

Épreuve de format «**Modélisation Mathématique et Informatique**» du concours A-ENV

Consignes La qualité de la rédaction, la clarté et la précision des raisonnements, entreront pour une part importante dans l'appréciation de la copie. Les abréviations, sigles ou phrases nominales sont à proscrire. La numérotation des exercices (et des questions) doit être respectée et mise en évidence. Les résultats doivent être encadrés proprement.

Chaque candidat est responsable de la vérification de son sujet d'épreuve : pagination et impression de chaque page. Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il convient de le signaler sur la copie et de poursuivre la composition en expliquant les raisons des initiatives qui ont été prises.

L'usage de la calculatrice est interdit.



Problème 1 Comparaisons de schémas numériques selon un critère énergétique

Partie I— Contexte et but de l'étude.

L'objectif de ce sujet est de présenter, dans un cas très simple, plusieurs modèles numériques permettant d'approcher les solutions d'une équation différentielle et de comprendre pourquoi on peut préférer un modèle numérique à un autre. On s'intéresse à un oscillateur harmonique (non amorti) régi par une équation différentielle :

$$\frac{d^2x}{dt^2} = -\omega^2 x, \quad (1)$$

où x désigne une certaine grandeur physique et t le temps (mesuré en secondes). La « pulsation » du mouvement est ω , qui est un réel strictement positif traduisant les caractéristiques du système oscillant. Le cas de l'oscillateur amorti sera abordé à la fin du sujet.

Pour étudier commodément cette équation différentielle, on introduit une seconde coordonnée :

$$y = \frac{1}{\omega} \frac{dx}{dt}.$$

Ainsi y est proportionnelle à la vitesse. L'équation différentielle initiale (1) est alors équivalente au système suivant :

$$(S) \begin{cases} \frac{dx}{dt} = \omega y \\ \frac{dy}{dt} = -\omega x. \end{cases}$$

L'ensemble $\{(x(t), y(t)), t \in \mathbf{R}\}$ s'appelle *l'espace des phases* associé à l'équation (1).

Une solution du système S est un couple de fonctions (x, y) définies et de classe \mathcal{C}^1 sur $[0, +\infty[$, et telles que :

$$\forall t \in [0, +\infty[, \begin{cases} x'(t) = \omega y(t) \\ y'(t) = -\omega x(t). \end{cases}$$

Dans tout le problème on notera $E(u, v)$ la quantité définie par :

$$\forall (u, v) \in \mathbf{R}^2, \quad E(u, v) = u^2 + v^2.$$

Par commodité, si $Z = \begin{pmatrix} u \\ v \end{pmatrix}$ est un élément de $\mathfrak{M}_{2,1}(\mathbf{R})$, on notera aussi :

$$E(Z) = E\left(\begin{pmatrix} u \\ v \end{pmatrix}\right) = u^2 + v^2.$$

Si (x, y) est une solution de (S), l'énergie du système à tout instant t positif ou nul est le réel :

$$E(x(t), y(t)) = x(t)^2 + y(t)^2.$$

La propriété fondamentale qui nous intéresse ici est une propriété physique : la conservation de l'énergie du système. On étudiera cette énergie dans un premier temps; on retiendra que la donnée de $x(t)$ et de $y(t)$ (position et vitesse) en un instant t fixe l'énergie du système à tout instant (cela provient du fait que le système est autonome et isolé).

On s'intéressera ensuite à trois méthodes numériques. L'idée, comme dans la méthode d'Euler pour les équations d'ordre 1, consiste à :

- discrétiser le temps en choisissant un pas de temps h et en posant, pour n entier : $t_n = n \times h$,
- approcher les quantités $x(t_n)$ et $y(t_n)$ par des valeurs x_n et y_n , en obtenant des relations de récurrence entre x_{n+1}, y_{n+1} et x_n, y_n .

Se posera alors la question du comportement de ces approximations par rapport à l'énergie. On cherche donc à comprendre comment l'analyse mathématique des modèles numériques nous conduit à privilégier ceux qui « conservent » les propriétés physiques d'un problème.

Partie II—Préliminaire : quelques propriétés théoriques.

1 — (Équivalence de (1) et (S))

Justifier le fait que les équations (1) et (S) sont équivalentes au sens suivant :

Pour toute solution x de l'équation (1), il existe une unique fonction y telle que (x, y) soit solution de (S);

\Leftrightarrow pour toute solution (x, y) de (S), la fonction x est solution de (1). On procédera, soigneusement, par double implication.

2 — (Résolution explicite) Soit x une solution de l'équation différentielle (1).

Résoudre l'équation différentielle et donner l'expression explicite de $x(t)$ en fonction de t et des valeurs initiales $x(0)$ et $x'(0)$.

3 — (Énergie) Soit (x, y) une solution de (S).

3.1. En dérivant, montrer que l'application $g \begin{cases} [0, \infty[& \longrightarrow & \mathbf{R} \\ t & \longrightarrow & E(x(t), y(t)) \end{cases}$ est constante.

3.2. En déduire que, lorsque t décrit $[0, +\infty[$, le point $M(t)$ de coordonnées $(x(t), y(t))$ parcourt un cercle \mathcal{C} dont on précisera le centre et dont on donnera le rayon en fonction des valeurs initiales $x(0)$ et $y(0)$. Dans quel sens le point $M(t)$ « tourne-t-il » sur le cercle \mathcal{C} ?

4 — (Étude d'une matrice)

Pour $a \in \mathbf{R}$ on définit $M_a \in \mathfrak{M}_2(\mathbf{R})$ par :

$$M_a = \begin{pmatrix} 1 & a \\ -a & 1 \end{pmatrix}.$$

4.1. Montrer que M_a est inversible et exprimer son inverse en fonction de a .

Soit $X = \begin{pmatrix} u \\ v \end{pmatrix}$ un élément de $\mathfrak{M}_2(\mathbf{R})$.

4.2. Montrer que $E(M_a X) = (1 + a^2)E(X)$. Exprimer aussi $E(M_a^{-1} X)$ en fonction de $E(X)$.

4.3. La matrice M_a est-elle diagonalisable sur \mathbf{R} ? sur \mathbf{C} ? On ne demande pas de diagonaliser de manière effective la matrice.

Dans toute la suite on pourra se servir des résultats de la partie précédente même s'ils n'ont pas été établis.

Partie III—Le problème de l'approximation.

5 — (Présentation des trois méthodes numériques)

Soient t un réel et f une fonction de classe \mathcal{C}^1 sur l'intervalle $[t, t+h]$. Par définition de l'intégrale on sait que l'on a :

$$f(t+h) - f(t) = \int_t^{t+h} f'(s) ds.$$

- La méthode des rectangles à gauche consiste à approcher l'intégrale ci-dessus par $f'(t) \times h$.
 - La méthode des rectangles à droite consiste à approcher cette intégrale par $f'(t+h) \times h$.
 - La méthode des trapèzes consiste à approcher l'intégrale par la demi-somme des approximations précédentes.
- Tracer la courbe d'une fonction simple $g = f'$ quelconque sur l'intervalle $[t, t+h]$ puis représenter sur ce graphique les trois aires qui viennent d'être décrites (i.e. les quantités $g(t) \times h$, $g(t+h) \times h$ et $\frac{1}{2}(g(t) \times h + g(t+h) \times h)$).

On considère maintenant (x,y) une solution de (S) et h un réel strictement positif (le pas de temps). On peut intégrer entre t et $t+h$ le système (S) pour obtenir :

$$(S_{int}) \begin{cases} x(t+h) - x(t) &= \omega \int_t^{t+h} y(s) ds \\ y(t+h) - y(t) &= -\omega \int_t^{t+h} x(s) ds. \end{cases}$$

Pour tout entier naturel n on pose $t_n = n \times h$. Toute méthode permettant d'approcher les deux intégrales précédentes donne alors une méthode numérique sur x,y , nous allons les étudier dans la suite.

6 — (Etude de la méthode d'Euler explicite)

Pour la méthode d'Euler dite **explicite** on utilise l'approximation selon la méthode des rectangles à **gauche**.

6.1. Justifier, en utilisant (S_{int}) et l'approximation de la méthode des rectangles à gauche, les approximations suivantes :

$$(S_g) \begin{cases} x(t_{n+1}) - x(t_n) &\approx h\omega y(t_n) \\ y(t_{n+1}) - y(t_n) &\approx -h\omega x(t_n) \end{cases}$$

6.2. En déduire que :

$$\begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix} \approx M_{h\omega} \begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix}.$$


La méthode d'Euler explicite nous amène donc à construire la suite $U_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$ définie par :


$$U_0 = \begin{pmatrix} a \\ 0 \end{pmatrix} \text{ avec } a > 0; \quad \forall n \in \mathbf{N}, \quad U_{n+1} = M_{h\omega} U_n.$$

6.3. Montrer que, pour tout $n \in \mathbf{N}$, $E(x_{n+1}, y_{n+1}) = (1 + h^2 \omega^2) E(x_n, y_n)$.

6.4. Pour tout $n \in \mathbf{N}$, exprimer $E(x_n, y_n)$ en fonction de n , puis calculer $\lim_{n \rightarrow \infty} E(x_n, y_n)$.

On souhaite maintenant programmer l'algorithme en langage Python. À cet effet, on supposera que la pulsation ω est prédéfinie comme une constante globale OMEGA.

6.5.  Créer une fonction appelée `eulerexp` prenant en entrée `x0`, `y0`, le pas `h` et un entier positif `N`, et qui renvoie les listes `[x_0, ..., x_N]`, `[y_0, ..., y_N]` issues de l'application de la méthode d'Euler explicite à partir de (x_0, y_0) avec le pas `h` avec `N` itérations.

6.6.  Créer une fonction appelée `eulerexp2` prenant en entrée `x0`, `y0`, le pas `h` et une erreur relative maximale `err`, et qui renvoie les listes `[x_0, ..., x_N]`, `[y_0, ..., y_N]` issues de l'application de la méthode d'Euler explicite à partir de (x_0, y_0) avec le pas `h` et allant jusqu'au dernier instant `N` où $E(x_N, y_N)$ diffère en valeur absolue de $E(x_0, y_0)$ d'au plus `err`.

7 — (Etude de la méthode d'Euler implicite)

Pour la méthode d'Euler dite **implicite** on utilise l'approximation selon la méthode des rectangles à **droite**. En utilisant (S_{int}) et l'approximation de la méthode des rectangles à droite, on justifie de la même manière l'introduction les approximations suivantes :

$$(S_d) \begin{cases} x(t_{n+1}) &\approx x(t_n) + h\omega y(t_{n+1}) \\ y(t_{n+1}) &\approx y(t_n) - h\omega x(t_{n+1}) \end{cases}$$

L'adjectif « implicite » provient du fait que l'approximation obtenue n'est pas une récurrence entre le terme d'ordre n et celui d'ordre $n+1$ mais une équation en le terme d'ordre $n+1$.

7.1. D duire une matrice C , exprim e sous la forme d'une matrice M_a , telle que :

$$C \begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix} \approx \begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix}.$$


Ainsi la m thode d'Euler implicite nous fait d finir la suite $U_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$ d finie par :

$$U_0 = \begin{pmatrix} a \\ 0 \end{pmatrix} \text{ avec } a > 0; \quad \forall n \in \mathbf{N}, \quad M_{-h\omega} U_{n+1} = U_n.$$

7.2. Exprimer U_{n+1} en fonction de U_n pour tout entier $n \in \mathbf{N}$. Justifier alors que la suite $(U_n)_n$ est bien d finie et exprimer, pour tout $n \in \mathbf{N}$, x_{n+1} et y_{n+1} en fonction de x_n et y_n .

7.3. Montrer que, pour tout $n \in \mathbf{N}$, $E(x_{n+1}, y_{n+1}) = \frac{1}{1+h^2\omega^2} E(x_n, y_n)$.

7.4. D terminer $\lim_{n \rightarrow +\infty} E(x_n, y_n)$.

7.5.  Cr er une fonction appel e `euler imp` prenant en entr e x_0, y_0 , le pas h et un entier positif N , et qui renvoie les listes $[x_0, \dots, x_N], [y_0, \dots, y_N]$ issues de l'application de la m thode d'Euler implicite   partir de (x_0, y_0) avec le pas h avec N it rations.

8 — (Etude de la m thode d'Euler trap zo dale)

Pour la m thode d'Euler **trap zo dale** on utilise l'approximation selon la m thode des **trap zes**.

8.1. En s'appuyant sur l' nonc , justifier l'approximation suivante : pour tout $n \in \mathbf{N}$,

$$(S_n) \begin{cases} x(t_{n+1}) - x(t_n) & \approx \frac{h\omega}{2} \times (y(t_n) + y(t_{n+1})) \\ y(t_{n+1}) - y(t_n) & \approx \times -\frac{h\omega}{2} (x(t_n) + x(t_{n+1})) \end{cases}$$

8.2. En d duire que pour tout $n \in \mathbf{N}$:

$$M_{-h\omega/2} \begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix} \approx M_{h\omega/2} \begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix}.$$

On construit alors la suite $U_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$ d finie par :

$$U_0 = \begin{pmatrix} a \\ 0 \end{pmatrix} \text{ avec } a > 0; \quad \forall n \in \mathbf{N}, \quad M_{-h\omega/2} U_{n+1} = M_{h\omega/2} U_n.$$

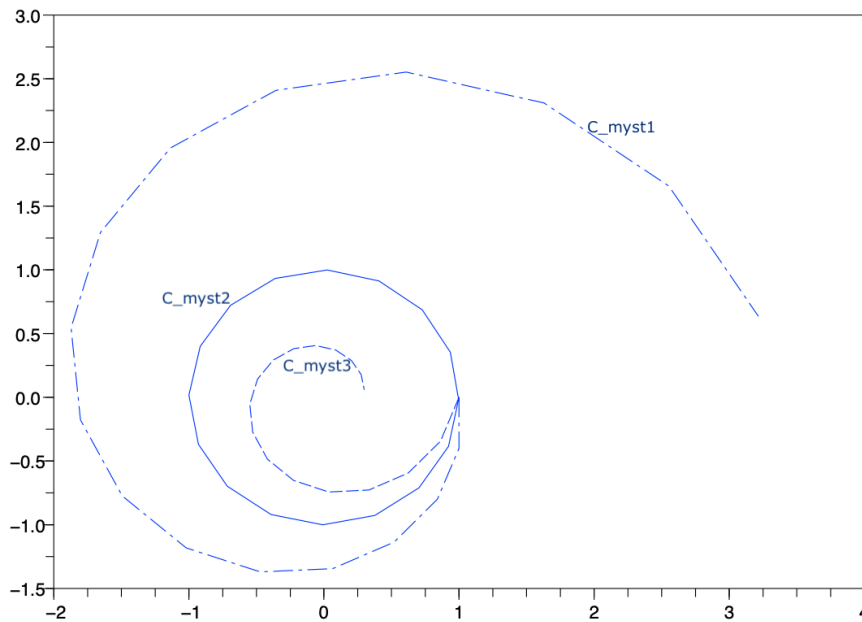
8.3. Exprimer U_{n+1} en fonction de U_n pour tout entier $n \in \mathbf{N}$. Justifier alors que la suite $(U_n)_n$ est bien d finie et exprimer, pour tout $n \in \mathbf{N}$, x_{n+1} et y_{n+1} en fonction de x_n et y_n .

8.4. Comparer alors $E(x_{n+1}, y_{n+1})$ et $E(x_n, y_n)$.

Partie IV— Interpr tation.

9 — Quels commentaires vous inspirent les questions 6.4., 7.4. et 8.4.?

10 — Sur l'image ci-dessous (o  les x sont repr sent s en abscisse et les y en ordonn e), on a pr sent  trois courbes obtenues en reliant les points obtenus par la m thode d'Euler explicite (courbe \mathcal{C}_g), par la m thode d'Euler implicite (courbe \mathcal{C}_d) et par la m thode d'Euler trap zo dale (courbe \mathcal{C}_r).



Associer les courbes \mathcal{C}_{myst1} , \mathcal{C}_{myst2} , \mathcal{C}_{myst3} aux courbes \mathcal{C}_g , \mathcal{C}_d et \mathcal{C}_l . Expliquer les raisons qui vous ont amené à faire ces choix.

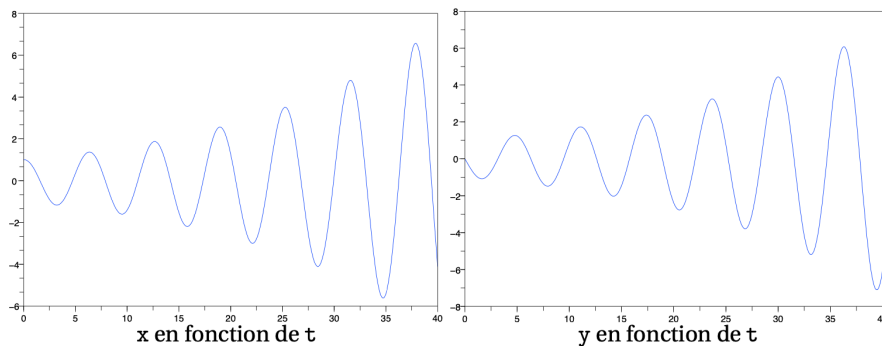
Partie V— Période des solutions approchées.

Cette partie propose une étude informatique analysant les solutions approchées de l'équation différentielle (S), en suivant un point de vue purement empirique. En utilisant une certaine méthode de calcul numérique, on a créé trois listes t , x et y , où

— t est la liste $(0, h, 2h, \dots, Nh)$,

— x et y sont les listes (x_0, \dots, x_N) et (y_0, \dots, y_N) qui fournissent une approximation de la solution de l'équation (S).

On observe alors une sorte de cyclicité dans ce mouvement et on cherche à évaluer les durées des cycles successifs.



11 — On propose la fonction Python suivante :

```

>_ Script
1 def fun1(L) :
2     '''
3     pas de docstring, sinon ce serait trop facile
4     '''
5     taille = len(L)
6     index = 0
7     continuer = True
8     while (index < taille-1) and continuer :
9         if L[index] * L[index+1] < 0 :
10            continuer = False
11            index = index + 1
12    return index

```

11.1. Expliquer, de manière succincte, ce que fait cette fonction.

11.2. Décrire ce qu'elle renvoie dans les cas suivants : $L = [1, 3, 7, 2]$, $L = [-1, -5, 2, 7]$, $L = [1, 5, 1, -1]$.

12 — Proposer une fonction Python appelée `pos_zeros` qui prend en entrée une liste L et renvoie la liste de toutes les positions i telles que $L[i]$ et $L[i+1]$ soient de signes différents.

13 — Proposer une fonction Python `maxima` qui prend en entrée une liste de nombres L et renvoie la liste des positions des maxima locaux.

Par exemple, `maxima([1, 0.1, -1, 2, 1, 0, 1, 3, 4, 3])` renverra `[0, 3, 8]`.

14 — Proposer un protocole, bâti à l'aide d'un ou plusieurs des outils précédents, pour calculer les durées successives des différents cycles du mouvement approché. On ne demande pas l'écriture précise d'un programme.

Partie VI — La méthode d'Euler dans le cas dissipatif.

On s'intéresse dans cette dernière partie au problème de l'oscillateur harmonique amorti, auquel on peut adapter les méthodes de résolution approchées développées dans le début du problème. L'oscillateur amorti est décrit par l'équation

$$\frac{d^2x}{dt^2} + \alpha \frac{dx}{dt} + \omega^2 x = 0, \quad (2)$$

où α et ω sont deux constantes strictement positives. Le terme du premier ordre est appelé *terme d'amortissement*.

15 — Résoudre l'équation différentielle (2).

16 — Justifier que les solutions de (2) sont toutes périodiques si et seulement si $\alpha < 2\omega$.

On supposera dans la suite que α est suffisamment petit pour que cette condition soit réalisée.

17 — Soit x une solution de (2). On introduit une fonction y comme précédemment, *i.e.* en posant :

$$y = \frac{1}{\omega} \frac{dx}{dt}.$$

17.1. Justifier que x est solution de (2) si et seulement si (x, y) solution de

$$(S_{amor}) \begin{cases} \frac{dx}{dt} = \omega y \\ \frac{dy}{dt} = -(\alpha y + \omega x). \end{cases}$$

17.2. Montrer que la fonction énergie du système $t \rightarrow E(x(t), y(t))$ décroît et tend vers 0 quand le temps t tend vers $+\infty$.

17.3. On met en place la méthode d'Euler explicite (ou rectangle gauche) avec un pas $h > 0$ pour ce nouveau problème.

Expliquer et justifier l'introduction d'une suite de vecteurs colonnes $(X_n)_n$ vérifiant :

$$X_{n+1} = AX_n \quad \text{avec} \quad A = \begin{pmatrix} 1 & h\omega \\ -h\omega & 1 - \alpha h \end{pmatrix}.$$

17.4. Exprimer alors X_n en fonction de n , A et X_0 .

18 — On admettra qu'il existe une matrice $P \in \mathcal{M}_2(\mathbb{C})$ inversible, et deux complexes μ_1 et μ_2 de module 1 tels que :

$$A = \sqrt{1 - \alpha h + h^2 \omega^2} P \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} P^{-1}.$$

On appellera (\star) cette propriété d'existence.

18.1. On pose $B = P \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} P^{-1}$. Montrer que la suite de terme général $E(B^n X_0)_{n \in \mathbb{N}}$ est bornée. Indication : On pourra

commencer par justifier que les coefficients de la matrice B^n sont bornés.

18.2. En déduire qu'il existe un $h_0 > 0$ tel que

$$\forall h \in]0, h_0[, \quad \lim_{n \rightarrow +\infty} E(X_n) = 0.$$

18.3. Que dire alors de ce modèle numérique ?

19 — Comment démontrer (\star) ?

Solution (Problème 1)

1 — Soit $t \mapsto x(t)$ une solution de l'équation (1).

\Rightarrow Posons $y(t) = \frac{1}{\omega} x'(t)$. Alors $x' = \omega y$ et $y' = \frac{1}{\omega} x'' = -\omega x$. De plus y est unique puisque la ligne 1 de (S) impose $y = \frac{1}{\omega} x'$.

\Leftarrow Soit (x, y) une solution de (S). Alors $x''(t) = \omega y'(t) = -\omega^2 x(t)$, donc x est solution de (1). En conclusion :

Les équations (1) et (S) sont équivalentes.

2 — La fonction x est solution de l'équation différentielle homogène d'ordre 2 à coefficients constants : $x'' + \omega^2 x = 0$, d'équation caractéristique associée : $X^2 + \omega^2 = 0 \iff X = \pm i\omega$.

Donc il existe deux constantes a et b telles que : $x(t) = a \cos(\omega t) + b \sin(\omega t)$.

On a alors : $x(0) = a$ et $x'(0) = b\omega$, donc : $x(t) = x(0) \cos(\omega t) + \frac{x'(0)}{\omega} \sin(\omega t)$.

3 — 3.1. Posons $g(t) = E(x(t), y(t)) = x(t)^2 + y(t)^2$. La fonction g est dérivable et, en utilisant (S), nous obtenons :

$$g'(t) = 2x'(t)x(t) + 2y'(t)y(t) = 2\omega y(t)x(t) + 2(-\omega x(t))y(t) = 0.$$

Ainsi g est constante :

La fonction $t \mapsto E(x(t), y(t))$ est constante sur \mathbf{R}_+ .

3.2. On a pour tout t : $x(t)^2 + y(t)^2 = x(0)^2 + y(0)^2$. Ainsi

$M(t) = (x(t), y(t))$ parcourt le cercle de centre $O(0,0)$ et de rayon $r = \sqrt{x(0)^2 + y(0)^2}$.

On a : $y' = -\omega x$ donc quand $x > 0$ alors $y' < 0$ et y décroît. Donc :

$M(t)$ tourne sur \mathcal{C}_a dans le sens anti-trigonométrique.

4 — 4.1. C'est une matrice de format 2×2 , donc on calcule son déterminant. On a $\det(M_a) = 1 + a^2 > 0$. Donc

M_a est inversible, et $M_a^{-1} = \frac{1}{1+a^2} \begin{pmatrix} 1 & -a \\ a & 1 \end{pmatrix} = \frac{1}{1+a^2} M_{-a}$.

4.2. Soit $X = \begin{pmatrix} u \\ v \end{pmatrix}$. Alors $E(X) = E(u, v) = u^2 + v^2$.

Et $M_a X = \begin{pmatrix} u + av \\ -au + v \end{pmatrix}$ donc $E(M_a X) = (u + av)^2 + (-au + v)^2 = (1 + a^2)(u^2 + v^2)$.

Ainsi $E(M_a X) = (1 + a^2)E(X)$. En appliquant cette égalité avec $M_a^{-1} X$ en remplacement de X ¹⁶ : $E(M_a M_a^{-1} X) =$

$(1 + a^2)E(M_a^{-1} X)$. Autrement dit : $E(M_a^{-1} X) = \frac{1}{1+a^2} E(X)$.

4.3. Soit $\lambda \in \mathbf{C}$. On a $\det(M_a - \lambda_2) = (1 - \lambda)^2 + a^2 = \lambda^2 - 2\lambda + 1 + a^2$. Le discriminant de ce polynôme en λ est $-4a^2$.

① Si $a \neq 0$: la matrice possède deux valeurs propres complexes distinctes, donc elle est

diagonalisable sur \mathbf{C} mais pas sur \mathbf{R} .

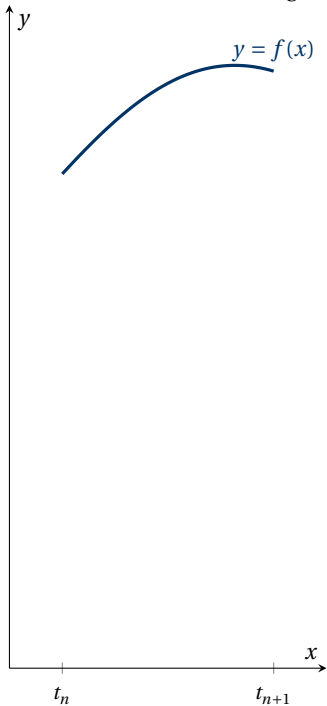
② Si $a = 0$, alors $M_0 = I_2$ donc la matrice est diagonalisable sur \mathbf{R} (car déjà diagonale!).

Partie I— Le problème de l'approximation.

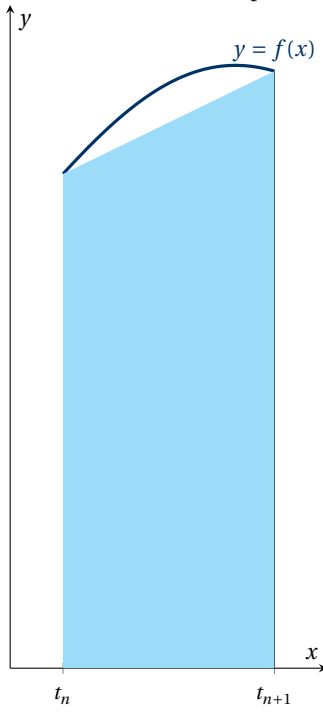
5 —

¹⁶Un calcul direct fonctionne aussi

Pour la méthode des rectangles.



Pour la méthode des trapèzes.



6 — 6.1. Dans (S_{int}) on approche les intégrales selon les rectangles à gauche :

$$\int_t^{t+h} y(s) ds \approx (t+h-t)\omega y(t_n),$$

et de-même pour la seconde intégrale. On obtient alors les relations de récurrences proposées par l'énoncé :

$$(S_g) \begin{cases} x(t_{n+1}) \approx x(t_n) + h\omega y(t_n), \\ y(t_{n+1}) \approx y(t_n) - h\omega x(t_n). \end{cases}$$

6.2. Ceci donne, directement par définition du produit matriciel :

$$\begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix} \approx B \begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix} \text{ avec } B = M_{h\omega}.$$

6.3. Nous avons calculé dans la partie précédente $E(M_a X)$ pour toute colonne X_n en choisissant $a = h\omega$, on trouve

$$E(U_{n+1}) = (1 + h^2\omega^2)E(U_n).$$

6.4. Et ainsi : $E(x_n, y_n) = (1 + h^2\omega^2)^n E(U_0)$. Puisque $1 + h^2\omega^2 > 0$, les résultats de convergence sur les suites géométriques nous donnent :

$$\lim_{n \rightarrow +\infty} E(x_n, y_n) = +\infty.$$

6.5. >_ Script

```

1 OMEGA = 1
2 import numpy as np
3 def eulerexp(x0, y0, h, N) :
4     x = np.zeros(N+1)
5     y = np.zeros(N+1)
6     t = np.zeros(N+1)
7     x[0] = x0
8     y[0] = y0
9     for k in range(N) :
10        t[k] = k*h
11        x[k+1] = x[k] + h*OMEGA*y[k]
12        y[k+1] = y[k] - h*OMEGA*x[k]
13     return x,y

```


6.6. >_ Script

```

1 import numpy as np
2
3 def E(u,v) :
4     '''
5     (u,v)->E(u,v)
6     '''
7     return u**2 + v**2
8
9 def eulerexp2(x0, y0, h, err) :
10    N = 1
11    X,Y = eulerexp(x0, y0, h, N)
12    while np.abs(E(X[-1], Y[-1])-E(x0,y0)) > err :
13        N += 1
14        X,Y = eulerexp(x0, y0, h, N)
15    return X,Y

```

Prenons un exemple : `eulerexp2(0, 1, 10**(-2), 0.01)` renvoie??.

7 — 7.1. En écrivant (S_d) sous la forme :

$$(S_d) \begin{cases} x(t_{n+1}) - h\omega y(t_{n+1}) \approx x(t_n), \\ y(t_{n+1}) + h\omega x(t_{n+1}) \approx y(t_n), \end{cases}$$

nous obtenons : $\begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix} \approx C \begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix}$ avec $C = M_{-h\omega}$.

7.2. Puisque $M_{-h\omega}$ est inversible la suite (U_n) est bien définie. On a : $U_{n+1} = M_{-h\omega}^{-1} U_n$. Or, par une question précédente,

$$M_{-h\omega}^{-1} = \frac{1}{1+h^2\omega^2} \begin{pmatrix} 1 & h\omega \\ -h\omega & 1 \end{pmatrix}. \text{ Et ainsi :}$$

$$x_{n+1} = \frac{1}{1+h^2\omega^2} (x_n + h\omega y_n), \quad y_{n+1} = \frac{1}{1+h^2\omega^2} (-h\omega x_n + y_n).$$

7.3. On a d'après la partie préliminaire : $E(U_n) = E(M_{-h\omega} U_{n+1}) = (1+h^2\omega^2)E(U_{n+1})$. Donc $E(x_{n+1}, y_{n+1}) = \frac{1}{1+h^2\omega^2} E(x_n, y_n)$.

7.4. Ainsi $(E(x_n, y_n))$ est une suite géométrique de raison $\frac{1}{1+h^2\omega^2} \in]0, 1[$. Donc $\lim_{n \rightarrow +\infty} E(x_n, y_n) = 0$.

7.5. >_ Script

```

1 def eulerimp(x0, y0, h, N) :
2     x = np.zeros(N+1)
3     y = np.zeros(N+1)
4     t = np.zeros(N+1)
5     x[0] = x0
6     y[0] = y0
7     for k in range(N) :
8         t[k] = k*h
9         x[k+1] = (1/(1+h**2*omega**2))*(x[k]+h*omega*y[k])
10        y[k+1] = (1/(1+h**2*omega**2))*(-h*omega*x[k]+y[k])
11    return x,y

```

8 — (Méthode d'Euler trapézoïdale)

8.1. Avec la méthode des trapèzes, on fait simplement la demi-somme des deux approximations précédentes. On obtient

pour x : $x(t_{n+1}) - x(t_n) \approx \frac{1}{2} (x'(t_n) \times h + x'(t_{n+1}) \times h) = \frac{h\omega}{2} (y(t_n) + y(t_{n+1})).$

De même pour y : $y(t_{n+1}) - y(t_n) \approx \frac{1}{2} (y'(t_n) \times h + y'(t_{n+1}) \times h) = \frac{-h\omega}{2} (x(t_n) + x(t_{n+1})).$

8.2. On regroupe simplement à gauche les termes en t_n et ceux en t_{n+1} à droite.

$$(S_r) \begin{cases} x(t_{n+1}) - \frac{h\omega}{2} y(t_{n+1}) \approx x(t_n) + \frac{h\omega}{2} \times y(t_n) \\ y(t_{n+1}) + \frac{h\omega}{2} x(t_{n+1}) \approx y(t_n) - \frac{h\omega}{2} x(t_n) \end{cases}$$

On déduit alors :

$$M_{-h\omega/2} \begin{pmatrix} x(t_{n+1}) \\ y(t_{n+1}) \end{pmatrix} \approx M_{h\omega/2} \begin{pmatrix} x(t_n) \\ y(t_n) \end{pmatrix}.$$

8.3. La suite (U_n) est bien définie puisque la matrice $M_{-h\omega/2}$ est inversible, et :

$$\forall n \in \mathbf{N}, \quad U_{n+1} = M_{-h\omega/2}^{-1} M_{h\omega/2} U_n = \frac{1}{1 + \frac{h^2\omega^2}{4}} \begin{pmatrix} 1 - \frac{h^2\omega^2}{4} & h\omega \\ -h\omega & 1 - \frac{h^2\omega^2}{4} \end{pmatrix} U_n.$$

8.4. D'après la partie préliminaires :

$$E(U_{n+1}) = E(M_{-h\omega/2}^{-1} M_{h\omega/2} U_n) = \frac{1}{1 + \frac{h^2\omega^2}{4}} E(M_{h\omega/2} U_n) = \frac{1}{1 + \frac{h^2\omega^2}{4}} \times (1 + \frac{h^2\omega^2}{4}) E(U_n) = E(U_n).$$

Ainsi $E(x_{n+1}, y_{n+1}) = E(x_n, y_n)$.

Partie II— Interprétation.

9 — On remarque que l'énergie $E(x_n, y_n)$:

- augmente et tend vers $+\infty$ avec la méthode d'Euler explicite ;
- diminue et tend vers 0 avec la méthode d'Euler implicite ;
- est constante avec la méthode d'Euler trapézoïdale.

Or d'après la partie préliminaires, l'énergie est constante pour une solution exacte, donc

la méthode d'Euler trapézoïdale a l'air ici plus adaptée.

10 — Pour les mêmes raisons qu'au-dessus :

- \mathcal{C}_1 est la courbe en spirale qui se déroule avec des bras de plus en plus longs i.e. \mathcal{C}_{myst1} ;
- \mathcal{C}_2 est la courbe en spirale qui s'enroule avec des bras de plus en plus courts, i.e. \mathcal{C}_{myst3} ;
- \mathcal{C}_3 est la courbe circulaire (avec donc des bras de longueur constante), i.e. \mathcal{C}_{myst2} .

Partie III— Période des solutions approchées.

11 — La fonction `fun1` renvoie le premier indice i pour lequel $L[i]$ a changé de signe par rapport aux précédents $L[k]$, si elle est appliquée à une liste L .

Par exemple `fun1([4, 2, 9, -7, -1, 8, -6]) = 3`.

Si tous les termes de L sont de même signe alors elle donne la longueur de L .

12 — Pour la fonction `pos_zeros` demandée on prend la trame de la fonction `fun1` mais on remplace le `while` par `for` :

```
>_ Script
1 def pos_zeros(L) :
2     '''
3     L->positions des changements de signe dans L
4     '''
5     taille = len(L)
6     liste = []
7     for index in range(taille-1) :
8         if L[index] * L[index+1] < 0 :
9             liste.append(index)
10    return liste
```

13 — Pour `maxima` c'est le même principe, avec deux cas particuliers pour la première et la dernière valeur.

```

1 def maxima(L) :
2     '''
3     L->maxima locaux de L
4     '''
5     taille = len(L)
6     liste = [ ]
7     if L[0]>L[1] :
8         liste.append(0)
9     for index in range(1,taille-1) :
10        if L[index] > L[index+1] and L[index]>L[index-1] :
11            liste.append(index)
12    if L[taille-1] > L[taille-2] :
13        liste.append(taille-1)
14    return liste

```

14 — Plusieurs protocoles sont possibles :

- on commence par calculer les positions des maxima locaux avec maxima, puis on mesure l'écart moyen entre deux maxima successifs. Cette quantité sera une bonne approximation de la période des oscillations.
- On peut utiliser aussi la fonction pos_zeros. On ne conserve ensuite qu'un élément sur deux, et la période correspondra là encore à l'écart moyen entre deux points de cette nouvelle liste.

Partie IV— La méthode d'Euler dans le cas dissipatif.

15 — L'équation différentielle (2) a pour équation caractéristique associée : $x^2 + \alpha x + \omega^2 = 0$, qui a pour discriminant : $\Delta = \alpha^2 - 4\omega^2$.

— Premier cas : si $\alpha > 2\omega$, alors $\Delta > 0$, $X = \frac{-\alpha \pm \sqrt{\alpha^2 - 4\omega^2}}{2}$, et :

$$\text{Si } \alpha > 2\omega, \quad \exists(a,b) \in \mathbf{R}^2, \forall t \in \mathbf{R}_+, \quad x(t) = a \exp\left(\frac{-\alpha + \sqrt{\alpha^2 - 4\omega^2}}{2} t\right) + b \exp\left(\frac{-\alpha - \sqrt{\alpha^2 - 4\omega^2}}{2} t\right).$$

— Deuxième cas : si $\alpha = 2\omega$, alors $\Delta = 0$, $X = \frac{-\alpha}{2}$, et :

$$\text{Si } \alpha = 2\omega, \quad \exists(a,b) \in \mathbf{R}^2, \forall t \in \mathbf{R}_+, \quad x(t) = a \exp\left(\frac{-\alpha}{2} t\right) + bt \exp\left(\frac{-\alpha}{2} t\right).$$

— Troisième cas : $\alpha < 2\omega$, alors $\Delta < 0$, $X = \frac{-\alpha \pm i\sqrt{4\omega^2 - \alpha^2}}{2}$, et :

$$\text{Si } \alpha < 2\omega : \exists(a,b) \in \mathbf{R}^2, \forall t \in \mathbf{R}_+, \quad x(t) = \exp\left(\frac{-\alpha}{2} t\right) \left(a \cos\left(\frac{\sqrt{4\omega^2 - \alpha^2}}{2} t\right) + b \sin\left(\frac{\sqrt{4\omega^2 - \alpha^2}}{2} t\right) \right).$$

16 — Il y a des oscillations si et seulement si $\alpha < 2\omega$, puisque c'est le seul cas où la solution est composée de cos et de sin.

17 —17.1.

\Rightarrow Supposons que x est solution de (2). Alors on a déjà par définition de y : $\frac{dx}{dt} = \omega y$. Et $\frac{dy}{dt} = \frac{1}{\omega} x''(t) = \frac{1}{\omega} (-\alpha x'(t) - \omega^2 x(t)) = -(\alpha y + \omega x)$. D'où le système

$$(S_{amor}) \begin{cases} \frac{dx}{dt} = \omega y \\ \frac{dy}{dt} = -(\alpha y + \omega x). \end{cases}$$

\Leftarrow Supposons que (x,y) est solution du système. Alors en remplaçant dans la deuxième ligne y par $\frac{1}{\omega} \frac{dx}{dt}$, on obtient l'équation (2). On a montré l'équivalence annoncée.

17.2. Posons $h(t) = E(x(t),y(t)) = x(t)^2 + y(t)^2$. La fonction h est dérivable et : $h'(t) = 2x(t)x'(t) + 2y(t)y'(t)$ pour tout $t \in \mathbf{R}$.

Or $y = \frac{1}{\omega} x'$ et $y' = \frac{1}{\omega} x'' = -\frac{\alpha}{\omega} x' - \omega x$.

Donc : $h'(t) = 2x(t)x'(t) + 2\frac{x'(t)}{\omega} \left(-\frac{\alpha x'(t)}{\omega} - \omega x(t)\right) = -\frac{2\alpha x'(t)^2}{\omega^2} \leq 0$.

Ainsi h est décroissante.

De plus x (et donc aussi y) est le produit de fonctions bornées cos ou sin par un terme exponentiel $t \mapsto \exp\left(\frac{-\alpha}{2} t\right)$ qui tend vers 0 lorsque t tend vers $+\infty$. Donc $h(t)$ tend lui aussi vers 0. En conclusion :

$t \mapsto E(x(t), y(t))$ décroît et tend vers 0 quand $t \rightarrow +\infty$.

17.3. On rappelle que : $x' = \omega y$ et $y' = -\omega x - \alpha y$.

Avec la méthode d'Euler explicite :

$$x(t_{n+1}) - x(t_n) \approx hx'(t_n) \approx h\omega y(t_n) \text{ et } y(t_{n+1}) - y(t_n) \approx hy'(t_n) \approx -h\omega x(t_n) - \alpha hy(t_n).$$

Et ainsi :

$$x(t_{n+1}) \approx x(t_n) + h\omega y(t_n) \text{ et } y(t_{n+1}) \approx -h\omega x(t_n) + (1 - \alpha h)y(t_n).$$

Ce qui s'écrit matriciellement :

Avec $X_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix}$ on a : $X_{n+1} = AX_n$ avec $A = \begin{pmatrix} 1 & h\omega \\ -h\omega & 1 - \alpha h \end{pmatrix}$.

18—18.1. Soit $B = P \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} P^{-1}$. Alors $B^n = P \begin{pmatrix} \mu_1^n & 0 \\ 0 & \mu_2^n \end{pmatrix} P^{-1}$. Puisque $|\mu_1^n| = |\mu_2^n| = 1$, la suite des coefficients de la matrice

B^n est bornée, donc il en est de même pour les coefficients de $B^n X_0$.

Ainsi la suite $(E(B^n X_0))$ est bornée.

18.2. On a : $X_n = A^n X_0 = (1 - \alpha h + h^2 \omega^2)^n B^n X_0$.

Pour h petit, $1 - \alpha h + h^2 \omega^2 \approx 1 - \alpha h < 1$. Donc $(1 - \alpha h + h^2 \omega^2)^n \rightarrow 0$ quand $n \rightarrow +\infty$. Puisque de plus $E(B^n X_0)$ est bornée, on en déduit que : $E(X_n) \rightarrow 0$.

18.3. Ce modèle se comporte alors comme le modèle implicite vu précédemment, car l'énergie tend vers 0.

19— Le fait (★) se démontre en diagonalisant la matrice.