TP Matlab 1

Introduction et rappel

1.1 Des valeurs propres ?

Les valeurs propres et vecteurs propres sont particulièrement utiles dans nombre de domaines. Un exemple parmi tant d'autres : la RATP a longuement étudié (avec un succès tout relatif), les fréquences de résonances de ses trains pour éviter que des fréquences désagréables pour les utilisateurs ne soient excitées lorsque le train roule. Comment faire pour cela ? Il "suffit" de modéliser le train – c'est-à-dire trouver sa fonction de transfert– puis de chercher les fréquences propres entre 10Hz et 20000Hz excitables, c'est-à-dire qui seraient associées à un module supérieur à l'unité.

Autre exemple particulièrement important : on évite, dans les machines outils, que les fréquences de résonance d'une machine soit un harmonique bas des fréquences de fonctionnement, pour éviter un vieillissement prématuré, voir même une dégradation, de la machine, due aux vibrations.

Enfin, les valeurs propres permettent d'extraire facilement les solutions des systèmes d'équations différentiels linéaires.

En fait, on résout ici des problèmes aux valeurs propres.

1.2 Faire une décomposition aux valeurs propres

Soit M une matrice de $\mathcal{M}_n$. Pour la suite du TP, posons $n = 4$.
On prendra au choix une matrice définie aléatoirement, par exemple faite de nombre entier entre -10 et 10 – utiliser alors la fonction `rand` et la fonction `floor`– ou la matrice M suivante :

$$M = \begin{pmatrix} 2 & 4 & 8 & 1 \\ -5 & -4 & 7 & -10 \\ 6 & 1 & 6 & -2 \\ 9 & -8 & -5 & -4 \end{pmatrix}$$

On pourra voir M comme la matrice d'un système différentiel $\mathcal{S}$:

$$\mathcal{S} : \left\{ \begin{array}{rcrrrr} \dot{x}_1 & = & 2x_1 & +4x_2 & +8x_3 & +x_4 \\ \dot{x}_2 & = & -5x_1 & -4x_2 & +7x_3 & -10x_4 \\ \dot{x}_3 & = & 6x_1 & +x_2 & +6x_3 & -2x_4 \\ \dot{x}_4 & = & 9x_1 & -8x_2 & -5x_3 & -4x_4 \end{array} \right.$$

Divers rappels

Valeur propre Une valeur propre de M est un scalaire λ tel qu'il existe un vecteur $\boldsymbol{x}$ – nommé vecteur propre de M– de $\mathbb{K}^n$ qui vérifie :

$$M\boldsymbol{x} = \lambda\boldsymbol{x}.$$

Intuitivement, un vecteur propre est donc une direction privilégiée de l'espace lorsque $|\lambda| > 1$.

Polynôme caractéristique Le polynôme caractéristique de M est le polynôme ayant pour racines les valeurs propres de M, c'est à dire :

$$\mathbb{P}(X) = \prod (X - \lambda_i)^{m_i}$$

où m_i est la multiplicité de chaque valeur propre (Il y a m_i vecteurs propres indépendants associés à la valeur propre). On a donc $\sum m_i = n$.

Trouver ce polynôme peut se faire grâce au déterminant, en notant $\mathbb{I}d_n$ la matrice unité de $\mathcal{M}_n$:

$$\mathbb{P}(X) = \det(\mathbb{I}d_n X - M)$$

Développement d'un déterminant Pour trouver le déterminant d'une matrice carrée de dimension supérieur à 3, on peut développer le déterminant suivant une ligne.Si on a M :

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix}$$

alors, en développant sur la première ligne, on a $\det(M) = m_{11}\Delta_{11} - m_{12}\Delta_{12} + m_{13}\Delta_{13} - m_{14}\Delta_{14}$, avec Δ_{ij} est le déterminant de la matrice M où l'on a enlevé la $i$ème ligne et la $j$ème colonne. Par exemple, on a :

$$\Delta_{13} = \begin{vmatrix} m_{21} & m_{22} & m_{24} \\ m_{31} & m_{32} & m_{34} \\ m_{41} & m_{42} & m_{44} \end{vmatrix}$$

Méthode de Newton Pour trouver les racines (les zéros) d'une fonction f, une méthode itérative instinctive est la méthode de Newton : proche de la racine, la fonction est monotone. Suffisamment proche, on peut même approximer la fonction par sa tangente. L'intersection de la tangente avec l'axe des abscisses doit donc donner une meilleure approximation du zéro. En pratique, partant du point X_0, on calcul X_1 tel que :

$$0 = f(X_0) + f'(X_0)(X_1 - X_0)$$

On itère jusqu'à la précision voulue. Attention, néanmoins, lorsque le zéro est complexe, la méthode ne peut pas converger.

Décomposition QR La décomposition qr d'une matrice A est :

$$A = QR$$

Où Q est une matrice orthogonale et R est triangulaire supérieure.

Méthode QR pour trouver les valeurs propres d'une matrice On pose $M_0 = M$. Ensuite, on construit la suite

$$\begin{aligned} Q_i R_i &= M_i \\ M_{i+1} &= R_i Q_i, \end{aligned}$$

jusqu'à ce que M_{i+1} soit triangulaire supérieure. Il se trouve que M_i a toujours les mêmes valeurs propres que M. On a en effet :

$$M_i = Q_i R_i.$$

D'où, en multipliant à droite par Q_i :

$$M_i Q_i = Q_i R_i Q_i.$$

En multipliant à gauche par la transposée de Q_i et en se rappelant que Q_i est orthogonale, on obtient :

$$Q_i^T M_i Q_i = R_i Q_i,$$

donc $R_i Q_i = M_{i+1}$ est juste l'expression de M_i dans une autre base. Par récurrence, on voit donc que les valeurs propres de tout M_i sont donc les mêmes que celles de M !

1.3 Au boulot !

1.3.1 Premier job : passer par le polynôme caractéristique

Identifier le polynôme caractéristique

Mettre en place un algorithme pour identifier le polynôme caractéristique de M – c'est à dire les coefficients du polynôme. Pour se faire, utiliser la méthode de dévellopement du déterminant. Noter que `eye(n)` rend $\mathbb{I}d_n$.
Comparer avec la fonction `poly` de matlab, qui renvoie les coefficients du polynôme caractéristique.

Calculer ses racines

Mettre en place un algorithme pour calculer les racines λ du polynôme caractéristique de M, par exemple une méthode de Newton. Penser à vérifier les possibles multiplicités du zéro. Comparer le résultat avec la fonction `roots` puis avec la fonction `fzero` (qui nécessite de créer une fonction).

Calculer les vecteurs propres de la matrice M

Identifier les vecteurs propres $\boldsymbol{x}$ associés à chaque valeur propre identifiée précédemment, en résolvant $M\boldsymbol{x} = \lambda\boldsymbol{x}$. Penser à trouver autant de vecteur propre que la multiplicité de la valeur propre.

1.3.2 Second job : passer par la méthode QR

Utiliser la fonction `qr` de matlab pour faire la décomposition qr. Trouver les valeurs propres, puis les vecteurs propres en utilisant la méthode qr décrite précédemment.

1.3.3 Dernier job : Vérification

Tester les résultats avec la fonction `eig` de Matlab, qui donne vecteurs propres et valeurs propres d'une matrice.

1.4 Pour les plus rapides : parlons de conditionnement...

Section à ne traiter qu'après avoir terminé les parties précédentes.
On va s'intéresser ici à des questions de conditionnement et d'erreur. On utilisera la matrice :

$$M = \begin{pmatrix} -10 & -8 & -1 & 2 \\ -7 & 4 & 9 & 1 \\ -1 & 3 & -8 & -1 \\ 6 & -10 & 4 & 1 \end{pmatrix}$$

Vecteurs propres et valeurs propres Calculer les valeurs propres λ_i et vecteurs propres $\boldsymbol{x}_i$ de M.

Vecteurs propres perturbés Supposons que la précision du calcul (ou des données qui ont permis de calculer M) ne permettent pas d'avoir une bonne identification de $\boldsymbol{x}_i$, mais une valeur approchée $\tilde{\boldsymbol{x}}_i$. On simule cette erreur en rajoutant un peu de bruit $\boldsymbol{\epsilon}$:

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{\epsilon}.$$

Par exemple, on peut utiliser la fonction `rand` de matlab ($\boldsymbol{\epsilon} = 0.01 * \mathrm{rand}\,(4, 1)$).
Comparer les vecteurs $\frac{1}{\lambda_i} M \tilde{\boldsymbol{x}}_i$ et $\boldsymbol{x}_i$. Conclusions ?

Ces différences proviennent du conditionnement de la matrice : la différence entre les petites valeurs propres et les grandes valeurs propres est très importante. Une petite erreur le long d'un vecteur propre (et donc une petite composante selon un autre vecteur propre) va en fait être multiplié par la valeur propre importante.

Estimation de la précision Estimer la précision minimale qu'il faut sur le vecteur propre pour ne pas avoir cette explosion de l'erreur.

On reparlera du conditionnement lors du TP 3.