

Matrices

1 Se familiariser avec les matrices

► **Question 1** Charger le package `LinearAlgebra` avec `with(LinearAlgebra);`.

► **Question 2** Créer deux matrices A et B 3×3 de votre choix avec la fonction `Matrix`.

► **Question 3** Créer astucieusement une matrice identité, une matrice diagonale.

► **Question 4** Créer une superbe matrice C 10×10 aléatoirement remplie par Maple.

► **Question 5** Les objets `Matrix` sont mutables : pour les copier, il faut utiliser la commande `copy` et non pas faire une simple assignation `:=`. Copier A dans D .

► **Question 6** Comparer les matrices A et B d'une part, A et D d'autre part, à l'aide de la fonction `Equal`

► **Question 7** Expérimentez les opérations matricielles : addition, produit par un scalaire, produit matriciel (avec l'opérateur `.`), exponentiation, opérations sur chaque coefficient (fonction `Map`).

► **Question 8** Extraire le bloc 4×4 en haut à gauche de C .

► **Question 9** Calculer les traces de A et C .

► **Question 10** Calculer les déterminants de A et C .

► **Question 11** Calculer les inverses de A et C (si possible, si pas possible changer A et C).

► **Question 12** Effectuer un pivot de Gauss sur C .

► **Question 13** Calculer la transposée de C .

► **Question 14** Calculer les rangs de A et C .

► **Question 15** Calculer les noyaux de A et C .

► **Question 16** Calculer les images de A et C (hint : en anglais, l'image est le column space).

► **Question 17** Calculer les polynômes caractéristiques de A et C .

► **Question 18** Calculer les polynômes minimaux de A et C .

► **Question 19** Calculer les valeurs propres et vecteurs propres de A et C (une valeur propre en anglais, c'est une eigen value).

► **Question 20** Diagonaliser A et C si possible (ou trouver une jolie matrice à diagonaliser) à l'aide de la fonction `JordanForm` : $D, P := \text{JordanForm}(A, \text{output} = ['J', 'Q'])$; donne D et P tels que $A = P^{-1}DP$ si A est diagonalisable. Notons que la même chose marche pour la trigonalisation.

► **Question 21** Calculer la matrice adjointe de C .

► **Question 22** Expérimenter la résolution de problèmes du type $Ax = b$, A et b fixés, avec la fonction `LinearSolve`.

► **Question 23** Noter qu'on peut bien évidemment faire tout ça avec des variables formelles en lieu et place des nombres.

► **Question 24** Lire la doc du package `LinearAlgebra`. Comment ça, c'est déjà fait ?

2 De la pratique

► **Question 25** Donner une base du sous-espace de \mathbb{R}^5 défini par :

$$\begin{cases} x_1 + 2x_2 - x_3 + 3x_4 + x_5 = 0 \\ x_2 + x_3 - 2x_4 + 2x_5 = 0 \\ 2x_1 + x_2 - 5x_3 - 4x_5 = 0 \end{cases}$$

3 De la dynamique

► **Question 26** Implémenter une fonction donnant la plus grande sous-suite croissante d'une suite de n entiers. Par exemple, pour la suite 3, 1, 6, 2, 5, 3, 9, 8, 4, 9, 1, 7, 2, 10, cette sous-suite est 1, 2, 3, 4, 7, 10. On pourra, dans un premier temps, ne s'intéresser qu'à la longueur de cette sous-suite. Dans les deux cas, la fonction devra s'effectuer en $O(n^2)$ opérations.

Matrices

Un corrigé

- **Question 1** `with(LinearAlgebra);`
- **Question 2** `A := Matrix(3,3,[2,23,57,18,6,4,90,32,17]);`
`B := Matrix(3,3,[65,37,94,33,96,73,4,49,59]);`
- **Question 3** `I := IdentityMatrix(3);`
- **Question 4** `C := RandomMatrix(10);`
- **Question 5** `d := copy(A);`
- **Question 6** `Equal(A,d); Equal(A,B);`
- **Question 7** `A+B; A*2+B; A.B; Matrix(2,2,[0,1,1,1])^10; Map(x->x^3,A);`
- **Question 8** `C[1..4,1..4];`
- **Question 9** `Trace(A); Trace(C);`
- **Question 10** `Determinant(A);`
- **Question 11** `A^(-1); C^(-1);`
- **Question 12** `with(linalg); gausselim(C);`
- **Question 13** `transpose(C);`
- **Question 14** `rank(A); rank(C);`
- **Question 15** `NullSpace(A); NullSpace(C);`
- **Question 16** `ColumnSpace(A); ColumnSpace(C);`
- **Question 17** `Determinant(A-X.IdentityMatrix(3));`
`Determinant(C-X.IdentityMatrix(10));`
- **Question 18** `MinimalPolynomial(A,x);`
`MinimalPolynomial(C,x);`

- **Question 19** `Eigenvalues(A);`
`Eigenvalues(A);`
`Eigenvalues(C);`
`Eigenvalues(C);`
- **Question 20** `JordanForm(A,output = ['J','Q']);`
- **Question 21** `Adjoint(C); Adjoint(C).C;`
- **Question 22** `LinearSolve(<<0,1>|<1,1>>,<8,13>);`
- **Question 23** `LinearSolve(<<0,1>|<1,1>>,<X,Y>);`
- **Question 25** `NullSpace(<<1,0,2>|<2,1,1>|<-1,1,-5>|<3,-2,0>|<1,2,-4>>);`

$$\begin{pmatrix} 3 & 3 \\ -2 & -1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$
- **Question 26**

```

pluslongue := proc(v, n, min)
local p1,p2;
option remember;
if(n>nops(v)) then
return op([]);
else
if(v[n]<=min) then
return pluslongue(v,n+1,min);
else
p1 := pluslongue(v,n+1,min);
p2 := v[n],pluslongue(v,n+1,v[n]);
if(nops(p1)>nops(p2)) then
return p1;
else
return p2;
fi;
fi;
return 42;
end;
pluslongue([3, 1, 6, 2, 5, 3, 9, 8, 4, 9, 1, 7, 2, 10],1,-1);

```