

On s'intéresse dans cet exercice à calculer la puissance n-ième d'une matrice le plus rapidement possible.

**Question 1)**

Programmer le produit entre deux matrices carrées. En déduire un algorithme qui permette de calculer la puissance n-ième d'une matrice.

**Question 2)**

Afin d'accroître les performances, on se propose dans un premier temps d'utiliser l'identité suivante :

$$\forall M, \forall n, M^{2n} = M^n \times M^n$$

Utiliser cette propriété pour calculer plus rapidement la puissance n-ième d'une matrice.

**Question 3)**

Si la matrice est diagonale, on remarque qu'il est très rapide de calculer sa puissance n-ième. Pour des matrices de taille 2, proposer un algorithme basé sur le court qui teste si la matrice est diagonalisable et dans un tel cas la diagonalise.

**Question 4)**

Utiliser cette transformation pour calculer la puissance n-ième des matrices diagonalisables. Proposer alors, avec des critères judicieux, une comparaison des performances pour le calcul de la puissance n-ième d'une matrice selon les trois méthodes possibles.

**Question 5)**

Proposer le dernier algorithme dans le cas des matrices de taille 3, voire plus. Comparer à nouveau les performances.

**Question 6)**

Ouverture: s'intéresser à un cas pratique où un tel calcul est justifié et montrer alors quelle méthode est la plus intéressante à utiliser, regarder l'impact des différentes méthodes sur les erreurs d'arrondis lors des calculs, laquelle des méthodes est préférable ?