

# Manipulation de suites P-récurives avec SageMath

Mathis CARISNTAN & Aurélien LAMOUREUX

sous la responsabilité de Marc MEZZAROBBA

09/03/2017

## Résumé

Ce rapport présente le travail que nous avons effectué au cours de ce projet. Nous présentons dans un premier temps ce que sont les suites P-récurives, ainsi que l'outil SageMath. Puis nous expliquons les motivations de ce projet. Enfin, nous détaillons les choix et détails de l'implémentation que nous avons réalisé, avant de discuter des limites de celle-ci et des possibles améliorations.

## 1 Introduction

Nous nous intéressons ici aux suites p-récurives, et comment les manipuler dans SageMath (ou Sage).

...TODO...

### 1.1 Suites p-récurives & Algèbre d'Ore

(Motivations : Repr exactes de suites, utilisée dans différents domaines des maths/sciences)

...TODO...

### 1.2 Python & Sage

...TODO...

## 2 Méthodologie de travail, et progression

La première tâche à laquelle nous nous sommes attelés, a été de chercher à comprendre notre sujet (les suites p-récurives) et nos outils (Python et Sage). Une fois cette étape effectuée, nous avons commencé à discuter de l'implémentation. Nous nous sommes rapidement mis d'accord avec notre encadrant, qu'il était plus pertinent d'un point de vue pédagogique de d'abord créer un module python, utilisant les fonctionnalités de Sage. Puis, une fois ce module éprouvé, le réécrire en utilisant la syntaxe de Sage. Cette manière de procéder nous a permis de nous concentrer initialement sur le fond, et non la forme, puisque nous étions plus familier avec Python.

## 2.1 Module Python

La base du module a été d'écrire une classe Python (**init. n'étend aucun classes**). Cette classe devait notamment permettre d'utiliser la représentation basée sur la relation de récurrence, et des conditions initiales. Immédiatement après, nous avons surchargé l'opérateur `__getitem__` pour accéder au n-ième terme de la suite. Initialement, nous calculions tous les termes de la suite, jusqu'à celui voulu, que nous renvoyions, mais cette méthode est très inefficace. Nous avons donc résolu d'utiliser la fonction `forward_matrix` du module `ORE_ALGEBRA` à la place (**exemple et comparaison complex avec Fibo ?**)  
**calculer ts les elts vs calculer que le bon élément**)

Par la suite, nous avons également surcharger les opérateurs d'addition, soustraction et multiplication, en accord avec les lois de l'algèbre d'Ore.

## Références

- [1] A. Bohr and B.R. Mottelson, Nuclear Structure, vol. 2, Benjamin, New York, 1975.
- [2] <http://ipnweb.in2p3.fr>
- [3] Nick Park, *A Grand Day Out*, 1989, [http://en.wikipedia.org/wiki/A\\_Grand\\_Day\\_Out](http://en.wikipedia.org/wiki/A_Grand_Day_Out)