Travaux dirigés n°4

Xavier JUVIGNY

12 février 2022

1 Exercice 1 : Algèbre linéaire

On veut écrire une petite bibliothèque d'algèbre linéaire dans les complexes permettant de faire différents produits matrice-vecteur avec les matrices suivantes :

- Une matrice pleine ;
- Une matrice tridiagonale ;
- Une matrice triangulaire inférieure.

ainsi qu'une fonction permettant d'approcher $y = e^{iA}.x \approx I + i.A.x - \frac{A^2}{2}.x - i\frac{A^3}{6}.x + \frac{A^4}{24}.x + i\frac{A^5}{120}.x$

On écrira les méthodes adéquates pour la classe vecteur et les classes représentant les matrices afin d'appeler la fonction approchant le produit de l'exponentielle complexe d'une matrice par un vecteur ainsi que l'affichage du résultat avec les diverses matrices.

Pour vérifier la fonction exponentielle, on peut prendre pour matrice A la matrice $-i.Id$ et vérifier si le produit exponentielle matrice-vecteur donne bien à la fin l'exponentielle de chaque composante du vecteur.

2 Exercice 2 : Gestion d'un nuage de points

On reprendra les classes de l'exercice sur le nuage de points du TD précédent.

L'idée de l'exercice est de pouvoir afficher "dynamiquement" des nuages de point sous n'importe quel angle. L'affichage dans cet exercice ne consistera qu'à afficher la boîte englobante alignée avec les axes (le min et max pour chaque composantes de tous les points compris dans un nuage). de chaque nuage de points.

La visualisation sera dynamique, c'est à dire qu'à chaque "affichage à l'écran", on avancera ensuite d'un petit pas de temps Δt. Ce petit pas de temps servira pour "mettre à jour" les nuages de points.

Il faudra gérer trois types de nuages de points :

- Les nuages de points statiques : une mise à jour du nuage de points ne fera rien.
- Les nuages de points dynamiques : chaque point du nuage est doté d'une accélération et d'une vitesse (la vitesse et l'accélération initiaux sont données à la construction). A chaque mise à jour, on met à jour pour chaque point sa vitesse et sa position (l'accélération n'est pas modifiée ici).
- Les nuages "fonctionels" : Une fonction donnée en argument du constructeur (cela peut être une classe ou un pointeur de fonction) permet de calculer à chaque instant la position de chaque point (par exemple, on peut penser à un nuage de point décrivant les sommets d'un triangle rectangle faisant une rotation autour de l'axe Oz par pas de 90 degré).

On écrira un programme qui visualisera plusieurs nuages de point en effectuant à chaque pas de temps une rotation autour de l'axe $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. Votre programme devra afficher à chaque pas de temps les boîtes englobantes de chaque nuage de points.

Rappel : Une rotation selon un axe unitaire u et un angle θ est décrit par la matrice de rotation suivante :

$$R = \begin{bmatrix} \cos(\theta) + u_x^2(1-\cos(\theta)) & u_xu_y(1-\cos(\theta)) - u_z\sin(\theta) & u_xu_z(1-\cos(\theta)) + u_y\sin(\theta) \\ u_xu_y(1-\cos(\theta)) + u_z\sin(\theta) & \cos(\theta) + u_y^2(1-\cos(\theta)) & u_yu_z(1-\cos(\theta)) - u_x\sin(\theta) \\ u_xu_z(1-\cos(\theta)) - u_y\sin(\theta) & u_yu_z(1-\cos(\theta)) + u_x\sin(\theta) & \cos(\theta) + u_z^2(1-\cos(\theta)) \end{bmatrix} \quad (1)$$