TP Maple 8 | Outils de calcul et de réduction matriciels

Le lecteur est renvoyé au TP n° 5 pour la description de la bibliothèque **LinearAlgebra**. *Nous commencerons par exposer revenir sur quelques commandes essentielles avant d'exposer les outils de réduction des endomorphismes. Nous évoquerons pour finir la décomposition de* **Dunford** *et détaillerons un algorithme efficace dû à* **Claude Chevalley** *permettant de la calculer.*

1. Calcul matriciel avec LinearAlgebra

Maple met à disposition de l'utilisateur deux bibliothèques d'Algèbre linéaire : LINALG et LINEARALGEBRA. Cette dernière a été introduite depuis la version 7 du logiciel et vise à supplanter la précédente. Nous ne détaillerons ici que les commandes offertes par LINEARALGEBRA.

```
> with(LinearAlgebra);
```

[‘& x‘, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, LA Main, LUDecomposition, LeastSquares, LinearSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRDecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

1.1. Définition d'une matrice

Matrices remarquables

► `ZeroMatrix(n,p)` : crée la matrice nulle de taille (n,p) mais le résultat n'est accessible qu'en lecture.

► `Matrix(n,p)` : crée la matrice nulle de taille (n,p), accessible en lecture et en enregistrement contrairement au cas pérédent.

► `IdentityMatrix(n)` : crée la matrice I_n mais le résultat n'est accessible qu'en lecture.

► `DiagonalMatrix([`$d_1,\dots,d_n$`])` : crée la matrice diagonale dont les coefficients diagonaux sont égaux à $d_1,\dots,d_n$. Le résultat est accessible en lecture et en enregistrement.

```
> M:=ZeroMatrix(2): M[1,1]:=1: M;
Error, attempt to assign non-zero to zero Matrix entry
> M:=Matrix(2): M[1,1]:=1: M;
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Méthodes pour définir une matrice

► *Définition d'une matrice par une liste de lignes :*

`M := Matrix(L)`

où L est la liste des lignes (données sous forme de listes) de la matrice M.

► *Définition d'une matrice par une liste de colonnes :*

`M := convert(L,Matrix)`

où L est la liste des vecteurs colonnes de la matrice M (attention : ces vecteurs doivent être du type *Matrix*).

► *Définition d'une matrice par l'expression générale de ses coefficients :*

`M := Matrix(n,p,f)`

où f est une fonction de deux variables $(i,j) \mapsto f(i,j)$, n et p les dimensions de la matrice M de coefficients $f(i,j)$.

Extraction d'une sous-matrice

► La ligne de commande

```
SubMatrix(M,li..lf,ci..cf)
```

crée la matrice extraite de M définie par $li \leqslant i \leqslant lf$ et $ci \leqslant j \leqslant cf$.

► Si M est un objet du type *Matrix*,

```
M[li..lf,ci..cf]
```

désigne le bloc de M définie par $li \leqslant i \leqslant lf$ et $ci \leqslant j \leqslant cf$. Les lignes de commandes

`A := M[li..lf,ci..cf]` et `M[li..lf,ci..cf] := B`

ont pour effets respectifs d'enregistrer le bloc dans la variable A et de modifier le bloc de M en l'écrasant par une matrice B donnée dont la taille doit correspondre à celle du bloc sous peine d'un message d'erreur.

Ces outils permettent de définir des matrices par blocs sous Maple à partir de la matrice nulle. Voici comment procéder :

Définir une matrice par blocs

On commence par définir la matrice nulle et puis on la « *remplit* » par blocs :

`M := Matrix(n,p)` puis `M[l1..l2,c1..c2] := A`

Le bloc de la matrice M défini par $li \leqslant i \leqslant lf$ et $ci \leqslant j \leqslant cf$ est alors remplacé par A.

La commande `SubMatrix` admet une syntaxe plus générale permettant l'extraction d'un ensemble de lignes et de colonnes n'appartenant pas à un intervalle. On donne dans cas souys forme de listes les indices des lignes et des colonnes à extraire.

```
> M:=Matrix(4): A:=Matrix([[1,1],[1,2]]):
  M[1..2,1..2]:=A: M[3..4,1..2]:=A: M[3..4,3..4]:=A^(-1): A,M;
```

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 1 & 1 & 2 & -1 \\ 1 & 2 & -1 & 1 \end{bmatrix}$$

La commande `piecewise` est très utile pour définir des matrices particulières. Elle permet de définir une fonction par morceaux. On retiendra la syntaxe suivante pour définir une fonction f valant f_i si la condition cond_i est vérifiée (i variant de 1 à n) et valant f_{sinon} dans les autres cas [1].

La commande `piecewise` de définition « *par morceaux* »

$$\texttt{f := x -> piecewise(cond}_1\texttt{,f}_1\texttt{,cond}_2\texttt{,f}_2\texttt{,...,cond}_n\texttt{,f}_n\texttt{,f}_{\texttt{sinon}}\texttt{)}$$

Attention, la commande **piecewise** ne définit qu'une expression ! On retiendra également que les conditions $cond_i$ sont testées successivement dans l'ordre croissant des indices [2].

```
> f:=(i,j)->piecewise(i=j,1,i=1 or j=1,2,0): Matrix(3,3,f);
```

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}$$

1.2. Quelques outils

Quelques commandes utiles

Soit M un objet du type *Matrix*.

- `Determinant(M)` : renvoie le déterminant de M ;
- `Adjoint(M)` : renvoie la comatrice de M ;
- `Transpose(M)` : renvoie la matrice transposée de M ;
- `Convert(M,set)` : renvoie l'ensemble dont les éléments sont les coefficients de M.

Exercice 1. *Polynômes de matrices*

Il n'existe pas sous Maple de commande permettant le calcul de polynômes de matrices. Ecrire une procédure `PolyMat(P,M)` prenant en arguments un polynôme P et une matrice M et renvoyant la matrice $P(M)$.

1. La valeur par défaut de f_{sinon} est zéro.
2. Ce qui permet de définir une fonction par morceaux d'une autre manière qu'en mathématiques (où l'on utilise une partition de l'ensemble de définition pour définir une fonction par morceaux).

Exercice 2.

Soit A la matrice
$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$
Ecrire une procédure `psi(X,Y)` prenant en arguments deux vecteurs colonnes de taille 3 (du type *Matrix*) et renvoyant le déterminant suivant :
$$\begin{vmatrix} A & X \\ {}^tY & 0 \end{vmatrix}$$

Exercice 3. *Posé à Centrale dans la filière MP en 2010*

Soit $(F_n)_{n\in\mathbb{N}}$ la suite de Fibonacci :
$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n \quad \text{pour tout } n \in \mathbb{N} \end{cases}$$
Soit $n \in \mathbb{N}^*$. Calculer le déterminant de la matrice $\left(F_{|i-j|}\right)_{1\leqslant i,j\leqslant n}$.

Exercice 4. *D'après une planche posée à Centrale dans la filière MP en 2010*

Pour toute matrice A de $\mathfrak{M}_n(\mathbb{C})$, on note $Com(A)$ la comatrice de A.

1. Soient $n \in \mathbb{N}^*$ et A_n la matrice $\left((i+j)^3\right)_{1\leqslant i,j\leqslant n}$.

1.a. Ecrire une procedure `A(n)` prenant en argument $n \in \mathbb{N}^*$ et renvoyant la matrice A_n.

1.b. Calculer le rang de A_n et de $Com(A_n)$ pour $2 \leqslant n \leqslant 7$.

2. Pour $A \in \mathfrak{M}_n(\mathbb{C})$, déterminer le rang de $Com(A)$ en fonction de celui de A.

3. Soient A et B dans $\mathfrak{M}_n(\mathbb{C})$. Etablir que $Com(A\cdot B) = Com(A)\cdot Com(B)$.

4. Soient
$$A = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad \text{et} \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4.a. Trouver, sans utiliser le logiciel, P et Q dans $GL_3(\mathbb{C})$ telles que $B = P\cdot A\cdot Q$.

4.b. Reprendre la question précédente en utilisant Maple.

4.c. Montrer l'existence de $M \in \mathfrak{M}_3(\mathbb{R})$ telle que $A = Com(M)$.

5. Soit $M \in GL_n(\mathbb{C})$. Montrer l'existence de $A \in GL_n(\mathbb{C})$ telle que $M = Com(A)$.

6. Déterminer
$$\left\{ Com(A) \mid A \in \mathfrak{M}_n(\mathbb{C}) \right\}$$

2. Réduction des endomorphismes

Le logiciel permet le calcul des éléments propres d'une matrice.

```
> A := Matrix([[1,-1,-1],[-2,2,3],[2,-2,-3]]):
   A, CharacteristicMatrix(A,lambda),
   factor(CharacteristicPolynomial(A,lambda)),
   Eigenvalues(A); Eigenvectors(A);
```

$$\begin{bmatrix} 1 & -1 & -1 \\ -2 & 2 & 3 \\ 2 & -2 & -3 \end{bmatrix}, \begin{bmatrix} -1+\lambda & 1 & 1 \\ 2 & -2+\lambda & -3 \\ -2 & 2 & 3+\lambda \end{bmatrix}, (\lambda-1)\lambda(\lambda+1), \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix}$$

Outils de réduction

- ► **CharacteristicMatrix(A,lambda)** : $\lambda I_n - A$.
- ► **CharacteristicPolynomial(A,lambda)** : $\det(\lambda I_n - A)$.
- ► **Eigenvalues** : renvoie les valeurs propres (comptées avec leur multiplicité algébrique, i.e. leur multiplicité en tant que racine du polynôme caractéristique) sous la forme d'un vecteur-colonne.
- ► **Eigenvectors** : renvoie les valeurs propres sous la forme d'un vecteur-colonne (cf. ci-dessus) *puis*, sous forme d'une matrice de taille n, des familles génératrices des sev propres.
- ► **MinimalPolynomial(A,lambda)** : calcule le polynôme minimal de A.

Les familles génératrices seront toujours composées d'une base du sous-espace propre correspondant complétée par des vecteurs nuls pour atteindre la multiplicité algébrique lorsque la multiplicité géométrique (i.e. la dimension du sev propre considéré) est strictement inférieure à la multiplicité algébrique.

```
> M:=Matrix([[1,1,0,0],[0,1,0,0],[0,0,2,1],[0,0,0,2]]):
  Eigenvalues(M), Eigenvectors(M);
```

$$\begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Dans le cas d'une matrice diagonalisable, on se souviendra que la matrice renvoyée en deuxième argument par **Eigenvectors** est la matrice de passage de la base canonique de $\mathbb{R}^n$ à la base de vecteurs propres trouvée par le logiciel.

```
> m:=Matrix([[-2,1,1],[1,-2,1],[1,1,-2]]);
```

$$m := \begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

```
> d, p := Eigenvectors(m)[1], Eigenvectors(m)[2];
```

$$d, p := \begin{bmatrix} 0 \\ -3 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

```
> p.DiagonalMatrix(Transpose(d)).p^(-1);
```

$$\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

On vérifie bien que $m = p \cdot d \cdot p^{-1}$.

Exercice 5.

Soient a un réel non nul et f l'endomorphisme de $\mathbb{R}^3$ dont la matrice dans la base canonique est

$$A = \begin{pmatrix} -1 & a & a \\ 1 & -1 & 0 \\ -1 & 0 & -1 \end{pmatrix}.$$

1. f est il diagonalisable sur $\mathbb{R}$, trigonalisable sur $\mathbb{R}$?

2. Au moyen du logiciel, trouver une base $\mathscr{B}$ de $\mathbb{R}^3$ dans laquelle la matrice de f soit égale à

$$M = \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

3. On se propose de généraliser cette étude en considérant, pour $n \geqslant 2$, les matrices de la forme

$$A_n = \begin{pmatrix} -1 & a & a & \dots & a \\ 1 & -1 & 0 & \dots & 0 \\ -1 & 0 & -1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ (-1)^n & 0 & 0 & \dots & -1 \end{pmatrix} \in \mathfrak{M}_n(\mathbb{R})$$

avec $a \in \mathbb{R}$.

3.a. Etudier le cas où $a = 0$ (diagonalisabilité ou, à défaut, trigonalisabilité, comment obtenir une matrice de passage).

3.b. Dorénavant $a \neq 0$. Ecrire une procédure `A(n,a)` qui prend en argument un entier $n \geqslant 3$, un paramètre a, et retourne la matrice A_n donnée ci-dessus. Utiliser alors le logiciel pour émettre quelques conjectures sur la digonalisabilité et la trigonalisabilité de A_n sur $\mathbb{R}$ puis les démontrer.

Exercice 6. *D'après une planche posée à l'oral de Centrale dans la filière MP en 2010*

Pour tout entier naturel $n \in \mathbb{N}^*$, on note A_n la matrice de $\mathfrak{M}_n(\mathbb{R})$ la matrice de diagonale nulle et dont, pour tout $1 \leqslant j \leqslant n$, les termes de la j-ème colonne autres que le terme diagonal sont tous égaux à j.

1. Ecrire une procédure `A(n)` prenant en argument $n \in \mathbb{N}^*$ et renvoyant la matrice A_n.

2. Calculer les valeurs propres de A_n pour $n \in \{2,3,4,10\}$.

3. Montrer que les valeurs propres de A_n sont les solutions de l'équation

$$\sum_{k=1}^{n} \frac{k}{x+k} = 1$$

4. La matrice A_n est-elle diagonalisable ?

5. Soit $n \geqslant 2$. On note x_n l'unique valeur propre de A_n appartenant à $]-2,-1[$. Etudier la monotonie et la convergence de $(x_n)_{n\geqslant 2}$.

6. Donner un développement asymptotique à deux termes de x_n.

3. Décomposition de Dunford

Dans ce paragraphe, E désigne un espace vectoriel de dimension finie sur le corps $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$.

3.1. La théorie

Soit $f \in \mathscr{L}(E)$ de polynôme minimal

$$\mu_f = \prod_{\ell=1}^{k} (X - \lambda_\ell)^{\alpha_\ell}$$

avec les λ_ℓ, $1 \leqslant \ell \leqslant k$ deux à deux distincts. Puisque les polynômes $(X-\lambda_\ell)^{\alpha_\ell}$ sont deux à deux premiers, on déduit du théorème chinois dans $\mathbb{K}[X]$ l'existence de polynômes $P_1, P_2, \ldots, P_k$ de $\mathbb{K}[X]$ tels que

$$\forall 1 \leqslant \ell \leqslant k, \quad \forall 1 \leqslant j \leqslant k, \quad P_\ell \equiv \delta_{\ell,j} \left[\left(X - \lambda_j\right)^{\alpha_j} \right]$$

On vérifie que les endomorphismes $P_\ell(f)$ sont les projections sur $\operatorname{Ker}\left(\left(f - \lambda_\ell id\right)^{\alpha_\ell}\right)$ parallélement à

$$\bigoplus_{j\neq\ell} \operatorname{Ker}\left(\left(f - \lambda_j id\right)^{\alpha_j}\right)$$

On prouve alors facilement qu'en posant

$$\begin{cases} d = \sum_{\ell=1}^{k} \lambda_\ell \cdot P_\ell(f) \\ \\ n = f - d \end{cases}$$

n est nilpotent, d diagonalisable, $d \circ n = n \circ d$ et $f = d + n$.

Nous venons d'établir le théorème suivant :

Théorème 1. (Décomposition de Dunford)
Soient E un espace vectoriel de dimension finie et f un endomorphisme de E dont le polynôme caractéristique est scindé. Alors, il existe un unique couple (d, n) d'endomorphismes de E tels que d est diagonalisable, n est nilpotent, d et n commutent et $f = d + n$. De plus, d et n sont des polynômes en f : $f = d + n$ est la décomposition de Dunford de f.

Mettons en œuvre cette méthode sur un exemple.

```
> A := Matrix([[5/2,1,1/2],[1/2,2,-1/2],[-1/2,-1,3/2]]);
```

$$A := \begin{bmatrix} 3/2 & 1 & -1/2 \\ 2/2 & 2 & 1/2 \\ -1/2 & -1/2 & 3/2 \end{bmatrix}$$

```
> factor(CharacteristicPolynomial(A,X));
```

$$(X-1)^2(X-3)$$

```
> gcdex((X-1)^2,X-3,X,'U','V'):
  P3,P1:=U*(X-1)^2,V*(X-3):
  Diago:=1*PolyMat(P1,A)+3*PolyMat(P3,A):Nilpo:=A-Diago:Diago,Nilpo;
```

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \quad \begin{bmatrix} -1/2 & 0 & -1/2 \\ 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}$$

```
> Nilpo^3,Eigenvectors(Diago)[2];
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Dans cette séquence, on a commencé par définir une matrice A dont on calcule ensuite le spectre. La commande `gcdex` renvoie les coefficients U et V d'une relation de Bezout

$$(X-1)^2 \cdot U + (X-3) \cdot V = 1$$

Les polynômes $P_3 = U\cdot(X-1)^2$ et $P_1 = V\cdot(X-3)$ sont donc solutions du système de congruences suivant [3] :

$$\begin{cases} P_1 & \equiv 1\ \left[(X-1)^2\right] \\ P_1 & \equiv 0\ [(X-3)] \end{cases},\quad \begin{cases} P_3 & \equiv 0\ \left[(X-1)^2\right] \\ P_3 & \equiv 1\ [(X-3)] \end{cases}$$

Le reste n'est que l'application des formules données dans la preuve de ci-dessus. On vérifie pour finir que les matrices $Nilpo$ et $Diago$ sont respectivement nilpotente et diagonalisable. Remarquons que cette démonstration n'est *constructive* que si les valeurs propres de f sont connues. Dans la pratique, c'est loin d'être le cas car on ne sait pas calculer explicitement les racines du polynôme caractéristique. Il reste donc à élaborer une méthode constructive de la décomposition de Dunford.

3.2. L'algorithme de Claude Chevalley

L'algorithme que nous présentons ici est adapté de la célèbre méthode de Newton pour le calcul approché d'une solution de $\phi(x) = 0$ ou ϕ est une fonction dérivable de dérivée ne s'annulant pas. Il est dû à Claude Chevalley [4]. On définit une suite par récurence par la fomule :

$$x_{n+1} = x_n - \frac{\phi(x_n)}{\phi'(x_n)}$$

Dans les bons cas $(x_n)_{n\geqslant 0}$ converge. La limite est alors une solution de l'équation $\phi(x) = 0$.

Revenons à la décomposition de Dunford de $f \in \mathscr{L}(E)$: le point de départ est la constation que les racines du polynôme minimal de d sont simples et sont les valeurs propres de f. Notons $P = \chi_f$ le polynôme caractéristique de f et $Q = \frac{P}{P\wedge P'}$. On a alors $Q(d) = 0$.

Théorème 2. (Décomposition effective de Dunford)
Soient E un $\mathbb{K}$-ev de dimension finie, $f \in \mathscr{L}(E)$, $P = \chi_f$ et $Q = \dfrac{P}{P\wedge P'}$. La suite d'endormorphismes définie par

$$\begin{cases} u_0 & = f \\ u_{n+1} & = u_n - Q(u_n)\cdot Q'(u_n)^{-1} \end{cases}$$

est bien définie et stationnaire. On note d sa limite et on pose $n = f - d$. Alors n est nilpotent, d est diagonalisable et ces deux endomorphismes commutent.

Exercice 7. *Preuve de la décomposition de Dunford effective*

On reprend les notations précédentes.

1. Montrer qu'il existe un couple $(U, V) \in \mathbb{K}[X]$ tel que $UP + VQ' = 1$.
2. On définit une suite $(S_n)_{n\in\mathbb{N}}$ de polynômes de $\mathbb{K}[X]$ en posant :

$$\begin{cases} S_0 & = X \\ S_{n+1} & = S_n - (Q\circ S_n)\cdot(V\circ S_n) \end{cases}$$

3. Il existe une procédure pour le calcul des restes chinois (`chrem`, pour *chinese remainder* mais elle n'effectue que des caculs dans l'anneau $\mathbb{Z}$.

4. Elève de l'Ecole Normale Supérieure, Claude Chevalley (1909-1984) fût un des membres fondateurs du groupe Bourbaki.

2.a. Etablir que

$$\forall n \in \mathbb{N},\ \ Q \circ S_n \equiv 0 \left[Q^{2^n}\right]$$

2.b. Montrer que

$$\forall n \in \mathbb{N},\ \ X - S_n \equiv 0\,[Q]$$

2.c. En déduire l'existence d'un entier naturel n_0 tel que

$$\begin{cases} f - S_{n_0}\left(f\right) \text{ est nilpotent} \\ Q\left(S_{n_0}\left(f\right)\right) = 0 \end{cases}$$

3. En déduire le théorème 2.

Exercice 8. *Calcul de la décomposition de Dunford*

Ecrire une procédure `Dunford(A)` d'argument A, matrice carrée, renvoyant la décomposition de Dunford de A.