Université Lyon 1 Année 2020-2021
M1G - UE Logiciels

TP6 – Bases de Sage 2

1 Algèbre linéaire

Exercice 1 - Corps de base

SAGE utilise les notations suivantes :
- `ZZ` : anneau $\mathbb{Z}$ des entiers relatifs
- `QQ` : corps $\mathbb{Q}$ des nombres rationnels,
- `RR` : corps $\mathbb{R}$ des nombres réels (avec 53 bits de précision par défaut),
- `CC` : corps $\mathbb{C}$ des nombres complexes (avec 53 bits de précision par défaut),
- `SR` : "corps"des expressions symboliques.

Le choix peut être imposé lors de la création de la matrice. Si le corps de base n'est pas précisé, SAGE choisit le plus "petit" possible.
On peut ainsi définir les matrices et les vecteurs comme suit

```
matrix(2,2,[1,2,3,4])
matrix(QQ,2,2,[1,2,3,4])
matrix(RR,2,2,[1,2,3,4])
matrix(2,2,[1,I,-1,1+I])
matrix(CC,2,2,[1,I,-1,1+I])
```

Vérifier le type de chacun de ses matrices à l'aide de la fonction `parent`.

Exercice 2 - Outils de construction de matrices

Comment sont construites les matrices suivantes ?

```
identity_matrix(4)
identity_matrix(QQ,4)
identity_matrix(RR,4)

matrix(QQ,3, lambda i,j: i-j)
matrix(QQ,3,4, lambda i,j: i-j)
```

On peut également construire des matrices sur des corps finis connus. On rappelle la notation $GF(2) = \mathbb{Z}/2\mathbb{Z}$ le corps à 2 éléments.
Que font les commandes suivantes ?

```
random_matrix(ZZ,10,10)
random_matrix(RR,10,10)
random_matrix(GF(2),10,10);

matrix(3, 4, range(12))
matrix(GF(2), 3, 4, range(12))
```

On peut aussi définir les matrices comme suit

```
M3Q = MatrixSpace(QQ,3)
M3Q([1,2,3,3,2,1,1,1,1])
```

Avec cette définition de `M3Q`, que font les commandes suivantes

```
M3Q(0)
M3Q(1)
M3Q(2)
M3Q.random_element()
M3Q.random_element(num_bound=10,den_bound=10)
```

Exercice 3 - Manipulation des matrices

On considère les matrices suivantes :

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, B = \begin{bmatrix} 0 & 1 \\ -1 & 5 \end{bmatrix}$$

1. Définir les matrices A et B.
2. Regarder la liste les méthodes disponibles pour les objets de type matrice en entrant `A.` et en appuyant sur la touche `<Tab>`.
3. Calculer $2A - B$, AB, ${}^tB$, $\det(A)$ et $\det(B)$.
4. Calculer le noyau de A et l'inverse de B.
5. Que fait la commande `A.augment(B)` ?
6. Calculer A^5 et A^n pour un n quelconque. Évaluer cette dernière expression pour $n = 5$ et comparer avec la première expression.

Exercice 4 - Manipulation des vecteurs

On considère les vecteurs et la matrice suivants :

$$u = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, v = \begin{bmatrix} 2 \\ -3 \\ 1 \end{bmatrix}, A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

1. Définir les vecteurs u et v ainsi que la matrice A avec la commande `matrix`.
   $\hookrightarrow$ *Utiliser les fonctions* `vector` *et* `matrix`.
2. Calculer $u \cdot v$, $u \wedge v$, Au et $A^3(u - v)$.

Exercice 5 - Résolution de systèmes linéaires

La résolution d'un système linéaire $Ax = b$ s'effectue de trois manières différentes :
- En utilisant la méthode `solve_right` : `A.solve_right(b)`
- En utilisant l'opérateur d'inversion de matrice : `A\b`
- En inversant la matrice : `A^(-1)*b` ou `~A*b`

Les deux dernières méthodes font exactement la même chose, et sont à éviter en général car le coût de cette résolution est très élevé. En revanche, elles fonctionnent à coup sûr (!!) pour des matrices contenant des variables symboliques.

Résoudre les systèmes d'équations pour les A et b suivants :

1. $A = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -1 & -2 \\ 1 & -2 & 4 \end{bmatrix}$ et $b = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}$.
2. $A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ et $b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.
   Que peut-on dire de la solution obtenue ?
3. $A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$ et $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
4. $A = \begin{bmatrix} 1 & -a & a^2 \\ a & -a^2 & a \\ a & 1 & -a^3 \end{bmatrix}$ et $b = \begin{bmatrix} a \\ 1 \\ 1 \end{bmatrix}$.
   Résoudre et discuter sur $\mathbb{C}$.

Exercice 6 - Éléments propres

On considère la matrice suivante

$$A = \begin{bmatrix} 3 & -2 & 1 & -1 \\ 0 & 1 & 0 & 0 \\ -4 & 2 & -1 & 2 \\ -2 & 1 & -1 & 2 \end{bmatrix}$$

1. Déterminer les valeurs et les vecteurs propres de la matrices A.
   $\hookrightarrow$ *Utiliser les méthodes* `eigenvalues` *et* `eigenvectors_right`.
2. Calculer le polynôme minimal et le polynôme caractéristique associés à la matrice A, et les factoriser pour retrouver le spectre de A.
   $\hookrightarrow$ *Utiliser les fonctions* `minpoly` *et* `charpoly`.
3. Calculer la décomposition de Jordan de A.
   $\hookrightarrow$ *Utiliser la méthode* `jordan_form`.

---

2 Approximations numériques

Exercice 1 - Développement de Taylor

On étudie la fonction $f(x) = x\cos(x)$ sur l'intervalle $[-5, 5]$.

1. Calculer le développement de Taylor de f à l'ordre 10 en 0.
   $\hookrightarrow$ *Utiliser la commande* `taylor`.
2. Tracer simultanément les graphes de f et de son développement de Taylor sur l'intervalle $[-3, 3]$, puis sur l'intervalle $[-5, 5]$. Commenter les résultats obtenus.

Exercice 2 - Interpolation de Lagrange

Considérons une fonction f définie, continue sur un intervalle $[a, b]$, et soient x_i, $i = 0, ..., N$, des points distincts appartenant à cet intervalle.
On considère $N = 10$.

1. Générer une liste `xset` de $N+1$ points x_i équirépartis sur l'intervalle $[-5, 5]$
2. Générer la liste `yset` des points $f(x_i)$, avec $f : x \mapsto x\cos(x)$.
3. Créer une liste `L` telle que $L = [(x_0, f(x_0)), ..., (x_i, f(x_i)), ..., (x_N, f(x_N))]$.
4. Calculer le polynôme d'interpolation P_N de Lagrange associé à la liste de points `L`.
   $\hookrightarrow$ *Utiliser la méthode* `lagrange_polynomial` *associée à la classe* `PolynomialRing`.
5. Représenter sur un même graphique les fonctions $f(x)$ et son polynôme d'interpolation, sur l'intervalle $[-5, 5]$ puis sur l'intervalle $[-7, 7]$. Commenter les résultats obtenus.

Exercice 3 - Intégration numérique

On considère l'intégrale $I = \displaystyle\int_0^{+\infty} \frac{\mathrm{d}x}{\sqrt{x^3 + 3x^2 + 2x}}$

1. Est-il possible de calculer I de manière exacte ?
2. Calculer I de manière approchée à au moins 10^{-10} près.
   $\hookrightarrow$ *Utiliser la fonction* `numerical_integral` et ses options.

Exercice 4 - EDO

1- À l'aide de la commande `desolve`, résoudre les équations différentielles suivantes

1. $(x+1)y' - xy = 0$
2. $(1+x^2)y' + xy = 1, y(0) = 1$
3. $\begin{cases} y'' - 3y' + 2y = xe^x + \sin(x) \\ y(0) = y(1) = 1 \end{cases}$
4. $y'\sqrt{1-x^2} - y^2 - 1 = 0$

Que pouvez-vous dire de la réponse du logiciel donnée pour la dernière équation ?

2- À l'aide de la commande `desolve_system`, résoudre le système d'équations différentielle suivant :

$$\left\{\begin{array}{l} x'(t) = 3x(t) - 4y(t) \\ y'(t) = x(t) + 3y(t) \\ x(0) = 2,\ y(0) = 0 \end{array}\right.$$

3- Lorsqu'il n'est pas possible de trouver une solution exacte à une EDO, il est possible d'utiliser un solveur numérique, par exemple `desolve_rk4` qui utilise la méthode de Runge-Kutta d'ordre 4.
On considère l'équation différentielle suivante :

$$\left\{\begin{array}{l} x^2y'(x) - 2y(x) = 0 \\ y(t_0) = y_0 \end{array}\right. .$$

1. Tracer le champ de vecteur $[x^2, 2y]$ pour $(x, y) \in [-2, 2] \times [-2, 2]$ en utilisant la commande `plot_vector_field`.
2. Sur la figure précédente, ajouter les trajectoires obtenues avec la commande `desolve` pour $t_0 = 1$ et $t_0 = -1$, avec y_0 entre -2 et 2 dans les deux cas, avec par exemple un pas de 0.5. Attention lors du tracé, la solution y obtenue est définie pour les $x > 0$ dans le premier cas, et pour les $x < 0$ dans le deuxième cas.
3. Même question en utilisant `desolve_rk4` avec un pas de 0.05. Utiliser la commande `line` pour tracer chacune des trajectoires.
4. Commenter les différences.
