TP1 : Introduction à Sage

Démarrer SAGE : Au CMI, taper `sage -n jupyter` dans la console. A la maison, possibilité de travailler en ligne sur les serveurs de *cocalc.com*, ou d'installer sa propre distribution à partir de *sagemath.org*.

Bibliographie : Calcul mathématique avec Sage, nombreux coauteurs.

1 Avant de commencer

Aide en ligne et complétion automatique

1. Imaginons que vous ne vous souvenez plus comment construire une matrice. Tapez `matr` puis TAB, sélectionnez `matrix` et exécutez finalement la commande `matrix?`.Comparez avec `Matr`.
2. Vous voulez à présent construire un anneau de polynômes. Testez `polyn` TAB `polynomial?` Retentez votre chance avec `Polyn` ... Morale ?

Soyez clair :

— Commentez votre code #.
— Ajoutez des titres "markdown".
— Illustrez autant que possible avec des graphiques.
— Réfléchissez à vos noms de variables/fonctions (ex : p pour un nombre premier, *cpt* pour un compteur, *publickey* pour RSA, etc.)
— Utilisez `print` pour mettre en valeur vos résultats finaux.

2 Rappels de programmation

Assurez-vous sur des exemples simples, et en vous aidant au besoin de l'aide en ligne, que vous savez manipuler :

— les listes [ ] (obtenir la longueur, ajouter/retirer un élément, extraire le 3ieme élement, liste vide, concaténer deux listes...)
— les tuples ( )
— les conditions et les booléens (`if, elif, else, true, false`)
— les boucles `for, while`.
— les fonctions `def`.
— les objets graphiques : graphe d'une fonction prédéfinie `plot`, graphe d'une fonction empirique `point` (nuage de points) ou `line` (ligne brisée), graphe d'une fonction implicite, courbe paramétrée, histogramme `bar_chart`. Modifier les titres, couleurs, nom des axes, superposer deux courbes, etc. Utiliser `show( )` pour gérer les paramètres d'affichage.

Quelques exercices

1. Ecrire une fonction *f_syra* qui, à tout entier $n \in \mathbb{N}$, associe $n/2$ si n est pair, $3n+1$ sinon.
2. Ecrire une fonction *premier_element* qui, à toute liste (éventuellement vide), associe son premier élément.

3. Ecrire une fonction $nb\_apparition3$ qui, à toute liste, indique le nombre de 3 qu'elle contient. La tester sur des listes d'entiers aléatoires entre 0 et 9 (inclus) de taille 10, 100, 1000 et 10000. Illustrer par un histogramme.
4. Tracer les 100, 1000 puis 10000 premiers termes de la suite $(cos(n))_n$. Que remarquez-vous ? Est-ce vraiment périodique ?
5. On rappelle que la suite de Fibonacci est donnée par : $u_0 = 1$, $u_1 = 1$ puis $u_{n+2} = u_{n+1} + u_n$ pour tout $n \geq 2$. Ecrire deux fonctions Fibo_recur et Fibo_iter qui calculent le n-ieme terme de la suite de Fibonacci de manière respectivement récursive et itérative. Comparer leurs exécutions pour $n = 10, 35, 100, 1000$. Pourquoi ?

♠ Pour aller plus loin : la conjecture de Syracuse

On considère la fonction f_syra décrite ci-dessus. Il semblerait que pour tout entier naturel de départ n, l'orbite de n pour f_syra, c'est-à-dire la suite $(f\_syra^k(n))_{k\in\mathbb{N}}$ soit périodique à partir d'un certain rang, de période $1-4-2$. Prospectez et illustrez vos résultats.

Quelques pistes :
— Tracer l'orbite de quelques points au hasard.
— Tracer "le temps de vol" de n (c'est-à-dire le nombre d'étapes avant d'atteindre la période), en fonction de n.
— Tracer "la hauteur de vol" (le plus grand entier de l'orbite de n.)
— Projeter sur $\mathbb{N}/m\mathbb{N}$
— etc.

3 Spécificités de SAGE

3.1 Calcul exact vs calcul approché, instabilité numérique

1. Tester les commandes suivantes : `20/6`, `20.0/6`, `numerical_approx(20/6)`, `numerical_approx(20/6, digits=60)` ; puis 2^{1000} et `numerical_approx(2`^{1}
Que constatez-vous ?
2. Comparer les résultats de $(x+10^{50})-10^{50}$ avec $x = 1$ et $x = 1.0$; puis, avec précision explicite, refaire le calcul pour `x=numerical_approx(1,digits=49)` et `x=numerical_approx(1,digits=48)`.

3.2 Calcul symbolique :

♣ `var('n'), sum, expand(f), f.collect(a), f.substitute(a=i), f.coefficient()`

1. Écrire puis simplifier $1 + 4 + \cdots + n^2$.
2. Calculer
$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$
3. Développer $(1 + x)^{15}$ puis dériver et factoriser le résultat obtenu.
4. Développer $(x^2 + ax + b)^5$. Organiser les termes suivant les puissances de a. Remplacer la valeur de a par 5. Extraire le coefficient en x^3.

3.3 Structures algébriques :

Polynômes sur un anneau

♣ `anneau_de_pol.<variable(s)_polynomiales>=anneau_de_base[ ]`

Ex : Pols_entiers.<X>=ZZ[], Pols_rationnels_multivariés.<X,Y,Z>=QQ[], etc

1. Développer le polynôme $P = X(2X-1)(X^2-2)(X^2+1)$.
2. Calculer ses racines sur $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ puis sur $\mathbb{C}$.
3. Le factoriser dans chacun de ces anneaux.

Matrices

Construire la matrice M

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

1. Extraire les coefficients 1,1, puis 3,2.
2. Calculer sa transposée, son rang, son déterminant.
3. Calculer son polynôme minimal, ses valeurs propres et vecteurs propres associés.

3.4 Résolution de systèmes d'équations

♣ `x,y,z = var('x,y,z'), solve` ♣

Soit le système :

$$\begin{cases} x+y+z & = & 1 \\ x-y+z & = & 2 \\ x^2+1/y-1/z & = & 2 \end{cases}$$

1. Le résoudre dans $\mathbb{C}$ en utilisant `solve`.