

TD 4 : Algèbre linéaire

Ce TD présente les fonctions de Maple relatives à l'algèbre linéaire. L'algèbre linéaire est traditionnellement un domaine de prédilection pour les logiciels de calcul. Malheureusement, si de très nombreuses fonctions sont disponibles, elle sont aussi généralement très complexes, et il est ainsi difficile de les maîtriser.

1. Le package linalg

Maple possède un ensemble assez important de fonctions spécialement dédiées à l'algèbre linéaire. Ces fonctions sont réunies dans le 'package' `linalg`. On charge ce 'package' en entrant `with(linalg)`:

1.1 Matrices et vecteurs

Il y a plusieurs façons de rentrer une matrice/un vecteur. Par exemple (essayez ces exemples avec Maple):

```
A := array(1..2, 1..3, [[a, b, c],[d, e, f]]);  
v := vector([1, 2, 3]);  
C := matrix(4, 4, (i, j) -> i^(j-1));
```

D'autres fonctions permettent de rentrer des matrices d'un type particulier (de Vandermonde, antisymétriques, etc. On se reportera à l'aide de `array`), ou encore de construire des matrices à partir d'autres matrices (voir `diag`, `band`).

On peut aussi créer directement la matrice associée à un système d'équations linéaires à l'aide de la fonction `genmatrix`.

1.2 Arithmétique de base

Pour effectuer un calcul sur des matrices, il est nécessaire d'utiliser la fonction `evalm`. Ainsi, si A et B sont deux matrices de même taille, $A + B$; ne réalisera pas le calcul souhaité: il faut écrire `evalm(A + B)`. De plus, la multiplication des matrices doit être écrite à l'aide de `&*` et non `*`. Exemple: `evalm(A &* B)`. On peut aussi utiliser la fonction puissance `^` avec des matrices (et même des puissances négatives, si la matrice est inversible). Attention néanmoins: `evalm(A^0)`; renvoie `1`, et non pas `&*( )` qui est la notation utilisée par Maple pour la matrice identité.

1.3 Fonctions de Base

`trace` et `det`, `rank` calculent respectivement la trace, le déterminant et le rang d'une matrice; `charpoly` sont polynôme caractéristique, `eigenvalues`, ses valeurs propres, `eigenvectors`, ses vecteurs propres etc. Comme il y a plusieurs dizaines de fonctions de ce type, on évitera de les apprendre par cœur, et on se référera à l'aide.

1.4 Map

`map` est une fonction indispensable pour manipuler des matrices. Cette fonction permet d'appliquer une fonction à tous les éléments d'une matrice. Par exemple, si M est une matrice, `map(e -> e^2, M)`; mets tous les éléments de M au carré.

2. Exercices

2.1 Noyau, Image d'une application linéaire

Soit $A = X^4 - 1$, $B = X^4 - X$ et u l'application qui à un polynôme y fait correspondre le reste de la division euclidienne de Ay par B .

Montrer (avec Maple) que u est linéaire. Déterminer la matrice associée à u dans la base canonique de $R^3[X]$. On pourra utiliser pour cela la fonction suivante qui renvoie les coefficients d'un polynôme p de degré d :

```
poly2vect := (p, d) -> [seq(coeff(p, indets(p)[1], i), i=0..d)]:
```

Déterminer alors le noyau de u , son image, ses valeurs propres et ses vecteurs propres.

2.2 Calcul des puissances d'une matrice diagonalisable

$$\text{Soit : } M = \frac{1}{4} \begin{bmatrix} 0 & 1 & 3 \\ 3 & 0 & 1 \\ 1 & 3 & 0 \end{bmatrix}.$$

Calculer quelques puissances de cette matrice. En déduire une hypothèse sur $\lim_{n \rightarrow \infty} M^n$.

Afin de vérifier cette hypothèse, montrer que M est diagonalisable (avec Maple, pas à la main !), la diagonaliser (on appellera N la matrice diagonale) et déterminer une matrice de passage entre M et N (à l'aide des fonctions `eigenvector`, `transpose`, et `diag`).

Regarder la documentation de la fonction `jordan` et en déduire une méthode plus rapide pour réaliser l'opération précédente.

Calculer les puissances de N et en déduire celles de M .

Calculer $\lim_{n \rightarrow \infty} M^n$. On remarquera que Maple a bien du mal à réaliser ce calcul. Pour l'aider, penser à utiliser

`evalc` qui permet de transformer les expressions du type $(a + I \cdot b)^n$ en $c(n) + I \cdot d(n)$.

2.3 Commutant d'une matrice

$$\text{Soit : } A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix}.$$

Déterminer le commutant de la matrice A (c'est à dire l'ensemble des matrices commutant avec A). Montrer que ce commutant est l'ensemble des polynômes en A . Enfin, trouver les droites et les plans stables par A .

Penser à `convert(., set)` pour convertir une matrice en un ensemble d'équations que l'on peut résoudre. Réinjecter des solutions s'effectue à l'aide de la fonction `assign`.

2.4 Sommes de séries dont le terme général est un polynôme.

(Cet exercice traite bien d'algèbre linéaire et pas de sommation de séries)

Soit P un polynôme de degré d . Soit $Q(n) = \sum_{i=0}^n P(i)$. Sachant que Q est un polynôme de degré $d+1$ (vous pouvez

essayer de démontrer cette propriété, mais ce n'est pas indispensable pour réaliser l'exercice), déterminer les coefficients de Q à partir de ceux de P . Pour cela, on cherchera à obtenir un système d'équations linéaires dont la solution donne les coefficients recherchés.

Mettre en pratique cette méthode avec Maple. On créera pour cela une fonction qui attends P et renvoie Q (`vandermonde`, `linsolve`, `degree`, et `indets` si nécessaire). On pourra créer plusieurs fonctions auxiliaires, et notamment, utiliser la fonction suivante qui renvoie un polynôme en l'indéterminée donnée par x et dont les coefficients sont donnés par le vecteur v :

```
vect2poly := (v, x) -> add(v[i]*x^(i-1), i=1..op(2, op(2, eval(v)))):
```

(Note : Il existe une méthode analogue pour une large classe de somme de séries dont le terme général est une fraction rationnelle : la méthode de Gosper)