



RAPPORT DE STAGE

**Aspects effectifs des équations de
Mahler : recherche de solutions,
relations linéaires et transcendance**

Réalisé par Nicholas Rumiz

Supervisé par Frédéric Chyzak

Stage réalisé à l'INRIA Saclay, de mars à juillet 2020

Abstract

The core of my internship consists in the study of two articles written by Boris Adamczewski (boris.adamczewski@math.cnrs.fr) and Colin Faverjon (colin.faverjon@ac-creteil.fr) that discuss the Mahler method for deciding the transcendence of the evaluation of a Mahlerian function at an algebraic number. A Mahlerian function f is a function which, given an integer b , satisfies a polynomial equation between $f(x), f(x^b), f(x^{b^2}), \dots$. The purpose of the article ([Adamczewski et Faverjon, 2018](#)) is to show that this problem is decidable. Indeed the authors give an explicit algorithmic procedure to solve it. The article ([Adamczewski et Faverjon, 2017](#)) provides and demonstrates the needed theoretical results to validate the procedure. The developed method is effective but needs too many calculations to be used in practice. Typically, it needs to do linear algebra on matrices with more than 10^{100} columns. So the aim of my internship is to find some algorithmic improvements in order to make the method feasible. The analysis of the method has led me to study new subjects. Thus, in this report, I will also present effective algorithms to calculate the coefficients of the expansion in entire series of Mahlerian functions. This section is based on the results of the article ([Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016](#)) which deals with the solutions of linear Mahler equations. An algorithm to find a basis of rational solutions from a Mahlerian equation will also be studied. Finally, I will focus on the linear dependence relations between Mahlerian functions solving the same mahlerian system. The report will end with a presentation, thanks to the new tools introduced, of a randomized version of the Mahler method initially formulated by Boris Adamczewski and Colin Faverjon.

Résumé

Le cœur de mon stage consiste à étudier deux articles publiés par Boris Adamczewski (boris.adamczewski@math.cnrs.fr) et Colin Faverjon (colin.faverjon@ac-creteil.fr) et traitant de la méthode de Mahler pour décider la transcendance de l'évaluation en un nombre algébrique d'une fonction mahlérienne. Une fonction mahlérienne f est une fonction qui, pour un entier b donné, vérifie une équation reliant $f(x), f(x^b), f(x^{b^2}), \dots$. L'article ([Adamczewski et Faverjon, 2018](#)) a pour objectif de montrer que ce problème est décidable, les auteurs fournissant pour cela une procédure algorithmique explicite. L'article ([Adamczewski et Faverjon, 2017](#)) fournit et démontre les résultats théoriques nécessaires pour y parvenir. La méthode présentée est une méthode effective mais requiert des calculs beaucoup trop importants pour pouvoir l'utiliser concrètement. Typiquement, on est amené à faire de l'algèbre linéaire sur des matrices à plus de 10^{100} colonnes. L'objectif de ce stage est donc de trouver des améliorations algorithmiques pour rendre la méthode praticable. La recherche de telles améliorations m'a conduit à de nouveaux sujets d'étude. Ainsi, dans ce rapport, je présenterai aussi des algorithmes efficaces de recherche de coefficients dans les développements en série de nos fonctions mahlériennes. Cette partie reposera notamment sur les résultats de l'article ([Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016](#)) qui traite de la recherche de solutions d'une équation mahlérienne linéaire. Un algorithme de recherche d'une base de solutions rationnelles d'une équation mahlérienne sera également étudié. Enfin, on s'intéressera aux relations de dépendance linéaire entre fonctions issues d'un même système mahlérien. Le rapport proposera alors, à l'aide des outils introduits, une version randomisée de la méthode de Mahler proposée initialement par Boris Adamczewski et Colin Faverjon.

Avant propos

J'ai débuté mon stage le lundi 2 mars 2020 dans l'équipe SPECFUN de l'INRIA. J'ai été placé en télétravail dès le premier jour car j'habite dans un des premiers clusters du virus *Covid-19*. La suite de ce stage s'est également déroulée en télétravail.

Sujet du stage : Étude des équations mahlériennes, des relations de dépendance linéaire entre fonctions mahlériennes, application à un test de transcendance et optimisations.

Établissement : INRIA Saclay (équipe SPECFUN), bâtiment Alan Turing, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau. Page web : <https://specfun.inria.fr/>.

Superviseur : Frédéric CHYZAK, contact : frederic.chyzak@inria.fr.

Le code source et les rapports écrits en lien avec le sujet présent sont disponibles à l'adresse suivante : <https://rznicholas.wixsite.com/website> voir la section Publications, au paragraphe Méthode de Mahler.

Sommaire

| | |
|--|-----------|
| Introduction | 1 |
| 1 Algèbre des équations mahlériennes | 2 |
| 1.1 Définitions et premières propriétés | 2 |
| 1.2 Arithmétique euclidienne non commutative | 4 |
| 1.2.1 L'algèbre $\mathbb{K}(x)[M]$ | 4 |
| 1.2.2 Division euclidienne et pgcd | 5 |
| 2 Résolution d'équations mahlériennes : recherche effective de solutions développables en série et rationnelles | 6 |
| 2.1 Détermination des solutions développables en série d'une équation mahlérienne, aspects théoriques | 6 |
| 2.2 Détermination des solutions développables en série d'une équation mahlérienne, aspects effectifs | 8 |
| 2.2.1 Un premier exemple | 8 |
| 2.2.2 Dualité système linéaire-diagramme de Newton | 9 |
| 2.2.3 Algorithme de recherche de solutions mahlériennes dans $\mathbb{K}[[x]]$ | 12 |
| 2.3 Recherche de solutions rationnelles | 19 |
| 3 Étude des relations de dépendance linéaire de fonctions mahlériennes | 24 |
| 3.1 Transcendance de valeurs mahlériennes et transcendance de fonctions mahlériennes | 24 |
| 3.2 Test de dépendance linéaire et extraction d'un sous-système vérifié par des fonctions linéairement indépendantes | 25 |
| 3.3 Application : obtention d'une équation minimale | 29 |
| 3.4 Limites et améliorations du modèle | 32 |
| 3.4.1 L'algorithme <code>one_term</code> | 32 |
| 4 Test de transcendance sur les fonctions mahlériennes | 36 |

| | | |
|---|---|----|
| A | Démonstrations de la sous-section 2.3 (Recherche de solutions rationnelles) | I |
| B | L'algorithme <code>one_term</code> | IV |

Liste des algorithmes

| | | |
|----|--|----|
| 1 | REEMPLIR LA MATRICE $(R_{m,n})_{m,n}$ | 14 |
| 2 | CALCUL DE TOUTES LES SOLUTIONS POLYNOMIALES DE DEGRÉ MAJORÉ D'UN SOUS-SYSTÈME DU SYSTÈME INITIAL | 15 |
| 3 | CALCUL DES PREMIERS TERMES DES SOLUTIONS SÉRIES | 16 |
| 4 | PROLONGEMENT D'UNE SOLUTION DÉVELOPPABLE EN SÉ À TOUT ORDRE | 17 |
| 5 | CALCUL DE SOLUTIONS POLYNOMIALES DE DEGRÉ MAJORÉ | 17 |
| 6 | CALCUL D'UNE BASE DE SOLUTIONS POLYNOMIALES | 18 |
| 7 | RECHERCHE D'UN MULTIPLE COMMUN À TOUS LES DÉNOMINATEURS POS- SIBLES PARMIS LES SOLUTIONS RATIONNELLES | 22 |
| 8 | CALCUL D'UNE BASE DE SOLUTIONS RATIONNELLES | 23 |
| 9 | CALCUL DE LA DIMENSION DE L'ESPACE VECTORIEL ENGENDRÉ PAR UNE FAMILLE DE FONCTIONS MAHLÉRIENNES | 26 |
| 10 | DÉTERMINATION DE L'ÉQUATION INHOMOGÈNE VÉRIFIÉE PAR UNE FONC- TION MAHLÉRIENNE DONNÉE | 30 |
| 11 | TEST DE TRANSCENDANCE DE $f(\alpha)$ | 36 |

Introduction

Dans ce rapport, je m'intéresse à des équations mahlériennes et à leurs solutions. Les *fonctions mahlériennes*, introduites en 1929 par Mahler, sont les fonctions développables en série entière autour de l'origine qui vérifient une équation du type

$$p_{-1}(x) + p_0 f(x) + p_1(x) f(x^b) + p_2(x) f(x^{b^2}) + \dots = 0.$$

Ici, b est un entier supérieur ou égal à 2 qui s'appelle la *base* et les p_i sont des polynômes. Je fais également l'hypothèse que les fonctions étudiées sont à coefficients dans un corps de nombre.

Dans ce cadre d'étude, il y a un résultat important, dû à Bézivin en 1994, qui établit une dichotomie sur la nature de l'évaluation, en un nombre algébrique, d'une fonction mahlérienne. C'est le résultat suivant.

Théorème (Bézivin, 1994). *Soit $f(x)$ une fonction b -mahlérienne et $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$, qui n'est pas un pôle de f . Soit \mathbf{k} un corps de nombres contenant α et les coefficients de f . On a l'alternative suivante : soit $f(\alpha)$ est transcendant, soit $f(\alpha) \in \mathbf{k}$.*

Un fait remarquable avec les fonctions mahlériennes, est que ces dernières ne prennent pas de valeurs algébriques sur \mathbf{k} . À partir de ce résultat, [Adamczewski et Faverjon](#) ont développé une procédure algorithmique permettant de trancher cette dichotomie. D'une part, ils ont montré que c'était donc un problème décidable, et d'autre part, l'accès à ce type d'information est intéressant en théorie des automates et plus généralement en théorie des nombres.

La solution apportée par [Adamczewski et Faverjon \(2018\)](#) possède néanmoins une limite : elle n'est pas exécutable en pratique, parce qu'elle nécessite des calculs d'algèbre linéaire sur des matrices, ou des polynômes, de tailles trop grands. Le travail qui suit consiste donc à proposer des améliorations algorithmiques, pour rendre le test plus praticable.

Dans la section [1](#), je m'intéresse aux différentes représentations possibles d'une équation mahlérienne, je déduis de premiers résultats qui me permettent de travailler par la suite uniquement avec des équations homogènes et de terme p_0 non nul. Je définis ensuite une algèbre non commutative dans laquelle je peux réinterpréter mes équations mahlériennes. Ce formalisme est utilisé par la suite dans l'implémentation de certaines procédures.

La variante de la méthode de Mahler présentée en [2018](#) nécessite d'accéder à des coefficients de grand ordre du développement en série de nos fonctions mahlériennes. Ainsi, dans la section [2](#) j'étudie différents procédés algorithmiques me permettant de trouver le développement en série d'une solution d'une équation mahlérienne. Cette recherche se fait essentiellement récursivement, à partir de la résolution d'un « petit » système linéaire correspondant aux conditions initiales possibles. Enfin, je reprends de ([Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016](#)) un algorithme de recherche des solutions rationnelles. Cette classe de solution est en effet intéressante pour notre objectif parce que pour une fonction rationnelle, et avec les notations du théorème de Bézivin, on sait que $f(\alpha) \in \mathbf{k}$.

Un autre ingrédient de la méthode Mahler [2018](#), est l'obtention d'une équation mahlérienne minimale. La section [3](#), en étudiant les relations de dépendances linéaires entre fonctions mahlériennes issues d'un même système, aboutit sur un algorithme permettant d'obtenir une équation minimale. Ici on évoquera un résultat majeur de ([Adamczewski et Faverjon, 2017](#)) qui fait le lien entre l'espace vectoriel des relations linéaires de fonctions mahlériennes et le noyau d'une

certaine matrice qu'on précisera.

Enfin, la section 4, est consacrée à l'algorithme qui permet de tester la transcendance des évaluations d'une fonction mahlérienne en un point algébrique. Je présente alors une version randomisée (sur l'obtention de l'équation minimale) qui permet de décider la dichotomie établie par Bézivin et je la mettrai à l'œuvre sur divers exemples issus de la théorie des nombres et des suites automatiques.

1 Algèbre des équations mahlériennes

L'objectif de cette partie est d'introduire la notion de fonction mahlérienne et de passer en revue les différentes représentations possibles d'une équation ou d'un système mahlérien ainsi que leurs liens.

1.1 Définitions et premières propriétés

Commençons par donner la définition d'une fonction mahlérienne.

Définition 1.1. On note $\overline{\mathbb{Q}}\{x\}$ l'ensemble des fonctions admettant un développement de Taylor dans un disque ouvert centré en l'origine et à coefficients dans $\overline{\mathbb{Q}}$.

Définition 1.2. Soit $b \geq 2$ un entier. Une fonction $f \in \overline{\mathbb{Q}}\{x\}$ est dite *b-mahlérienne* s'il existe des polynômes $p_{-1}, p_0, \dots, p_n \in \overline{\mathbb{Q}}[x]$ non tous nuls avec $p_n \neq 0$ tels que :

$$p_{-1}(x) + p_0(x)f(x) + p_1(x)f(x^b) + \dots + p_n(x)f(x^{b^n}) = 0. \quad (1)$$

Remarque 1.3. L'équation (1) est une équation *inhomogène* vérifiée par une fonction mahlérienne f . Si $p_{-1} = 0$, on parle alors d'équation *homogène*. Pour chacun des cas, homogène ou inhomogène, on parlera d'équation minimale si de plus l'ordre n qui intervient est minimal.

Propriété 1.4. Dans l'équation (1), on peut sans perte de généralité supposer que p_0 est non nul.

Démonstration. En effet, supposons qu'on ait une équation du type (1). Supposons également $p_0 = 0$. Par hypothèse, soit $j > 0$ minimal tel que $p_j \neq 0$. On décompose alors les polynômes p_i en sections selon les puissances de x modulo b^j . On obtient alors :

$$\forall i \geq -1, p_i(x) = \sum_{k=0}^{b^j-1} p_{i,k}(x^{b^j})x^k.$$

L'équation (1) implique alors que :

$$\forall 0 \leq k < b^j : p_{-1,k}(x^{b^j}) + \sum_{l=j}^n p_{l,k}(x^{b^l})f(x^{b^l}) = 0.$$

On applique le changement de variable $y = x^{b^j}$ dans cette équation et on obtient :

$$\forall 0 \leq k < b^j : p_{-1,k}(y) + p_{j,k}(y)f(y) + \dots + p_{n,k}(y)f(y^{b^{n-j}}) = 0.$$

Comme $p_j \neq 0$, il existe au moins un k_0 tel que $p_{j,k_0} \neq 0$. On pose alors $\tilde{p}_{-1} := p_{-1,k_0}$ et pour $0 \leq i \leq n - j$, $\tilde{p}_i := p_{i+j,k_0}$. D'où, après avoir remplacé y par x :

$$\tilde{p}_{-1}(x) + \tilde{p}_0(x)f(x) + \cdots + \tilde{p}_{n-j}(x)f(x^{b^{n-j}}) = 0.$$

□

Dans la suite on supposera donc que p_0 est non nul.

Propriété 1.5. • Une série formelle $f \in \overline{\mathbb{Q}}[[x]]$ solution d'une équation du type (1) est convergente au voisinage de l'origine. De plus, elle admet un prolongement méromorphe sur le disque unité ouvert et le cercle unité forme une frontière naturelle dès que f n'est pas une fraction rationnelle.

- Les coefficients de Taylor d'une fonction mahlérienne engendrent une extension finie de \mathbb{Q} .

Démonstration. On renvoie à la preuve d'un résultat, énoncé au chapitre 3, théorème 6, dans (Dumas, 1993). Ce dernier est plus général car il précise le rayon de convergence de la série formelle f en fonction des racines du polynôme p_0 qui intervient dans l'équation (1) qu'elle vérifie. □

Remarque 1.6. On peut donc supposer sans perte de généralité que f appartient à $\mathbf{k}\{x\}$ où \mathbf{k} est un corps de nombre (i.e. une extension finie de \mathbb{Q}).

Pour étudier une fonction b -mahlérienne f il peut être commode de considérer un système d'équations fonctionnelles de la forme :

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = A(x) \begin{pmatrix} f_1(x^b) \\ \vdots \\ f_n(x^b) \end{pmatrix}, \quad (2)$$

où $A(x)$ est une matrice de $Gl_n(\overline{\mathbb{Q}}(x))$ et conçu de sorte que la fonction f apparaissent comme coordonnée d'une solution.

Proposition 1.7. Une fonction est solution d'un système du type (1) si, et seulement si, elle est la coordonnée d'un vecteur solution d'un système de type (2).

Remarque 1.8. Désormais une fonction mahlérienne sera donnée par une expression de type (1) ou de type (2).

Démonstration. Pour passer de l'expression (1) à une expression de type (2) on se ramène au système compagnon suivant :

$$\begin{pmatrix} 1 \\ f(x) \\ \vdots \\ \vdots \\ f(x^{b^{n-1}}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\frac{p_{-1}(x)}{p_0(x)} & \frac{p_1(x)}{p_0(x)} & \cdots & \cdots & \frac{p_n(x)}{p_0(x)} \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ \vdots \\ f(x^{b^n}) \end{pmatrix} =: A(x) \begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ f(x^{b^n}) \end{pmatrix}.$$

En développant le déterminant de $A(x)$ par rapport à la dernière colonne, comme $p_n \neq 0$, on a $A(x) \in Gl_n(\overline{\mathbb{Q}}(x))$. Concernant la réciproque, nous présenterons dans la section 3 un algorithme permettant, à partir d'une équation de type (2), de déterminer une équation homogène et une inhomogène minimales pour chaque fonction mahlérienne coordonnée. □

Donnons un premier exemple pour fixer les idées.

Exemple 1.9. Considérons l'équation fonctionnelle suivante :

$$xf(x) - (1 - 2x)f(x^2) + (1 + x)f(x^4) = 0. \quad (3)$$

C'est bien une équation du type (1) qui peut se réécrire comme suit :

$$\begin{pmatrix} f(x) \\ f(x^2) \end{pmatrix} = \begin{pmatrix} \frac{1+2x}{x} & -\frac{1+x}{x} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f(x^2) \\ f(x^4) \end{pmatrix}. \quad (4)$$

On vérifie que la fonction constante égale à 1 est solution. Par ailleurs, si on s'intéresse aux solutions développables en série entière, une étude sur les coefficients nous permet de trouver une solution série dont les premiers termes sont :

$$f(x) = x + 2x^2 + x^3 + 3x^4 + 2x^5 + 2x^6 + x^7 + 4x^8 + \dots$$

Donc on n'a pas unicité de la fonction mahlérienne en se donnant uniquement une équation (1) ou un système (2).

Ce premier exemple est là pour motiver la recherche d'une équation minimale pour l'équation (1), homogène ou inhomogène. Plus tard, nous aurons besoin de pouvoir les retrouver, à partir d'une équation ou d'un système mahlérien. Mais avant ça, décrivons un autre formalisme utile pour écrire et étudier de telles équations mahlériennes.

1.2 Arithmétique euclidienne non commutative

1.2.1 L'algèbre $\mathbb{K}(x)[M]$

La construction qui est faite ici est générale en algèbre différentielle et en algèbre à différence.

On note $M : f(x) \mapsto f(x^b)$ l'opérateur de Mahler. Les opérateurs linéaires sur les séries formelles,

$$f(x) \mapsto p_0(x)f(x) + p_1(x)f(x^b) + \dots + p_n(x)f(x^{b^n}),$$

forment une algèbre non commutative. Ces derniers sont alors combinaisons linéaires de l'opérateur M et de la multiplication par x lesquels ne commutent pas. Néanmoins, on a la relation suivante $Mx = x^qM$ et on peut considérer une forme dite normale : $p_0(x) + p_1(x)M + \dots + p_n(x)M^n$. Prenons toujours \mathbf{k} un corps de nombres, on voit qu'on peut traduire toute équation mahlérienne homogène en un polynôme dans l'anneau $\mathbf{k}(x)[M]$.

Remarque 1.10. On appellera opérateurs les éléments de $\mathbf{k}(x)[M]$.

La relation de pseudo-commutativité $Mx = x^qM$ nous fournit le résultat suivant.

Proposition 1.11. *Tout élément de $\mathbf{k}(x)[M]$ s'écrit de manière unique comme $\sum_{k=0}^N c_k(x)M^k$ où les c_k sont des fractions rationnelles.*

Démonstration. Supposons qu'on ait deux telles écritures, en multipliant l'égalité par le ppcm de tous les dénominateurs qui interviennent et en faisant la différence, on se ramène à une égalité du type $\sum_{k=0}^N p_k(x)M^k = 0$. Appelons m un majorant du degré des polynômes p_k qui interviennent. On applique alors cette dernière égalité à la fonction $f(x) = x^{m+1}$ et on obtient que les p_k sont tous nuls. C'est donc qu'on avait égalité entre les deux écritures. \square

Définition 1.12. Le *degré* par rapport à M d'un opérateur $P = \sum_{k=0}^N c_k(x)M^k$ est le plus grand indice k tel que $c_k \neq 0$. On prend la convention $\deg_M(0) = -\infty$.

On peut vérifier que ce degré vérifie bien les propriétés d'un degré. Munis de ce degré, nous donnons la proposition suivante.

Proposition 1.13. *L'anneau $\mathbf{k}(x)[M]$ est intègre et ses éléments inversibles sont les fractions rationnelles non nulles de $\mathbf{k}(x)$.*

Proposition 1.14. *Toute équation mahlérienne inhomogène peut se réécrire comme une équation homogène.*

Démonstration. En effet soit f une fonction b -mahlérienne qui vérifie une égalité de type (1). Soit p_{-1} le terme inhomogène. L'opérateur $P_1 = p_{-1}(x)M - p_{-1}(x^b)$ s'annule en $p_{-1}(x)$. Soit $P_2 = p_0(x) + p_1(x)M + \dots + p_n(x)M^n$. On a $P_2(f) = -p_{-1}(x)$. Donc $(P_1P_2)(f) = 0$. Donc on a ainsi obtenu une équation homogène vérifiée par f dont la traduction dans l'algèbre $\mathbf{k}(x)[M]$ s'écrit P_1P_2 . \square

1.2.2 Division euclidienne et pgcd

Nous allons voir que l'anneau $\mathbf{k}(x)[M]$ possède une division euclidienne.

Proposition 1.15. *Soient F et G deux opérateurs non nuls. Alors il existe un unique couple (Q, R) d'opérateurs tels que $F = QG + R$, avec $\deg_M(R) < \deg_M(G)$.*

Démonstration. Prenons $F = f_0 + f_1M + \dots + f_nM^n$ et $G = g_0 + g_1M + \dots + G^mM^m$ avec f_n, g_m non nuls et $n \geq m$. On pose alors $Q = \frac{f_n}{M^{n-m}g_m}M^{n-m}$ et $R = F - QG$. Tant que le R construit ne vérifie pas la propriété voulue, on recommence avec R en place de F . Comme à chaque étape le degré en M du reste décroît strictement et est entier ou $-\infty$, la procédure se termine. Pour l'unicité, si $GQ_1 + R_1 = GQ_2 + R_2$, alors $R_1 - R_2 = G(Q_2 - Q_1)$. Or $\deg_M(R_1 - R_2) < \deg_M(G)$ donc nécessairement $Q_2 = Q_1$ puis $R_1 = R_2$. \square

Les propositions 1.13 et 1.15 montrent que l'anneau $\mathbf{k}(x)[M]$ est euclidien à gauche. On a alors les résultats suivants :

Corollaire 1.16. *Les idéaux (à gauche) de $\mathbf{k}(x)[M]$ sont principaux.*

Corollaire 1.17. *Le pgcd (à droite) entre deux opérateurs se calcule de manière classique par l'algorithme d'Euclide, il est unique à multiplication par une fraction rationnelle non nulle près.*

Remarque 1.18. Le fait que l'anneau $\mathbf{k}(x)[M]$ soit euclidien à gauche et en particulier noethérien fournit une preuve pour le sens réciproque de la proposition (1.7).

Nous avons vu qu'on pouvait toujours se ramener à travailler avec une équation homogène, et que cette dernière pouvait s'écrire en un polynôme quand on travaillait dans l'algèbre $\mathbf{k}(x)[M]$. Pour certains algorithmes présentés dans cet article, nous avons fait ce choix pratique de représentation. En particulier, ce sera le cas de l'algorithme `one_term` que nous verrons à la section 3.

2 Résolution d'équations mahlériennes : recherche effective de solutions développables en série et rationnelles

Notre point de départ est la donnée d'une équation du type (1) ou du type (2). Dans ce chapitre et les suivants, nous considérerons principalement des solutions développables en série entière. L'objectif qu'on suit ici est de pouvoir accéder aux coefficients présents dans le développement, de la manière la plus efficace possible. En parallèle de cet objectif, nous présenterons également un algorithme de recherche d'une base de solutions rationnelles.

2.1 Détermination des solutions développables en série d'une équation mahlérienne, aspects théoriques

Pour commencer, on se place dans le cas d'une fonction mahlérienne donnée par une équation de type (1). On définit les quantités suivantes :

- v_i la valuation en 0 du polynôme p_i .
- $d := \max \left\{ \left\lfloor \frac{v_0 - v_i}{b^i - 1} \right\rfloor, 1 \leq i \leq n \right\}$.

Lemme 2.1. Soit $f(x) = \sum_{k=0}^{\infty} f_k x^k$ une solution de l'équation (1).

- Si $d < 0$, alors la solution f est unique et l'équation permet de déterminer tous les coefficients f_k . De plus, f est nulle si, et seulement si, $p_{-1} = 0$.
- Si $d \geq 0$, alors les coefficients f_k pour $k > d$ sont déterminés de manière unique par les coefficients f_k pour $k \leq d$.

Démonstration. On pose $f_k = 0$ quand $k \in \mathbb{Q} \setminus \mathbb{N}$. Comme f vérifie l'équation (1), on obtient (puisque p_0 est non nul, ne l'oublions pas) :

$$f(x) = -\frac{p_{-1}(x)}{p_0(x)} - \frac{p_1(x)}{p_0(x)} f(x^b) - \dots - \frac{p_n(x)}{p_0(x)} f(x^{b^n}). \quad (5)$$

Pour chaque fraction rationnelle $-\frac{p_k(x)}{p_0(x)}$ on considère son développement en série de Laurent, ce qui nous donne :

$$-\frac{p_k(x)}{p_0(x)} = \sum_{k \geq v_i - v_0} p_{i,k} x^k.$$

Soit $k \geq 0$. Pour $i \geq 1$, on a l'égalité $\frac{p_i(x)}{p_0(x)} f(x^{b^i}) = \sum_{j \geq v_i - v_0} p_{i,j} x^j \sum_{l \geq 0} f_l x^{lb^i}$ et l'équivalence $j + lb^i = k \Leftrightarrow l = \frac{k-j}{b^i}$. Donc on obtient que le terme de valuation k dans l'équation (5) vérifie :

$$f_k = p_{-1,k} + \sum_{i=1}^n \sum_{j=v_i-v_0}^k p_{i,j} f_{\frac{k-j}{b^i}}. \quad (6)$$

Maintenant, si $k > d$, alors pour chaque i allant de 1 à n :

$$k > \frac{v_0 - v_i}{b^i - 1} \implies b^i k - k > v_0 - v_i \implies \forall j \geq v_i - v_0, k > \frac{k-j}{b^i}.$$

On en déduit que, pour $k > d$, f_k est déterminé de manière unique par les coefficients f_l pour $l < k$.

En particulier, si $d < 0$, les f_k sont tous uniquement déterminés par l'équation (6). On voit également à partir de (6) que les f_k sont tous nuls si, et seulement si, pour tout k , $p_{-1,k} = 0$. Ce qui revient à dire que f est nulle $\iff p_{-1} = 0$. \square

Remarque 2.2. Si $d < 0$, l'espace affine des solutions solutions développables en série entière ne contient qu'une seule fonction uniquement déterminée par (6). Si $d \geq 0$, on résout $d + 1$ équations affines de type (6) pour trouver toutes les solutions séries.

Ceci se traduit par le corollaire suivant :

Corollaire 2.3. *Étant donnée une équation (1), la dimension de l'espace affine des solutions développables en série entière de cette équation est d'au plus $1 + \lfloor \frac{v_0}{b-1} \rfloor$.*

On peut obtenir un résultat similaire si on s'intéresse à des fonctions mahlériennes données à partir d'un système de type (2). C'est l'objet du lemme suivant issu de (Adamczewski et Favrejon, 2018).

Notation 2.4. Soit f_1, \dots, f_n les coordonnées d'un vecteur solution pour le système (2). On note $\mathbf{f}(x)$ le vecteur colonne formé des fonctions f_i . Chaque fonction f_i étant développable en série entière, on écrira alors

$$\mathbf{f}(x) := \sum_{i=0}^{\infty} \mathbf{f}_i x^i,$$

où les \mathbf{f}_i sont les vecteurs colonnes comportant les coefficients d'ordre i des développements des fonctions f_1, \dots, f_n .

Lemme 2.5. *Soit $\mathbf{f}(x) = \sum_{k=0}^{\infty} \mathbf{f}_k x^k$ un vecteur non nul de fonctions développables en série entière solution d'un système de type (2). Soit ν le minimum des évaluations en $x = 0$ des coefficients de la matrice $A(x)$. Alors on a $\nu \leq 0$ et les \mathbf{f}_k pour $k > \lfloor \frac{-\nu}{b-1} \rfloor$ sont déterminés de manière unique par les \mathbf{f}_k pour $k \leq \lfloor \frac{-\nu}{b-1} \rfloor$.*

Démonstration. Les coefficients de $A(x)$ sont des fractions rationnelles, on peut comme précédemment considérer leur développement en série de Laurent :

$$A(x) = \sum_{k \geq \nu} A_k x^k$$

avec $A_k \in \mathcal{M}_n(\mathbf{k})$ et A_ν une matrice non nulle par définition. Le système (2) implique alors :

$$\sum_{k=0}^n \mathbf{f}_k x^k = \sum_{r \geq \nu} \sum_{m=0}^n A_r \mathbf{f}_m x^{qm+r}.$$

On identifie alors

$$\mathbf{f}_k = \sum_{r=\nu}^k A_r \mathbf{f}_{\frac{k-r}{b}}. \quad (7)$$

avec $\mathbf{f}_{\frac{k-r}{b}} = 0$ si b ne divise pas $k - r$.

Si $\nu > 0$, alors $\mathbf{f} = 0$. En effet, supposons le contraire : soit k_0 le plus entier vérifiant $\mathbf{f}_{k_0} \neq 0$. Dans ce cas, pour $k = k_0$ l'équation (7) serait contradictoire.

Supposons maintenant que $\nu \leq 0$. On a alors :

$$k > \frac{-\nu}{b-1} \implies kq > k - \nu \implies \forall r \geq \nu, k > \frac{k-r}{b}.$$

Ainsi l'équation (7) permet de retrouver de manière unique le vecteur \mathbf{f}_k connaissant les vecteurs \mathbf{f}_l pour $0 \leq l < k$. \square

Exemple 2.6. Reprenons notre exemple (1.9)

$$xf(x) - (1 - 2x)f(x^2) + (1 + x)f(x^4) = 0.$$

Avec les notations du lemme 2.5, on trouve $n = 2$, $b = 2$, $v_0 = 1$, $v_1 = v_2 = 0$ et $d = 1$. Donc la donnée des deux premiers coefficients du développement en série entière d'une fonction solution permet de la retrouver complètement. Ici on rappelle qu'on a deux fonctions développables en série entière solutions qui sont la fonction constante égale à 1 et la fonction dont les premiers termes sont les suivants :

$$f(x) = x + 2x^2 + x^3 + 3x^4 + 2x^5 + 2x^6 + x^7 + 4x^8 + \dots$$

2.2 Détermination des solutions développables en série d'une équation mahlérienne, aspects effectifs

2.2.1 Un premier exemple

Dans cette sous-section, nous allons considérer une classe plus grande de solutions que les solutions développables en série entière, celle des solutions développables en série de Puiseux dont nous rappelons la définition.

Définition 2.7. Une *série de Puiseux* est une série de la forme $\sum_{i \geq k} a_i x^{i/n}$ où $k \in \mathbb{Z}$, $a_i \in \mathbb{K}$ et $n \in \mathbb{N}^*$.

On note $\mathbb{K}((x^{1/n}))$ le corps des séries de Puiseux à coefficients dans \mathbb{K} .

Pour illustrer ce qui va suivre, appuyons-nous sur l'exemple suivant tiré de (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016).

Exemple 2.8. On choisit ici de travailler avec $b = 3$ et $n = 2$ et on cherche à résoudre $L(y) = 0$ avec

$$L = \underbrace{x^3(1 - x^3 + x^6)(1 - x^7 - x^{10})}_{:=p_2(x)} M^2 - \underbrace{(1 - x^{28} - x^{31} - x^{37} - x^{40})}_{:=p_1(x)} M + \underbrace{x^6(1 + x)(1 - x^{21} - x^{30})}_{:=p_0(x)}. \quad (8)$$

où M est l'opérateur de Mahler associé à $b = 3$: $M(x) = x^3$.

Supposons avoir une solution y série de Puiseux qui vérifie $p_2(x)y(x^9) + p_1(x)y(x^3) + p_0(x)y(x) = 0$. Notons v sa valuation à l'origine. Les valuations de $p_k M^k y$ pour k allant de 0 à 2 sont alors respectivement $6 + v$, $3v$ et $9v + 3$. Si une de ces valuations était strictement inférieure aux autres, on n'aurait pas de compensation possible et donc $L(y) = 0$ ne serait pas vérifiée. On a donc nécessairement une égalité entre deux de ces valuations. On résout ce système et on trouve alors que les seules valuations possibles pour y sont $v \in \{-1/2, 3\}$.

Par exemple, prenons le cas de $v = 3$. Pour $m \geq 0$, on peut regarder le coefficient devant x^m dans l'écriture de $y = \sum_{i \geq 3} y_i x^i$ sachant que y est solution de l'équation considérée. On obtient alors l'égalité suivante :

$$\begin{aligned} & (y_{m-6} + y_{m-7} - y_{m-27} - y_{m-28} - y_{m-36} - y_{m-37}) \\ & - \left(\frac{y_m}{3} - \frac{y_{m-28}}{3} - \frac{y_{m-31}}{3} - \frac{y_{m-37}}{3} - \frac{y_{m-40}}{3} \right) \\ & + \left(\frac{y_{m-3}}{9} - \frac{y_{m-6}}{9} + \frac{y_{m-9}}{9} - \frac{y_{m-10}}{9} - \frac{y_{m-19}}{9} \right) = 0 \end{aligned} \quad (9)$$

où on suppose que $y_i = 0$ quand $i < 0$.

Si on s'intéresse à une solution $y \in \mathbb{K}[[x]]$ on voit alors que $\forall n \geq 4$, y_n est uniquement déterminé par les y_j pour lesquels $j < n$. En d'autres termes, on a un seul degré de liberté qui est le choix de y_3 . Dit autrement, l'espace des solutions dans $\mathbb{K}[[x]]$ de l'équation (8) est de dimension 1 et une base peut être donnée par la série :

$$x^3 - x^4 + x^5 - 2x^6 + 2x^7 - 2x^8 + 3x^9 - 3x^{10} + 3x^{11} - 5x^{12} \dots$$

Le cas $v = -1/2$ correspond à chercher une solution dans $\mathbb{K}((x^{1/2}))$. Soit $y = \sum_{i \geq -1} y_i x^{i/2}$ une telle solution.

L'idée est de se ramener au cas correspondant à une valuation entière. On effectue donc le changement de variable $x = t^2$ et on pose ensuite $\tilde{y}(t) = ty(t)$. On obtient alors une solution $\tilde{y}(t) = \sum_{i \geq 0} \tilde{y}_i t^i$ telle que $\tilde{L}(\tilde{y}) = 0$. Alors les solutions de valuation $-1/2$ en l'origine pour $L(f) = 0$ sont les solutions de valuation 0 en l'origine de $\tilde{L}(f) = 0$. Et on applique ensuite le même raisonnement que pour le cas où $v = 3$ pour conclure.

2.2.2 Dualité système linéaire-diagramme de Newton

Je m'intéresserai désormais à une équation en particulier

$$p_r(x)y(x^{b^r}) + \dots + p_1(x)y(x^b) + p_0(x)y(x) = 0. \quad (\text{EQN})$$

et je définis

$$L = p_r M^r + \dots + p_1 M + p_0. \quad (\text{OPR})$$

Cadre : Soit $L = p_r M^r + \dots + p_1 M + p_0$ où M est l'opérateur de Mahler de raison b . Soit y une série de Puiseux qui vérifie $L(y) = 0$ et notons v sa valuation à l'origine. On note également d_k le degré du polynôme p_k et $d := \max\{d_k\}$.

Remarque 2.9. Les valuations à l'origine des termes $p_k M^k y$ sont $b^k v + v_k$.

Comme discuté dans l'exemple 2.8

$$L(y) = 0 \implies \exists (k_1, k_2) \in \llbracket 0, r \rrbracket^2, k_1 \neq k_2, \text{ tels que } v_{k_1} + b^{k_1} v = v_{k_2} + b^{k_2} v = \min_{0 \leq k \leq r} v_k + b^k v. \quad (10)$$

Définition 2.10. (*Diagramme/polygone de Newton*)

Considérons y une série de Puiseux qui vérifie $L(y) = 0$, et notons v sa valuation en 0. La condition $L(y) = 0$ peut s'interpréter géométriquement à l'aide d'un *diagramme de Newton*. L'idée est d'associer à chaque monôme $x^j M^k$ apparaissant dans L un point (b^k, j) . Le *diagramme de Newton* est alors la collection de ces points représentés dans un repère. Parmi ces points, je vais m'intéresser aux points d'ordonnée extrême pour chaque abscisse b^k . Le polygone obtenu en reliant ces points particuliers s'appelle le *polygone de Newton*. Je distinguerai *polygone de Newton supérieur* et *inférieur*.

Pour faire le lien avec ce qui précède, l'équation (10) se traduit en disant que les points (b^{k_1}, v_{k_1}) et (b^{k_2}, v_{k_2}) se trouvent sur un même côté du polygone inférieur de Newton, de pente $-v$.

De plus, on parle de « côté admissible » quand la somme des coefficients devant les monômes dont la représentation en point appartient à ce bord est nulle. Dans l'exemple, on trouve deux bords admissibles, les deux côtés du polygone inférieur de Newton, de pente respectives $-1/2$ et 3 et qui correspondent aux valuations $-1/2$ et 3 déjà calculées précédemment.

Si on se donne une solution y vérifiant $L(y) = 0$ avec L un certain opérateur, on a un résultat sur la valuation à l'origine de y .

Lemme 2.11. *Soit $L = p_r M^r + \dots + p_1 M + p_0$ où M est l'opérateur de Mahler de raison b . Soit y une série de Puiseux vérifiant $L(y) = 0$ de valuation v à l'origine. Alors v vérifie :*

$$-\frac{v_r}{b^r - b^{r-1}} \leq v = -\frac{v_{k_1} - v_{k_2}}{b^{k_1} - b^{k_2}} \leq \frac{v_0}{b - 1} \quad (11)$$

où les v_i sont les valuations à l'origine des p_i et où (b^{k_1}, v_{k_1}) et (b^{k_2}, v_{k_2}) sont deux points du polygone inférieur de Newton, comme dans l'équation (10).

Démonstration. L'égalité sur v découle de l'équation (10). Il reste donc à voir les bornes.

D'une part, la convexité du polygone de Newton nous permet de dire que la pente du côté le plus à gauche du diagramme inférieur de Newton possède l'opposé le plus grand parmi les opposés des pentes possibles en reliant deux points du diagramme inférieur de Newton. Autrement dit, il fournit la valuation la plus grande. En particulier, pour un certain $k \geq 1$ on a l'inégalité suivante :

$$v = -\frac{v_{k_1} - v_{k_2}}{b^{k_1} - b^{k_2}} \leq -\frac{v_k - v_0}{b^k - 1} \leq \frac{v_0}{b - 1}.$$

D'autre part, de manière analogue, l'opposé de la pente du côté le plus à droite du diagramme inférieur de Newton fournit la valuation la plus petite, c'est-à-dire que pour un certain $k < r$, on a :

$$v = -\frac{v_{k_1} - v_{k_2}}{b^{k_1} - b^{k_2}} \geq -\frac{v_r - v_k}{b^r - b^k} \geq -\frac{v_r}{b^r - b^{r-1}}.$$

□

De la même manière un résultat sur les degrés existe, pour rappel on note d le degré maximal des polynômes qui interviennent dans l'écriture de l'opérateur L .

Lemme 2.12. *Le degré d'une solution polynomiale de (EQN) correspond à l'opposé d'une pente (admissible) du polygone de Newton supérieur et vérifie*

$$\delta = -\frac{d_{k_1} - d_{k_2}}{b^{k_1} - b^{k_2}} \leq \frac{d}{b^{r-1}(b - 1)} \quad \text{pour certains } k_1 \neq k_2. \quad (12)$$

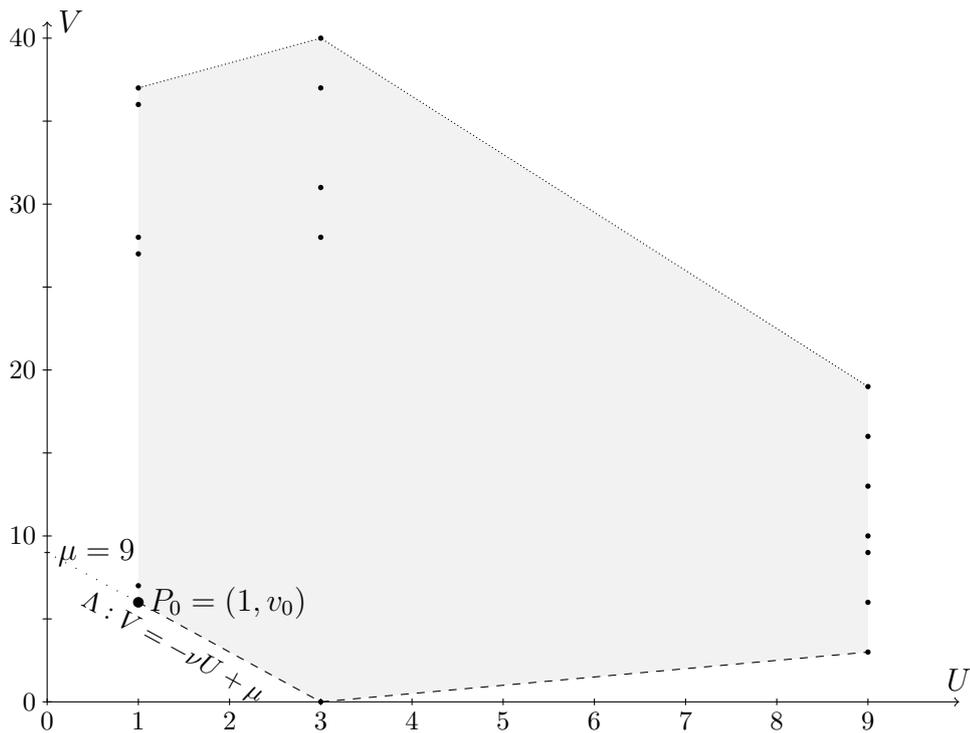


FIGURE 1 – Le diagramme de Newton de l'équation (8) traitée en exemple avec $b = 3$.

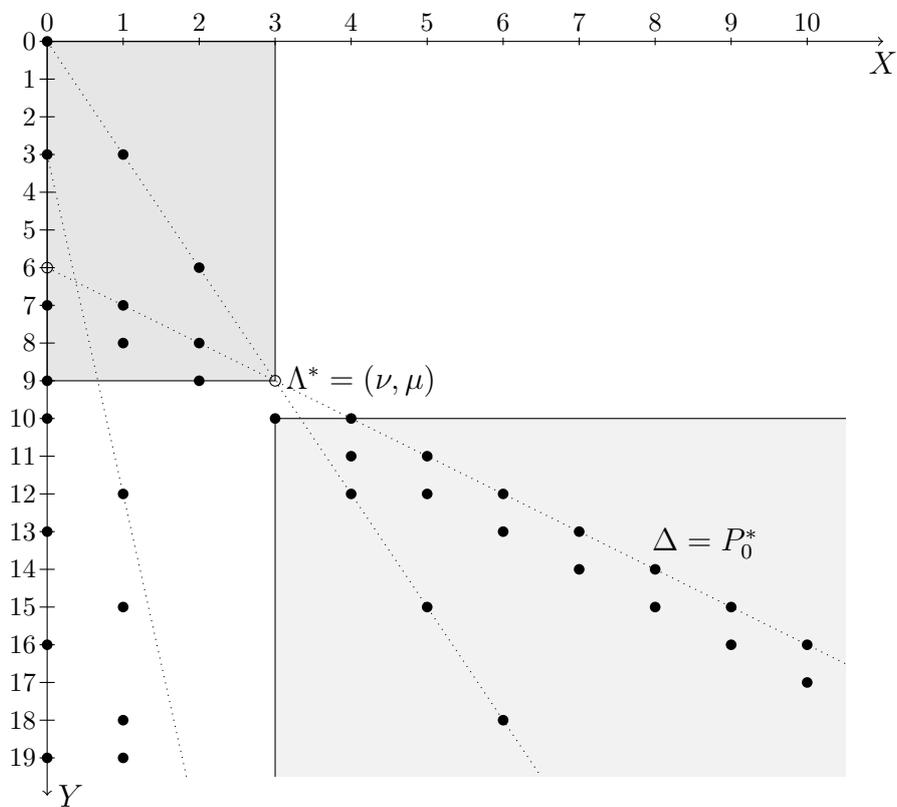


FIGURE 2 – Visualisation de la matrice infinie correspondant à notre exemple de départ. Les cercles pleins représentent des entrées non nulles et les cercles vides des recombinaisons à zéro.

Remarque 2.13. La Figure (2) représente le système linéaire associé à l'exemple (8), pour fixer les idées appelons (n, m) un point sur cette figure. La figure elle-même peut être vue comme une matrice (infinie) R où on a $R_{m,n} = [x^m]L(x^n)$.

Intéressons-nous maintenant aux liens que l'on peut faire entre les deux figures. On a déjà dit que pour une droite $Y = b^k X + j$ de la Figure (2) on associait le point (b^k, j) de la Figure (1) et chaque point (n, m) de la Figure (2) se voit attribuer une droite $V = m - nU$ de la Figure (1).

Voyons comment tirer parti de cette analogie pour élaborer un algorithme de recherche de solutions mahlériennes dans $\mathbb{K}[[x]]$.

2.2.3 Algorithme de recherche de solutions mahlériennes dans $\mathbb{K}[[x]]$

Soit une solution $y \in \mathbb{K}[[x]]$ vérifiant $L(y) = 0$, on s'intéresse au coefficient devant x^m dans l'écriture de $y = \sum_{n \geq 0} y_n x^n$. Cela nous donne une équation linéaire qu'on retrouve à la m -ième ligne de la matrice $R := R(L)$ définie plus haut.

Par ailleurs, comme par hypothèse $p_0 \neq 0$, la plus petite pente observable dans le diagramme de Newton, (*c.f.* Figure 2) vaut 1 et est celle de la droite $Y = X + v_0$. Dans notre exemple, $v_0 = 6$. Ceci nous permet de dire, que pour des Y suffisamment grands, la droite $Y = X + v_0$ est la plus « haute » quand on représente notre système linéaire comme dans la Figure 2. Autrement dit, pour $n = m - v_0$, le coefficient y_n peut être déterminé de manière unique par l'équation linéaire obtenue à partir de la ligne R_m de la matrice R . Ainsi, pour trouver toutes les solutions dans $\mathbb{K}[[x]]$, on est ramené à résoudre un sous-système linéaire, associé à une sous-matrice de R .

Guidés par ce qui se passe sur la Figure 2, intéressons-nous comme dans (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016) aux deux quantités suivantes :

$$\nu = \max_{k \geq 1} \frac{v_0 - v_k}{b^k - 1} \quad \text{et} \quad \mu = \nu + v_0. \quad (13)$$

Remarque 2.14. En faisant l'analogie avec le diagramme de Newton, ν peut être interprétée comme l'opposé de la pente du côté le plus à gauche du polygone inférieur de Newton et μ comme l'ordonnée à l'origine de cette droite. Notons également que si $\nu < 0$, alors il n'y a pas de solution dans $\mathbb{K}[[x]]$ autre que la fonction nulle. Ceci arrive si, et seulement si, v_0 est strictement à tous les autres v_k .

Remarque 2.15. Si on considère le point $(1, v_0)$ de la Figure 1, on lui associe sur la Figure 2 la droite duale $P_0^* := \Delta : Y = X + v_0$. De manière générale si on considère la droite passant par deux points dans la Figure 1, le dual de cette droite correspond à l'intersection des droites obtenues comme duales de chacun de ces deux points. Par exemple si on considère les points $(1, v_0)$ et $(3, 0)$, la droite passant par ces deux points a été appelée Λ , son dual Λ^* est bien l'intersection de Δ et de la droite $Y = 3X$.

De plus, on montre juste après que ce point Λ^* aussi l'intersection entre droites la plus à droite de l'origine que l'on peut trouver. C'est donc pour des $Y > \nu$ que la droite $Y = X + v_0$ est la plus « haute » dans la représentation de notre système linéaire comme fait sur la Figure 2.

On obtient le résultat important suivant.

Théorème 2.16. *Supposons $\nu \geq 0$. Alors $(y_0, \dots, y_{\lfloor \nu \rfloor})$ est un vecteur composé des premiers termes du développement en série entière d'une solution $y = y_0 + \dots + y_{\lfloor \nu \rfloor} x^{\lfloor \nu \rfloor} + \dots$ de l'équation (EQN) si, et seulement si, ce vecteur est solution du système linéaire obtenu en considérant la sous-matrice de la partie de taille $(\lfloor \mu \rfloor + 1) \times (\lfloor \nu \rfloor + 1)$ en haut à gauche de la matrice $R(L)$. De plus, sous ces conditions, toute solution y dont les $\lfloor \nu \rfloor + 1$ premiers termes sont donnés est déterminée de manière unique.*

Démonstration. Une solution $y = y_0 + y_1 x + \dots$ est solution si, et seulement si, ses coefficients vérifient le système linéaire $(R_m)_{m \geq 0}$. Par ailleurs

$$n > \nu \iff v_0 + n < v_1 + b^1 n < \dots < v_r + b^r n.$$

Si $n > \nu$ alors le premier terme non nul de la n -ème colonne de $R(L)$ se situe sur la $(n + v_0)$ -ième ligne, ou plus bas. Cette ligne permet d'exprimer y_n en fonction uniquement de y_0, \dots, y_{n-1} . Ainsi si on se donne $(y_n)_{0 \leq n \leq \nu}$, un seul choix est possible pour $(y_n)_{n > \nu}$ satisfaisant les équations R_m pour $m \geq \nu + v_0 = \mu$. Donc les équations $(R_m)_{0 \leq m \leq \mu}$ ne font intervenir que les variables $(y_n)_{0 \leq n \leq \nu}$. \square

Remarque 2.17. Pour calculer les coefficients successifs du développement en série entière d'une fonction mahlérienne, l'idée est de résoudre un système linéaire de petite taille, correspondant à la recherche des premiers termes puis une fois ces termes possibles calculés, de déduire tous les suivants en déroulant une récurrence. On peut formaliser ceci de deux manières et j'ai implémentés les deux codes. Soit on détermine les conditions initiales en calculant les valuations à l'origine possibles pour notre solution et on applique une formule de récurrence directement comme dans l'égalité (9). Cette idée est reprise par l'algorithme `calcul_coeff_N` qui calcule le n -ème coefficient d'une fonction mahlérienne en prenant en entrée les $\nu + 1$ premiers termes, en vertu du théorème 2.16. Soit on adopte le formalisme matriciel, on résout alors un sous-système linéaire pour déterminer les premiers termes, c'est l'objet de l'algorithme `approx_series_sol`. Ensuite on résout un système matriciel triangulaire pour étendre nos solutions séries. Ceci est effectué par l'algorithme `prolongement_polys`.

Remarque 2.18. Considérons maintenant une transformation \mathbb{K} -linéaire Φ par :

$$\Phi(x^j M^k) = x^{\alpha b^k + \beta j - \gamma} M^k \tag{14}$$

où $\alpha, \gamma \in \mathbb{Z}$, $\beta \in \mathbb{N}^*$ et $\beta \wedge b = 1$.

J'introduis également de nouvelles variables $\tilde{v}_k, \tilde{\nu}, \tilde{\mu}$ et \tilde{p}_k pour faire référence au système $\Phi(L)(f) = 0$ au lieu de $L(f) = 0$.

L'objectif, repris de (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016), est de pouvoir considérer également des solutions ramifiées, notamment si on veut exploiter ces algorithmes pour la recherche de solutions en série de Puiseux. Les codes présentés fournissent cette option, bien que dans cet exposé, seul le cas des séries entières nous intéressera. On pourra se référer à (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016) (section 2.7) pour les adaptations à faire pour la recherche de série de Puiseux solutions.

Je présente ci-dessous les algorithmes utiles pour mettre en place la recherche des coefficients des solutions développables en série entière et des solutions polynomiales.

Remarque 2.19. C'est la fonction `sous_matrice` dans le programme que j'ai développé.

Proposition 2.20. *L'algorithme présenté ci-dessus permet de remplir une sous-matrice R_E de $R(\Phi(L))$ avec $R_E := (R_{m,n})_{\substack{m \in E \\ 0 \leq n < w}}$.*

Algorithme 1 REMPLIR LA MATRICE $(R_{m,n})_{m,n}$

Entrées: • Une équation de Mahler dy type (EQN).

- Une transformation Φ du type (14).
- Un entier ω .
- Un ensemble $E = \{m_1, m_2, \dots, m_e\}$ d'indices tels que $m_0 < m_1 < \dots < m_e$.

Sortie(s): La sous-matrice $R_E = (R_{m,n})_{\substack{m \in E \\ 0 \leq n \leq \omega}}$ de la matrice (infinie) $R(\Phi(L))$.

1: Soit R_E matrice nulle de taille $e \times \omega$.

2: **pour** $1 \leq i \leq e$:

3: **pour** $0 \leq k \leq r$:

4: $B = m_i + \gamma - \alpha b^k$.

5: $j'_0 = \beta^{-1} B \bmod b^k$

.....
/* $0 \leq j'_0 < b^k$ */

6: $j' = j'_0$.

7: **tant que** $j' \leq d_k$ $\beta j' \leq B$:

8: **si** $\beta j' > B - b^k \omega$ **alors**

9: $(R_E)_{i, b^{-k}(B - \beta j')} = (R_E)_{i, b^{-k}(B - \beta j')} + p_{k, j'}$.

10: $j' = j' + b^k$.

11: **renvoyer** R_E

Démonstration. Rappelons brièvement quelques notations. On note $\tilde{L} := \Phi(L) = \sum_{k=0}^r \tilde{p}_k M^k$. Soit $y = \sum_{n \geq 0} y_n x^n$ solution de $\tilde{L}(y) = 0$. Par définition de la matrice $R(\phi(L))$, $R_{m,n}$ est le coefficient de $y_n x^m$ dans l'expression de $\tilde{L}(y)$. Or, précisément, on a :

$$\tilde{L}(y) = \sum_{k=0}^r \sum_{j=0}^d \tilde{p}_{k,j} x^j \sum_{n=0}^{\infty} y_n x^{b^k n} = \sum_{n=0}^{\infty} \sum_{m=0}^{d+b^r n} \left(\sum_{\substack{k,j \\ t.b. \ j+b^k n=m}} \tilde{p}_{k,j} \right) y_n x^m.$$

Par définition de Φ , $\tilde{p}_{k,j} := \Phi(p_{k,j}) = p_{k,j'}$ avec $j := \alpha b^k + \beta j' - \gamma$. En particulier, si β ne divise pas $\alpha b^k - \gamma$, alors $\tilde{p}_{k,j} = 0$. Par conséquent, $R_{m,n}$ est égal à la somme des $p_{k,j'}$ pour (k, j') vérifiant $\alpha b^k + \beta j' - \gamma = m - b^k n$.

On se fixe désormais m et k . Posons $B := m + \gamma + \alpha b^k$. Le coefficient $p_{k,j}$ intervient alors dans le calcul de $R_{m,n}$ quand $n = b^{-k}(B - \beta j')$. Comme on suppose $0 \leq n < \omega$, on a : $B - b^k \omega \leq \beta j' < B$. Par hypothèse, $\beta \wedge b = 1$ donc les j' qui nous intéressent vérifient $j' \equiv j'_0 \bmod b^k$ où $j'_0 := \beta^{-1} B$. \square

Exemple 2.21. On reprend notre exemple (8). La sous-matrice $(R_{m,n})_{m,n}$ comportant les 20 premières lignes et les 10 premières colonnes obtenue à l'aide de l'algorithme 1 qui s'appelle

`sous_matrice` dans le programme que j'ai développé est la suivante :

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

Ceci est à rapprocher de la Figure 2, avec, au lieu des cercles pleins pour les entrées non nulles, les valeurs des coefficients.

Algorithme 2 CALCUL DE TOUTES LES SOLUTIONS POLYNOMIALES DE DEGRÉ MAJORÉ D'UN SOUS-SYSTÈME DU SYSTÈME INITIAL

Entrées: • Une équation de Mahler du type (EQN).

- Une transformation Φ du type (14).
- Deux entiers h et w .
- Un ensemble $E = \{m_0, m_1, \dots, m_{w-1}\}$ d'indices tel que $m_0 < m_1 < \dots < m_{w-1} < h$ et tel que la sous-matrice $(R_{m_i, j})_{1 \leq i, j \leq w}$ de $R(\Phi(L))$ soit triangulaire avec au plus r zéros sur la diagonale.

Sortie(s): Un vecteur (f_1, \dots, f_σ) de polynômes solutions de (EQN) de degré inférieur à w .

- 1: Construire la matrice S_E à partir des données et en utilisant l'algorithme 1.
 - 2: Calculer $\ker S_E$ et construire la matrice $G = (G_{i,j}) \in \mathcal{M}(\mathbb{K})_{w \times \rho}$ dont les colonnes forment une base du noyau (à droite) de S_E .
 - 3: **pour** $0 \leq j \leq \rho - 1$:
 - 4: $g_j := G_{0,j} + G_{1,j}x + \dots + G_{w-1,j}x^{w-1} \in \mathbb{K}[x]$.
 - 5: Calculer $L(g_j(x)) \bmod x^h =: \sum_{0 \leq i < h} s'_{i,j} x^i$.
 - 6: Construire $S' := (s'_{i,j}) \in \mathcal{M}(\mathbb{K})_{h \times \rho}$.
 - 7: Calculer $\ker S'$ et construire matrice K dont les colonnes forment une base du noyau (à droite) de S' .
 - 8: Calculer $F = (F_{i,j}) := GK \in \mathcal{M}(\mathbb{K})_{w \times \sigma}$.
 - 9: **pour** $1 \leq j \leq \sigma$:
 - 10: $f_j := F_{0,j} + \dots + F_{w-1,j}x^{w-1}$.
 - 11: **renvoyer** (f_1, \dots, f_σ) .
-

L'algorithme 2 est implémenté par la fonction `sol_poly` dans le programme que j'ai développé.

Proposition 2.22. *L'algorithme 2 renvoie une base du noyau de l'application linéaire induite par $\Phi(L) : \mathbb{K}[x]_{<w} \mapsto \mathbb{K}[x]/(x^h)$.*

Démonstration. Notons S_E la sous matrice de $(R_{m,n})$ qui a été construite par l'algorithme 1 et notons S la sous-matrice de R contenant les h premières lignes et les w premières colonnes. S est alors de taille $h \times w$. Par construction, on a $S_E G = 0$. Mais par définition de S' on a aussi $S' = SG$ avec les notations de l'algorithme 2. Par ailleurs on a construit K et F de sorte que $S'K = 0$ et $F = GK$. On veut montrer que l'algorithme 2 renvoie une base du noyau à droite de S . Soit f un vecteur colonne de hauteur w tel que $Sf = 0$. Comme S_E est une sélection de certaines lignes de S uniquement, on a également $S_E f = 0$. Ceci implique alors qu'il existe un vecteur η tel que $f = G\eta$. Comme $S' = SG$, il suit que $S'\eta = 0$ et donc qu'il existe λ tel que $\eta = K\lambda$. D'où finalement $f = GK\lambda = F\lambda$. Réciproquement, $SF = SGK = S'K = 0$, donc tout vecteur de la forme $F\lambda$ est dans le noyau (à droite) de S . Il reste à voir que ces vecteurs forment bien une base de $\ker S$. Les colonnes de G et de K sont, par construction, linéairement indépendantes. Donc

$$F\lambda = 0 \Rightarrow GK\lambda = 0 \Rightarrow K\lambda = 0 \Rightarrow \lambda = 0.$$

Donc les colonnes de F sont bien linéairement indépendantes et forment ainsi une base de $\ker S$. \square

Algorithme 3 CALCUL DES PREMIERS TERMES DES SOLUTIONS SÉRIES

Entrées: Un opérateur L d'ordre r .

Sortie(s): Une base (f_1, \dots, f_σ) de solutions en série approchées de L .

- 1: Définir μ, ν comme dans (13).
 - 2: **si** $\nu < 0$ **alors**
 - 3: **renvoyer** la base vide, $()$.
 - 4: **sinon**
 - 5: Appeler l'algorithme 2 avec :
 - $h = \lfloor \mu \rfloor + 1$.
 - $w = \lfloor \nu \rfloor + 1$.
 - $\Phi = Id$.
 - $E = (\min_k (v_k + nb^k))_{0 \leq n \leq w}$.
 - 6: **renvoyer** le résultat obtenu.
-

C'est la fonction `approx_series_sol` dans le programme que j'ai développé.

Proposition 2.23. *Supposons $\nu \geq 0$. L'algorithme 2 appelé avec $h = \lfloor \mu \rfloor + 1$, $w = \lfloor \nu \rfloor + 1$, $\Phi = Id$ et $E = (\min_k (v_k + nb^k))_{0 \leq n \leq w}$ renvoie une base de l'espace des solutions en séries approchées de $L(y) = 0$.*

Remarque 2.24. Ici on ne traite que le cas où $\Phi = Id$ mais tout ce qui suit est général, il suffit d'étudier l'opérateur \tilde{L} à la place de L .

Démonstration. On reprend les notations de l'algorithme 2. Pour $m = m_i \in E$, aucun des termes de $p_k M^k$ de L n'intervient dans le calcul des entrées de S situées au dessus à gauche de $S_{m,n}$. Il suit que la matrice S_E est triangulaire inférieure. Par ailleurs pour $m \in E$: $R_{m,n} = 0 \iff -n$ est une pente admissible du polygone inférieur de Newton, *i.e.* il y a un croisement (compensation) au point de coordonnées (m, n) quand on trace, comme c'est fait dans la Figure 2, les droites $Y = b^k X + j$ pour tous les couples (b^k, j) présents sur le diagramme de Newton. Comme on peut le voir dans l'équation (10), il n'y a que r pentes (admissibles) possibles pour le polygone inférieur de Newton. Donc il y a au plus r termes nuls sur la diagonale de S_E . Les conditions d'appel à notre algorithme 2 sont donc respectées. D'après le théorème 2.16 et la proposition 2.22 il suit que les f_j construits par l'algorithme 2 forment une base de solutions approchées, au sens où on ne récupère que les premiers termes du développement, du système $L(y) = 0$. \square

Algorithme 4 PROLONGEMENT D'UNE SOLUTION DÉVELOPPABLE EN SE À TOUT ORDRE

Entrées: • Une équation de Mahler de type (EQN).

- Une transformation Φ .
- Un polynôme $\hat{y} = y_0 + \dots + y_{\lfloor \tilde{\nu} \rfloor} x^{\lfloor \tilde{\nu} \rfloor}$ vérifiant $\Phi(L)(\hat{y}) = \mathcal{O}(x^{\lfloor \tilde{\mu} \rfloor + 1})$.
- Un entier n .

Sortie(s): Un polynôme $y_0 + \dots + y_{\lfloor \tilde{\nu} \rfloor + n} x^{\lfloor \tilde{\nu} \rfloor + n}$ solution de (EQN).

1: Utiliser l'algorithme 1 avec :

- $E = \{\lfloor \tilde{\mu} \rfloor + 1, \dots, \lfloor \tilde{\mu} \rfloor + n\}$.
- $h = \lfloor \tilde{\mu} \rfloor + n + 1$.
- $\omega = \lfloor \tilde{\nu} \rfloor + n + 1$.

Obtenir ainsi une sous-matrice R_E de $R(\Phi(L))$.

2: Résoudre $R_E(y_0, \dots, y_{\lfloor \tilde{\nu} \rfloor + n})^T$ en les inconnues $(y_{\lfloor \tilde{\nu} \rfloor + 1}, \dots, y_{\lfloor \tilde{\nu} \rfloor + n})$.

3: **renvoyer** $y_0 + \dots + y_{\lfloor \tilde{\nu} \rfloor + n} x^{\lfloor \tilde{\nu} \rfloor + n}$.

C'est la fonction `prolongement_polys` dans le programme que j'ai développé.

Algorithme 5 CALCUL DE SOLUTIONS POLYNOMIALES DE DEGRÉ MAJORÉ

Entrées: • Un opérateur L d'ordre r .

- Un entier w .

Sortie(s): Une base (f_1, \dots, f_σ) de solutions polynomiales de L de degré inférieur ou égal à w .

1: Appeler l'algorithme 2 avec :

- $h = \max_k d_k + (w - 1)b^r + 1$.
- w .
- $\Phi = Id$.
- $E = (\max_k (d_k + nb^k))_{0 \leq n \leq w}$.

2: **renvoyer** le résultat obtenu.

C'est la fonction `poly_sol_borne` dans le programme que j'ai développé.

Proposition 2.25. *Supposons $\nu \geq 0$. L'algorithme 2 appelé avec $h = \max_k d_k + (w-1)b^r + 1$, $\Phi = Id$, $E = (\max_k (d_k + nb^k))_{0 \leq n \leq w}$ et un entier w qu'on a choisi, renvoie une base de solutions polynomiales pour l'équation (EQN) de degré majoré par w .*

Démonstration. On suit la même démarche que pour la preuve de la proposition 2.23. La sous-matrice qu'on étudie est cette fois triangulaire supérieure par choix des indices dans E et comme précédemment, il y a au plus r termes non nuls sur la diagonale de la matrice S_E car de tels zéros correspondraient à des pentes admissibles sur le polygone supérieur de Newton et on a vu qu'il y en avait au plus r . Donc les conditions d'appel de l'algorithme 2 sont réunies. Par ailleurs, chaque fois que $\deg_x(y) < w$, on a $\deg_x(L(y)) < h$ par choix de h . Donc d'après la proposition 2.22, la base renvoyée par l'algorithme 2 avec ces entrées, correspond, d'après ce qui précède, au noyau de l'application induite par L de $\mathbb{K}[x]_{<w}$ vers $\mathbb{K}[x]$. Donc on obtient bien ce qui était annoncé. \square

Algorithme 6 CALCUL D'UNE BASE DE SOLUTIONS POLYNOMIALES

Entrées: Un opérateur L d'ordre r .

Sortie(s): Une base (f_1, \dots, f_σ) de solutions polynomiales de L .

- 1: Appeler l'algorithme 5 avec $w = \lfloor \frac{\max_k d_k}{b^{r-1}(b-1)} \rfloor + 1$.
 - 2: **renvoyer** le résultat obtenu.
-

C'est la fonction `base_sol_poly` dans le programme que j'ai développé.

Exemple 2.26. On reprend l'exemple donné par l'équation (8) pour lequel l'exposant de Mahler est $b = 3$. On veut déterminer s'il existe des solutions polynomiales ou développables en série et dans ce dernier cas, en donner un développement jusqu'à un ordre souhaité.

L'opérateur L est donné par ses coefficients.

```
liste_polynomes := [-x^37 - x^36 - x^28 - x^27 + x^7 + x^6,
x^40 + x^37 + x^31 + x^28 - 1,
-x^19 - x^10 + x^9 - x^6 + x^3].
```

On calcule d'abord les valuations possibles en 0 des solutions et on calcule les conditions initiales possibles avant de dérouler la récurrence.

```
valuations_possibles(3, liste_polynomes) := [3].
```

```
condi_ini(3, liste_polynomes) :=
r(0) == r(2) + r(0)
r(1) == -r(0)
r(2) == -r(1)
r(3) == r(3) - r(2) - r(0).
```

```
approx_series_sol(3, liste_polynomes) := [x^3].
```

On sait désormais que le seul degré de liberté réside dans le choix du coefficient y_3 de notre solution $L(y) = 0$. On se donne pour conditions initiales $[0, 0, 0, 1]$. Je demande le développement de la solution correspondante jusqu'à l'ordre 50.

```
prolongement_polys(3, liste_polynomes,
phi = [0,1,0], 50, [0,0,0,1]) :=
```

```
-63 x^50 + 63 x^49 - 63 x^48 + 54 x^47 - 54 x^46 +
54 x^45 - 47 x^44 + 47 x^43 - 47 x^42 + 40 x^41 - 40
x^40 + 40 x^39 - 33 x^38 + 33 x^37 - 33 x^36 + 28
x^35 - 28 x^34 + 28 x^33 - 23 x^32 + 23 x^31 - 23
x^30 + 18 x^29 - 18 x^28 + 18 x^27 - 15 x^26 + 15
x^25 - 15 x^24 + 12 x^23 - 12 x^22 + 12 x^21 - 9
x^20 + 9 x^19 - 9 x^18 + 7 x^17 - 7 x^16 + 7 x^15 -
5 x^14 + 5 x^13 - 5 x^12 + 3 x^11 - 3 x^10 + 3 x^9 -
2 x^8 + 2 x^7 - 2 x^6 + x^5 - x^4 + x^3.
```

```
base_sol_poly(3, liste_polynomes) := [].
```

On constate donc qu'il n'y a pas de solutions polynomiales pour cet exemple.

2.3 Recherche de solutions rationnelles

Dans cette sous-section, je m'intéresse aux solutions qui sont dans $\mathbb{K}(x)$ où \mathbb{K} est un corps de nombre. Les résultats qui suivent sont repris de (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016). Ma part de travail a essentiellement consisté à implémenter les procédures présentées ci-dessous.

Cadre : On s'intéresse ici à des solutions de (EQN) de la forme $y = p/(x^{\bar{v}}q)$ en demandant que :

- $\bar{v} \geq 0$,
- $p \wedge q = 1 \in \mathbb{K}[x]$,
- $q(0) \neq 0$,
- $p(0) = 0 \Rightarrow \bar{v} = 0$.

On définit également $T_a := \bigvee_{i=0}^{r-1} M^i a$ où \bigvee désigne le ppcm.

Remarque 2.27. Soit $y = p/(x^{\bar{v}}q)$ solution de

$$\ell_r(x)M^r y + \dots + \ell_1(x)M y + \ell_0(x)y = 0.$$

En multipliant cette équation par $(M^r x^{\bar{v}})(M^j q) \bigvee_{i \neq j} M^i q$ avec $0 \leq j \leq r$ et réduisant modulo $M^j q$, on obtient

$$M^j q \mid x^{(br-bj)\bar{v}} \ell_j \times (M^j p) \bigvee_{i \neq j} M^i q.$$

En prenant $j = r$, comme p et q sont premiers entre eux, on obtient alors

$$M^r q \mid \ell_r T_q. \tag{15}$$

Maintenant, soit α une racine de q et soit β tel que $\alpha = \beta^{br}$. On a alors

$$\begin{aligned} q(\alpha) = 0 &\implies M^r q(\beta) = 0. \\ \text{D'où } q(\alpha) = 0 &\implies \ell_r(\beta) = 0 \quad \text{ou} \quad T_q(\beta) = 0. \\ \text{Donc } q(\alpha) = 0 &\implies \ell_r(\beta) = 0 \quad \text{ou} \quad (q(\alpha') = 0 \text{ pour } \alpha' = \beta^{b^i}, 0 \leq i < r). \end{aligned}$$

Ceci se réécrit

$$q(\alpha) = 0 \implies (\ell_r(\beta) = 0 \text{ pour } \beta^{br} = \alpha) \quad \text{ou} \quad (q(\alpha') = 0 \text{ pour } (\alpha')^{b^k} = \alpha, k > 0).$$

q étant un polynôme non nul, on ne pourra pas construire une infinité de α' . Qui plus est, si on met de côté le cas où α est une racine de l'unité, qu'on traitera à part, la démarche exposée met donc en évidence une suite finie de racines distinctes de q obtenues à partir de α . Il suit alors, qu'après un nombre fini d'étape, on trouvera nécessairement un nombre β tel que $\beta^{br} = \alpha$ et $\ell_r(\beta) = 0$.

Notre premier objectif dans cette sous-section est de trouver un multiple de tous les dénominateurs possibles parmi les solutions rationnelles solutions de l'équation considérées, écrites sous la forme souhaitée. La remarque 2.27 fournit quelques conditions que doit vérifier le dénominateur q de toute solution rationnelle. Notons également que les calculs qu'on a considérés n'ont fait intervenir que le polynôme de tête ℓ_r de l'équation mahlérienne considérée. On formalisera et complètera cette remarque dans le lemme 2.37.

Commençons par définir un nouvel opérateur : *l'opérateur de Gräffe*.

$$\begin{aligned} G : \mathbb{K}[x] &\rightarrow \mathbb{K}[x] \\ p &\mapsto \text{Res}_y(y^b - x, p(y)) \end{aligned}$$

Remarque 2.28. $G(p) = p(x^{1/b})p(\zeta x^{1/b}) \dots p(\zeta^{b-1} x^{1/b})$ où ζ est une racine primitive b -ième de l'unité quelconque.

Remarque 2.29. Les zéros de $M(p)$ sont les racines b -ièmes des zéros de p . Les zéros de $G(p)$ sont les puissances b -ièmes des zéros de p . Il vient alors

$$\deg Mp = \deg p, \quad \deg Gp = \deg p.$$

On donne ici quelques lemmes qui nous seront utiles par la suite, sur l'utilisation et les propriétés de *l'opérateur de Gräffe*.

Lemme 2.30. *On a les propriétés suivantes : $\forall i \in \mathbb{N}$ et $\forall p, q \in \mathbb{K}[x]$,*

- (a) $G^i M^i p = p^{b^i}$,
- (b) $p \mid M^i G^i p$,
- (c) $p \mid q \iff M^i p \mid M^i q$.

Démonstration. Voir [ici](#). □

Lemme 2.31. (a) *Si $p \in \mathbb{K}[x]$ est unitaire et irréductible, alors pour $i \in \mathbb{N}$, on a $G^i p = q^e$ pour un certain $q \in \mathbb{K}[x]$ unitaire irréductible et $e \in \mathbb{N}$.*

- (b) $G^i p = p \iff p \mid M^i p$.

(c) Si $G^i p = p$ pour un $i > 0$, alors $\forall j \in \mathbb{N}$, $G^j p$ est unitaire et irréductible.

Démonstration. Voir [ici](#). □

Lemme 2.32. Soit $f \in \mathbb{K}[x]$ polynôme non constant vérifiant $f(0) \neq 0$. Si f et f' sont premiers entre eux, alors (Mf) et $(Mf)'$ le sont aussi.

Démonstration. Voir [ici](#). □

Lemme 2.33. Soit $p \in \mathbb{K}[x]$ unitaire irréductible. On a

$$\forall i \geq 1, G^i p \neq p \implies \forall i > j \geq 0, M^i p \wedge M^j p = 1.$$

Démonstration. Voir [ici](#). □

Lemme 2.34. Soit \mathbb{K} un corps parfait. Soit $f \in \mathbb{K}[x]$ polynôme non constant vérifiant $f(0) \neq 0$. Il existe un polynôme unitaire irréductible $q \in \mathbb{K}[x]$ facteur de Mf tel que $\forall k \in \mathbb{N}^*$, $G^k q \neq q$.

Démonstration. Voir [ici](#). □

Lemme 2.35. Soit $f \in \mathbb{K}[x]$ unitaire et irréductible qui vérifie $\forall i > 0$, $G^i f \neq f$. Soit $g \in \mathbb{K}[x]$ unitaire et irréductible également, diviseur de $M^j f$ pour un certain $j \geq 0$. Alors on a $\forall i > 0$, $G^i g \neq g$.

Démonstration. Voir [ici](#). □

Lemme 2.36. Soient $s \geq r - 1$, $m \geq 1$ deux entiers, $f \in \mathbb{K}[x]$ unitaire irréductible, $q \in \mathbb{K}[x]$ polynôme non constant et $\ell \in \mathbb{K}[x]$ non nul tel que :

- $x \nmid q$,
- $M^s f^m \mid M^r q \mid \ell T_q$,
- $\forall 0 \leq i < r$, $M^{s-i} f \wedge q = 1$.

Alors $M^s f^m \mid \ell$.

Démonstration. Voir [ici](#). □

Lemme 2.37. Soit $\ell \in \mathbb{K}[x]$ un polynôme non nul et $q \in \mathbb{K}[x]$ un polynôme non constant tels que $x \nmid q$ et $M^r q \mid \ell T_q$. Alors il existe un polynôme non constant $u \in \mathbb{K}[x]$ vérifiant :

- $M^r u \mid \ell$,
- ou bien $M^{r-1} u \mid \ell$ et $q \mid Gu$.

Démonstration. Voir [ici](#). □

Lemme 2.38. Soient $f_0, \dots, f_{b-1}, u \in \mathbb{K}[x]$. Le polynôme $\ell = Mf_0 + xMf_1 + \dots + x^{b-1}Mf_{b-1}$ est divisible par Mu si, et seulement si, chacun des f_i est divisible par u .

Démonstration. Voir [ici](#). □

Algorithme 7 RECHERCHE D'UN MULTIPLE COMMUN À TOUS LES DÉNOMINATEURS POSSIBLES PARMIS LES SOLUTIONS RATIONNELLES

Entrées: • Le polynôme ℓ_r de tête dans l'équation mahlérienne de type (EQN).

- L'ordre r de cette équation.
- La base b de Mahler.

Sortie(s): $q^* \in \mathbb{K}[x]$ multiple des dénominateurs possibles pour nos solutions rationnelles.

1: Poser $\ell := \ell_r$.

2: **répéter pour** $k = 1, 2, \dots$:

3: Écrire $\ell = \sum_{i=0}^{b^r-1} x^i M^r f_i$ avec $f_i \in \mathbb{K}[x]$.

4: Calculer $u_k = \bigwedge_{i=0}^{b^r-1} f_i$.

5: Remplacer $\ell = (\ell / M^r u_k) \bigvee_{i=0}^{r-1} M^i u_k$.

6: **jusqu'à** $\deg(u_k) = 0$.

7: $t = k - 1$.

8: Calculer $\tilde{u} = \bigvee_{i=0}^{b^{r-1}-1} f_i$ avec $\ell = \sum_{i=0}^{b^{r-1}-1} x^i M^{r-1} f_i$.

9: **renvoyer** $u_1 \dots u_t(G\tilde{u})$ où G est l'opérateur de Gräffe.

Remarque 2.39. C'est la fonction `multiple_denominateur` dans le programme que j'ai développé.

Remarque 2.40 (esquisse de preuve). Les lemmes 2.37 et 2.38 forment la base théorique de l'algorithme de recherche d'un dénominateur commun à toutes les solutions rationnelles de notre équation mahlérienne. Le lemme 2.37 nous conduit à chercher des facteurs de la forme $M^r u$ ou de la forme $M^{r-1} u$ qui divisent ℓ_r , le terme de tête. Cette recherche exploite le lemme 2.38 pour les bases respectives b^r et b^{r-1} , c'est pour cela que l'on recherche les facteurs u sous forme de ppcm de polynômes f_i . L'algorithme 7 commence par chercher des polynômes u tels que $M^r u$ divise ℓ_r . À chaque étape, après avoir calculé un u_k comme souhaité, on attribue une nouvelle valeur à ℓ dans l'algorithme 7. Cette actualisation rend compte qu'à l'étape suivante ce n'est plus la solution y qu'on traite mais la solution $\tilde{y} := y u_k$. En effet on a déjà trouvé un terme u_k qui est par construction un facteur du multiple commun des dénominateurs possibles renvoyé à la sortie. Enfin, l'existence d'un u tel que $M^r u \mid \ell$ n'est pas toujours vraie donc l'algorithme 7 calcule également dans un second temps un \tilde{u} tel que $M^{r-1} \tilde{u} \mid \ell$ et on multiplie le résultat qu'on renvoie par $G\tilde{u}$ en vertu du second point du lemme 2.37 car on sait qu'alors $q \mid G\tilde{u}$. Ce dernier q , si on appelle y notre solution rationnelle initiale, correspond à la solution $y' := u_1 \dots u_t y$, donc le résultat renvoyé, à savoir $u_1 \dots u_t(G\tilde{u})$, est bien un multiple commun à tous les dénominateurs.

L'obtention de l'algorithme 8 pour le calcul d'une base des solutions rationnelles exploite l'algorithme 7 et l'algorithme 5 sur les solutions polynomiales. Les spécificités propres aux fonctions mahlériennes n'apparaissant qu'à travers les appels à ces deux algorithmes, nous ne détaillons donc pas sa preuve ici. Sa démonstration ainsi que l'étude de sa complexité sont traitées dans la proposition 3.22 de (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016). Nous donnons ici l'implémentation qui en a été faite.

Algorithme 8 CALCUL D'UNE BASE DE SOLUTIONS RATIONNELLES

Entrées: Une équation de Mahler de type (EQN). On note L son opérateur associé.

Sortie(s): Une base de solutions rationnelles.

- 1: $\delta := \max_k \deg(p_k)$.
 - 2: **si** $\delta < b^{r-1}$ **alors**
 - 3: **si** $L(1) = 0$ **alors**
 - 4: **renvoyer** (1)
 - 5: **sinon**
 - 6: **renvoyer** ()
 - 7: Calculer q^* à l'aide de l'algorithme 7 et poser $\bar{v} = \lfloor \frac{\delta}{(b^r - b^{r-1})} \rfloor$.
 - 8: **pour** $0 \leq k \leq r$:
 - 9: $e_k = \lfloor \frac{b\delta}{(b-1)} \rfloor - b^k \bar{v}$
 - 10: $\tilde{p}_k = x^{e_k} p_k \prod_{0 \leq i \leq r, i \neq k} M^i q^*$.
 - 11: $\tilde{L} = \tilde{p}_r M^r + \dots + \tilde{p}_0$.
 - 12: Appeler l'algorithme 5 pour le système $\tilde{L}(p) = 0$ avec $w = \deg(q^*) + 2\bar{v} + 1$.
.....
/* On obtient une base (p_1, \dots, p_σ) de solutions polynomiales de degré majoré par w . */
.....
 - 13: **renvoyer** $(p_k / (x^{\bar{v}} q^*))_{1 \leq k \leq \sigma}$.
-

Remarque 2.41. C'est la fonction `sol_ratio` dans le programme que j'ai développé.

Exemple 2.42. Donnons un exemple d'une recherche de solutions rationnelles. Soit

$$(x^2 + x - 1)f(x) + (-x^4 - x^2 + 1)f(x^2) = 0$$

une équation mahlérienne homogène, d'exposant $b = 2$ et d'ordre $r = 1$.

`liste_polynomes := [x^2 + x - 1, -x^4 - x^2 + 1].`

`multiple_denominateur(liste_polynomes[-1], b = 2, r = 1) :=
x^4 - 2x^3 - 3x^2 + 4x - 1.`

`_.factor() := (x^2 - 3x + 1).(x^2 + x - 1).`

`sol_ratio(b = 2, liste_polynomes) := [1/(x^2 + x - 1)].`

Exemple 2.43. Si on reprend l'exemple 8, on a :

`liste_polynomes := [-x^37 - x^36 - x^28 - x^27 + x^7 + x^6,`

`x^40 + x^37 + x^31 + x^28 - 1,`

`-x^19 - x^10 + x^9 - x^6 + x^3].`

`multiple_denominateur(liste_polynomes[-1], b = 3, r = 2) :=
x^3 + 2x^2 + x.`

`sol_ratio(3, liste_polynomes) := [].`

3 Étude des relations de dépendance linéaire de fonctions mahlériennes

On choisit le cadre où notre fonction est donnée comme la coordonnée d'un système du type (2). L'objectif désormais est de trouver une équation minimale pour une fonction mahlérienne fixée. Pour y parvenir, nous allons étudier les relations de dépendance linéaire entre les fonctions coordonnées.

3.1 Transcendance de valeurs mahlériennes et transcendance de fonctions mahlériennes

Dans cette première partie, nous présentons certains résultats théoriques importants pour justifier les procédures algorithmiques qui vont suivre. On renvoie le lecteur à (Adamczewski et Faverjon, 2017) pour les démonstrations des résultats suivants.

Définition 3.1. Le $\mathbf{k}(x)$ -espace vectoriel des relations linéaires fonctionnelles entre les $f_i(x)$ est défini par :

$$\text{Rel}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)) := \left\{ (w_1(x), \dots, w_n(x)) \in \mathbf{k}(x)^n \mid \sum_{i=1}^n w_i(x) f_i(x) = 0 \right\}.$$

Définition 3.2. Un point α est une *singularité* du système (2) (point singulier) s'il existe un entier positif ℓ tel que α^{b^ℓ} soit un pôle d'un coefficient de la matrice $A(x)$ ou une racine de son déterminant. Dans le cas contraire, on parle de point régulier.

Définition 3.3. Soient f_1, \dots, f_r des fonctions b -mahlériennes, chacune donnée par une équation de type (1) ou un système de type (2) et soit α , $0 < |\alpha| < 1$, un nombre algébrique. Le $\overline{\mathbb{Q}}$ -espace vectoriel des relations linéaires entre les $f_i(\alpha)$ est défini par :

$$\text{Rel}_{\overline{\mathbb{Q}}}(f_1(\alpha), \dots, f_r(\alpha)) := \left\{ (\lambda_1, \dots, \lambda_r) \in \overline{\mathbb{Q}}^r \mid \sum_{i=1}^r \lambda_i f_i(\alpha) = 0 \right\}.$$

Enfin, on note ev_α l'application d'évaluation en $x = \alpha$.

Théorème 3.4 (Philippon). Soient $f_1(x), \dots, f_n(x) \in \overline{\mathbb{Q}}\{x\}$ des fonctions solutions d'un système de type (2). Soit $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$, un point régulier pour ce système. Alors, pour tout $P \in \overline{\mathbb{Q}}[X_1, \dots, X_n]$, de degré total d , tel que $P(f_1(\alpha), \dots, f_n(\alpha)) = 0$, il existe $Q \in \overline{\mathbb{Q}}(x)[X_1, \dots, X_n]$, de degré total d en X_1, \dots, X_n , tel que $Q(x, f_1(x), \dots, f_n(x)) = 0$ et $Q(\alpha, X_1, \dots, X_n) = P(X_1, \dots, X_n)$.

Corollaire 3.5. Soit $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$, un point régulier pour le système (2). On a :

$$\text{Rel}_{\overline{\mathbb{Q}}}(f_1(\alpha), \dots, f_n(\alpha)) = ev_\alpha \left(\text{Rel}_{\overline{\mathbb{Q}}(x)}(f_1(x), \dots, f_n(x)) \right).$$

En particulier, si les fonctions $f_1(x), \dots, f_n(x)$ sont linéairement indépendantes sur $\overline{\mathbb{Q}}(x)$, alors les nombres $f_1(\alpha), \dots, f_n(\alpha)$ sont linéairement indépendants sur $\overline{\mathbb{Q}}$.

Théorème 3.6. Soient $f_1(x), \dots, f_n(x)$ des fonctions b -mahlériennes. Soient $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$, un nombre qui n'est pôle d'aucune de ces fonctions et soit \mathbf{k} un corps de nombres contenant α et les coefficients des f_i . On a alors :

$$\text{Rel}_{\overline{\mathbb{Q}}}(f_1(\alpha), \dots, f_n(\alpha)) = \text{Vect}_{\overline{\mathbb{Q}}} \{ \text{Rel}_{\mathbf{k}}(f_1(\alpha), \dots, f_n(\alpha)) \}.$$

Remarque 3.7. En d'autres termes, ce théorème montre que si les nombres $f_1(\alpha), \dots, f_n(\alpha)$ sont linéairement indépendants sur $\overline{\mathbb{Q}}$, alors ils le sont aussi dans \mathbf{k} .

Comme la fonction identiquement égale à 1 est b -mahlérienne pour tout $b \geq 2$, on retrouve le théorème de Bézivin présenté en introduction.

Corollaire 3.8. *Soit $f(x)$ une fonction b -mahlérienne et $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$, qui n'est pas un pôle de f . Soit \mathbf{k} un corps de nombres contenant α et les coefficients de f . On a l'alternative suivante : soit $f(\alpha)$ est transcendant, soit $f(\alpha) \in \mathbf{k}$.*

On se donne maintenant un réel $0 < \rho < 1$, strictement inférieur au minimum des modules des pôles (on ne considérera que les pôles non nuls) de la matrice $A(x)$ du système (2) et des racines (de même on ne s'intéresse qu'aux racines non nulles) de son déterminant. Ainsi la matrice $A(x)$ est définie et inversible sur le disque épointé $D(0, \rho)^*$.

Notation : Soit $\ell \geq 0$, on note $A_\ell(x) = A(x)A(x^b) \dots A(x^{b^{\ell-1}})$.

Théorème 3.9. *Soient $f_1(x), \dots, f_n(x)$ des solutions d'un système de type (2). Soient $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$ et \mathbf{k} un corps de nombres contenant α et les coefficients des f_i . Soit ℓ un entier tel que $|\alpha|^{b^\ell} < \rho$. Si α n'est pas un pôle de $A_\ell(x)$, alors*

$$\text{Rel}_{\mathbf{k}}(f_1(\alpha), \dots, f_n(\alpha)) = \ker_{\mathbf{k}} A_\ell(\alpha) + \text{ev}_\alpha \left(\text{Rel}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)) \right).$$

Théorème 3.10. *Soient $f_1(x), \dots, f_n(x)$ des solutions d'un système de type (2). Soient $\alpha \in \overline{\mathbb{Q}}$, $0 < |\alpha| < 1$ et \mathbf{k} un corps de nombres contenant α et les coefficients des f_i . Supposons que les f_i soient définis au point α . Soit ℓ un entier tel que $|\alpha|^{b^\ell} < \rho$. Alors, il existe une matrice $B(x) \in GL_n(\mathbf{k}(x))$ satisfaisant aux conditions suivantes.*

(i) On a :

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = B(x) \begin{pmatrix} f_1(x^{b^\ell}) \\ \vdots \\ f_n(x^{b^\ell}) \end{pmatrix}.$$

(ii) Le point α n'est pas un pôle pour $B(x)$.

(iii) Le point α^{b^ℓ} est régulier pour le système du point (i).

De plus, si les fonctions $f_1(x), \dots, f_n(x)$ sont linéairement indépendantes sur $\mathbf{k}(x)$, on a alors $B(x) = A_\ell(x)$.

3.2 Test de dépendance linéaire et extraction d'un sous-système vérifié par des fonctions linéairement indépendantes

Soit un système du type (2). D'une part, on souhaite connaître la dimension, qu'on notera r , de l'espace vectoriel (sur $\mathbf{k}(x)$) engendré par les fonctions intervenant dans (2). D'autre part, on veut aussi déterminer les r -uplets possibles de fonctions linéairement indépendantes, et pour un r -uplet de notre choix, renvoyer un sous-système vérifié par ces r fonctions seulement. Ce sera l'objet de l'algorithme 9 qui suit.

Algorithme 9 CALCUL DE LA DIMENSION DE L'ESPACE VECTORIEL ENGENDRÉ PAR UNE FAMILLE DE FONCTIONS MAHLÉRIENNES

Entrées: Une matrice $A(x)$ associée à un système du type (2) avec $\mathbf{f} = {}^t(f_1, \dots, f_n)$ solution de ce système. La base b de Mahler correspondante.

Sortie(s): $r := \dim(\text{Vect}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)))$. Une matrice $B(x)$ représentant le système associé à un r -uplet de fonctions linéairement indépendantes sur $\mathbf{k}(x)$.

CALCULS PRÉLIMINAIRES :

1: Calculer : $b(x), d, \nu, h, M$ tels que :

- $b(x) := \text{ppcm}$ des dénominateurs des coefficients de $A(x)$.
- $d := \text{maximum}$ des degrés des coefficients de $b(x)A(x)$.
- $\nu := \text{valuation}$ en $x = 0$ de $b(x)$.
- $h := 4^n d$.
- $M := \left\lceil \frac{b^{n\left(\frac{bh+d+1}{b-1}+b+1\right)(h+b)+\nu-\frac{h+d}{b-1}}{b-1} \right\rceil$.

CALCUL DE LA DIMENSION :

2: Soit $\mathbf{f}(x) := \sum_{i=0}^{\infty} \mathbf{f}_i x^i$ où $\mathbf{f}_i = [x^i](\mathbf{f})$.

3: Calculer $\mathcal{S}(\mathbf{f}, h, M) := \begin{pmatrix} \mathbf{f}_0 & \mathbf{f}_1 & \dots & \mathbf{f}_h & \dots & \mathbf{f}_M \\ 0 & \mathbf{f}_0 & \ddots & \ddots & \ddots & \mathbf{f}_{M-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mathbf{f}_0 & \dots & \mathbf{f}_{M-h} \end{pmatrix} \in \text{Mat}_{n(h+1), (M+1)}(\mathbf{k})$.

4: Calculer une base \mathcal{B} de $\ker \mathcal{S}(\mathbf{f}, h, M) := \{\boldsymbol{\lambda} \in (\mathbf{k}^n)^{h+1} \mid \boldsymbol{\lambda} \mathcal{S}(\mathbf{f}, h, M) = 0\}$.

5: Soit σ la dimension de ce noyau.

6: **pour** chaque élément $\mathbf{w} = (w_0, \dots, w_h) \in \mathcal{B}$:

7: Calculer le polynôme $\mathbf{w}(x) := \sum_{i=0}^h w_i x^i$.

8: Écrire $\Lambda(x) := \begin{pmatrix} \mathbf{w}_1(x) \\ \vdots \\ \mathbf{w}_\sigma(x) \end{pmatrix}$.

9: Calculer le rang s de $\Lambda(x)$.

10: **renvoyer** $r := n - s$.

TEST D'INDÉPENDANCE LINÉAIRE :

Objectif : tester si f_{i_1}, \dots, f_{i_r} sont linéairement indépendantes avec $1 \leq i_1 < \dots < i_r \leq n$.

- 11: Soit $J := \{1, \dots, n\} \setminus \{i_1, \dots, i_r\}$.
- 12: Calculer Δ_J le mineur de Λ associé à J .
- 13: **si** $\Delta_J \neq 0$ **alors**
- 14: **renvoyer** vrai.
- 15: **sinon**
- 16: **renvoyer** faux.

.....
 /* On suppose pour la suite que f_1, \dots, f_r sont linéairement indépendantes dans $\mathbf{k}(x)$. */

EXTRACTION D'UN SOUS-SYSTÈME :

- 17: Calculer la forme échelonnée de $\Lambda(x)$.
- 18: Construire :

$$S(x) := \left(\begin{array}{cccccccc} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{array} \right).$$

$\Lambda_s(x)$

avec $\Lambda_s(x)$ le bloc composé des s premières lignes de la forme échelonnée de $\Lambda(x)$.

.....
 /* On obtient alors :

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_r(x) \end{pmatrix} = B(x) \begin{pmatrix} f_1(x^b) \\ \vdots \\ f_r(x^b) \end{pmatrix}.$$

avec $B(x)$ le bloc principal $r \times r$ de la matrice $S(x)A(x)S(x^b)^{-1}$. */

- 19: Calculer $B(x) := (S(x)A(x)S(x^b)^{-1})[:, :r]$, le bloc principal $r \times r$ de cette matrice.
 - 20: **renvoyer** $B(x)$.
-

Remarque 3.11. La matrice $\mathcal{S}(\mathbf{f}, j, M)$ est la matrice sur les bases monomiales de l'application :

$$\begin{aligned} \mathbf{k}[x]_{\leq h}^n &\longrightarrow \mathbf{k}[x]_{\leq M} \\ \mathbf{w}(x) &\longmapsto \langle \mathbf{w}(x) | \mathbf{f}(x) \rangle \text{ mod } x^M. \end{aligned}$$

Remarque 3.12. J'ai décomposé cette procédure en plusieurs fonctions. La fonction présente dans le programme que j'ai développé `liste_fonctions_independantes` renvoie une liste composée de tous les r -uplets d'indices correspondant aux fonctions linéairement indépendantes du système. Je précise aussi que dans la version implémentée, la matrice $\mathcal{S}(\mathbf{f}, h, M)$ qui intervient n'est pas calculée, l'utilisateur entre une sous-matrice de \mathcal{S} calculée au préalable. Si cette dernière est de rang maximal parmi les sous-matrices possibles, la terminaison de l'algorithme reste correcte. Pour la deuxième partie de l'algorithme, à savoir l'extraction d'un sous-système, on suppose que les r fonctions qui nous intéressent et dont a montré qu'elles étaient linéairement indépendantes, sont les r premières à intervenir dans le système (2).

L'algorithme précédent repose sur le théorème suivant, issu de ([Adamczewski et Faverjon, 2017](#)).

Théorème 3.13. *Soient \mathbf{k} un corps de nombres et n fonctions b -mahlériennes $f_1, \dots, f_n \in \mathbf{k}\{x\}$ données par un système de type (2). On reprend les mêmes notations que dans l'algorithme 9. Posons*

$$\begin{aligned} \phi : (\mathbf{k}^n)^{h+1} &\longrightarrow (\mathbf{k}[x]_h)^n \\ (\mathbf{w}_0, \dots, \mathbf{w}_h) &\longmapsto \sum_{i=0}^h \mathbf{w}_i x^i. \end{aligned}$$

Alors :

$$\begin{aligned} \text{Rel}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)) &= \text{Vect} \left(\text{Rel}_{\mathbf{k}[x]_h}(f_1(x), \dots, f_n(x)) \right) \\ &= \text{Vect}(\phi(\ker_{\mathbf{k}} \mathcal{S}(\mathbf{f}, h, M))). \end{aligned}$$

Démonstration. La preuve se trouve chez ([Adamczewski et Faverjon, 2017](#)), section 7, théorème 7.1. Une légère amélioration de la borne M est possible, je la donne dans la sous-section 3.4. \square

Démonstration. (de l'algorithme 9) On reprend les mêmes notations.

Le théorème 3.13 permet de montrer que le r obtenu est bien la dimension de l'espace vectoriel des relations linéaires entre les fonctions f_i . On démontre maintenant la partie concernant les tests d'indépendance linéaire. Supposons que $\Delta_J = 0$. Par définition, on peut trouver $\mu(x) \in \mathbf{k}[x]^s$ non nul tel que $v(x) := \mu(x)\Lambda(x)$ vérifie $v(x) = (v_1(x), \dots, v_n(x))$ avec $v_i(x) = 0$ pour tout i dans J . Par ailleurs, $\Lambda(x)$ est de rang s par définition des $(\mathbf{w}_0, \dots, \mathbf{w}_n)$. Donc $v(x)$ n'est pas nul (sinon il appartiendrait au noyau de Λ qui est réduit au vecteur nul). De plus, $\langle \mathbf{w}(x) \mid \mathbf{f}(x) \rangle = 0$ d'après le Théorème 3.13.

On obtient alors :

$$\sum_{l=1}^r v_{i_l}(x) f_{i_l}(x) = v(x) \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \mu(x)\Lambda(x) \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = 0.$$

D'où f_{i_1}, \dots, f_{i_r} sont linéairement indépendantes dans $\mathbf{k}(x)$.

Réciproquement, supposons que f_{i_1}, \dots, f_{i_r} sont linéairement dépendantes dans $\mathbf{k}(x)$. Par définition, il existe un vecteur $v(x) \in \mathbf{k}[n]^n$ tel que $\sum_{i=1}^n v_i(x) f_i(x) = 0$ et $v_i(x) = 0$ pour tout i dans J . Or d'après le Théorème 3.13, les vecteurs $\mathbf{w}_1(x), \dots, \mathbf{w}_s(x)$ forment une base de $\text{Rel}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x))$, il existe donc un vecteur non nul $\mu(x)$ tel que $\mu(x)\Lambda(x) = v(x)$. Ainsi $\Delta_J = 0$. Pour la dernière partie, on peut supposer sans perte de généralité que les r fonctions dont a montré la linéaire indépendance sont les r premières. On considère alors la matrice

$$S(x) := \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 1 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ \hline & & & & \Lambda(x) & & \end{pmatrix}.$$

On a alors :

$$S(x) \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} f_1(x) \\ \vdots \\ f_r(x) \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

De plus, $S(x)$ est inversible car $\Delta_J \neq 0$ par hypothèse. (Ici $J = \{1, \dots, n\} \setminus \{1, \dots, r\}$).
Finalement en appliquant le résultat précédent aux $f_i(x)$ et aux $f_i(x^b)$ on obtient :

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_r(x) \end{pmatrix} = B(x) \begin{pmatrix} f_1(x^b) \\ \vdots \\ f_r(x^b) \end{pmatrix}.$$

avec $B(x)$ le bloc principal $r \times r$ de la matrice $S(x)A(x)S(x^b)^{-1}$. □

Le théorème 3.13 que nous avons énoncé ci-dessus permet d'exprimer l'ensemble des relations linéaires (sur $\mathbf{k}(x)$) entre les fonctions mahlériennes issues d'un même système de type (2). Néanmoins, si on veut juste tester si certaines de ces fonctions sont linéairement indépendantes, on peut le faire de manière plus efficace (avec une borne M plus petite). On énonce le résultat suivant.

Théorème 3.14. *On reprend les mêmes notations que dans le pseudo-code 9. Cette fois on pose $h := \lfloor \frac{d}{b-1} \rfloor$, le reste des variables étant inchangé dans les définitions. On a les équivalences suivantes :*

- (i) $\text{Rel}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)) \neq \{0\}$.
- (ii) $\text{Rel}_{\mathbf{k}[x]_h}(f_1(x), \dots, f_n(x)) \neq \{0\}$.
- (iii) $\ker_{\mathbf{k}} \mathcal{S}(\mathbf{f}, h, M) \neq \{0\}$.

Pour la démonstration, voir (Adamczewski et Faverjon, 2017) section 7.

3.3 Application : obtention d'une équation minimale

L'étude des relations de dépendance linéaire va nous permettre dans cette sous-section de pouvoir exprimer une procédure pour obtenir une équation inhomogène minimale d'une fonction mahlérienne à condition d'être capable de pouvoir mettre en oeuvre l'algorithme 3.13 sur le système mahlérien étudié.

Définition 3.15. Soit f une fonction b -mahlérienne définie par une équation du type (1). On définit $V_f := \text{Vect}_{\overline{\mathbb{Q}(x)}} \{1, f(x), f(x^b), f(x^{b^2}), \dots\}$. Cet espace vectoriel est donc de dimension finie.

Plus précisément, on a le résultat suivant :

Lemme 3.16. *Soit f une fonction b -mahlérienne d'équation inhomogène minimale*

$$p_{-1}(x) + p_0(x)f(x) + p_1(x)f(x^b) + \dots + p_n(x)f(x^{b^n}) = 0. \quad (16)$$

La dimension de V_f est $n + 1$.

Démonstration. On note $(V_m)_{m \geq 1}$ la suite d'espaces emboîtés

$$V_m := \text{Vect}_{\overline{\mathbb{Q}}(x)} \{1, f(x), f(x^b), \dots, f(x^{b^m})\}.$$

Soit m_0 tel que $V_{m_0+1} = V_{m_0}$. Par définition il existe des fractions rationnelles $q_i(x)$ pour i allant de -1 à m_0 telles que

$$f(x^{b^{m_0+1}}) = q_{-1}(x) + q_0(x)f(x) + \dots + q_{m_0}(x)f(x^{b^{m_0}}).$$

En appliquant cette égalité à $x = x^b$, on trouve :

$$f(x^{b^{m_0+2}}) = q_{-1}(x^b) + q_0(x^b)f(x^b) + \dots + q_{m_0}(x^b)f(x^{b^{m_0+1}}) \in V_{m_0+1}.$$

Donc on vient de montrer que $V_{m_0+2} \subset V_{m_0+1}$. Par minimalité de l'équation (16), on a donc $V_m \subsetneq V_{m+1}$ pour tout $m < n-1$. Et d'autre part, cette même équation implique que $V_n = V_{n-1}$. D'où finalement, $V_f = V_{n-1}$ et $\dim(V_f) = n+1$. \square

Je présente ici la procédure qui nous intéresse.

Algorithme 10 DÉTERMINATION DE L'ÉQUATION INHOMOGÈNE VÉRIFIÉE PAR UNE FONCTION MAHLÉRIENNE DONNÉE

Entrées: Une matrice $B(x)$ correspondant à un système du type (2) dont la première coordonnée correspond à f et la dernière coordonnée correspond à la fonction 1. On suppose aussi que toutes les fonctions coordonnées sont linéairement indépendantes. La base de Mahler b associée.

.....
 /* On suppose ici que f n'est pas rationnelle comme fonction, c'est-à-dire que les fonctions f et 1 sont linéairement indépendantes sur $\mathbf{k}(x)$. Typiquement, la matrice $B(x)$ donnée en entrée est le résultat de l'algorithme 9. On a alors des fonctions $g_1(x) =: f(x), g_2(x), \dots, g_s(x), g_{s+1}(x) := 1$ linéairement indépendantes sur $\mathbf{k}(x)$ qui vérifient

$$\begin{pmatrix} g_1(x) \\ \vdots \\ g_{s+1}(x) \end{pmatrix} := B(x) \begin{pmatrix} g_1(x^b) \\ \vdots \\ g_{s+1}(x^b) \end{pmatrix}. \quad (17)$$

*/

.....
Sortie(s): Des polynômes $p_{-1}(x), \dots, p_{r-1}(x)$ tels que $p_{-1}(x) + p_0(x)f(x) + \dots + p_{r-1}(x)f(x^{b^{r-1}}) = 0$, où $r := \dim(\text{Vect}_{\mathbf{k}(x)}(f_1(x), \dots, f_n(x)))$ a été calculé grâce à l'algorithme 9.

INVERSION DU SYSTÈME (17) :

- 1: Soit $s+1$ le nombre de colonnes de $B(x)$.
- 2: Poser $\lambda_0 := (0, \dots, 0, 1)$, de taille $s+1$.
- 3: **pour** $1 \leq \ell \leq s$:
- 4: Calculer $B_\ell(x) := B(x)B(x^b) \dots B(x^{b^\ell})$.
- 5: Calculer $B_\ell(x)^{-1}$ et λ_ℓ son premier vecteur ligne.
- 6: Construire $C(x) := \begin{pmatrix} \lambda_0 \\ \vdots \\ \lambda_s \end{pmatrix}$.

/* On obtient ainsi le système suivant

$$\begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ f(x^{b^s}) \end{pmatrix} = C(x) \begin{pmatrix} g_1(x) \\ \vdots \\ g_{s+1}(x) \end{pmatrix}. \quad (18)$$

*/

OBTENTION D'UNE ÉQUATION INHOMOGÈNE MINIMALE :

- 7: Calculer le rang ρ de $C(x)$.
 8: Résoudre la système linéaire $\boldsymbol{\mu}(x)C_{[:,\rho]}(x) = (1, 0, \dots, 0)$ avec $\boldsymbol{\mu}(x) = (\mu_1(x), \dots, \mu_\rho(x))$.
 9: Poser $\boldsymbol{\mu}(x) = (\mu_1(x), \dots, \mu_\rho(x), 0, \dots, 0)$ de taille $s + 1$.

/* On a alors

$$\boldsymbol{\mu}(x) \begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ f(x^{b^s}) \end{pmatrix} = f(x) (= g_1(x)).$$

*/

- 10: Calculer $p_0(x)$, le ppcm des dénominateurs des fractions rationnelles des $\mu_i(x)$.
 11: Calculer $p_{-1}(x) := -p_0(x)\mu_1(x)$.
 12: **pour** $1 \leq i \leq r - 1$:
 13: Calculer $p_i(x) := -p_0(x)\mu_{i+1}(x)$.
 14: **renvoyer** $[p_{-1}, \dots, p_{r-1}]$.

Démonstration. (de l'algorithme 10) On suppose ici que f est donnée comme coordonnée d'un vecteur solution de type (2). On note $(f_1(x), \dots, f_n(x))$ la transposée d'un tel vecteur solution. Quitte à effectuer des permutations sur la matrice $A(x)$ et après l'ajout éventuel de la fonction identiquement égale à 1, on peut supposer que $f_1(x) = f(x)$ et que $f_n(x) = 1$. On suppose également que f est transcendante, *i.e.* que 1 et f sont linéairement indépendantes, ceci peut se tester grâce à l'algorithme 9. Toujours d'après cet algorithme, on trouve également une matrice $B(x)$ comme décrite dans le pseudo-code ci-dessus. En reprenant les notations du pseudo-code, on a que les fonctions g_i pour i allant de 1 à s sont linéairement indépendantes sur $\mathbf{k}(x)$, donc en notant ρ le rang de $C(x)$, on a que $\rho = \dim(\text{Vect}_{\mathbb{Q}(x)}(1, f(x^b), \dots, f(x^{b^s})))$. On reprend les notations du lemme 3.16, notons $g(x) = f(x^b)$, on a alors $V_g \subset V_f$. Donc on peut déjà en déduire que $\dim V_f \geq \rho$. Toujours d'après le lemme 3.16, il existe $\rho + 1$ polynômes $q_{-1}(x), q_1(x), \dots, q_\rho(x)$ tels que $q_{-1}(x) + q_1(x)f(x^b) + \dots + q_\rho(x)f(x^{b^\rho}) = 0$. En ne regardant que les puissances de x modulo b , on peut extraire ρ polynômes de sorte à avoir

$$\tilde{b}_{-1}(x) + \tilde{b}_0(x)f(x) + \dots + \tilde{b}_{\rho-1}(x)f(x^{b^{\rho-1}}) = 0. \quad (19)$$

Il suit donc que $\dim V_f \leq \rho$ et enfin $\dim V_f = \rho$. L'égalité (19) implique l'existence d'un vecteur

$\boldsymbol{\mu}(x) = (\mu_1(x), \dots, \mu_\rho(x), 0, \dots, 0)$ tel que

$$\boldsymbol{\mu}(x) \begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ f(x^{b^s}) \end{pmatrix} = f(x).$$

Comme les fonctions g_i avec $1 \leq i \leq s + 1$ sont linéairement indépendantes, on a aussi que $\boldsymbol{\mu}(x)C(x) = (1, 0, \dots, 0)$. Réciproquement si on trouve un tel vecteur $\boldsymbol{\mu}(x)$ tel que $\boldsymbol{\mu}(x)C_{[\rho+1, \rho+1]}(x) = (1, 0, \dots, 0)$, alors

$$\mu_1(x) + \mu_2(x)f(x^b) + \dots + \mu_r(x)f(x^{b^{\rho-1}}) = f(x).$$

Les μ_i sont *a priori* des fractions rationnelles, donc on multiplie par le ppem des dénominateurs de ces μ_i pour obtenir les polynômes finaux p_i qui conviennent. \square

3.4 Limites et améliorations du modèle

Le théorème 3.13 permet de faire le lien entre l'espace des relations linéaires entre les f_i et le noyau de la matrice \mathcal{S} de l'algorithme 9 (à valeurs dans $\overline{\mathbb{Q}}$). Néanmoins, une des limites importantes de ce résultat réside dans le fait qu'on doit calculer le noyau (à gauche) de la matrice $\mathcal{S}(f, h, M)$ qui possède $M + 1$ colonnes où M en fonction des exemples peut valoir $M \approx 10^{280}$. Donc bien que la procédure soit effective, on ne peut pas l'implémenter en l'état.

Une première solution que j'ai cherchée a été de baisser la borne théorique M . J'ai repris la preuve de Adamczewski et Faverjon (2017) et par des majorations un peu plus fines, on peut montrer que la borne $M = \left\lceil \frac{b^{n(bm+h)}(be-d-1)+\nu-be+1}{b-1} \right\rceil$ avec $e := \left\lfloor \frac{d+h}{b} \right\rfloor + 1$ et $m := \left\lfloor \frac{e}{b-1} \right\rfloor + 1$ est suffisante. Cela étant dit, même si cette nouvelle borne est strictement plus petite, dans les exemples usuels, les valeurs obtenues restent inaccessibles (réduction moyenne de 20% du nombre de colonnes) pour faire tourner l'algorithme présenté plus haut.

Le problème que j'ai eu réside dans le trop grand nombre de colonnes de cette matrice \mathcal{S} , une solution a alors consisté à partir sur un algorithme probabiliste, c'est-à-dire ne calculer qu'un certain nombre de colonnes, tirées au hasard, calculer le rang associé à chaque fois et rajouter des colonnes jusqu'à un nombre limite qu'on se fixe. On perd certes l'assurance d'avoir un résultat prouvé sûr, mais la procédure peut être implémentée.

Pour remplir ces colonnes, nous avons besoin d'accéder aux coefficients du développement en série entière des fonctions mahlériennes étudiées, et en pratique on veut accéder à des coefficients d'ordre très éloigné (du même ordre que M). L'algorithme 4 a une complexité au minimum linéaire par rapport à l'ordre n du coefficient qu'on veut calculer. Cette complexité n'est pas suffisamment basse quand on veut pouvoir remplir des colonnes très à droite dans la matrice \mathcal{S} , idéalement on aimerait une complexité plutôt logarithmique. Après discussion avec Frédéric Chyzak et Philippe Dumas, membre de l'équipe SpecFun également, nous proposons ici un algorithme efficace pour le calcul des coefficients du développement en série entière d'une fonction mahlérienne vérifiant une certaine équation homogène, minimale ou non.

3.4.1 L'algorithme one_term

On commence par définir la notion de section.

Définition 3.17 (*section/opérateur de Cartier*). Soit une fonction f admettant un développement en série de Laurent. À la manière de l'opérateur de Mahler $M : f(x) \mapsto f(x^b)$, on définit

un opérateur de b -section comme suit :

Pour chaque $0 \leq j < b$, on définit

$$S_{b,j}(f(x)) = \sum_n f_{bn+j}x^n. \quad (20)$$

On l'appelle la j -ème b -section de f . Par la suite on omettra de préciser l'indice b .

On a immédiatement les propriétés suivantes.

Propriété 3.18. *Les égalités suivantes $S_0(Mf) = f$ et $\forall 0 < j < b$, $S_j(Mf) = 0$ sont vérifiées.*

Ce qui se réécrit :

$$f(x) = \sum_{j=0}^{b-1} S_j(f(x^b))x^j. \quad (21)$$

Soit une équation homogène d'ordre r à coefficients polynomiaux de degré au plus d et une solution f développable en série entière. Soit $N \geq 0$. On cherche à calculer efficacement le coefficient f_N sans avoir à calculer tous les précédents. On rappelle que ceci est nécessaire pour pouvoir remplir la matrice \mathcal{S} qui apparaît dans l'algorithme 9. La méthode qui suit est inspirée d'une discussion avec Philippe Dumas et utilise des résultats sur les propriétés de base des sections présents dans la section 1.3 de Dumas (1993).

Notons $w_k w_{k-1} \dots w_1 w_0$ le développement en base b de l'entier N . Comme f est solution d'une équation homogène, on peut réécrire l'équation obtenue sous la forme

$$f(x) = \ell_1(x)f(x^b) + \dots + \ell_r(x)f(x^{b^r}). \quad (22)$$

Avec les ℓ_i des fractions rationnelles. Notons $L := (\ell_1 \dots \ell_r)$ et $F(x) := \begin{pmatrix} f(x) \\ \vdots \\ f(x^{b^{r-1}}) \end{pmatrix}$.

Je traduis l'équation 22 sous forme d'un système compagnon, d'où $F(x) = A(x)F(x^b)$ où A est la matrice compagnon dont la première ligne est formée par $L(x)$.

En particulier, l'équation (22) se traduit par $f(x) = LF(x^b)$ ou encore $f(x) = E_1 A(x)F(x^b)$ où E_1 est l'opérateur qui renvoie la valeur de la première coordonnée du vecteur sur lequel on l'applique.

Proposition 3.19. *L'évaluation en $x = 0$ de $S_{w_k} \circ S_{w_{k-1}} \circ \dots \circ S_{w_1} \circ S_{w_0}(f(x))$ est égale à f_N .*

Démonstration. Par définition, on a

$$\begin{aligned} S_{w_0}(f(x)) &= \sum_n f_{bn+w_0}x^n. \\ S_{w_1} \circ S_{w_0}(f(x)) &= \sum_n f_{b(bn+w_1)+w_0}x^n. \\ &\dots \\ S_{w_k} \circ \dots \circ S_{w_1} \circ S_{w_0}(f(x)) &= \sum_n f_{b^{k+1}n+b^k w_k + \dots + bw_1 + w_0}x^n. \end{aligned}$$

Par évaluation en $x = 0$ de l'égalité précédente, on obtient :

$$S_{w_k} \circ S_{w_{k-1}} \circ \dots \circ S_{w_1} \circ S_{w_0}(f(x))(0) = f_{b^k w_k + \dots + bw_1 + w_0} = f_N.$$

□

Je peux réécrire la proposition 3.19 comme suit : $E_1 \circ S_{w_k} \circ \dots \circ S_{w_1} \circ S_{w_0}(F(x)) = f_N$. L'intérêt de la démarche réside dans le fait que $S_{w_0}(F(x)) = S_{w_0}(AF(x^b))$. Or, pour deux fonctions f et g admettant un développement de Laurent et pour $0 \leq j < b$, on a $S_j(fg) = \sum_{s+t \equiv j [b]} x^{\lfloor \frac{s+t}{b} \rfloor} S_s(f)S_t(g)$ (voir Dumas (1993)) et $S_j(Mf) = 0$ pour $j \neq 0$. Donc $S_{w_0}(AF(x^b)) = (S_{w_0}(A)) \cdot F(x)$.

Si on note $\Lambda_0 := S_{w_0}(A)$, on a

$$\begin{aligned} S_{w_1} \circ S_{w_0}(F(x)) &= S_{w_1} \circ \Lambda_0 \cdot F(x) \\ &= S_{w_1} \circ \Lambda_0 \cdot AF(x^b) \\ &= S_{w_1}(\Lambda_0 A) \cdot F(x) \\ &=: \Lambda_1 \cdot F(x) \end{aligned}$$

On continue ce procédé pour tout $0 < i \leq k$, avec donc $\Lambda_i := S_{w_i}(\Lambda_{i-1}A)$. À la fin on obtient

$$f_N = E_1 \circ \Lambda_k(0)F(0).$$

On a alors effectué $\log_b(N)$ étapes avec à chaque étape un calcul de section de fractions rationnelles. Les degrés intervenant aux numérateurs et dénominateurs de nos fractions rationnelles étant bornés par d la complexité théorique attendue, en terme d'opérations arithmétiques, est donc en $\mathcal{O}(\log(N))$. La méthode vue à l'algorithme 4 avait une complexité plus importante car le calcul de f_N nécessitait de calculer certains coefficients précédents et potentiellement de calculer f_{N-1} ou f_{N-2} ce qui correspondrait plutôt à une complexité en $\mathcal{O}(N)$. Notons également que contrairement à l'algorithme 4, le calcul d'un coefficient ne nécessite ici aucun calcul de coefficient d'ordre plus petit. L'implémentation de `one_term` reprend la démarche ainsi décrite, je fournis le code en annexe B.

Remarque 3.20. On a vu que `one_term` avait besoin de calculer des sections de fractions rationnelles, j'en explique ici la démarche générale. C'est l'objet de la sous-fonction `section_rat_fun` présente dans l'annexe B. On veut calculer la section de $A(x)/B(x)$ une fraction rationnelle. Soit $C(x)$ le résultant par rapport à y des polynômes $y^b - x$ et $B(y)$. Par propriété du résultant, on a donc que les zéros de C sont exactement les puissances b -ième des racines de B . En particulier, $B(x) \mid C(x^b) = \text{Res}_y(y^b - x^b, B(y))$. On se ramène alors à calculer la section d'une fraction de la forme $\tilde{A}(x)/C(x^b)$. Ce qu'on appelle *section d'une fraction rationnelle* c'est en fait la section de la série qui développe la fonction rationnelle considérée. Le lemme 3 (section 1.3) de (Dumas, 1993) permet d'affirmer que la j -ième b -section de $\tilde{A}/C(x^b)$ est le quotient de la j -ième b -section de \tilde{A} par $C(x)$, et $C(x)$ peut être vu comme la 0-ième b -section de $C(x^b)$ d'où la justification du code écrit.

Le travail restant consiste à mettre en place un algorithme randomisé, permettant de remplir des colonnes de la matrice \mathcal{S} vue dans l'algorithme 9, prises au hasard, d'en calculer le rang et ce jusqu'à un nombre limite de colonnes qu'on se fixe. On retient une sous-matrice qui a un rang maximal parmi les rangs calculés avec un nombre de colonnes égal au rang. Ce rang sera éventuellement strictement inférieur au rang de la matrice \mathcal{S} . On effectue le calcul du noyau à partir de cette matrice.

Donnons une illustration complète de l'étude des relations de dépendance linéaire pour l'obtention d'une équation mahlérienne minimale.

Exemple 3.21. Soit $(a_n)_{n \geq 0}$ la suite définie par $a_n = 0$ si n a un nombre pair de 2 dans son développement triadique et $a_n = 1$ sinon. On définit $f_1(x) := \sum_n a_n x^n$ et $f_2(x) := \sum_n (1 - a_n) x^n$.

Ces deux fonctions vérifient le système mahlérien suivant :

$$\begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} 1+x & x^2 \\ x^2 & 1+x \end{pmatrix} \begin{pmatrix} f_1(x^3) \\ f_2(x^3) \end{pmatrix}.$$

Avec les notations de l'algorithme 9, la matrice $\mathcal{S}(\mathbf{f}, h, M)$ associée comporte 3.24×10^{284} colonnes. Grâce à la fonction `one_term`, on peut calculer le coefficient d'ordre 10^{100} en 0.09 secondes et remplir les 10^5 premières colonnes de \mathcal{S} en environ 700 secondes. Heuristiquement, on trouve que le rang de \mathcal{S} est de 259. Dans la suite on travaille avec une sous-matrice de rang 259 avec uniquement 259 colonnes et on applique les algorithmes décrits pour trouver une équation inhomogène puis homogène minimales pour f_1 et f_2 , on trouve alors

`liste_fonctions_independantes(S, A) := [[0,1], [0,2], [1,2]].`

L'argument A correspond à la matrice du système mahlérien étudié et S à une sous-matrice de S dont le rang est maximal parmi les sous-matrice qu'on a pu construire aléatoirement. Ici on a introduit la solution égale à 1 dans le système en deuxième place (f_1 en première et f_2 en troisième) et les sorties sont tous les 2-uplets d'indices correspondant aux fonctions linéairement indépendantes. 0 pour la fonction entrée en première coordonnée du système, 1 pour la deuxième, etc.

`systeme_fcts_indep(S, A, b=3) :=`
`[-x^2 + x + 1 -x^2/(x^3 - 1)]`
`[0 1].`

Le résultat est une matrice qui représente le système mahlérien vérifié par les fonctions f_1 et 1 données dans cet ordre.

`equa_inhom_min(matrice, b=3) := [x^2, x^3 - 1, x^5 - x^4 - x^3 - x^2 + x + 1].`

Ici la matrice donnée en entrée est le résultat de la fonction précédente. Le résultat est la liste des polynômes intervenant dans une équation inhomogène minimale vérifiée par f_1 .

`inhom_to_hom(liste, b=3) := [x^4, - x^5 - x^4 - x^3 - 1, - x^12 + x^6 + 2x^3 + 1].`

On a ici renvoyé la liste des polynômes intervenant dans une équation minimale, homogène cette fois, vérifiée par f_1 , à partir de la liste des polynômes de l'équation inhomogène minimale, calculée juste au dessus. En d'autres termes, on a maintenant

$$x^4 f_1(x) - (x^5 + x^4 + x^3 + 1) f_1(x^3) - (x^{12} - x^6 - 2x^3 - 1) f_1(x^9) = 0.$$

Remarque 3.22. Pour calculer une sous-matrice de la matrice \mathcal{S} qui est inaccessible, on utilise la fonction `calcul_rang_alea(b, fonction, A, nbre_max, last_column, p)`, b est la base de Mahler, `fonction` est la fonction qui nous permet de calculer n'importe quel f_i de la matrice \mathcal{S} , à l'aide de la fonction `one_term`. Ensuite, A est la matrice associée au système mahlérien étudié, `nbre_max` est le nombre maximal de colonnes qu'on voudra remplir, `last_column` est l'indice de la colonne la plus à droite de \mathcal{S} qu'on voudra calculer (car les colonnes les plus à droites sont les plus longues à remplir) et p est un nombre premier, on remplit les matrices mod p .

Je précise aussi que `one_term` a besoin d'une équation homogène vérifiée par la fonction dont on veut calculer les coefficients de son développement en série. À partir d'un système mahlérien on peut toujours en trouver une, la difficulté est d'en avoir une prouvée minimale.

4 Test de transcendance sur les fonctions mahlériennes

Dans cette section, plus courte, l'objectif est de rassembler les résultats prouvés précédemment pour présenter un test de transcendance sur les évaluations algébriques d'une fonction mahlérienne. La procédure développée ici reprend les idées théoriques de l'article ([Adamczewski et Faverjon, 2018](#)), et par souci d'efficacité j'ai remplacé certains calculs lourds sur des représentations polynomiales exactes de nombres algébriques par du calcul numérique adapté. Je présente ci-après la procédure telle qu'elle est implémentée.

Algorithme 11 TEST DE TRANSCENDANCE DE $f(\alpha)$

Entrées: • Une équation inhomogène **minimale** $[p_{-1}, \dots, p_n]$ vérifiée par la fonction f .

- La base b de Mahler de f .
- α un nombre algébrique vérifiant $0 < |\alpha| < 1$.
- Π_α le polynôme minimal unitaire de α .

Sortie(s): • Soit : " α est un pôle de f ".

- Soit : " $f(\alpha)$ est transcendant".
- Soit : Expression de $f(\alpha)$, correspondant au cas où $f(\alpha)$ est algébrique.

RÉÉCRITURE EN SYSTÈME :

1: Calculer $A(x) := \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -\frac{p_{-1}(x)}{p_0(x)} & \frac{p_1(x)}{p_0(x)} & \dots & \dots & \frac{p_n(x)}{p_0(x)} \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 1 & 0 \end{pmatrix}$.

.....
/* On obtient ainsi le système suivant

$$\begin{pmatrix} 1 \\ f(x) \\ \vdots \\ \vdots \\ f(x^{b^{n-1}}) \end{pmatrix} = A(x) \begin{pmatrix} 1 \\ f(x^b) \\ \vdots \\ \vdots \\ f(x^{b^n}) \end{pmatrix}.$$

*/

CALCULS DE ρ ET DE ℓ :

2: Calculer les modules des pôles non nuls des coefficients de $A(x)$ et des racines non nulles de son déterminant. Soit ρ ce minimum quand il existe.

.....
/* On sait que $A(x)$ est inversible et à coefficients dans $\overline{\mathbb{Q}}(x)$ */

3: Calculer le plus petit entier $\ell \geq 1$ tel que $|\alpha^{b^\ell}| < \rho$. Si à l'étape précédente aucun ρ n'a été calculé, prendre $\ell := 1$.

4: Calculer $A_\ell(x) := A(x) \dots A(x^{b^{\ell-1}})$.

/* On a maintenant le système suivant

$$\begin{pmatrix} 1 \\ f(x) \\ \vdots \\ f(x^{b^{n-1}}) \end{pmatrix} = A_\ell(x) \begin{pmatrix} 1 \\ f(x^{b^\ell}) \\ \vdots \\ f(x^{b^{\ell+n-1}}) \end{pmatrix}.$$

*/

- 5: **pour** chaque dénominateur de la seconde ligne de $A_\ell(x)$:
- 6: Tester s'il est divisible par Π_α .
- 7: **si** oui **alors**
- 8: **renvoyer** " α est un pôle de f ".
- 9: Tester si Π_α divise le déterminant de $A_\ell(x)$ (*i.e.* si α est une racine de ce déterminant).
- 10: **si** non **alors**
- 11: **renvoyer** " $f(\alpha)$ est transcendant".
- 12: Calculer le noyau à gauche de $A_\ell(x)$.
- 13: Tester si un vecteur (de taille $n + 1$) de la forme $(a_1, a_2, 0, \dots, 0)$, avec a_1 et a_2 non tous les deux nuls appartient à ce noyau.
- 14: **si** oui **alors**
- 15: **renvoyer** $-\frac{a_1}{a_2}$.
-
- /* Dans ce cas, $f(\alpha)$ est rationnel et on a $a_1 + a_2 f(\alpha) = 0$. De plus, $-\frac{a_1}{a_2}$ est à valeur dans $\mathbf{k}[x]/(\Pi_\alpha)$ où \mathbf{k} est un corps de nombre contenant α et les coefficients de f qu'on a fixé au départ. */
-
- 16: **renvoyer** " $f(\alpha)$ est transcendant".
-

Remarque 4.1. Notons que l'algorithme trouve son intérêt quand la fonction donnée en entrée n'est pas rationnelle. Ce dernier cas est en effet simple à traiter car pour f rationnelle et α algébrique, on a $f(\alpha)$ algébrique aussi. Par ailleurs, pour savoir si notre fonction est rationnelle ou non, il y a plusieurs solutions, on peut tester si 1 et f sont linéairement indépendantes, l'algorithme 9 le permettra, mais on peut aussi calculer une base des solutions rationnelles pour une équation mahlérienne donnée, c'est l'objet de l'algorithme 8 qui reprend la procédure détaillée dans (Chyzak, Dreyfus, Dumas, et Mezzarobba, 2016).

Pour implémenter cet algorithme, j'ai fait le choix de rechercher sur $\overline{\mathbb{Q}}$ les pôles non nuls des coefficients de la matrice $A(x)$ et des racines non nulles de son déterminant (éventuellement du numérateur du déterminant quand ce dernier est une fraction rationnelle sous forme irréductible) et d'en calculer les modules à chaque fois. Pour la suite des opérations, afin de ne plus travailler avec des valeurs approchées, on introduit le polynôme minimal de α , demandé en entrée du programme. Je ramène ainsi la question de savoir si α est racine d'un polynôme à un test de divisibilité entre ce polynôme et le polynôme minimal unitaire de α .

Remarque 4.2. Quand on ne connaît pas une écriture explicite du nombre algébrique α demandé en entrée de l'algorithme 11 on peut se contenter à la place, d'entrer une valeur approchée qui majore strictement le module de α , la terminaison de l'algorithme reste correcte.

Exemple 4.3. On reprend l'exemple 3.21. Dans (Adamczewski et Faverjon, 2017), on a la proposition suivante.

Proposition 4.4. *Les seuls nombres algébriques α de module compris entre 0 et 1 strictement tels que $f_1(\alpha)$ soit algébrique sont les $\phi^{1/3^l}$ avec $l \geq 0$ et où $\phi := \frac{1-\sqrt{5}}{2}$.*

Vérifions ce que nous donne notre algorithme.

```
liste_polynomes := [x^2, x^3 - 1, x^5 - x^4 - x^3 - x^2 + x + 1].
```

```
phi := QQbar((1-sqrt(5))/2).
```

```
phi.abs() := 0.618033988749895.
```

```
P_min := x^2 - x - 1.
```

```
test_transcendance(P_min, alpha = phi, b=3, liste_polynomes) :=  $-\frac{1}{2}\alpha$ .
```

```
test_transcendance(P_min, alpha = 0.7, b=3, liste_polynomes) :=  $-\frac{1}{2}\alpha$ .
```

```
test_transcendance(x - 1/2, alpha = 1/2, b=3, liste_polynomes) :=  
"f(alpha) est transcendant".
```

```
test_transcendance(P_min(x^3), alpha = phi^(1/3), b=3, liste_polynomes) :=  
 $-\frac{3}{2}\alpha^5 + \frac{1}{2}\alpha^4 + \frac{1}{2}\alpha^3$ .
```

L'algorithme retrouve bien, et précise, les résultats attendus théoriquement.

Conclusion

Ce rapport propose une implantation d'une version randomisée de l'approche théorique instaurée par Boris Adamczewski et Colin Faverjon sur la transcendance des valeurs prises par une fonction mahlérienne. Pour cela j'ai réuni un nouvel algorithme efficace pour la recherche des coefficients des solutions séries d'une équation mahlérienne, l'algorithme `one_term`, et un algorithme de recherche d'une base de solutions rationnelles pour une équation mahlérienne donnée. La randomisation intervient au moment de déterminer le rang de la matrice \mathcal{S} qui a été présentée et une erreur sur le rang est rapidement détectable en testant sur les premiers termes l'équation inhomogène renvoyée par la procédure. Sur le plan théorique, des réductions de la borne du nombre de colonnes de la matrice \mathcal{S} sont possibles, mais aucune pour l'heure n'a permis une implantation efficace et exacte de la méthode de Mahler.

Références

Boris ADAMCZEWSKI et Colin FAVERJON : Méthode de Mahler : relations linéaires, transcendance et applications aux nombres automatiques. *Proc. Lond. Math. Soc.*, 3(115):55–90, 2017.

Boris ADAMCZEWSKI et Colin FAVERJON : Méthode de Mahler, transcendance et relations linéaires : aspects effectifs. *Journal de Théorie des Nombres de Bordeaux*, 30(2):557–573, 2018.

Frédéric CHYZAK, Thomas DREYFUS, Philippe DUMAS et Marc MEZZAROBBA : Computing solutions of linear mahler equations. *CoRR*, abs/1612.05518, 2016. URL <http://arxiv.org/abs/1612.05518>.

Philippe DUMAS : *Réurrences mahlériennes, suites automatiques, études asymptotiques. (Mahlerian recurrences, automatic sequences, asymptotic studies)*. Thèse de doctorat, University of Bordeaux, France, 1993. URL <https://tel.archives-ouvertes.fr/tel-00614660>.

A Démonstrations de la sous-section 2.3 (Recherche de solutions rationnelles)

Démonstration de 2.30 : Sans perte de généralité on peut supposer $i = 1$, quitte à remplacer b par b^i .

Pour 1) on a $Mp(x) = p(x^b)$, puis à l'aide de la Remarque 2.28, on a

$$GMp(x) = p(x)p(\zeta^b x) \dots p(\zeta^{b(b-1)} x) = p^b.$$

Ensuite 2) découle de 1). Pour 3), le sens direct est clair. Pour l'autre, on effectue la division euclidienne $q = up + v$. Si $Mq = sMp$ pour un certain polynôme s , alors $Mq = (Mu)(Mp) + (Mv) = sMp$. Or $\deg v < \deg p$ donc $\deg Mv < \deg Mp$. Ainsi $Mv = 0$ puis $v = 0$.

Démonstration de ?? : Considérons la décomposition en irréductibles $G^i p = cq_1^{e_1} \dots q_s^{e_s}$ où $c \in \mathbb{K}^*$ et les q_i sont unitaires et deux à deux premiers entre eux. D'après le Lemme 2.30(c), les polynômes $M^i q_1^{e_1} \dots M^i q_s^{e_s}$ sont également deux à deux premiers entre eux. Nous avons

$$M^i G^i p = c(M^i q_1^{e_1}) \dots (M^i q_s^{e_s}),$$

Comme p est unitaire irréductible, d'après le Lemme 2.30(b), $p \mid M^i q_j^{e_j}$ pour un certain j . Ainsi $G^i p \mid G^i M^i q_j^{e_j} = q_j^{e_j b^i}$ d'après le Lemme 2.30(a). Ceci démontre la première assertion.

Si $G^i p = p$, alors d'après le Lemme 2.30(b), $p \mid M^i p$. Réciproquement, si $p \mid M^i p$, alors d'après le Lemme 2.30(a), $G^i p \mid p^{b^i}$ et donc $\exists e \in \mathbb{N}$ tel que $G^i p = p^e$. Par ailleurs, la Remarque 2.29 et le fait que p est unitaire irréductible nous assurent que $e = 1$.

Enfin, supposons $G^i p = p$ pour un $i > 0$. Soit $j \in \mathbb{N}$, choisissons m tel que $m \times i \geq j$. Dans ces conditions, $p = G^{mi} p = G^{mi-j} (G^j p)$ est unitaire irréductible et donc $G^j p$ l'est aussi.

Démonstration de 2.32 : Comme $f \wedge f' = 1$, on peut trouver une relation de Bézout $uf + vf' = 1$ avec $u, v \in \mathbb{K}[x]$. On applique alors M à cette identité et on obtient aussi que $Mf \wedge M(f') = 1$ à l'aide encore d'une relation de Bézout. Maintenant, $(Mf)' = bx^{b-1} M(f')$. Si s est un diviseur commun de (Mf) et de (Mf') , alors d'après ce qui précède s est premier avec $M(f')$ et donc nécessairement, $s \mid bx^{b-1}$. Mais par hypothèse $x \nmid f$ donc $x \nmid Mf$ et alors s est constant.

Démonstration de 2.33 : Raisonnons par contraposée. On suppose $\exists i > j \geq 0$ tels que $M^i p \wedge M^j p \neq 1$. On pose $k = i - j \geq 1$. D'après le Lemme 2.30(c), $M^k p$ et p ne sont pas premiers entre eux. Comme p est unitaire irréductible, $p \mid M^k p$. Alors d'après le Lemme 2.31(b) $G^k p = p$.

Démonstration de 2.34 : Soit p un facteur unitaire irréductible de f . On s'intéresse à Mp qu'on écrit sous sa décompositions en irréductibles $Mp = q_1 \dots q_s$. Raisonnons par l'absurde et supposons que pour tout $0 \leq i \leq s$, il existe $k_i \in \mathbb{N}^*$ tel que $G^{k_i} q_i = q_i$. Posons $k := k_1 \dots k_s$, il suit que $G^k q_i = q_i$ pour tout i . D'après le Lemme 2.30(a), on a alors $p^b = GMp = (Gq_1) \dots (Gq_s)$ et d'après le Lemme 2.31(c), on a que pour tout i , Gq_i est unitaire irréductible. En combinant ces deux résultats, on montre que pour tout i , $\exists e_i \in \mathbb{N}^*$ tel que $Gq_i = p^{e_i}$ (*) et b vérifie $b = e_1 + \dots + e_s$. Par conséquent, en multipliant (*) par G^{k-1} , on trouve que pour tout i , $q_i = G^{k-1} p^{e_i}$. Comme q_i est irréductible, p^{e_i} doit aussi l'être ce qui impose $e_i = 1$. On vient donc de montrer que $\forall i q_i = G^{k-1} p := \tilde{q}$. Il vient que $Mp = \tilde{q}^b$. Comme p est irréductible,

$p \wedge p' = 1$ et par ailleurs $p(0) \neq 0$ car $p \mid f$, le Lemme 2.32 assure donc que $(Mp) \wedge (Mp)' = 1$, ce qui n'est manifestement pas le cas.

Démonstration de 2.35 : Supposons par l'absurde $\exists i \geq 1$ tel que $G^i g = g$. D'après le Lemme 2.31(c), on a alors que $G^j g$ est unitaire irréductible. Par ailleurs, $G^j g \mid G^j M^j f = f^{bj}$, on voit alors que $G^j g = f$. Ainsi, $G^i f = G^{i+j} g = G^j g = f$ ce qui constitue une contradiction.

Démonstration de 2.36 : On s'intéresse à la décomposition en irréductibles de $M^s f^m$. Soit h^k un tel facteur (h unitaire irréductible) avec $k > 0$. Par hypothèse, on a donc $h^k \mid \ell T_q$. Montrons alors que h et T_q sont premiers entre eux. Supposons que ce ne soit pas le cas, cela signifie, comme h est irréductible, que $\exists 0 \leq i < r$ tel que $h \mid M^i q$. Par conséquent, on a $G^i h \mid G^i M^i q$ et $G^i h \mid G^i M^s f$ ce dernier point car $h \mid M^s f$ car $h \mid M^s f^m$ et h, f sont unitaires irréductibles. Comme $G^i M^i q$ et $G^i M^s f$ sont respectivement des puissances de q et de $M^s s - if$ d'après le Lemme 2.30(a), on obtient une contradiction car alors q et $M^{s-i} f$ ne seraient plus premiers entre eux. Par lemme de Gauß, on conclut que $h^k \mid \ell$. Comme h était un diviseur irréductible quelconque de $M^s f^m$, on obtient finalement que $M^s f^m \mid \ell$.

Démonstration de 2.37 : Nous allons procéder par disjonction de cas.

Premier cas : $\exists f \in \mathbb{K}[x]$ unitaire irréductible tel que $f \mid b$ et $\forall i > 0, G^i f \neq f$.

Montrons qu'on peut sans perte de généralité se ramener au cas où $M^j f \wedge q = 1, \forall j > 0$. Si ce n'est pas le cas, leur pgcd est alors non trivial pour au moins un certain j . D'après le Lemme 2.33 on a que les $M^j f$ sont deux à deux premiers entre eux pour tout entier j . Comme par ailleurs on a un nombre fini de diviseurs de q , l'identité $M^j f \wedge q \neq 1$ ne peut être vérifiée que pour un nombre fini de j . On notera désormais j le maximum de ces entiers. Considérons également g un facteur unitaire irréductible de $M^j f \wedge q$. On applique alors le Lemme 2.35 qui nous précise que $G^i g \neq g, \forall i > 0$. On peut donc remplacer f par g comme on a maintenant $M^j g \wedge q = 1$ car on avait choisi j maximal et $g \mid M^j f \xrightarrow{2.30(c)} M^j g \mid M^{2j} f \implies (M^j g \wedge q) \mid (M^{2j} f \wedge q) = 1$.

Second cas : On suppose maintenant que $q = c \prod_k f_k^{m_k}$ décomposition en irréductibles, avec c constante non nulle et de sorte que pour tout k , il existe un indice $i_k > 0$ tel que l'on ait $G^{i_k} f_k = f_k$. Fixons un entier k . Le Lemme 2.34 nous fournit un polynôme unitaire irréductible $\tilde{f}_k \in \mathbb{K}[x]$ diviseur de $M f_k$ tel que $G^i \tilde{f}_k \neq \tilde{f}_k$ pour tout $i > 0$. Maintenant, on s'intéresse à $M^i \tilde{f}_k \wedge q$. Si ce pgcd est différent de 1 pour un certain i alors il existe un diviseur unitaire irréductible g de ce dernier qui est nécessairement l'un des $f_{k'}$ d'après l'écriture de q . Mais dans ce cas, le Lemme 2.35 nous montre alors qu'on aurait $G^i g \neq g$ pour tout $i > 0$ ce qui est contradictoire car on a vu qu'il existait un indice $i_{k'}$ tel que cette égalité est vérifiée justement. Donc $M^i \tilde{f}_k \wedge q = 1 \forall i \in \mathbb{N}$.

Posons désormais $s = r - 1, m = m_k, g = \tilde{f}_k$. On a $M^s g^m = M^{r-1} \tilde{f}_k^{m_k} \mid M^r f_k^{m_k} \mid M^r q$ et de plus $M^{s-i} g = M^{r-1-i} \tilde{f}_k$ et est premier avec q pour tout $0 \leq i < r$ comme on vient de le montrer. Par conséquent, le Lemme 2.36 prouve que $M^{r-1} \tilde{f}_k^{m_k} = M^s g^m$ divise ℓ . On a également $Gg = G\tilde{f}_k \mid GMf_k = f_k^b$ d'où Gg est une puissance de f_k . Ainsi $f_k \mid Gg = G\tilde{f}_k$ puis $f_k^{m_k} \mid G\tilde{f}_k^{m_k}$. En itérant ce raisonnement pour tous les k , comme les f_k sont deux à deux premiers entre eux, les \tilde{f}_k le sont aussi d'après le Lemme 2.30(c). Donc tous les $M^{r-1} \tilde{f}_k^{m_k}$ divisent ℓ et sont deux à deux premiers entre eux. Finalement en posant $u = \prod_k \tilde{f}_k^{m_k}$ on a que $M^{r-1} u \mid \ell$ et que $q \mid c.Gu$.

Démonstration de 2.38 : Le sens réciproque est évident. Pour le sens direct, fixons $i < b$ et supposons que $Mu \mid \ell$. Soit ω une racine primitive b -ème de l'unité. En particulier, pour tout entier $j, Mu(\omega^j x) \mid \ell(\omega^j x)$. Il suit alors que, pour tout $0 \leq i \leq b - 1,$

Mu divise

$$\sum_{j=0}^{b-1} \omega^{-ij} \ell(\omega^j x) = bx^i Mf_i.$$

Par ailleurs, $Mu \in \mathbb{K}[x^b]$ et $i < b$ donc nécessairement cela implique que Mu divise Mf_i et d'après le lemme 2.30(c) on a que pour tout $0 \leq i \leq b-1$, u divise f_i .

B L'algorithme one_term

```

@cached_function
def QxM(Qx):
    return PolynomialRing(Frac(Qx), 'M')

def section_i_M(poly, b, i):
    """ Calcul de la ième b-section de poly. """

    Qx = poly.parent()
    return Qx(list(poly)[i::b])

def expression_y_M(liste_polynomes):
    """ liste_polynomes représente une équation homogène donnée
    sous forme d'une liste qui commence par le terme P_0.
    Soit y(x) une solution. On veut exprimer y(x)
    en fonction de y(x^b), y(x^(b^2)), etc. On renvoie
    la nouvelle équation obtenue sous forme de liste. """

    p_0 = liste_polynomes[0]
    return [- p/p_0 for p in liste_polynomes[1:]]

@cached_function
def rat_section_aux(b):
    (x, y) = PolynomialRing(QQ, ['x', 'y']).gens()
    return y, x**b - y**b

def section_rat_fun_M(b, frac, j):
    """ L'objectif est de calculer la j-ème b-section
    d'une fraction rationnelle. """

    num = frac.numerator()
    denom = frac.denominator()
    y, aux = rat_section_aux(b)
    Qx = num.parent()
    R = aux.resultant(denom(y), y).univariate_polynomial(Qx)
    Q = R//denom
    a_1 = section_i_M(num*Q, b, j)
    a_2 = section_i_M(R, b, 0)
    return a_1/a_2

def expression_poly_y_M(liste_polynomes):
    """ Dans expression_y_M, on demandait l'équation
    vérifiée par y(x), une solution de l'équation
    homogène représentée par liste_polynomes, cette fois
    on demande à l'écrire sous forme d'un polynome en la
    variable M, M représentant l'opérateur de Mahler x |-> x^b. """

```

```
Qx = liste_polynomes[0].parent()
Qx_M = QxM(Qx)
L = [Qx.zero()] + expression_y_M(liste_polynomes)
return Qx_M(L)

def section_comb_poly_M(b, liste_polynomes, expression_poly_M, j):
    """ On a un polynôme en M donné par expression_poly_M.
    On veut renvoyer un nouveau polynôme en M, dont le
    coefficient devant  $M^l$  est la  $j$ -ème section du
    coefficient devant  $M^{l+1}$  du polynôme
    expression_poly_M. """

    Qx = liste_polynomes[0].parent()
    Qx_M = QxM(Qx)
    P = expression_poly_M
    return Qx_M([section_rat_fun_M(b, P[k], j) for k
    in range(1, len(liste_polynomes))])

def push_M(liste_polynomes, expr):
    """ On a un polynôme expr en la variable M.
    On rappelle qu'avec ce choix de notations, on a M qui
    représente  $y(x^b)$ ,  $M^2$  qui représente  $y(x^{b^2})$ , etc.
    Le terme constant de expr représente donc le
    coefficient devant  $y(x)$  dans cette équation.
    On veut remplacer ce  $y(x)$  par
    expression_poly_y_M(liste_polynomes). """

    temp = list(expr)
    temp[0] = expr.base_ring().zero()
    temp = expr.parent()(temp)
    return temp + expr[0]*(expression_poly_y_M(liste_polynomes))

def iteration_M(b, liste_polynomes, n, expr):
    """ Rappel: expr est un polynôme en M, où M
    représente l'opérateur de Mahler. """
    while True:
        expr = section_comb_poly_M(b, liste_polynomes,
            push_M(liste_polynomes, expr), n%b)
        n = n//b
        if n == 0:
            return expr

def one_term_M(b, liste_polynomes, n, liste_poly_ini):
    Qx = liste_polynomes[0].parent()
    x = Qx.gen()
    Qx_M = QxM(Qx)
    expression = iteration_M(b, liste_polynomes, n, Qx_M(1))
    deg = expression.degree()
```

```
POLY_x = sum(expression[i]*sum(liste_poly_ini[j]
(x^(b^i)) for j in range(len(liste_poly_ini)))
for i in [0..deg])
return POLY_x(0)
```

```
def donnees_ini(liste_ini, liste_valuations_possibles):
return [sum(liste_ini[i]*x^i
for i in [liste_valuations_possibles[j]..
liste_valuations_possibles[j+1] - 1])
for j in range(len(liste_valuations_possibles)-1)]
+[liste_ini[-1]*x^(liste_valuations_possibles[-1])]
```