E_{n} et F_{n} les valeurs de $x(t)$, $\dot{x}(t)$, $\ddot{x}(t)$, $E(t)$ et $F_{\text{exc}}(t)$ à $t = n\Delta t$.

À chaque pas, les équations du mouvement reliant $x_{\text{n+1}}$ et $v_{\text{n+1}}$ à x_{n} et v_{n} sont utilisées afin d'obtenir les valeurs de x, v et E. Les conditions initiales x_0 et v_0 sont connues et permettent de démarrer le processus d'intégration numérique. Deux algorithmes distincts (Euler et Leapfrog) vont être utilisés dans la suite.

Pour l'écriture du code, on se place dans le cas le plus général, c'est-à-dire avec amortissement et excitation harmonique externe. Les variables et tableaux suivants sont notamment choisis :

`N`	nombre de points sur l'axe des temps utilisés pendant toute la simulation
`t[]`	tableau des temps (s), de dimension N
`x[]`	tableau des positions (m), de dimension N
`v[]`	tableau des vitesses ($m.s^{-1}$), de dimension N
`E[]`	tableau des énergies totales (J), de dimension N
`F[]`	tableau des forces d'excitation (N), de dimension N
`dt`	pas de temps (s)
`tmax`	temps total de la simulation (s)
`k`	constante de raideur du ressort ($N.m^{-1}$)
`m`	masse du système (kg)
`om0`	ω_0 (s^{-1})
`zeta`	ζ (sans unité)

Tableau 1 – Principaux tableaux et principales variables utilisés pour la résolution numérique

On rappelle qu'un aide-mémoire sur `numpy` est fourni en **Annexe**, page 10.

Q21. Écrire les lignes de code permettant de définir l'entier N. On suppose t_{max} et Δt connus et fixés par l'utilisateur en début de code.

Q22. Écrire alors l'instruction permettant de définir le tableau `t` qui contient toutes les valeurs de t telles que : $0 \leq t \leq t_{\text{max}}$. On rappelle que le temps est discrétisé en N points $t = 0, \Delta t, 2\Delta t, \ldots, (N-1)\Delta t$ avec un pas de temps constant Δt et que $N-1$ pas sont effectués.

Q23. En effectuant des développements de Taylor de x et v tronqués à l'ordre 1, on obtient l'algorithme d'Euler, où x et v sont évalués au même temps t selon le schéma donné **figure 3**, page 7 :

$$\begin{cases} x_{n+1} \approx x_n + v_n \cdot \Delta t \\ v_{n+1} \approx v_n + a_n \cdot \Delta t \end{cases} . \tag{13}$$

Donner les expressions de $x_{\text{n+1}}$ et $v_{\text{n+1}}$ en fonction de x_{n}, v_{n} et F_{n}.

Téléchargé gratuitement sur Doc-Solus.fr .

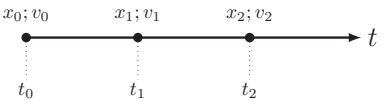

Figure 3 – Discrétisation en temps utilisée dans le cas de l'algorithme d'Euler. Les conditions initiales correspondent à $(x_0; v_0)$

Q24. Écrire la boucle en `i` permettant d'obtenir toutes les valeurs de `x[i+1]`, `v[i+1]` et `E[i+1]` où `i` correspond à un point sur l'axe des temps. On précisera les variables éventuellement introduites en supposant qu'elles ont été définies dans le code.

Q25. Dans l'algorithme de Leapfrog, les x sont évalués aux temps entiers, c'est-à-dire à $t = 0, \Delta t, 2\Delta t, \ldots, (N-1)\Delta t$, alors que les v sont évalués à $t = -\Delta t/2, \Delta t/2, 3\Delta t/2, \ldots, (N-1)\Delta t - \Delta t/2$ selon le schéma donné **figure 4**. Ainsi, pour cet algorithme, `x[i]` représente de façon approchée la position à l'instant $i\Delta t$, et `v[i]` représente de façon approchée la vitesse à l'instant $(i - 1/2)\Delta t$.

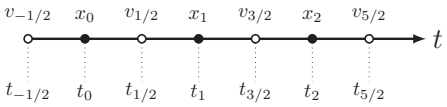

Figure 4 – Discrétisation en temps utilisée dans le cas de l'algorithme de Leapfrog. Les conditions initiales correspondent à $(x_0; v_{-1/2})$

Pour le système considéré, montrer alors que x_{n+1} et $v_{n+1/2}$ prennent les formes suivantes :

$$\begin{cases} x_{n+1} \approx x_n + v_{n+1/2} \cdot \Delta t \\ v_{n+1/2} \approx -\dfrac{\omega_0^2 \Delta t}{1 + \zeta\omega_0\Delta t} x_n + \dfrac{1 - \zeta\omega_0\Delta t}{1 + \zeta\omega_0\Delta t} v_{n-1/2} + \dfrac{F_n}{m(1 + \zeta\omega_0\Delta t)}\Delta t. \end{cases} \tag{14}$$

Q26. Quel problème pose l'évaluation de `E[i+1]` ?
Dans la suite, on préfèrera ainsi évaluer `E[i]`.

Q27. Comment obtenir `E[i]` à partir de `x[i]` et `v[i]` ?

Q28. En introduisant deux variables `fac1` et `fac2` correspondant respectivement à $1 - \zeta\omega_0\Delta t$ et $\dfrac{1}{1 + \zeta\omega_0\Delta t}$, écrire la boucle en `i` permettant d'obtenir toutes les valeurs de `x[i+1]` et `v[i+1]`.

Q29. Compléter la boucle de la question **Q28** avec le calcul du terme d'énergie totale.

Téléchargé gratuitement sur Doc-Solus.fr .

Q30. Écrire une fonction `integration(F)` qui prend en argument le tableau `F[]` des forces d'excitation et renvoie les tableaux `x[]`, `v[]` et `E[]` complétés.
On introduira pour cela une variable `algo` supposée définie en global, permettant d'appliquer l'algorithme d'Euler si `algo==0` ou de Leapfrog si `algo==1`.
On considèrera également le cas $t = 0$.

Q31. Écrire une fonction `force(f,t,w)` qui prend en argument F_0, t, et ω de l'équation (9) et retourne la valeur de $\mathbf{F}_{\mathbf{exc}}$ pour un temps `t` donné.

Q32. Écrire une fonction `force_exc()` qui complète et retourne le tableau `F[]` des forces d'excitation en fonction du booléen `exc` défini globalement valant `True` si une excitation est appliquée au système, `False` sinon.
Dans le cas où `exc==True`, on appellera la fonction `force` définie précédemment en question **Q31**. Dans le cas où `exc==False`, on prendra alors : `F[i]=0`, $\forall i$.

Q33. Donner alors les lignes de code permettant de réaliser la simulation numérique à partir des fonctions précédentes.

Q34. On souhaite désormais estimer la qualité des résultats numériques par rapport aux données analytiques de référence. Écrire une fonction `ema(d,dref)` qui calcule l'erreur maximale absolue entre un jeu de données numériques `dn` et analytiques `da`. On supposera que les tableaux `dn` et `da` sont de même dimension n.

Q35. Pour le problème particulier qui nous intéresse, si l'on souhaite appliquer cette fonction aux tableaux contenant les données numériques et analytiques pour E, quel est l'indice maximal de ces tableaux à considérer ?
Réécrire alors la fonction `ema` pour qu'elle soit applicable aux deux algorithmes considérés.

Partie III - Modélisation : analyse des résultats d'un cas simple

Toutes les données numériques suivantes ont été obtenues avec : $t_{\max} = 10$ s, $m = 110$ tonnes, $x_0 = 0{,}02$ m et $v_0 = 0$ m.s^{-1} et la valeur de k obtenue en question **Q15**, dans le cas d'un système sans amortissement et sans excitation externe. Le **tableau 2** en page 9 présente les erreurs maximales absolues (EMA) calculées avec la fonction `ema` des énergies obtenues numériquement par rapport à celles obtenues analytiquement, pour les deux algorithmes envisagés et divers pas de temps.

Q36. Justifier l'ordre de grandeur des Δt considérés du **tableau 2** pour la discrétisation en temps utilisée.

Q37. En utilisant les données numériques du **tableau 2** donner l'ordre approximatif de l'erreur globale sur E des deux algorithmes considérés. Justifier votre réponse.

Q38. Dans le cas simple de l'algorithme d'Euler, comment augmente E lorsqu'on passe de t_n à t_{n+1} ?
Relier alors ce résultat aux données du **tableau 2**.

Téléchargé gratuitement sur Doc-Solus.fr .

Δt (s)	EMA	
	Euler	Leapfrog
0,050	128,0203160	0,0837314
0,010	15,1373529	0,0033484
0,005	7,1159053	0,0008371
0,001	1,3556230	0,0000335

Tableau 2 – EMA obtenues pour E (en J) par rapport à la valeur analytique, pour différents pas de temps Δt, dans le cas où il n'y a pas d'amortissement et d'excitation externe

Une propriété importante que devrait vérifier un algorithme d'intégration est la réversibilité dans le temps : en partant des positions et vitesses d'un temps $t + \Delta t$ et en appliquant un pas de temps $-\Delta t$, un algorithme réversible en temps devrait redonner les positions et vitesses du temps t.
En pratique, en partant d'un couple $(x_n; v_n)$, on applique donc tout d'abord un pas Δt pour déterminer $(x_{n+1}; v_{n+1})$, puis on applique un pas $-\Delta t$ afin d'obtenir $(\bar{x_n}; \bar{v_n})$. Si $(\bar{x_n}; \bar{v_n}) = (x_n; v_n)$, alors l'algorithme est dit réversible en temps.

Q39. Pourquoi s'agit-il d'une propriété importante à vérifier pour le problème considéré ?

Q40. Donner l'expression de $\bar{x_n}$ en fonction de x_n pour les deux algorithmes considérés. Peut-on déjà conclure sur la réversibilité en temps de chacun de ces algorithmes ?

Q41. Donner alors l'expression de $\bar{v_n}$ en fonction de v_n lorsque nécessaire et conclure sur la réversibilité en temps.

Q42. On s'intéresse enfin à une simulation de plus grande durée (t_{max} = 60 s). La **figure 5** donne l'évolution de l'erreur absolue sur x entre données numériques et analytiques ($|e_x|$) pour les deux algorithmes choisis. Que peut-on mettre en évidence sur cette figure ?

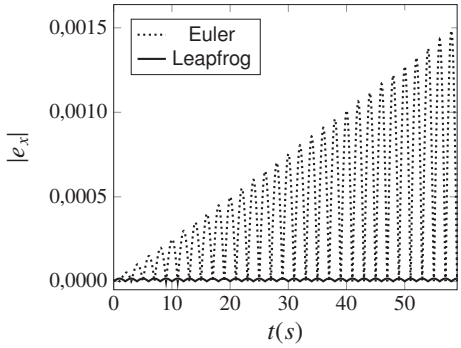

Figure 5 – Erreurs absolues calculées sur la position de la masse à Δt = 0,001 s, pour les deux algorithmes considérés

Q43. Conclure sur les avantages et les inconvénients de chacun de ces algorithmes et sur leur adéquation pour le traitement numérique de ce problème dans le cas où on envisage une simulation de plusieurs heures.

Téléchargé gratuitement sur Doc-Solus.fr .

ANNEXE

Aide-mémoire sur numpy

Les bibliothèques sont importées de la façon suivante :

```
from math import *
import numpy as np
```

La création d'un tableau numpy `tab` à une dimension possédant n éléments, tous initialisés à 0, est réalisée à l'aide de l'instruction :

```
>>> tab=np.zeros(n)
```

Celle d'un tableau numpy `tab` à une dimension possédant n éléments, uniformément répartis entre deux valeurs `debut` et `fin`, se fait avec :

```
>>> debut=0; fin=10; n=5
>>> tab=np.linspace(debut,fin,n)
>>> print tab
array([  0.   2.5  5.   7.5  10. ])
```

L'accès à un élément du tableau `tab` (en lecture ou en écriture) se fait par `tab[i]`, la numérotation des indices se faisant à partir de 0 :

```
>>> tab=np.zeros(4)
[ 0.  0.  0.  0. ]
>>> tab[1]=2; tab[2]=6; print tab
[ 0.  2.  6.  0. ]
```

La sélection de l'ensemble des `j` premiers éléments du tableau `tab` est possible avec :

```
>>> print tab[:3]
[ 0.  2.  6. ]
```

Le maximum des éléments d'un tableau `tab` s'obtient avec :

```
>>> np.max(tab)
```

FIN

Téléchargé gratuitement sur Doc-Solus.fr .