Modes propres de vibration

10 juin 2025

1 Modes propres associés à une assemblée de ressorts

Le but de cette étude est de calculer les fréquences propres et les modes propres d'un oscillateur linéaire formé de quatre sphères couplées deux à deux par des ressorts.

Pour simplifier, on suppose que les sphères sont de masse identique m et que les ressorts ont tous la même constante de raideur k.

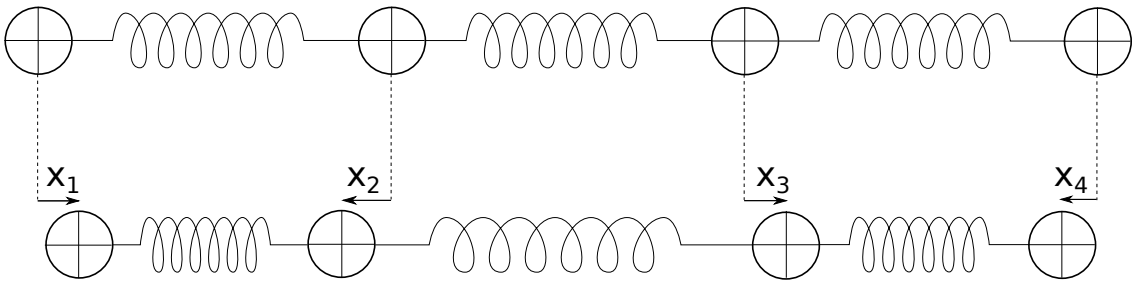

FIGURE 1 – ressorts couplés

Le principe fondamental de la dynamique appliqué successivement à chaque masse mène à un système de quatre équations différentielles couplées :

$$\left\{\begin{array}{lcl} m\ddot{x}_1 &=& -k\,(x_1-x_2) \\ m\ddot{x}_2 &=& -k\,(x_2-x_1)-k\,(x_2-x_3) \\ m\ddot{x}_3 &=& -k\,(x_3-x_2)-k\,(x_3-x_4) \\ m\ddot{x}_4 &=& -k\,(x_4-x_3) \end{array}\right. \Longleftrightarrow \left(\begin{array}{c} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \end{array}\right) = -\frac{k}{m}\left(\begin{array}{cccc} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\right)$$

En se plaçant dans le régime harmonique et en utilisant la notation complexe, on écrit : $x_i(t) = a_i\, exp(i\omega t)$ où $i \in [1,4]$ et $a_i \in \mathbb{C}$. Noter que les déplacements physiques coïncident avec les parties réelles de ces quantités.

On obtient, en notant $\omega_0^2 = k/m$, le système linéaire suivant :

$$\omega_0^2 \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \omega^2 \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$$

Le problème revient à diagonaliser une matrice de couplage que l'on note K.

$$K = \omega_0^2 \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

où chaque valeur propre λ correspond au carré de la pulsation propre ω.

1.1 Mise en application

Développer un script Python permettant de calculer les pulsations propres ainsi que les modes associés. *Remarque : en Python, l'indexation des tableaux commence à 0.*

- La matrice K est une matrice pleine stockée dans un tableau numpy.ndarray. Les valeurs propres et vecteurs propres sont calculées avec la fonction `linalg.eig` de la librairie `numpy`.
- La matrice K est une matrice creuse. On utilisera la fonction `sparse.csr_matrix` pour stocker les termes non nuls de la librairie `scipy`. Les valeurs propres et vecteurs propres sont calculées avec la fonction `sparse.linalg.eigsh` de la librairie `scipy`.

2 Application de conditions de déplacement nul en certains noeuds

On présente ici une méthode générale pour calculer les modes propres en s'appuyant sur l'étude précédente lorsque certaines masses du systèmes sont fixes. Ces conditions s'apparentent aux conditions de dirichlet du type $x_i = 0$.

En notant N_d le nombre de masses où la condition de dirichlet est imposée, le nombre de degrés de liberté se réduit alors à $N_{dof} = N - N_d$ et correspond ici au nombre de masses pouvant osciller.

2.1 Algorithme

L'algorithme consiste à former une liste l où le numéro de chaque masse pouvant osciller, apparaît de façon unique. Sa taille est $N_{dof} = N - N_d$ et s'identifie, ici, au nombre de

degrés de liberté.

On forme ensuite une matrice de projection P permettant de passer de l'espace des solutions à N degrés de liberté (étude précédente) à celui restreint à N_{dof} degrés de liberté. Les dimensions de P sont $N_{dof} \times N$ (nombre de lignes × nombre de colonnes). On initialise d'abord la matrice P à zéro ; puis en parcourant séquentiellement la liste l, on impose $P[i, l[i]] = 1$ avec $[\![0, N_{dof} - 1]\!]$. On remarque que $PP^t = \texttt{Id}(N_{dof})$.

Une autre façon de construire la matrice P est de l'initialiser avec la matrice identité $\texttt{Id}(N)$ puis de supprimer toutes les lignes correspondant aux noeuds de dirichlet. Cette technique est bien adaptée à Numpy/Scipy en masquant les numéros de lignes correspondant aux noeuds de dirichlet.

```
def delete_rows(mat, indices):
    indices = list(indices) # transformation en liste
    mask = np.ones(mat.shape[0], dtype=bool) # ligne de True
    mask[indices] = False # True->False pour les noeuds de dirichlet
    return mat[mask]
```

$$P = \begin{pmatrix} \cancel{1} & \cancel{0} & \cancel{0} & \cancel{0} \\ 0 & 1 & 0 & 0 \\ \cancel{0} & \cancel{0} & \cancel{1} & \cancel{0} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

FIGURE 2 – exemple de matrice P avec $ld = [0, 2]$

On construit ensuite K_p qui est la matrice projetée de K dans l'espace solution à N_{dof} degrés de liberté : $K_p = PKP^t$. C'est cette matrice qui est finalement diagonalisée. On obtient alors les valeurs propres λ_p associées aux vecteurs propres a_p. Autrement dit, cela revient à résoudre : $PKP^t a_p = \lambda_p a_p$.

On reconstruit la solution dans l'espace à N corps en appliquant $a = P^t a_p$. Elle vérifie automatiquement, $a[i] = 0$ pour tout noeud i de dirichlet.

2.2 Mise en application

• Utiliser sous Numpy, la fonction `ld=numpy.unique(ld)` pour éliminer tout doublon dans la liste `ld`.

• Calculer numériquement les modes propres du système lorsque les masses 0 et 2 sont fixes.

• Vérifier vos résultats numériques avec une approche analytique.

3 Modes propres d'une corde

Le but de l'exercice est de caractériser les modes de vibration $u_k(x)$ selon Oz d'une corde. Dans la limite des faibles amplitudes de vibration, ils sont solutions de l'équation d'Helmholtz stationnaire :

$$\partial_x^2 \, u_k + k^2 \, u_k = 0 \tag{1}$$

On considère que la corde est fixée à ses extrémités.

3.1 Partie Mathématique

La corde de longueur L_x est dicrétisée en différences finies avec une grille comportant N_x noeuds. On note Δx et le pas de la grille.

1. On numérote de manière unique les noeuds dans $[\![0, N_x - 1]\!]$. L'abscisse d'un noeud est donnée par la relation : $x = ix * \Delta x$. On en déduit que $\Delta x = L_x/(N_x - 1)$.
2. L'expression discrète en différences finies, de l'opérateur laplacien appliqué à $u(x)$ en un point **intérieur** x de la grille est donnée par :

$$\partial_x^2 \, u_k(x) = \frac{u_k(x + \Delta x) + u_k(x - \Delta x) - 2u_k(x)}{\Delta x^2} + \vartheta(\Delta x)^2 \tag{2}$$

3. L'équation (**??**) s'écrit après discrétisation, sous la forme matricielle suivante :

$$\sum_j K_{i,j} \, u_j = k^2 u_i = \lambda_k u_i \tag{3}$$

   Préciser la forme générale de la matrice K sans se soucier des bords.

4. On tient maintenant compte du déplacement nul des noeuds du bord. Dans le cas d'une grille 4×4 de pas de maille Δx, donner précisément le remplissage de K avec l'expression des termes non-nuls. Dans la suite, on verra comment éliminer les degrés de liberté liés aux noeuds de dirichlet. On admet que pour tout noeud i du bord, on laisse vide la ligne i dans la matrice K. Utiliser le patron de la matrice vide fournie ci-après.

	0	1	2	3
0				
1				
2				
3				

3.2 Mise en application

On vous fournit un embryon de programme `FD_helm.py` à compléter, utilisant une structure de données `fdm` pour stocker tous les paramètres de la simulation.

1. Compléter la fonction `__dirichlet` permettant de construire la liste *ld* des noeuds du bord où sont appliquées la conditions de dirichlet. On rappelle que l'instruction `ld.append(e)` permet d'insérer l'élément e dans la liste ld. Utiliser sous Numpy, la fonction `ld=numpy.unique(ld)` pour éliminer tout doublon dans la liste `ld`.
2. Compléter la fonction `__build_K` permettant de remplir la matrice K pour une grille de taille N_x. On rappelle que toute ligne n de K correspondant à un noeud de dirichlet n'est pas remplie.
   Pour tester l'appartenance d'un noeud n dans la liste `ld`, on utilisera l'expression booléenne `n in ld`.
3. Compléter la fonction `solve` permettant de calculer la n^{eme} plus petite valeur propre en module ainsi que le mode propre associé. On utilisera la fonction Numpy `eig`
4. Pour un système de taille $L_x = 2m$, déterminer les 4 premiers modes de basse énergie et comparer à la solution analytique en $\sin(k_x x)$ de l'équation d'Helmholtz (**??**). Reporter les valeurs propres et les contours des modes associés.