Manipulation de suites P-récursives avec SageMath

Encadrant : Marc Mezzarobba
Nombre d'étudiants : 2

Une suite (disons, de nombres complexes) $(u_n)_{n\in\mathbb{N}}$ est dite *P-récursive* (ou *P-finie*) si elle est solution d'une récurrence linéaire dont les coefficients sont des polynômes en n :

$$p_s(n)\, u_{n+s} + \cdots + p_1(n)\, u_{n+1} + p_0(n)\, u_n = 0.$$

Cette récurrence permet de calculer u_n pour tout n à partir d'un nombre fini de conditions initiales. Ainsi, bien qu'une suite P-récursive comporte un nombre infini de termes, il est possible de la représenter exactement : il suffit de donner la récurrence et les conditions initiales correspondantes.

Par exemple, la suite de Fibonacci $(F_n) = (0, 1, 1, 2, 3, 5, 8, \ldots)$ est P-récursive puisqu'elle satisfait

$$F_{n+2} - F_{n+1} - F_n = 0, \qquad F_0 = 0, F_1 = 1.$$

De même, la suite $(n!)_{n\in\mathbb{N}}$ est P-récursive et peut être définie par la récurrence $(n+1)! = (n+1)\cdot n!$ accompagnée de la condition initiale $0! = 1$. De très nombreuses autres suites classiques qui apparaissent en combinatoire et dans divers autres domaines des mathématiques sont aussi P-récursives.

Les suites P-récursives forment un anneau : on peut montrer que la somme $(u_n + v_n)_{n\in\mathbb{N}}$ et le produit $(u_n\, v_n)_{n\in\mathbb{N}}$ de deux suites P-récursives $(u_n)_{n\in\mathbb{N}}$ et $(v_n)_{n\in\mathbb{N}}$ sont encore des suites P-récursives. Par exemple, la suite $(w_n)_{n\in\mathbb{N}}$ définie par $w_n = n! + F_n$ vérifie

$$n\,(n+2)\, w_{n+3} - (n^3 + 6\, n^2 + 9\, n + 3)\, w_{n+2} + (n^3 + 4\, n^2 + 5\, n + 3)\, w_{n+1} + (n+3)\,(n+1)^2\, w_n = 0.$$

De plus, à partir de représentations de $(u_n)_{n\in\mathbb{N}}$ et $(v_n)_{n\in\mathbb{N}}$ par récurrence et conditions initiales, on peut calculer automatiquement la récurrence somme ou produit et les conditions initiales correspondantes.

L'objet de ce projet est de développer un module de calcul avec les suites P-récursives pour SageMath (ou simplement Sage), un système de calcul formel libre écrit en langage Python utilisé par des milliers de mathématiciens. Vous devrez dans un premier temps vous familiariser avec les propriétés mathématiques des suites P-récursives et avec la programmation en Python dans le cadre de Sage. Vous ajouterez ensuite à Sage un type d'objet « suite P-récursive » qui devra rendre aussi naturels que possible pour l'utilisateur les calculs avec ces suites. Par exemple, si `u` et `v` sont des objets de type « suite P-récursive », l'expression `u + v` devra calculer un objet « suite P-récursive » représentant la somme des deux suites, `u[42]` devra calculer (aussi efficacement que possible) le terme d'indice 42 de la suite $(u_n)_{n\in\mathbb{N}}$, etc. Vous pourrez vous appuyer pour cela sur la bibliothèque Sage `ore_algebra`, qui fournit des implémentations de tous les algorithmes de base nécessaires (comme le calcul d'une récurrence pour la somme à partir de deux récurrences).

Suivant son état d'avancement, le code développé pourra être soumis pour inclusion dans SageMath à la fin du projet.

Ce stage demande une bonne connaissance du langage Python, et s'adresse en priorité à des étudiants curieux de contribuer au développement d'un « gros » logiciel (le code source de Sage fait plusieurs centaines de milliers de lignes).

Bibliographie

[1] Alexandre Casamayou, Nathann Cohen, Guillaume Connan, Thierry Dumont, Laurent Fousse, François Maltey, Matthias Meulien, Marc Mezzarobba, Clément Pernet, Nicolas M. Thiéry, and Paul Zimmermann. *Calcul mathématique avec Sage*. 2013. URL: http://sagebook.gforge.inria.fr/.

[2] Manuel Kauers, Maximilian Jaroschek, and Fredrik Johansson. `ore_algebra`, 2013–. URL: http://kauers.de/software.html.

[3] Manuel Kauers, Maximilian Jaroschek, and Fredrik Johansson. Ore polynomials in Sage. In Josef Schicho Gutierrez, Jaime and Martin Weimann, editors, *Computer Algebra and Polynomials*, pages 105–125. Springer, 2015. URL: http://www.algebra.uni-linz.ac.at/people/mkauers/publications/kauers14b.pdf.

[4] Manuel Kauers and Peter Paule. *The Concrete Tetrahedron: Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Springer, 2011.

[5] Richard P. Stanley. *Enumerative combinatorics*, volume 2. Cambridge University Press, 1999.

[6] The SageMath Developers. SageMath mathematics software, 2005-. URL: http://sagemath.org/.