FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

ALGORITHMS FOR FACTORING LINEAR RECURRENCE OPERATORS

By

YI ZHOU

A Dissertation submitted to the
Department of Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2022

Yi Zhou defended this dissertation on April 8, 2022.
The members of the supervisory committee were:

Mark van Hoeij
Professor Directing Dissertation

Philip Sura
University Representative

Ettore Aldrovandi
Committee Member

Paolo Aluffi
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

ACKNOWLEDGMENTS

Foremost, I am extremely grateful to Prof. van Hoeij for his truly tireless effort and guidance throughout this project. I greatly benefited from his knowledge and insights.

I deeply appreciate all professors who taught me at FSU, especially Dr. Aluffi and Dr. Aldrovandi for their helpful feedback on the dissertation. Special thanks to Dr. Kirby for her professional advice on teaching and Dr. Sura for serving in my committee.

Thanks to the friends I made at FSU during my PhD for a lot of fun times and beneficial discussions.

Finally, I wish to thank my grandparents, my parents, my cousin and my wife for their support and encouragement.

TABLE OF CONTENTS

ABSTRACT

In this thesis we develop a few algorithms that are useful for factoring linear recurrence operators. We approach the factorization problem from three directions.

First, considering reduction modulo a prime leads to the study of recurrence operators in positive characteristic in Chapter 3, where we prove an unexpected relation between the singularities and the p-curvature. The result suggests desingularization accelerates the algorithm for computing the characteristic polynomial of the p-curvature. Useful information about factorization of a recurrence operator can be obtained by factoring the characteristic polynomial of its p-curvature. In particular, the irreducibility of the characteristic polynomial implies that of the operator.

The second strategy is to imitate the Beke-Bronstein algorithm for factoring differential operators. This requires solving recurrence systems for hypergeometric solutions, which can be done by a generalization of Petkovšek's algorithm. We work out the details in Chapter 4 and add several improvements, such as minimizing the number of candidates that the algorithm needs to consider.

Third, we consider solutions of recurrence operators. Some solutions of an operator are also solutions of a proper factor but most are not, even when the operator is reducible. We present an heuristic algorithm for constructing certain sequence solutions that likely lead to non-trivial factors in Chapter 5. This method raises the question of how to find the operator with the minimal order that a solution satisfies, which is solved in Chapter 6 by bounding degrees of factors.

CHAPTER 1

INTRODUCTION

1.1 Linear Recurrence Operators

A *homogeneous linear recurrence equation*, or *recurrence equation* for short, is an equation about a function $f(x)$ in the form

$$a_n(x)f(x+n) + a_{n-1}(x)f(x+n-1) + \cdots + a_0(x)f(x) = 0. \tag{1.1}$$

Many common functions that naturally arise in combinatorics and physics are solutions of linear recurrence equations. Among those functions, sequences are a strong motivation to study recurrence equations.

Example 1 (Fibonacci sequence). The well-known Fibonacci sequence is defined by the recurrence equation

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

and the initial values

$$f(0) = 0, \quad f(1) = 1.$$

▲

A number of sequences that satisfy a linear recurrence equation can be found at the On-Line Encyclopedia of Integer Sequences (abbr. OEIS, [14]).

Example 2 (OEIS A002777, [11]). The entry A002777 on OEIS satisfies the following relation when $x \geqslant 7$:

$$\begin{aligned}(3x^2 - 17x + 23)a(x) - (3x^2 - 17x + 21)a(x-1) + (3x^4 - 23x^3 + 63x^2 - 74x + 34)a(x-2)\\ - 4(x-3)(x-2)a(x-3) + 2(x-4)(x-3)(3x^2 - 11x + 9)a(x-4) = 0.\end{aligned}$$

▲

Introduce the *shift operator* τ, which acts on a function by

$$\tau(f(x)) = f(x+1).$$

Then (1.1) can be written as

$$L(f) = 0,$$

where $L = \sum_{i=0}^{n} a_i \tau^i$. Call L a *linear recurrence operator*, or *recurrence operator* for short. The collection of recurrence operators with coefficients from a field $\mathbb{K}$ is denoted by $\mathbb{K}[\tau]$. Equipped with the addition and multiplication (composition) of operators, $\mathbb{K}[\tau]$ is a ring. In particular, the multiplication (composition) is ruled by

$$\tau \cdot a(x) = a(x+1)\tau,$$

and hence is non-commutative. For an operator $L = \sum_{i=m}^{n} a_i \tau^i \in \mathbb{K}[\tau]$ where $a_m a_n \neq 0$, denote $\operatorname{ord}(L) = n - m$ and call it the *order* of L. It is easy to verify that $\operatorname{ord}(L_1 L_2) = \operatorname{ord}(L_1) \operatorname{ord}(L_2)$. An operator $L_2 \in \mathbb{K}[\tau]$ is called a *right-hand factor* of L if $L = L_1 L_2$ for some $L_1 \in \mathbb{K}[\tau]$. We say the factor L_2 is *non-trivial* if $0 < \operatorname{ord}(L_2) < \operatorname{ord}(L)$.

1.2 The Factorization Problem

The objective of this thesis is to develop algorithms that are useful for factoring recurrence operators, or in other words, computing non-trivial right-hand factors of given operators if they exist or proving the irreducibility otherwise. Factoring recurrence operators is helpful for solving recurrence equations and finding the minimal recurrence operator that a sequence satisfies, if we know it is a solution of some recurrence operator a priori.

The approach for factoring linear differential operators given by Beke ([10]) and Bronstein ([5]) applies to linear recurrence operators as well, with which the problem of computing order-m right-hand factors of an order-n recurrence operator converts to finding *hypergeometric solutions* of a recurrence system of rank $\binom{n}{m}$. Together with Petkovšek's algorithm ([17]) for computing hypergeometric solutions, we then have a complete algorithm for factoring recurrence operators. However, solving a system of rank $\binom{n}{m}$ for hypergeometric solutions is sometimes beyond the capacity of a normal computer even when n and m are small, since $\binom{n}{m}$ can be quite large. Plus Petkovšek's algorithm was originally designed for operators; to apply it on systems, some extra work has to be done to convert a system into an operator. These motivate us to find efficient algorithms for factoring recurrence operators and solving recurrence systems for hypergeometric solutions.

CHAPTER 2

PRELIMINARIES

2.1 Difference Rings

Definition 3. *A (commutative and unital) ring R equipped with an automorphism $\tau : R \to R$ is called a* difference ring*. In this case we also say (R, τ) is a difference ring. If R is a field then call it a* difference field*.*

Definition 4. *Suppose (R, τ) is a difference ring. A τ-*constant *is an element $c \in R$ such that $\tau(c) = c$. The subset of constants is denoted by R^τ.*

Lemma 5. *If $(\mathbb{K}, \tau)$ is a difference field, then $\mathbb{K}^\tau$ is a subfield of $\mathbb{K}$.*

Proof. It is clear that τ-constants are closed under addition, multiplication and inverse. □

Definition 6. *Suppose $(R_1, \tau_1), (R_2, \tau_2)$ are difference rings. Say R_2 is a* difference extension *of R_1 if there is an embedding $\phi : R_1 \to R_2$ such that $\tau_2 \circ \phi = \phi \circ \tau_1$. In other words, R_1 is a subring of R_2 and the restriction of τ_2 on R_1 is the same automorphism as τ_1.*

Remark 7. If (R, τ) is a difference ring and R is an integral domain, then τ can be extended to an automorphism of $\mathrm{Frac}(R)$, the fraction field of R, in a natural and unique way, which makes $\mathrm{Frac}(R)$ a difference extension of R.

In this thesis we deal with the *recurrence* case exclusively, where the difference ring is a field of functions in x equipped with the *shift operator* $\tau : f(x) \mapsto f(x + 1)$. Particularly, polynomials and rational functions are of our major interest.

Let F be a field. It is easy to see the shift operator τ is indeed an automorphism of $F[x]$. Thus $(F[x], \tau)$ is a difference ring and, according to Remark 7, $(F(x), \tau)$ is a difference field. When $\mathrm{char}(F) = 0$, the field of τ-constants for $F(x)$ is F; when $\mathrm{char}(F) = p$ for some prime p, the field of τ-constants is $F(x^p - x)$.

In order to study $(F(x), \tau)$, sometimes it is helpful to embed $F(x)$ into $F((x^{-1}))$. The action of τ is defined by

$$\tau(x^{-1}) = (1 + x)^{-1} = x^{-1}\frac{x}{1 + x} = x^{-1}\frac{1}{1 + x^{-1}} = x^{-1}\sum_{i=0}^{\infty}(-x^{-1})^i. \qquad (2.1)$$

By the definition $F((x^{-1}))$ is a difference extension of $F(x)$. If $\mathrm{char}(F) = 0$, the field of τ-constants of $F((x^{-1}))$ is F.

2.2 Difference Operators

Definition 8. *For a difference ring (R, τ), a* (linear) difference operator *over R is an expression*

$$\sum_{i=0}^{n} a_i \tau^i, \quad n \in \mathbb{N}, a_i \in R.$$

The addition of difference operators is the same as that of polynomials, namely

$$\sum_{i=0}^{n} a_i \tau^i + \sum_{i=0}^{n} b_i \tau^i = \sum_{i=0}^{n} (a_i + b_i) \tau^i.$$

The multiplication is ruled by

$$\tau \cdot a = \tau(a)\tau, \quad a \in R. \tag{2.2}$$

In the shift case, also call a difference operator a recurrence operator*.*

Example 9. In this example we demonstrate how difference operators multiply. Suppose $L_1, L_2 \in \mathbb{Q}(x)[\tau]$ where $L_1 = \tau - 1, L_2 = x\tau - 1$. Then

$$L_1 L_2 = (\tau - 1)(x\tau - 1) = \tau \cdot (x\tau) - \tau \cdot (-1) - x\tau + 1 = (x+1)\tau^2 + \tau - x\tau + 1 = (x+1)\tau^2 + (1-x)\tau + 1.$$

▲

The ring of difference operators over (R, τ) is denoted by $R[\tau]$. It naturally acts on R by

$$(\sum_{i=0}^{n} a_i \tau^i)(r) = \sum_{i=0}^{n} a_i \tau^i(r).$$

The action is compatible with addition and multiplication of difference operators. In fact any difference extension of R is a natural left $R[\tau]$-module.

It also makes sense to allow negative powers of τ, since automorphisms have inverses. The ring of such difference operators is denoted by $R[\tau, \tau^{-1}]$. Usually it suffices to study operators in $R[\tau]$ only. In fact, any non-zero difference operator in $R[\tau, \tau^{-1}]$ can be reduced into the so-called *normal* form (defined below) by left multiplying by a unique power of τ.

Definition 10 (Normal)**.** *The operator $\sum_{i=0}^{n} a_i \tau^i \in R[\tau]$ is called* normal *if $a_0 \neq 0$. Also assume the 0 operator is normal.*

Definition 11 (Order)**.** *Suppose $L = \sum_{i=m}^{n} a_i \tau^i \in K[\tau]$ where $a_m a_n \neq 0$. Call $n-m$ the* order *of L and n the* τ-degree*. Denote them by* $\operatorname{ord}(L)$ *and* $\deg_\tau(L)$*, respectively. Assume* $\operatorname{ord}(0) = \deg_\tau(0) = -\infty$*. In the shift case $K = F(x)$, if $L \in F[x][\tau]$, denote* $\deg_x(L) = \max\{a_i : i = 0, 1, \ldots, n\}$ *and call it the* x-degree *of L.*

An operator is normal if and only if its order and τ-degree are equal.

Definition 12 (Coefficients)**.** *For $L = \sum_{i=0}^{\infty} a_i \tau^i \in K[\tau]$ where $a_i \neq 0$ for finitely many i, call a_i a* τ-coefficient *or simply* coefficient *of L. If $L = \sum_{i=m}^{n} a_i \tau^i \in K[\tau]$ where $a_n a_m \neq 0$, call a_n the* leading coefficient *and a_m the* trailing coefficient *of L, denoted by* $\operatorname{lc}(L), \operatorname{tc}(L)$*, respectively.*

Definition 13 (Content and primitive part)**.** *Suppose (A, τ) is a difference ring where A is a UFD. Let K be the fraction field of A. Suppose $L \in K[\tau]$. If the coefficients of L are in A and their* gcd *is 1, L is called* primitive *over A or simply* primitive *when it is clear from the context what A is. Assume the zero operator is primitive. If $k^{-1}L$ is primitive for some $k \in K$, k is called a* content *of L and kL a* primitive part.

Remark 14. Content and primitive part of an operator are unique up to a unit in A. In the case $A = F[x], K = F(x)$, the contents of $L \in A[\tau]$ are polynomials. Denote by $\operatorname{Cont}(L)$ the monic content and $\operatorname{Prim}(L) = \operatorname{Cont}(L)^{-1} L$.

Theorem 15 (Right-division with remainder)**.** *Suppose $L, R \in K[\tau] - \{0\}$. There exists a unique pair $q, r \in K[\tau]$ such that*

$$L = qR + r$$

and $\deg_\tau(r) < \deg_\tau(R)$*.*

Definition 16 (GCRD and LCLM)**.** *Suppose $L_1, L_2 \in K[\tau]$. Their* least common left multiple *(abbr. LCLM) is an operator L such that*

- *L is a left multiple of both L_1 and L_2;*
- *L has the minimal order among all operators satisfying the previous condition.*

The greatest common right divisor *(abbr. GCRD) is an operator L such that*

- *L is a right divisor of both L_1 and L_2;*
- *L has the maximal order among all operators satisfying the previous condition.*

LCLM and GCRD are not unique, since multiplying them by an element in K yields another LCLM or GCRD. To make them unique, we introduce the notations $\operatorname{LCLM}(L_1, L_2), \operatorname{GCRD}(L_1, L_2)$ *for monic and normal LCLM and GCRD, respectively.*

2.3 Difference Modules

Definition 17 (Difference modules)**.** *A* difference module *over a difference field F is a left $F[\tau, \tau^{-1}]$ module of finite type. Equivalently, M is a finitely dimensional F-vector space equipped with the action of τ that satisfies*

- $\tau(m_1 + m_2) = \tau(m_1) + \tau(m_2)$ *for* $m_1, m_2 \in M$*;*
- $\tau(fm) = \tau(f)\tau(m)$ *for* $f \in F$ *and* $m \in M$*;*
- *for* $m \in M$ *there exists* $n \in M$ *such that* $\tau(n) = m$*.*

Given an F-basis B of a difference module M in the form of a column vector, then there exists an invertible matrix A such that

$$\tau(B) = AB.$$

The matrix A is called the *representation matrix* of M with respect to the basis B.

Conversely, given an invertible matrix with a suitable order, a finitely dimensional F-vector space with basis B can be made a difference module by assigning

$$\tau(B) = AB.$$

Two matrices are called *gauge equivalent* if modules defined by them are isomorphic.

A normal difference operator $L \in F[\tau]$ defines a difference module $F[\tau]/F[\tau]L$. Suppose $\operatorname{ord}(L) = n$. Then a basis of this module is

$$(1, \tau, \cdots, \tau^{n-1})^\intercal.$$

The representation matrix under such a basis is the *companion matrix* of L.

2.4 Solutions of Difference Operators

Definition 18 (Universal extension)**.** *Suppose (K, τ) is a difference field with the field of τ-constants C. A* universal extension *of (K, τ) is a difference extension (Ω, τ) such that*

1. *for* $L \in K[\tau]$, $\operatorname{ord}(L) = \dim_C \ker(L)$*, where L is viewed as a C-linear map over Ω;*
2. *any element in Ω is a solution of some difference operator.*

If a difference extension (Ω, τ) satisfies 1 but potentially not 2, then it is called a pre-universal extension*.*

Remark 19. If Ω' is a pre-universal extension, then the subspace

$$\Omega = \{w \in \Omega : w \text{ is a solution of some operator in } K[\tau]\}$$

is a universal extension. The solution space of L in Ω is the same as that in Ω'.

The universal extension is unique up to isomorphism if it exists. Any difference field of characteristic 0 has a universal extension ([19, pp. 20–21]).

For an operator L, denote $\mathrm{Sol}(L)$ its solution space in the universal extension, if the universal extension exists.

CHAPTER 3

DESINGULARIZATION AND P-CURVATURE

3.1 Introduction

Singularities of linear difference operators can be divided into two groups, true (i.e. non-removable) singularities, and apparent (i.e. removable) singularities. Desingularization (detecting or removing apparent singularities) can expedite various algorithms for difference or differential equations. An early application [22] appeared in `DEtools[Homomorphisms]` in Maple 10. Other algorithms that benefit from reducing the number of singularities include finding closed form solutions and factoring, e.g. `LREtools[RightFactors]` in Maple 2021.

In characteristic p, difference operators can be classified by the so-called p-curvature. Our main result gives a relation between $\chi(L)$, the characteristic polynomial of the p-curvature of L, and the true singularities of L. We prove that the denominator of $\chi(L)$ determines the true singularities, including their multiplicities, up to shift equivalence.

The algorithm from [3] computes $\chi(L)$, multiplied by a denominator bound, by computing its Z-adic expansion. One application of our theorem is that we can replace the denominator bound by the exact denominator. This lowers the required Z-adic precision, which can speed up the computation, see subsection 3.5.2.

We want desingularization to take less time than the time it saves in applications. Then it is useful to compute a *partial desingularization* (where the goal is to remove most apparant singularities, at a fraction of the cost of a full desingularization). We give various algorithms for this in section 3.6.

3.2 Preliminaries

3.2.1 Desingularization

Let F be a field. Let $P = F[x][y]$ and $D = F(x)[y]$. If $f \in P$, then f is called *primitive* if the gcd of its coefficients in $F[x]$ is 1. If $f \in D - \{0\}$, then there is $c \in F(x) - \{0\}$, unique up to a factor in F, for which $c^{-1}f \in P$ is primitive. The content of f, denoted $\mathrm{Cont}(f)$, is this c, while the *primitive part* of f is $\mathrm{Prim}(f) = c^{-1}f \in P$. A version of Gauss's lemma says $\mathrm{Cont}(f_1 f_2) = \mathrm{Cont}(f_1)\mathrm{Cont}(f_2)$.

Let τ be the shift-operator. If $r(x)$ is a rational function then the product $\tau \cdot r(x)$ equals $r(x+1) \cdot \tau$. This product turns $\mathcal{P} := F[x][\tau]$ and $\mathcal{D} := F(x)[\tau]$ into non-commutative rings. The product corresponds to compositions of operators, where $L = \sum_{i=0}^{n} a_i(x)\tau^i \in \mathcal{D}$ operates on $y(x)$ as $L(y(x)) = \sum_{i=0}^{n} a_i(x)y(x+i)$.

If $L \in \mathcal{D} - \{0\}$ we can define $\mathrm{Cont}(L) \in F(x)$ and $\mathrm{Prim}(L) \in \mathcal{P}$ in the same way as before.

Definition 20. *An operator $L \in \mathcal{P}$ is called* Gaussian *if*

$$\forall_{A \in \mathcal{D}}\ AL \in \mathcal{P} \Longrightarrow A \in \mathcal{P}.$$

If L is Gaussian then L is primitive. In the commutative case, the two properties are equivalent by Gauss's lemma. It is known that Gauss's lemma does not hold in the non-commutative case, which is illustrated in Example 22 below (see also [15, p. 27]).

An element $L = \sum_{i=0}^{n} a_i(x)\tau^i \in \mathcal{P}$ corresponds to a recurrence relation $L(y(x)) = 0$, i.e.

$$a_n(x)y(x+n) + a_{n-1}(x)y(x+n-1) + \cdots + a_0(x)y(x) = 0. \tag{3.1}$$

So we can express $y(r)$ in terms of $y(r-1), \ldots, y(r-n)$ where $r = x+n$. The expression is not defined when r is a root of the denominator, which is $a_n(r-n)$. Hence we define:

Definition 21. *Let $L = \sum_{i=0}^{n} a_i(x)\tau^i \in \mathcal{P}$ be primitive. If $a_n \neq 0$ then define* $\mathrm{ord}(L) := n$ *and let $\mathfrak{lc}(L)$ be $a_n(x-n)$ divided by its leading coefficient (to make it monic). The* singularities *of L are the monic irreducible factors of $\mathfrak{lc}(L)$ in $F[x]$ (or equivalently, their roots in $\overline{F}$) with their multiplicities.*

Example 22. Let

$$L = x^2(x^2+1)\tau - (x+1)(x^2+2x+2) \in \mathbb{Q}[x][\tau].$$

Substituting $x \mapsto x - \mathrm{ord}(L)$ in the leading coefficient we obtain $\mathfrak{lc}(L) = (x-1)^2((x-1)^2+1)$. With repetition indicating multiplicity, the singularities are $x-1$, $x-1$, $(x-1)^2+1$, or equivalently 1, 1, $1 \pm \sqrt{-1}$. Let

$$A = \frac{1}{(x+1)\,(x^2+2\,x+2)}(10\,\tau + 11\,x^2 + 15\,x + 14).$$

Then

$$AL = 10\,(x+1)\,\tau^2 + \left(11\,x^3 - 18\,x^2 + 35\,x - 50\right)\tau - 11\,x^2 - 15\,x - 14.$$

Now $\mathfrak{lc}(AL) = x + 1 - \mathrm{ord}(AL) = x - 1$ (we divided by 10 to make it monic). Compared to L, the singularity $(x-1)^2+1$ disappeared, as well as one of the two copies of $x-1$. ▲

Lemma 23 ([15, Theorem 4.1.7, Corollary 4.1.9]). *Let $L \in \mathcal{P}$. For $k = 0, 1, 2, \ldots$ let*

$$\mathcal{I}_k = \{0\} \cup \{\mathfrak{lc}(AL) \ : \ A \in \mathcal{D}, AL \in \mathcal{P}, \mathrm{ord}(A) = k\} \tag{3.2}$$

and let $\mathcal{I}_\infty$ be their union. Then $\mathcal{I}_0 \subseteq \mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq \cdots \subseteq \mathcal{I}_\infty$ are ideals in $F[x]$.

Proof. To show $\mathcal{I}_k \subseteq \mathcal{I}_{k+1}$, take a non-zero $a \in \mathcal{I}_k$. So $a = \mathfrak{lc}(AL)$ for some A of order k. Replacing A by τA shows that $a \in \mathcal{I}_{k+1}$. Clearly $\mathcal{I}_k$ is closed under $F[x]$-multiplication, and it is not difficult to show that it is closed under addition as well. □

Definition 24 (Essential parts, removable parts, [15, Definitions 4.1.8 and 4.1.10]). *With notations as in Lemma 23, let $\mathfrak{lc}_k(L)$ be the monic generator of $\mathcal{I}_k$ for $k = 0, 1, 2, \ldots, \infty$. Call $\mathfrak{lc}_k(L)$ the* essential part of the leading coefficient at order k. *Note that $\mathfrak{lc}_0(L) = \mathfrak{lc}(\mathrm{Prim}(L))$ and $\mathfrak{lc}_l(L)$ divides $\mathfrak{lc}_k(L)$ if $l \geqslant k$. Let $\mathfrak{rp}_k(L) = \frac{\mathfrak{lc}(L)}{\mathfrak{lc}_k(L)} \in F[x]$ and call it the* removable part of the leading coefficient at order k.

Factors (or roots) of $\mathfrak{lc}(L)$ are divided into two (possibly overlapping) sublists: Factors of $\mathfrak{lc}_\infty(L)$ are the *true singularities* of L. Factors of $\mathfrak{rp}_\infty(L)$ are the *apparent singularities*. In Example 22 the true singularity is $x - 1$ and the apparant singularities are $x - 1$, $(x-1)^2 - 1$.

Definition 25. *Let $L \in \mathcal{P}$ be primitive. We call $A \in \mathcal{D}$ a* desingularizer *if $AL \in \mathcal{P}$. Such A is* trivial *if $A \in \mathcal{P}$ (that implies $\mathfrak{lc}(L) \mid \mathfrak{lc}(AL)$, so no singularities were removed). A desingularizer A is* optimal at order k *if $\mathfrak{lc}(AL) = \mathfrak{lc}_k(L)$ and $\mathrm{ord}(A) \leqslant k$; when $k = \infty$ say A is* optimal *since it removes all apparant singularities while introducing no new singularities.*

An optimal desingularizer at order k exists because $\mathcal{I}_k$ is a principal ideal; $F[x]$ is a PID.

3.2.2 LCLM method for desingularization

Desingularization of recurrence operators has been studied in [2], [1], [7], [15], [8] and [24]. Papers [2], [1], [7], [15] aim for full desingularization; [24] focuses on desingularization over $R[x]$ where R is not a field. Here we describe the so-called *LCLM method*, published in [8]. LCLM and GCRD are defined in section 2.2. The main result of [8] is restated below, where we focus on the recurrence case while the original version also applies to other types of Ore operators.

Theorem 26 (Reformulation of Theorem 6 in [8]). *Suppose $L \in \mathcal{P}$. Introduce new constants $c_0, c_1, \ldots, c_k$ that are algebraically independent over F. Denote $A = c_0 + c_1\tau + \cdots + c_k\tau^k$ and $L' = \mathrm{Prim}(\mathrm{LCLM}(L, A)) \in F(c_0, c_1, \ldots, c_k)[x][\tau]$. Then $\mathfrak{lc}(L') = \mathfrak{lc}_k(L)f$ where $f \in F(c_0, c_1, \ldots, c_k)[x]$ has no non-trivial factor in $F[x]$.*

The original form of Theorem 6 in [8]:

Let q be an irreducible polynomial which appears with multiplicity e in $\mathfrak{lc}(L)$ and let $m \leqslant e$ be maximal such that $q^m \mid \frac{\mathfrak{lc}(L)}{\mathfrak{lc}_k(L)}$ for $k \in \mathbb{N}$. Let $A = c_0 + c_1\tau + \cdots + c_k\tau^k$ in $F(c_0, \ldots, c_k)[x][\tau]$, where $c_0, \ldots, c_k$ are new constants that are algebraically independent over F. Denote $L' = \mathrm{Prim}(\mathrm{LCLM}(L, A))$. Then the multiplicity of q in $\mathfrak{lc}(L')$ is $e - m$.

The proof in [8] also holds when $e = m = 0$, which results in Theorem 26. It was stated for the case $\mathrm{char}(F) = 0$, but the proof is valid for positive characteristic as well.

Remark 27. Theorem 26 implies $\mathfrak{lc}_k(L)$ stays the same if L is viewed as an operator in $E(x)[\tau]$ where E is a field extension of F, as the field extension does not affect $L' = \mathrm{Prim}(\mathrm{LCLM}(L, A))$.

Theorem 26 implies the following desingularization algorithm.

```
Algorithm 1: LCLM_Method
Input  : a primitive operator $L = \sum_{i=0}^{n} a_i\tau^i \in F[x][\tau]$ and positive integer $k$
Output: $\mathfrak{lc}_k(L)$
$A \leftarrow \sum_{i=0}^{k} c_i\tau^i$, where $c_0, c_1, \ldots, c_k$ are new constants that are algebraically independent over $F$;
$L' \leftarrow \mathrm{Prim}(\mathrm{LCLM}(A, L))$;
return $\gcd(\mathfrak{lc}(L), \mathfrak{lc}(L'))$;
```

The discussion following the main theorem in [8] states that in characteristic 0, instead of new constants, we can let $c_0, c_1, \ldots, c_k$ be random elements in F. In this case the algorithm is Monte-Carlo, meaning it returns the desired result with a high probability. The Monte-Carlo version is much faster since it avoids computations in a transcendental extension of F. It was implemented in [22] with $k = 1$ by the second author in 2004.

We refer to the algorithm where $k = 1$ as *order-1 LCLM method*. The Monte-Carlo version of order-1 LCLM method strikes a good balance between benefit and cost, since the LCLM computation is much faster for $k = 1$ than for larger k, and $\mathfrak{lc}_1(L)$ is often very close to $\mathfrak{lc}_\infty(L)$, and hence is often used in practice. We will further speed up the algorithm in Section 3.6.

3.2.3 The p-characteristic polynomial

From here until Section 3.6, F will be a field of characteristic p, where p is a prime number.

A general theory of linear difference equations in positive characteristic is developed in [19, Chapter 5]. In [3], p-characteristic polynomials of recurrence operators (and differential operators) over $\mathbb{F}_p[x]$ are studied and an algorithm for computing them is given. An algorithm for computing p-characteristic polynomials of operators in $\mathbb{Z}[x][\tau]$ for a number of p is presented in [16], based on the algorithm from [3]. We will give more information about these algorithms in subsection 3.2.4

Let $Z = x^p - x = x(x+1)\cdots(x+p-1)$. Clearly Z is fixed by τ and hence elements of $F(Z)$ are τ-constant. In fact, $F(Z)$ *is* the field of τ-constants; to see this, notice that $F(x)$ is a degree p field extension of $F(Z)$ and hence there is no proper intermediate field.

Let $\mathcal{N} : F(x) \to F(Z)$ denote the norm map of the field extension $F(x)/F(Z)$. It is given by the formula

$$\mathcal{N} : f(x) \mapsto f(x)f(x+1)\cdots f(x+p-1).$$

Denote $T = \tau^p$. The center of $\mathcal{D}$ is $F(Z)[T]$. Since $F(x)[T] \subseteq \mathcal{D}$, any $\mathcal{D}$-module is naturally an $F(x)[T]$-module, that is, an $F(x)$-vector space equipped with an $F(x)$-linear map.

Definition 28. *For a $\mathcal{D}$-module M, call the $F(x)$-linear map induced by T the* p-curvature *of M.*

For an operator $L \in \mathcal{D}$, define its p-curvature to be that of $\mathcal{D}/\mathcal{D}L$. Denote $\chi(L) \in F(x)[T]$ its characteristic polynomial with T as its variable and call it the p-characteristic polynomial *of L.*

A characteristic polynomial is monic by definition so the leading coefficient of L is lost in $\chi(L)$. To reinsert it, denote $\tilde{\chi}(L) = \mathcal{N}(\mathfrak{lc}(L))\chi(L)$. It is called the reduced norm *of L in [3].*

Lemma 29. *Properties of p-characteristic polynomials.*

(i) For $L \in \mathcal{D}$, $\chi(L) \in F(Z)[T]$.

(ii) For $L \in \mathcal{D}$, $\chi(L) \in \mathcal{D}L$.

(iii) For $L_1, L_2 \in \mathcal{D}$, $\chi(L_1L_2) = \chi(L_1)\chi(L_2)$ and $\tilde{\chi}(L_1L_2) = \tilde{\chi}(L_1)\tilde{\chi}(L_2)$.

(iv) For $L_1, L_2 \in \mathcal{D}$, if $\mathrm{GCRD}(L_1, L_2) = 1$, then

$$\chi(\mathrm{LCLM}(L_1, L_2)) = \chi(L_1)\chi(L_2).$$

(v) For $L \in \mathcal{P}$, $\tilde{\chi}(L) \in F[Z][T]$ and $\deg_Z(\tilde{\chi}(L)) \leqslant \deg_x(L)$.

(vi) If $L \in F(Z)[T]$ then $\tilde{\chi}(L) = L^p$.

Proof. All except item (iv) are proved in [3, Section 3] for the case $F = \mathbb{F}_p$ and the proofs are valid for a general field F with positive characteristic. We now prove (iv). Denote $L = \mathrm{LCLM}(L_1, L_2)$. If $\mathrm{GCRD}(L_1, L_2) = 1$ then $\mathcal{D}/\mathcal{D}L \cong \mathcal{D}/\mathcal{D}L_1 \oplus \mathcal{D}/\mathcal{D}L_2$ as $\mathcal{D}$-modules (and hence as $F(x)[T]$-modules). Now (iv) follows from the fact that characteristic polynomials are multiplicative on direct sums. □

Lemma 29(iii) implies that an operator factors only when its p-characteristic polynomial factors (as a polynomial in $F(Z)[T]$). In fact, the p-characteristic polynomial tells us even more. See [9] and [20] for discussions on this topic in the differential case. The p-characteristic polynomial is also useful for testing or proving irreducibility of operators in $\mathbb{Q}(x)[\tau]$ by reduction modulo p.

3.2.4 BCS algorithm and Pagès' algorithm

Bostan, Caruso and Schost (2015) present an algorithm for computing the p-characteristic polynomial of an operator in $\mathbb{F}_p[x][\tau]$, called `Xi_theta_d` in [3]. We refer to it as the BCS algorithm. Their implementation in Magma is available at `https://github.com/schost`.

The BCS algorithm takes a prime p and a difference operator $L \in \mathbb{F}_p[x][\tau]$ as its input and computes $\tilde{\chi}(L) \in \mathbb{F}_p[Z][T]$ (making $\tilde{\chi}(L)$ monic gives $\chi(L)$). The algorithm computes $\tilde{\chi}(L)$ in $\mathbb{F}_p[[Z]][T]$ to precision $O(Z^{\deg_x(L)+1})$ which suffices by Lemma 29(v).

For $L \in \mathcal{P}$, the first part of Lemma 29(v) implies that $\mathcal{N}(\mathfrak{lc}(L))$ is a denominator bound for $\chi(L)$. The BCS algorithm uses this bound to ensure that what it computes in $\mathbb{F}_p[[Z]][T]$ is in $\mathbb{F}_p[Z][T]$, not just in $\mathbb{F}_p(Z)[T]$. We will show that (partial) desingularization leads to sharper denominator bounds. That reduces the required Z-adic precision, speeding up the computation. In fact, our main result Theorem 30 says that full desingularization gives the exact denominator.

For an operator $L \in \mathbb{Q}(x)[\tau]$, denote by $\chi_p(L)$ the p-characteristic polynomial of its reduction modulo p. Pagès (2021) gives an algorithm for computing $\chi_p(L)$ for a number of primes at the same time, if $L \in \mathbb{Z}[x][\tau]$ has a leading coefficient in $\mathbb{Z}$ ([16, Algorithm 3]). The algorithm is based on the BCS algorithm.

3.3 Main Theorem and Corollaries

Let denom$(\cdot)$ be the monic denominator of a rational function, or of a polynomial with rational function coefficients. We will use this notation in the cases $F[x] \subset F(x)$ and $F[Z][T] \subset F(Z)[T]$.

Theorem 30. *For $L \in F[x][\tau]$,* $\operatorname{denom}(\chi(L)) = \mathcal{N}(\mathfrak{lc}_\infty(L))$.

The theorem quickly implies two corollaries, expressed in terms of the following definition.

Definition 31. *Let $r_1, r_2 \in F(x) - \{0\}$. We say that r_1 and r_2 are* shift equivalent*, denoted $r_1 \sim r_2$, if $\tau - \frac{r_1}{r_2}$ has a non-zero solution in $F(x)$, in other words, if there exists $f \in F(x) - \{0\}$ for which $\frac{r_1}{r_2} = \frac{\tau(f)}{f}$.*

If $r(x)$ has a factor $q(x)$ in the numerator or denominator, and one replaces $q(x)$ by its shift $q(x+1)$, then the result is shift-equivalent to $r(x)$. Note that $r_1 \sim r_2$ if and only if $\mathcal{N}(r_1) = \mathcal{N}(r_2)$.

Corollary 32. *If $\mathcal{D}/\mathcal{D}L_1 \cong \mathcal{D}/\mathcal{D}L_2$ for $L_1, L_2 \in \mathcal{D}$, then $\mathfrak{lc}_\infty(L_1)$ and $\mathfrak{lc}_\infty(L_2)$ are shift equivalent, so L_1 and L_2 have the same true singularities up to shifts.*

Corollary 33. *For $L_1, L_2 \in \mathcal{D}$, if*

- *$L = L_1 L_2$, or*
- *$L = \mathrm{LCLM}(L_1, L_2)$ and $\mathrm{GCRD}(L_1, L_2) = 1$*

then $\mathfrak{lc}_\infty(L)$ and $\mathfrak{lc}_\infty(L_1)\mathfrak{lc}_\infty(L_2)$ are shift equivalent.

These corollaries are not true in characteristic 0, so we did not expect Theorem 30.

3.4 Proof of the Main Theorem

This section is devoted to the proof of Theorem 30. We start with an easy lemma.

Lemma 34. *For any $A, L \in \mathcal{D}$*

$$\mathrm{denom}(\chi(AL)) = \mathrm{denom}(\chi(A))\,\mathrm{denom}(\chi(L)).$$

Proof. Lemma 29(iii) and Gauss's lemma for $F(Z)[T]$ gives

$$\mathrm{Cont}(\chi(AL)) = \mathrm{Cont}(\chi(A))\mathrm{Cont}(\chi(L)).$$

But $\chi(\cdot)$ is monic by definition, and the denominator of a monic polynomial is the reciprocal of the content. □

3.4.1 Special case, Gaussian operators

Lemma 35. *Let $L \in \mathcal{P}$. The following are equivalent.*

1. *L is Gaussian, i.e. $\forall_{A \in \mathcal{D}}\ AL \in \mathcal{P} \implies A \in \mathcal{P}$.*
2. *Every desingularizer is trivial.*
3. *$\mathrm{Cl}(L) = \mathcal{P}L$, where $\mathrm{Cl}(L) := \mathcal{D}L \bigcap \mathcal{P}$. (This is called the* Weyl closure *in [18].)*
4. *$\mathfrak{lc}(L) = \mathfrak{lc}_\infty(L)$, i.e. there are no apparant singularities.*

Proof. Items 2 and 3 are reformulations of item 1, and immediately imply item 4. It remains to show that item 4 implies item 1. Suppose that $\mathfrak{lc}(L) = \mathfrak{lc}_\infty(L)$ and $AL \in \mathcal{P}$. To prove: $A \in \mathcal{P}$.

By partial fraction decomposition, $A = A_1 + A_2$ where $A_1 \in \mathcal{P}$ and $A_2 = \sum q_i \tau^i$ with the numerator of q_i having lower degree than its denominator. Since AL and A_1L are in $\mathcal{P}$, their difference A_2L is in $\mathcal{P}$ as well. If $A_2 \neq 0$, then the leading coefficient of A_2L will have lower degree than $\mathfrak{lc}(L)$, contradicting item 4. Thus $A_2 = 0$ and hence $A \in \mathcal{P}$. □

Lemma 29(v) says that $\tilde{\chi}(L) = \mathcal{N}(\mathfrak{lc}(L))\chi(L) \in F[Z][T]$ when $L \in \mathcal{P}$, in other words

$$\text{denom}(\chi(L)) \mid \mathcal{N}(\mathfrak{lc}(L)). \tag{3.3}$$

Here we give a sharper denominator bound.

Lemma 36. *For $L \in F[x][\tau]$, $\text{denom}(\chi(L)) \mid \mathcal{N}(\mathfrak{lc}_\infty(L))$.*

Proof. Let $A \in \mathcal{D}$ be an optimal desingularizer of L, then $\mathfrak{lc}(AL) = \mathfrak{lc}_\infty(L)$. From Lemma 34 and Equation (3.3) applied to AL, $\text{denom}(\chi(L)) \mid \text{denom}(\chi(AL)) \mid \mathcal{N}(\mathfrak{lc}(AL)) = \mathcal{N}(\mathfrak{lc}_\infty(L))$. □

Next we show that our denominator bound is exact for Gaussian operators. The next section will prove the general case by exploiting the fact that any operator has a Gaussian multiple.

Lemma 37. *If $L \in F[x][\tau]$ is Gaussian, then*

$$\text{denom}(\chi(L)) = \mathcal{N}(\mathfrak{lc}_\infty(L)).$$

Proof. Denote $f = \text{Prim}(\chi(L)) \in F[Z][T]$. By Lemma 29(ii), $f \in \mathcal{D}L$ so there exists $Q \in \mathcal{D}$ such that

$$QL = f. \tag{3.4}$$

In fact $Q \in \mathcal{P}$ since L is Gaussian. Lemma 29(v) says $\tilde{\chi}(Q), \tilde{\chi}(L) \in F[Z][T]$. Applying $\tilde{\chi}$ to Equation (3.4), and Lemma 29(vi), gives

$$\tilde{\chi}(Q)\tilde{\chi}(L) = \tilde{\chi}(f) = f^p. \tag{3.5}$$

Now $f^p \in F[Z][T]$ is primitive since f is primitive. Then Gauss's lemma implies $\tilde{\chi}(L) \in F[Z][T]$ is primitive. It follows that

$$\text{denom}(\chi(L)) = \mathfrak{lc}(\tilde{\chi}(L)) = \mathcal{N}(\mathfrak{lc}(L)) = \mathcal{N}(\mathfrak{lc}_\infty(L))$$

where the last equality comes from Lemma 35, part 4. □

3.4.2 Proof for the general case

Lemma 38. *Suppose $L \in \mathcal{P}$ and $A = \sum_{i=0}^{k} \frac{n_i}{d_i} \in \mathcal{D}$ is a desingularizer of L, where $\frac{n_i}{d_i} \in F(x)$ is in lowest terms for each i. Then $\mathcal{N}(d_k) \mid \mathcal{N}(\mathfrak{rp}_\infty(L))$.*

Proof. Let n be the order of L. The definition of $\mathfrak{lc}_\infty$ and the product of the leading terms of A and L gives

$$\mathfrak{lc}_\infty(L) \mid \tau^{-k-n}(\frac{n_k}{d_k})\mathfrak{lc}(L)$$

and hence $\tau^{-k-n}(\frac{n_k}{d_k})\mathfrak{rp}_\infty(L) \in F[x]$. Since $\frac{n_k}{d_k}$ is a reduced fraction, we have $\tau^{-k-n}(d_k) \mid \mathfrak{rp}_\infty(L)$, which leads to

$$\mathcal{N}(d_k) = \mathcal{N}(\tau^{-k-n}(d_k)) \mid \mathcal{N}(\mathfrak{rp}_\infty(L)).$$

□

Lemma 39. *Suppose $L \in \mathcal{P}$ and $A \in \mathcal{D}$ is an optimal desingularizer of L. Then there exists a positive integer N such that*

$$\text{denom}(\chi(A)) \mid (\mathfrak{rp}_\infty(L))^N.$$

Proof. Write $A = \sum_{i=0}^{k} \frac{n_i}{d_i}\tau^i$, where $\frac{n_i}{d_i} \in F(x)$ is a reduced fraction for each i. We deduce $n_k = 1, d_k = \tau^{n+k}(\mathfrak{rp}_\infty(L))$ from the fact that $\mathfrak{lc}(AL) = \mathfrak{lc}_\infty(L)$. Clearly $d_0 d_1 \cdots d_k A \in \mathcal{P}$. Then by Equation (3.3)

$$\text{denom}(\chi(A)) \mid \mathcal{N}(\mathfrak{lc}(d_0 d_1 \cdots d_k A)) = \mathcal{N}(d_0 d_1 \cdots d_{k-1}).$$

Now we bound $\mathcal{N}(d_i)$ in terms of $\mathfrak{rp}_\infty(L)$. Let

$$A_j = d_k d_{k-1} \cdots d_{j+1}(\sum_{i=0}^{j} \frac{n_i}{d_i}\tau^i)$$

for $j = 0, 1, \ldots, k-1$. Notice that

$$A_j - d_k d_{k-1} \cdots d_{j+1} A = -d_k d_{k-1} \cdots d_{j+1}(\sum_{i=j+1}^{k} \frac{n_i}{d_i}\tau^i) \in \mathcal{P}.$$

This implies $A_j L \in \mathcal{P}$, or equivalently, A_j is a desingularizer of L. Apply Lemma 38 to A_j:

$$\mathcal{N}(\text{denom}(d_k d_{k-1} \cdots d_{j+1}\frac{n_j}{d_j})) \mid \mathcal{N}(\mathfrak{rp}_\infty(L)).$$

Notice that

$$d_j \mid d_k d_{k-1} \cdots d_{j+1} \cdot \text{denom}(d_k d_{k-1} \cdots d_{j+1}\frac{n_j}{d_j}).$$

Therefore

$$\mathcal{N}(d_j) \mid \mathcal{N}(d_k d_{k-1} \cdots d_{j+1}) \cdot \mathcal{N}(\mathfrak{rp}_\infty(L)).$$

Recall that $\mathcal{N}(d_k) \mid \mathcal{N}(\mathfrak{rp}_\infty(L))$. By downward induction on j, we conclude that

$$\mathcal{N}(d_j) \mid (\mathcal{N}(\mathfrak{rp}_\infty(L)))^{k-j+1}.$$

□

We are now ready to finish the proof of Theorem 30.

Proof of Theorem 30. It remains to show that $\mathcal{N}(\mathfrak{lc}_\infty(L)) \mid \operatorname{denom}(\chi(L))$ for any $L \in \mathcal{D}$. There exists a sufficiently large k such that $\mathfrak{lc}_k(L) = \mathfrak{lc}_\infty(L)$. Introduce new constants $c_0, \ldots, c_k$ that are algebraically independent over F and denote $E = F(c_0, c_1, \ldots, c_k)$. Let

$$L' = \operatorname{Prim}(\operatorname{LCLM}(c_k\tau^k + \cdots + c_0, L)) \in E[x][\tau].$$

Theorem 26 says $\mathfrak{lc}(L') = \mathfrak{lc}_\infty(L)f$, where $f \in E[x]$ has no non-trivial factor in $F[x]$. It follows from Definition 24 (see also Equation 3.2) that $\mathfrak{lc}_\infty(L) \mid \mathfrak{lc}_\infty(L')$ and hence $\mathfrak{rp}_\infty(L') \mid f$. Remark 27 guarantees $\mathfrak{lc}_\infty(L)$ does not change as we shift from F to E. Let A be an optimal desingularizer of L'. By Lemma 29 ((iii) and (iv)), we have

$$\chi(AL') = \chi(A)\chi(c_k\tau^n + \cdots + c_0)\chi(L). \tag{3.6}$$

Lemma 29(vi) implies

$$\operatorname{denom}(\chi(c_k\tau^n + \cdots + c_0)) = 1.$$

Applying Lemma 34 to Equation (3.6) gives

$$\operatorname{denom}(\chi(AL')) = \operatorname{denom}(\chi(A)) \operatorname{denom}(\chi(L)).$$

Since AL' is Gaussian, we know from Lemma 37

$$\operatorname{denom}(\chi(AL')) = \mathcal{N}(\mathfrak{lc}_\infty(AL')),$$

which equals $\mathcal{N}(\mathfrak{lc}_\infty(L'))$ since A is an optimal desingularizer of L'. As a consequence,

$$\mathcal{N}(\mathfrak{lc}_\infty(L)) \mid \mathcal{N}(\mathfrak{lc}_\infty(L')) = \operatorname{denom}(\chi(A)) \operatorname{denom}(\chi(L)). \tag{3.7}$$

Lemma 39 says $\operatorname{denom}(\chi(A))$ is a factor of f^N for some sufficiently large N, so $\operatorname{denom}(\chi(A))$ has no non-trivial factor in $F[x]$. By taking only factors in $F[x]$ in Equation 3.7, we obtain the desired result $\mathcal{N}(\mathfrak{lc}_\infty(L)) \mid \operatorname{denom}(\chi(L))$.

□

3.5 Application to Computations

In this section $F = \mathbb{F}_p$.

3.5.1 Algorithm

Let $L \in \mathcal{P}$ and $\alpha = \mathfrak{rp}_k(L)$. Theorem 30 implies that $\mathcal{N}(\alpha)$, which is in $\mathbb{F}_p[Z]$, is a factor of $\tilde{\chi}(L)$. Dividing this factor away reduces the degree bound from Lemma 29(v) to

$$\deg_Z(\,\mathcal{N}(\alpha)^{-1}\tilde{\chi}(L)\,) \leqslant \deg_x(L) - \deg_x(\alpha) \tag{3.8}$$

which becomes an equality when k is sufficiently large. However, we use $k = 1$ to minimize the time spent computing α. The reduced degree bound allows us to recover $\chi(L)$ from a lower precision Z-adic expansion. That leads to the following algorithm.

```
Algorithm 2: Xi_p_desing
Input  : prime $p$ and $L \in \mathbb{F}_p[x][\tau]$
Output: $\mathrm{Prim}(\chi(L)) \in \mathbb{F}_p[Z][T]$
1 Pick $k \geqslant 1$ and compute $\mathfrak{lc}_k(L)$ and $\alpha := \mathfrak{rp}_k(L) \in \mathbb{F}_p[x]$. We use $k = 1$ to minimize the time
  spent in this step.
2 Compute $\mathcal{N}(\alpha) \in \mathbb{F}_p[Z]$. Let $v$ be its $Z$-adic valuation in $\mathbb{F}_p[[Z]]$ and let
  $\beta = Z^{-v}\mathcal{N}(\alpha) \in \mathbb{F}_p[Z]$.
  For computing $\mathcal{N}(\cdot)$ see Step 3 of Xi_theta_d in [3].
3 Let $d_1 = \deg_Z(\beta)$ and $d := \deg_x(L)$. Apply the BCS algorithm with $d$ replaced by $d - d_1$ to
  $L$ to obtain $\tilde{\chi}(L)$ up to the precision $O(Z^{d-d_1+1})$. Denote the result by $\chi_1$.
4 Compute $\beta^{-1}$ in $\mathbb{F}_p[[Z]]$ up to $O(Z^{d-d_1-v+1})$.
  This can be done by applying the extended Euclidean algorithm to $\beta$ and $Z^{d-d_1-v+1}$.
5 Compute $\beta^{-1} \cdot (Z^{-v}\chi_1)$ in $\mathbb{F}_p[[Z]][T]$ up to the precision $O(Z^{d-d_1-v+1})$. This gives
  $\mathcal{N}(\alpha)^{-1}\chi_1 \in \mathbb{F}_p[Z][T]$.
  Return its primitive part (with respect to $T$).
  Note: $\mathcal{N}(\alpha)^{-1}\chi_1$ and $\mathcal{N}(\alpha)^{-1}\tilde{\chi}(L)$ agree to precision $O(Z^{d-d_1-v+1})$ which suffices by (3.8).
```

Step 3 is where we save CPU time over the original algorithm from [3] if $d_1 > 0$. If $d_1 = 0$ then there is no improvement in efficiency. However, as we will see in the following section, the extra steps cost very little time.

3.5.2 Implementation and timings

Our Magma implementation of algorithm `Xi_p_desing` is available at `https://www.math.fsu.edu/~yzhou/magma/`, together with experiments on a variety of operators. One should load the implementation of [3] at `https://github.com/schost/pCurvature` (file `pCurvature.mgm`) prior to ours.

In the following table the data for two operators from OEIS ([11], [13]) is presented. Here d_1 is defined in the Step 3 of `Xi_p_desing`; each running time is the average of ten runs.

Table 3.1: Timings for operators from OEIS.

OEIS index	p	order	x-degree	d_1	BCS	`Xi_p_desing`
A151329	27457	9	18	10	17.2s	9.6s
A002777		4	3	0	6.97s	7.01s

For the recurrence for OEIS A002777, we expect `Xi_p_desing` to be slower than BCS since $d_1 = 0$. However, the running time difference between two algorithms is nearly unnoticeable.

We also tested our algorithm on operators that are LCLMs of two operators [25]. Such operators tend to have many apparent singularities and hence benefit more from our approach.

3.6 Fast Algorithms for Desingularization at Order 1

3.6.1 First algorithm

In this section we present our first speedup of the order-1 LCLM method. We used it for Step 1 of algorithm 2.

The order-1 LCLM method computes $L' = \mathrm{LCLM}(L, \tau - c)$ where c is a new constant (or a random number in the Monte-Carlo version). To speed this up, our idea is to obtain $\mathfrak{lc}_1(L)$ while only computing a portion of L'.

First we express of L' in terms of c and coefficients of L. Suppose $L = \sum_{i=0}^{n} a_i \tau^i \in F(x)[\tau]$. Then $L' = \sum_{i=0}^{n} c^i L_i$, where

$$L_i = a_i \tau L - \tau(a_{i-1}) L = (a_i \tau - \tau(a_{i-1})) L, \tag{3.9}$$

where $a_i = 0$ for $i < 0$ and $i > n$. Clearly this L' is a left multiple of L. To verify it is also a left-multiple of $\tau - c$, use the fact that the remainder of τ^i right-divided by $\tau - c$ is c^i. We skip the tedious computation. As a result L' is an LCLM of L and $\tau - c$.

The order-1 LCLM method computes L' which amounts to compute all L_i's. The following proposition shows that one can provably obtain $\mathfrak{lc}_1(L)$ from just a subset of the L_i's.

Proposition 40. *Let $L = \sum_{i=0}^{n} a_i \tau^i \in F[x][\tau]$. Let L_i be defined by Equation 3.9, where $a_i = 0$ for $i < 0$ and $i > n$. If*

$$\gcd(a_{i_1}, a_{i_2}, \ldots, a_{i_k}) = 1, \tag{3.10}$$

then

$$\gcd(\mathfrak{lc}_0(L_{i_1}), \mathfrak{lc}_0(L_{i_2}), \ldots, \mathfrak{lc}_0(L_{i_k})) = \mathfrak{lc}_1(L).$$

The proof will be given in the next section. Note that there exist $i_1, i_2, \ldots, i_k$ satisfying the gcd condition (Equation 3.10) if and only if L is primitive. The proposition immediately implies algorithm 3.

```
Algorithm 3: $\mathfrak{lc}_1$
Input  : a primitive operator $L = \sum_{i=0}^{n} a_i \tau^i \in F[x][\tau]$
Output: $\mathfrak{lc}_1(L)$
Find $I \subset \{0, 1, \ldots, n\}$ such that $a_i \neq 0$ for any $i \in I$ and $\gcd(a_i \mid i \in I) = 1$. Note: the
    algorithm is still correct if we allow $a_i = 0$, but that $i$ is redundant since it does not affect
    the gcd at all.
Compute $L_i$ for $i \in I$ by Equation 3.9.
Return $\gcd(\mathfrak{lc}_0(L_i) : i \in I)$.
```

Remark 41. Computing $\mathfrak{lc}_0(L_i) = \mathfrak{lc}(\mathrm{Prim}(L_i))$ is the most time-consuming part in the algorithm, because L_i has twice the x-degree as L.

3.6.2 Proof

Always assume $L = \sum_{i=0}^{n} a_i \tau^i \in F[x][\tau]$ is primitive and L_i is defined by Equation 3.9.

Lemma 42. *There exists $b \in F[x]$ such that*

$$\mathrm{Cont}((\tau - b)L) = \tau^{n+1}(\mathfrak{rp}_1(L)).$$

Proof. Let $A \in \mathcal{D}$ be an optimal desingularizer of L at order 1. Then $A = \frac{1}{d_1}\tau - \frac{n_2}{d_2}$, where $d_1 = \tau^{n+1}(\mathfrak{rp}_1(L))$ and $\frac{n_2}{d_2} \in F(x)$ is a reduced fraction. Let $b = d_1 \frac{n_2}{d_2}$. Observe that

$$bL = \tau \cdot L - d_1 A L \in \mathcal{P}. \tag{3.11}$$

Due to L being primitive, b has to be a polynomial. Since A is an optimal desingularizer of L at order 1, $AL \in \mathcal{P}$ is primitive; otherwise dividing out the content of AL yields a more optimal desingularizer. By rearranging Equation 3.11 we see that $\frac{1}{d_1}(\tau - b)L = AL$ is primitive, which completes the proof. □

Theorem 43. *Let $C = (c_0, c_1, \ldots, c_{n+1}) \in F^{n+2}$. Denote*

$$C_1 = \sum_{i=0}^{n+1} c_i a_i, \quad C_0 = \sum_{i=0}^{n+1} c_i \tau(a_{i-1}), \quad L' = (C_1\tau - C_0)L.$$

Then

$$\mathfrak{lc}_1(L) \mid \mathfrak{lc}_0(L') \mid \tau^{-n-1}(C_1)\mathfrak{lc}_1(L).$$

Proof. Assume $C_1 \neq 0$ since otherwise it is trivial.

The relation $\mathfrak{lc}_1(L) \mid \mathfrak{lc}_0(L')$ immediately follows from the definition of $\mathfrak{lc}_1$. By Lemma 42, there exists $b \in F[x]$ such that $\mathrm{Cont}((\tau - b)L) = \tau^{n+1}(\mathfrak{rp}_1(L))$. Since

$$(\tau - b)L = \sum_{i=0}^{n+1}(\tau(a_{j-1}) - ba_j)\tau^j,$$

we have

$$\gcd(\tau(a_{j-1}) - ba_j \mid j = 0, 1, \ldots, n+1) = \tau^{n+1}(\mathfrak{rp}_1(L)). \tag{3.12}$$

Notice that

$$L' = C_1(\tau - b)L + (bC_1 - C_0)L,$$

and in particular

$$bC_1 - C_0 = \sum_{i=0}^{n+1} bc_i a_i - \sum_{i=0}^{n+1} c_i\tau(a_{i-1}) = \sum_{i=0}^{n+1} c_i(ba_i - \tau(a_{i-1}))$$

is a multiple of $\tau^{n+1}(\mathfrak{rp}_1)$ due to Equation 3.12. Hence $\frac{1}{\tau^{n+1}(\mathfrak{rp}_1)}L' \in F[x][\tau]$. When $C_1 \neq 0$,

$$\mathfrak{lc}\left(\frac{1}{\tau^{n+1}(\mathfrak{rp}_1(L))}L'\right) = \frac{1}{\mathfrak{rp}_1(L)}\tau^{-n-1}(C_1)\mathfrak{lc}(L) = \tau^{-n-1}(C_1)\mathfrak{lc}_1(L).$$

Then we have

$$\mathfrak{lc}_0(L') = \mathfrak{lc}(\mathrm{Prim}(L')) \mid \mathfrak{lc}\left(\frac{1}{\tau^{n+1}(\mathfrak{rp}_1(L))}L'\right) = \tau^{-n-1}(C_1)\mathfrak{lc}_1(L).$$

□

Proof of Proposition 40. In Theorem 43, setting $c_i = 1$ for some i and $c_j = 0$ for any $j \neq i$ yields

$$\mathfrak{lc}_1(L) \mid \mathfrak{lc}_0(L_i) \mid \tau^{-n-1}(a_i)\mathfrak{lc}_0(L).$$

The desired result follows immediately. □

3.6.3 Desingularizing both leading and trailing coefficients

The variation in this section handles both leading and trailing singularities. It uses only one L_i (defined in Equation 3.9) without checking the gcd condition (Equation 3.10), since most apparant singularities are already detected with one L_i.

In the algorithm, $\mathfrak{tc}_1$ denotes the *essential part of trailing coefficient at order 1*, which is the counterpart of $\mathfrak{lc}_1$ for trailing coefficients.

```
Algorithm 4: $\mathfrak{lc}1\_\mathfrak{tc}1$
    Input  : a primitive operator $L = \sum_{i=0}^{n} a_i \tau^i \in F[x][\tau]$ with $a_0 a_n \neq 0$
    Output: $l, t \in F[x]$ such that $\mathfrak{lc}_1(L) \mid l \mid \mathfrak{lc}(L)$ and $\mathfrak{tc}_1(L) \mid l \mid \mathfrak{tc}(L)$
$i \leftarrow \lfloor \frac{n}{2} \rfloor$
$L_i \leftarrow (a_i \tau - \tau(a_{i-1}))L$
$l, t \leftarrow \mathfrak{lc}_0(L_i), \mathrm{TC}_0(L_i)$
$l, t \leftarrow \gcd(\tau^{-n}(a_n), l), \gcd(a_0, t)$
return $l, t$
```

3.6.4 Examples and comparisons

We have implemented algorithm 3 and algorithm 4 in Maple and SageMath, and done some experiments to compare the running time of our algorithm with the order-1 LCLM method. All can be found at [25]. Below we give an experiment we did in Maple.

Example 44. In this example the base field is $\mathbb{Q}$. We took random operators

$$L_1 = (26x^4 + 20)\tau^{11} - 96x^3\tau^9 + 64x^5\tau^8 + 45x^{11}\tau^4 - x^2\tau^3,$$

$$L_2 = -55x^3\tau^7 + 85x^3\tau^4 + 64x^4\tau^3 + (-14x^8 - 20x^4)\tau + 79x,$$

and then computed

$$L = \mathrm{Prim}(\mathrm{LCLM}(L_1, L_2)).$$

The x-degree of L is 109. We desingularize L using three different algorithms. For the LCLM method we used the Monte-Carlo version and randomly choose $c = 7$. The results are shown in the Table 3.2, where each time is the average of ten runs.

One might expect $\mathfrak{lc}1\_\mathfrak{tc}1$ to be slower than LC1 because it treats both the leading and trailing coefficient, however, we expected it to be faster because it corresponds to taking just one L_i in LC1.

▲

Table 3.2: Comparison of different desingularization algorithms

algorithms	running time	x-degree in output
Order-1 LCLM	1.191s	6
LC1	0.055s	6
lc1_tc1	0.092s	6

3.7 Future Work

3.7.1 Application to Pagès' algorithm

Pagès' Algorithm computes $\chi_p(L)$ for $L \in \mathbb{Z}[x][\tau]$ with $\mathrm{lc}(L) \in \mathbb{Z}$, but with minor adjustments it applies to all recurrence operators in $\mathbb{Z}[x][\tau]$. We expect desingularization to be beneficial here as well.

3.7.2 Differential case

The desingularization improvement should also work for the differential case or Ore operators. For a differential operator $L = \sum_{i=0}^{n} a_i \partial^i \in F[x][\partial]$, we can write $\mathrm{LCLM}(L, \tau - c) = \sum_{i=0}^{n} c^i L_i$, where

$$L_i = \left(a_i \partial - (a_{i-1} + a_i')\right) L.$$

We expect that there should also be a differential analog of our main result, Theorem 30.

CHAPTER 4

HYPERGEOMETRIC SOLUTIONS OF DIFFERENCE SYSTEMS

4.1 Introduction

Suppose C is a subfield of $\mathbb{C}$. A nonzero element y in the universal extension (Definition 18) of $C(x)$ is called *hypergeometric* if $\tau(y)/y \in C(x)$, in other words, y is a solution of a first order operator. If $r \in C(x) - \{0\}$ then r is hypergeometric of trivial type. If y_1, y_2 are hypergeometric then $y_1 y_2$ is hypergeometric as well. Denote by $\mathrm{hyp}(\lambda)$ a nonzero solution of $\tau - \lambda$ in the universal extension for $\lambda \in C(x)$. The notation is defined up to a nonzero τ-constant.

The first algorithm to compute hypergeometric solutions of an operator was given by Petkovšek. One writes hypergeometric solutions of $L \in C[x][\tau]$ over $C(x)$ in this format:

$$y = \mathrm{hyp}(\lambda)P$$

where $\lambda = c\frac{A}{B}$ with $A, B, P \in C[x]$, A and B monic, and $c \in C - \{0\}$. Note that y can be rewritten as $\mathrm{hyp}(\lambda')$ where $\lambda' = \lambda\frac{\tau(P)}{P}$. Allowing a polynomial factor P in candidate solutions $y = \mathrm{hyp}(\lambda)P$ makes it easier to restrict λ to a computable set. Petkovšek's algorithm works as follows:

Step P1 Petkovšek proves that it suffices to consider A, B where $A|a_0$ and $\tau^{n-1}(B)|a_n$. This leaves a finite set of candidates for A/B.

Step P2 Compute candidates for c.

Step P3 For each candidate $\lambda = cA/B$:
Construct an operator L_λ whose solutions are the solutions of L divided by $\mathrm{hyp}(\lambda)$.
For all polynomial solutions P of L_λ: join $\mathrm{hyp}(\lambda)P$ to the output.

If M is an $n \times n$ invertible matrix over $C(x)$, consider the following n-dimensional system:

$$\tau(Y) = MY.$$

A hypergeometric solution of the system is one in the form $\mathrm{hyp}(\lambda)P$ where P is an n-dimensional column vector over $C(x)$. If $\tilde{P} = cP$ then $\mathrm{hyp}(\lambda)\tilde{P} = \mathrm{hyp}(\lambda\frac{\tau(c)}{c})P$ so we may assume without loss of generality that P is in $C[x]^n$ and is primitive (the content, the gcd of entries, is 1), see section 4.2.

Before his tragic passing, Bronstein observed that Petkovšek's strategy works for such systems as well ([4]). Thus, the Bronstein-Petkovšek strategy for hypergeometric solutions is as follows:

Step BP1 Construct the set of candidates for A/B:

$$\mathcal{S} := \{\frac{A}{B} : A, B \in C[x] \text{ are monic, } A \mid \mathrm{denom}(M^{-1}),\ B \mid \mathrm{denom}(M)\}.$$

Step BP2 For each candidate A/B compute candidates for c.

Step BP3 For each $\lambda = cA/B$: compute all polynomial solutions $P \in C[x]^n$ of $\tau(P) = \lambda^{-1}MP$ and join $\mathrm{hyp}(\lambda)P$ to the output.

Remarkably, Bronstein's proof for the sufficiency of Step 1 (Theorem 46) is actually easier than Petkovšek's proof, despite the fact that it is more general. To obtain an algorithm, Bronstein still needed a way to find candidates for c, which we will give in subsection 4.3.2.

The BP-strategy applies to many cases (difference systems, q-difference systems, the multi-basic case) provided that one can compute (1) candidates for c, and (2) polynomial solutions.

For operators $L \in C(x)[\tau]$, the paper [21] addressed the following issues in Petkovšek's algorithm:

(a) The number of candidates A/B can be much larger than it needs to be.

(b) As a side effect, the algorithm can produce duplicate solutions.

For systems, the same issues arise in the BP-strategy. The goal in this chapter is to address these issues. Of course one might discard duplicate solutions, or take steps to prevent them, but that still leaves issue (a). Reducing the number of candidates as much as possible leads to a more efficient algorithm and eliminates issue (b) as a side effect.

A key idea is that rather than bounding A/B using the denominators of M^{-1} and M, we bound the *type* of A/B by bounding *local types*. Our bounds for the local types are sharper than the global bounds in Step 1 above, leading to fewer candidates. In fact, our bounds are almost as sharp as those for operators in [21], but take less time to compute, so an operator version of our approach may well be faster than [21]. Experiments show that our implementation can handle systems of high dimension, which will be useful for the application in factoring operators discussed in subsection 4.6.2.

4.2 Hypergeometric Solutions

Definition 45 (Hypergeometric). *Let $\mathbb{F}$ be a difference field that has a universal extension. In this Chapter we are interested in the cases where $\mathbb{F} = C(x)$ or $C((x^{-1}))$. A non-zero element γ in the universal extension of $\mathbb{F}$ is called* hypergeometric *if $\tau(\gamma) = f\gamma$ for some $f \in \mathbb{F}^*$. In this case let* $\mathrm{hyp}(f)$ *denote γ. The notation is unique up to a τ-constant. When γ is hypergeometric, call the column vector $\gamma R \, (R \in \mathbb{F}^n)$ a* hypergeometric vector*.*

It is not hard to verify the following properties of hypergeometric elements:

(i) $\mathrm{hyp}(f_1)\,\mathrm{hyp}(f_2) = \mathrm{hyp}(f_1 f_2)$ up to a constant where $f_1, f_2 \in \mathbb{F}^*$;

(ii) $\mathrm{hyp}(\frac{\tau(f)}{f}) = f$ up to a constant where $f \in \mathbb{F}^*$.

Consider the system

$$\tau(Y) = MY, \qquad \text{where} \quad M \in \mathrm{GL}_n(C(x)). \tag{(SYS)}$$

The goal of this chapter is to design efficient algorithms for finding its hypergeometric solutions (solutions that are hypergeometric vectors).

A consequence of properties of hypergeometric elements is

$$\mathrm{hyp}(f) = g\,\mathrm{hyp}(f\frac{g}{\tau(g)}),$$

which implies a hypergeometric vector has seemingly different representations. Using the properties, over $C(x)$, a hypergeometric vector can always be written into γP where γ is hypergeometric and $P \in C[x]^n$ is primitive. This is called the *standard representation* of the hypergeometric vector.

4.3 Algorithm Version I

In this section a basic version of the algorithm is presented, which follows the procedure in section 4.1. We first give the algorithm and then explain why it works in the follow-up sections.

Algorithm: Version I

Input: $\tau Y = MY$, where $M \in \mathrm{GL}(n, C(x))$

Output: hypergeometric solutions of the system

Step BP1 – Compute $\mathrm{denom}(M)$ and $\mathrm{denom}(M^{-1})$ and factor them in $C[x]$.

– Compute

$$\mathcal{S} := \{\frac{A}{B} : A, B \in C[x] \text{ are monic}, A \mid \text{denom}(M^{-1}), B \mid \text{denom}(M)\}. \tag{4.1}$$

In subsection 4.3.1 we justify that $\mathcal{S}$ contains all $\frac{A}{B}$ that are needed.

Step BP2 – Compute

$$G := \{\text{unramified generalized exponents of the system}\}.$$

Generalized exponents will be defined in section subsection 4.3.2. An unramified generalized exponents is in the form $cx^s(1 + dx^{-1})$, where $(c, s, d) \in \mathbb{C}^* \times \mathbb{Z} \times \mathbb{C}$.

– Let

$$\mathcal{H} := \{c\frac{A}{B} : cx^s(1 + dx^{-1}) \in G, \frac{A}{B} \in \mathcal{S}, \deg(A) - \deg(B) = s, d - \text{slc}(\frac{A}{B}) \in \mathbb{N}\},$$

where the notation slc will be defined in subsection 4.3.2.

Step BP3 – Let Sols $= \emptyset$.

– For each $c\frac{A}{B} \in \mathcal{H}$, solve the system $\tau P = c^{-1}\frac{B}{A}MP$ for (a basis of) polynomial solutions using the algorithm introduced in [2]. Add $\text{hyp}(c\frac{A}{B})P$ to Sols for any polynomial solution P.

– Return Sols as the output.

4.3.1 Step BP1

Theorem 46. *Suppose $Y = \text{hyp}(f)P$ is a hypergeometric solution of Equation (*SYS*) that is in the standard form, where $f \in C(x)$. Then*

$$\text{numer}(f) \mid \text{denom}(M^{-1}), \quad \text{denom}(f) \mid \text{denom}(M).$$

The theorem says that any hypergeometric solution Y can be written as $Y = \text{hyp}(c\frac{A}{B})P$ for some primitive polynomial vector P, some τ-constant c, and some $\frac{A}{B} \in \mathcal{S}$, where $\mathcal{S}$ is defined in Step BP1 by Equation 4.1.

We remark that the result holds not only for $C[x] \subseteq C(x)$, but a general difference ring that is a UFD and its field of fractions as well.

Proof. Write now $M = d^{-1}W$ where $d = \text{denom}(M)$ and $\gcd(W) = 1$. Then, substitution of this and $Y = \gamma P$ in standard form into Equation (SYS) allows us to rewrite the equation as

$$\frac{\tau(\gamma)}{\gamma}\tau(P) = \frac{1}{d}WP, \qquad \text{or,} \qquad dA\tau(P) = BWP$$

using $\tau(\gamma)/\gamma = A/B$ and clearing denominators. Let now $f \in C[x]$ be any prime factor of B. Then f divides the right-hand side of the equation. On the left-hand side, if we assume A and B are coprime, f cannot divide A. Moreover, there is at least one component of $\tau(P)$ which is not divisible by f since the entries of P are coprime. Consequently, f must be a factor of d. Dividing d and B in the equation by f and continuing the argument for the other factors of B implies that $B \mid d$.

Write the inverse of M as $M^{-1} = {d'}^{-1}W'$ where $d' \in C[x]$ and where $W' \in C[x]^{n\times n}$ has coprime entries. Then, multiplying the original Equation (SYS) by M^{-1} and substituting again $Y = \gamma P$ in standard form, we obtain

$$\frac{1}{d'}W'\tau(P) = \frac{B}{A}P \qquad \text{or} \qquad AW'\tau(P) = d'BP.$$

Similarly as before, we can thus derive that $A \mid d'$. □

Example 47. Let

$$M = $$
$$\begin{pmatrix} 0 & 0 & \frac{x+1}{x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{x+1}{x} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{x+1}{x} \\ \frac{(x+1)^2}{x(x+2)(x+3)} & 0 & -\frac{(x+1)(x^4+2x^3+x^2+x+4)}{x(x+2)^2(x+3)} & 0 & -\frac{x+1}{x(x+2)^2(x+3)} & 0 \\ 0 & \frac{(x+1)^2}{x(x+2)(x+3)} & \frac{(x+1)^2}{x(x+2)^2(x+3)} & 0 & 0 & -\frac{x+1}{x(x+2)^2(x+3)} \\ 0 & 0 & 0 & \frac{(x+1)^2}{x(x+2)(x+3)} & \frac{(x+1)^2}{x(x+2)^2(x+3)} & \frac{(x+1)(x^4+2x^3+x^2+x+4)}{x(x+2)^2(x+3)} \end{pmatrix}$$

be a 6×6 matrix over $\mathbb{Q}$. Then $\operatorname{denom}(M) = x(x+2)^2(x+3)$ and $\operatorname{denom}(M^{-1}) = (x+1)^2(x+2)$. Therefore,

$$\mathcal{S}(M) = \{x^{i_1}(x+1)^{i_2}(x+2)^{i_3}(x+3)^{i_4} : -1 \leqslant i_1 \leqslant 0, 0 \leqslant i_2 \leqslant 2, -2 \leqslant i_3 \leqslant 1, -1 \leqslant i_4 \leqslant 0\},$$

whose cardinality is

$$2 \cdot 3 \cdot 4 \cdot 2 = 48.$$

This number is more than necessary; a provably complete search can be done with just 8 cases, see Example 62 which addresses issues (a) and (b) stated in section 4.1. ▲

Remark 48. Finding hypergeometric solutions of the system in 47 is related to the factorization of the operator

$$(x+2)^2(x+3)\tau^4 + \tau^3 + (x^4 + 2x^3 + x^2 + x + 4)\tau^2 + (x+1)\tau + (x+1)(x+2).$$

It is explained in subsection 4.6.2 how to convert a factorization problem into solving a system for hypergeometric solutions.

4.3.2 Step BP2: generalized exponents

The key of Step BP2 is the notion of *generalized exponents*. Generalized exponents of difference operators have been defined in [6, Section 3.2] and will also be discussed later in this thesis (section 6.1). We can define generalized exponents of a system in a similar way. First some necessary knowledge on the difference field $\mathbb{C}((x^{-1}))$ is introduced.

The difference field K. Denote $t = 1/x$. Let $K = \mathbb{C}((t))$ and $K_r = \mathbb{C}((t^{\frac{1}{r}}))$ for $r = 1, 2, 3, \ldots$. According to [23] the algebraic closure of K is

$$K_\infty := \bigcup_{r=1}^{\infty} K_r.$$

The notations $\overline{K}$ and K_∞ will be used interchangeably. When we write K_r, r can be a positive integer or ∞ unless otherwise stated. The canonical t-adic valuation on K extends to K_r naturally; we denote it by $v : \overline{K} \mapsto \mathbb{Q} \cup \{\infty\}$. We remind the readers that for v to be a valuation, it has to satisfy the following properties:

- $v(a) = \infty$ if and only if $a = 0$,
- $v(ab) = v(a) + v(b)$,
- $v(a + b) \geqslant \min\{v(a), v(b)\}$, with equality if $v(a) \neq v(b)$.

For a vector with entries in $\overline{K}$, define its valuation to be the smallest valuation of the entries. The big O and little-o notations are used for elements in $\overline{K}$ with respect to the valuation v. In particular, when we write $f = g + o(t^u)$ for $f, g \in \overline{K}$, it means $v(f - g) > u$.

A general non-zero element in K_r factors into $ct^s(1 + \sum_{i=1}^{\infty} a_i t^{\frac{i}{r}})$. Call c its *leading coefficient* and ct^s the *leading term*.

The action of τ on K is ruled by

$$\tau(t) = \tau(\frac{1}{x}) = \frac{1}{x+1} = \frac{t}{1+t} = t - t^2 + t^3 - \cdots,$$

which extends to K_r as well as $\overline{K}$ following

$$\tau(t^{\frac{1}{r}}) = t^{\frac{1}{r}}(1+t)^{-\frac{1}{r}} = t^{\frac{1}{r}}(1 - \frac{1}{r}t + \frac{(-\frac{1}{r})(-\frac{1}{r} - 1)}{2}t^2 - \cdots).$$

We briefly describe the universal extension of $\overline{K}$. For more details see [19, Chapter 6]. Denote by $\overline{K}\{\text{hyp}\}$ the algebra over $\overline{K}$ generated by hypergeometric solutions. Let τ act naturally on

$\overline{K}\{\text{hyp}\}$. Then the polynomial ring $\overline{K}\{\text{hyp}\}[l]$ is a universal extension of $\overline{K}$, equipped with a τ-action following the rule

$$\tau(l) = l + t.$$

The valuation v extends to $\overline{K}[l]$ by setting $v(\sum_{i=0}^{n} a_i l^i) = \min\{v(a_i) : i = 0, 1, \ldots, n\}$.

Generalized exponents. This section will show how generalized exponents classify solutions of operators up to a factor of valuation 0. First consider the group

$$\{\text{hyp}(f) : f \in K_r^*\}/\{f \in K_r^* : v(f) = 0\}, \tag{4.2}$$

which classifies hypergeometric elements over K_r up to a factor of valuation 0. Applying the map $g \mapsto \frac{\tau(g)}{g}$, namely $\text{hyp}(f) \mapsto f$, the group (4.2) becomes

$$\mathcal{G}_r := K_r^*/K_{1,r},$$

where $K_{1,r} = \{\frac{\tau(f)}{f} : f \in K_r, v(f) = 0\}$.

Lemma 49. *Claim that*

$$K_{1,r} = \{g \in K_r^* : v(g - 1) > 1\}.$$

Proof. A straight-forward calculation shows that

$$\frac{\tau(f)}{f} = 1 - v(f)t + o(t). \tag{4.3}$$

Obviously when $v(f) = 0$ the right-hand side is $1 + o(t)$.

The other direction follows from the proof given in [6, Lemma 3.2.4]

□

Thus, $f, g \in K_r^*$ represent the same class in $\mathcal{G}_r$ when $v(\frac{f}{g} - 1) > 1$, which, by applying properties of valuations, happens if and only if $v(f - g) > v(f) + 1$. For

$$f = ct^s(1 + \sum_{i=1}^{\infty} a_i t^{\frac{i}{r}}) \in K_r^*, \quad \text{where } r < \infty,$$

denote

$$\text{Trunc}(f) = ct^s(1 + \sum_{i=1}^{r} a_i t^{\frac{i}{r}}) \in K_r^*.$$

Clearly Trunc : $K_\infty^* \to K_\infty^*$ is well-defined. Then $v(f - \text{Trunc}(f)) > v(f) + 1$, which means the image of f in $\mathcal{G}_r$ is represented by Trunc(f). Hence $\mathcal{G}_r$ can be identified with $E_r := \text{Trunc}(K_r^*)$, a set of representatives. For $r \leqslant \infty$,

$$\mathcal{G}_r = \{ct^s(1 + \sum_{i=1}^{r} a_i t^{\frac{i}{r}}) : c \in \mathbb{C}, s \in \frac{1}{r}\mathbb{Z}, a_i \in \mathbb{C}\}$$

and $\mathcal{G}_\infty = \bigcup_{r=1}^{\infty} \mathcal{G}_r$. The following short exact sequence of abelian groups is natural,

$$1 \to K_{1,r} \to K_r^* \xrightarrow{\text{Trunc}} \mathcal{G}_r \to 1,$$

where the group structure on E_r is given by

$$g_1 \circ g_2 = \text{Trunc}(g_1 g_2), \quad g_1, g_2 \in \mathcal{G}_r.$$

Denote

$$\overline{K}[l]_h = \{\text{hyp}(f)p : f \in \overline{K}, p \in \overline{K}[l] \setminus \{0\}\} \subseteq \overline{K}\{\text{hyp}\}[l]. \tag{4.4}$$

It is a multiplicative monoid. Using the properties of hypergeometric functions, $h \in \overline{K}[l]_h$ can be written uniquely in the form $h = \text{hyp}(g)p$, where $g \in E_\infty$ and $p \in \overline{K}[l]$ has valuation 0. Define gen$(h) = e$. Call gen(h) the *generalized exponent* of h, because gen : $\overline{K}[l]_h \to E_\infty$ is an extension of $v : \overline{K}[l] \to \mathbb{Q}$, where $(\mathbb{Q}, +)$ is embedded into $(\overline{K}[l]_h, \cdot)$ by $q \mapsto 1 + qt$, and hence a generalization of exponents. It is not hard to verify that gen preserves multiplication using the fact that hyp$(\frac{f}{\text{Trunc}(f)})$ lies in $\overline{K}$ and has valuation 0.

Generalized exponents of operators are defined in [6]. An order n operator has exactly n generalized exponents in E, counting with multiplicity. The relevant property for us is:

Proposition 50. *For a non-zero operator $L \in \overline{K}[\tau]$, $e \in E_\infty$ is a generalized exponent of L if and only if there exists a solution $h \in \overline{K}[l]_h$ with $e = \text{gen}(h)$.*

Definition 51. *Based on Proposition 50, say $e \in E$ is a* generalized exponent *of the system $\tau Y = MY$ if this system has a solution of the form* hyp$(e)S$ *where $S \in \overline{K}[l]^n$ and $v(S) = 0$.*

Remark 52 (Algorithms). For operators over $\mathbb{C}(x)$ we can quickly compute all generalized exponents with the program GeneralizedExponents in the LREtools package in Maple 2021. There is also an implementation for computing the unramified generalized exponents (i.e. those in $\mathcal{G}_1$) for systems over $\mathbb{C}(x) \subseteq K$. For this chapter, these suffice.

Remark 53. Denote $g = \mathrm{Trunc}(f)$ for $f \in \overline{K}^*$. Due to Lemma 49, $\mathrm{hyp}(\frac{f}{g})$ lies in $\overline{K}$ and has valuation 0. Hence if $\mathrm{hyp}(f)S$ is a hypergeometric solution of a system where $v(S) = 0$, then e is a generalized exponent.

Now we discuss the relation between hypergeometric solutions of systems over $\mathbb{C}(x)$ and their generalized exponents. Since $\mathbb{C}(x) \subseteq K$, $\mathrm{Trunc}(f) \in \mathcal{G}_1$ for $f \in \mathbb{C}(x)^*$. For a monic Laurent series

$$f = t^n + dt^{n+1} + \cdots \in K^*,$$

denote $\mathrm{slc}(f) = d$, where slc stands for *second leading coefficient.*

For a monic polynomial

$$A = x^n + dx^{n-1} + \cdots,$$

we have

$$\mathrm{Trunc}(A) = x^n + dx^{n-1} = t^{-\deg(A)}(1 + \mathrm{slc}(A)t),$$

and for a non-zero rational function $c\frac{A}{B}$ where A, B are monic polynomials, straight-forward computations show that

$$\mathrm{Trunc}(c\frac{A}{B}) = ct^{\deg(B)-\deg(A)}(1 + (\mathrm{slc}(A) - \mathrm{slc}(B))t).$$

The following lemma justifies the definition of $\mathcal{H}$ in Step BP2 (section 4.3).

Lemma 54. *Consider Equation (*SYS*). Suppose $A, B \in C[x]$ are monic. If there is a solution in the form* $\mathrm{hyp}(cA/B)P$ *where $P \in C(x)^n$ then the system has a generalized exponent $ct^s(1+dt) \in E_1$ that satisfies*

$$\deg(B) - \deg(A) = s, \quad \mathrm{slc}(A) - \mathrm{slc}(B) - d = v(P) \in \mathbb{Z}. \tag{4.5}$$

If we further require $P \in C[x]^n$ then the relations have a stronger form

$$\deg(B) - \deg(A) = s, \quad -\mathrm{slc}(A) + \mathrm{slc}(B) + d = -v(P) = \deg(P) \in \mathbb{N}. \tag{4.6}$$

We will refer to (4.5) *as the* weak compatibility relations *and* (4.6) *the* strong compatibility relations.

Proof. The proof is routine. Notice that

$$\mathrm{hyp}(c\frac{A}{B})P = \mathrm{hyp}(c\frac{A}{B})t^{v(P)}t^{-v(P)}P = \mathrm{hyp}(c\frac{A}{B}(1+t)^{-v(p)})t^{-v(P)}P,$$

where $v(t^{-v(P)}P) = 0$. By Remark 53 the existence of such a solution implies

$$\mathrm{Trunc}(c\frac{A}{B}) \circ \mathrm{Trunc}((1+t)^{-v(P)}) = ct^{\deg(B)-\deg(A)}(1 + (\mathrm{slc}(A) - \mathrm{slc}(B) - v(P))t)$$

is a generalized exponent in E_1. When $P \in C[x]^n$, $-v(P) = \deg(P) \in \mathbb{N}$. □

Example 55. Consider the same system as in Example 47. There are two unramified generalized exponents:

$$t^2(1+2t), \quad t^{-1}(1-4t).$$

Therefore $c = 1$. Eight A/B's in $\mathcal{S}(M)$ from Step 1 (Example 47) match $t^2(1+2t)$; none matches the other generalized exponent. ▲

4.3.3 Step BP3

In this step, for each potential $\lambda = cA/B$, compute polynomial solutions of $\lambda\tau(P) = MP$ using the algorithm given in [2].

Example 56 (Continued from Example 55). Our next step is to find all polynomial solutions of the system

$$\tau(P) = (c\frac{A}{B})^{-1}MP \tag{4.7}$$

for each $c\frac{A}{B}$. The algorithm finds the space of hypergeometric solutions is one dimensional over $\mathbb{Q}$, generated by the hypergeometric vector

$$\mathrm{hyp}(\frac{x+1}{x(x+2)(x+3)})\begin{pmatrix} (x+1)^3(x+2)x^2 \\ (x+2)(x+1)(-x-1) \\ (x+2)^2(x+1)^2 \\ -x^4-4x^3-3x^2+1 \\ -x-2 \\ (x+2)(x+3) \end{pmatrix}.$$

In fact, this solution is computed four times because four different $c\frac{A}{B}$'s all lead to it with seemingly different representations. If $\mathrm{hyp}(\lambda)P$ is a solution where $P \in C[x]^n$ and $\mathrm{hyp}(\lambda') = f\,\mathrm{hyp}(\lambda)$ for some $f \in C[x]$, then the algorithm will rediscover $\mathrm{hyp}(\lambda)P$ by computing $\mathrm{hyp}(\lambda')fP$. ▲

The issues (a),(b) in section 4.1, computing too many candidates and duplicate solutions, will be addressed in the next section.

4.4 Algorithm Version II

4.4.1 Type and local types

Definition 57 (Type)**.** *For $f_1, f_2 \in \mathbb{C}(x)^*$, say f_1 and f_2 have the same* type *if* $\operatorname{hyp}(f_1)\mathbb{C}(x) = \operatorname{hyp}(f_2)\mathbb{C}(x)$.

In the Step BP1 of Algorithm Version I, the set $\mathcal{S}$ may contain different $\frac{A}{B}$ with the same type. If we can select one single $\frac{A}{B}$ for each type, then repeated solutions are avoided. This can be achieved by bounding the *local types* of $\frac{A}{B}$.

Notation: For a prime polynomial $p \in C[x]$, denote $[p] := \{\tau^i(p) : i \in \mathbb{Z}\}$. Since p is irreducible, $[p]$ consists of all polynomials that are *shift equivalent* (Definition 31) to p.

We note that it is easy to detect if two irreducible polynomials are shift equivalent. It suffices to do so for two monic polynomials. When f, g are monic and $g(x+i) = f(x)$, we have $i\deg(g) + \operatorname{slc}(g) = \operatorname{slc}(f)$. Consequently, f, g are shift equivalent if and only if $\frac{\operatorname{slc}(f)-\operatorname{slc}(g)}{\deg(g)}$ is an integer and $g(x + \frac{\operatorname{slc}(f)-\operatorname{slc}(g)}{\deg(g)}) = f(x)$.

For a prime polynomial $p \in C[x]$, let $v_p : C(x) \to \mathbb{Z} \cup \{\infty\}$ be the p-valuation.

Definition 58 (Local Type)**.** *For a non-zero element $a \in C(x) \setminus \{0\}$ and a prime polynomial $p \in C[x]$, let*

$$g_p(a) = \sum_{k\in\mathbb{Z}} v_{\tau^k(p)}(a).$$

In other words, g_p is the sum of valuations with respect to all prime polynomials that are shift equivalent to p. We call $g_p(a)$ the local type *of a at $[p]$.*

The following theorem shows the relation between types and local types.

Theorem 59 ([21, Theorem 1])**.** *Suppose $c_1\frac{A_1}{B_1}, c_2\frac{A_2}{B_2} \in C(x)^*$ where $A_1, B_1, A_2, B_2 \in C[x]$ are monic and $c_1, c_2 \in C$. Then $c_1\frac{A_1}{B_1}$ and $c_2\frac{A_2}{B_2}$ have the same type if and only if*

- *$c_1 = c_2$, and*
- *$g_p(c_1\frac{A_1}{B_1}) = g_p(c_2\frac{A_2}{B_2})$ for any prime $p \in C[x]$.*

Let now $Y = \operatorname{hyp}(cA/B)P$ be once more a hypergeometric solution to Equation (SYS) where $c \in C^*$ and $A, B \in C[x]$ are monic. Then Theorem 46 yields $A \mid \operatorname{denom}(M^{-1})$ and $B \mid \operatorname{denom}(M)$. The first statement implies

$$0 \leqslant g_p(A) \leqslant g_p(\operatorname{denom}(M)^{-1})$$

for every prime polynomial $p \in C[x]$ while the second statement

$$0 \leqslant g_p(B) \leqslant g_p(\mathrm{denom}(M)).$$

Now we restate Theorem 46 in terms of local types.

Lemma 60. *For a hypergeometric solution $Y = \mathrm{hyp}(c\frac{A}{B})P$ of Equation (*SYS*) where $c \in C^*$ and $A, B \in C[x]$ are monic, we have*

$$-g_p(\mathrm{denom}(M)) \leqslant g_p(A/B) \leqslant g_p(\mathrm{denom}(M^{-1}))$$

for every prime $p \in C[x]$ where $\tau(\gamma)/\gamma = A/B$.

4.4.2 The algorithm

Theorem 59 and Lemma 60 lead to a second algorithm for computing hypergeometric solutions.

Algorithm: Version II

Input: $\tau Y = MY$, where $M \in \mathrm{GL}(n, C(x))$

Output: hypergeometric solutions of the system

Step BP1

- Compute $\mathrm{denom}(M)$ and $\mathrm{denom}(M^{-1})$.
- Factor $\mathrm{denom}(M)$ and $\mathrm{denom}(M^{-1})$. Say

$$\mathrm{denom}(M) = a_1 \prod_{i=1}^{m} p_i^{e_i}, \quad \mathrm{denom}(M^{-1}) = a_2 \prod_{i=1}^{m} p_i^{e_i},$$

where $a_1, a_2 \in C - \{0\}$ and p_i are monic irreducible polynomials.

- Sort $p_1, \ldots, p_m$ according to their shift equivalence classes. Suppose $[p_1], [p_2], \ldots, [p_l]$ are all the shift equivalence classes.
- For $i = 1, 2, \ldots, l$, calculate $g_{p_i}(\mathrm{denom}(M))$ and $p_{q_i}(\mathrm{denom}(M^{-1}))$.
- Return $\mathcal{S}_2 := \{\prod_{i=1}^{l} p_i^{f_i} : -g_{p_i}(\mathrm{denom}(M)) \leqslant f_i \leqslant g_{p_i}(\mathrm{denom}(M^{-1}))\}$.

Step BP2

- Compute

$$G := \{\text{unramified generalized exponents of the system}\}.$$

- Let

$$\mathcal{H}_2 := \{c\frac{A}{B} : cx^s(1 + dx^{-1}) \in G, \frac{A}{B} \in \mathcal{S}_2, \deg(A) - \deg(B) = s, d - \mathrm{slc}(\frac{A}{B}) \in \mathbb{Z}\}.$$

Step BP3

- Let $\mathrm{Sols} = \emptyset$.

- For each $c\frac{A}{B} \in \mathcal{H}_2$, solve the system $\tau P = c^{-1}\frac{B}{A}MP$ for (a basis of) *rational* solutions using the algorithm introduced in [2]. Add $\operatorname{hyp}(c\frac{A}{B})P$ to Sols for any rational solution P.
- Return Sols as the output.

Remark 61. Notice that the Step BP2 and Step BP3 are slightly different from the same steps in Version I. The reason is, if $\operatorname{hyp}(cA/B)P$ is a hypergeometric solution in the standard form, in Step BP1 Version II A/B is computed up to its type, since local types do not distinguish different functions of the same type. To be more precise, there exists a unique $A'/B' \in \mathcal{S}_2$ such that A'/B' has the same type as A/B, and the solution will be discovered is in the form $\operatorname{hyp}(c\frac{A'}{B'})P'$, where $P' = \operatorname{hyp}(A/B)\operatorname{hyp}(A'/B')^{-1}P$ is not guaranteed to have polynomial entries. Hence in Step BP2 the weak compatibility relations (4.5) are applied instead of the strong and in Step BP3 we compute rational solutions.

Example 62. Let M be the same matrix as in Example 47. Recall that $\operatorname{denom}(M) = x(x+2)^2(x+3)$ and $\operatorname{denom}(M^{-1}) = (x+1)^2(x+2)$. All factors of $\operatorname{denom}(M)$ and $\operatorname{denom}(M^{-1})$ are shift equivalent to x, and

$$g_x(\operatorname{denom}(M)) = 4, \quad g_x(\operatorname{denom}(M^{-1})) = 3.$$

Therefore,

$$\mathcal{S}_2(M) = \{x^i : -4 \leqslant i \leqslant 3\}.$$

The cardinality of $\mathcal{S}_2(M)$ is 8, much less than that of $\mathcal{S}_1(M)$(Example 47).

In Example 55 we calculated the unramified generalized exponents of the system:

$$t^2(1+2t), \quad t^{-1}(1-4t).$$

The elements in $\mathcal{S}_2(M)$ that match one of them are x^{-2}, x and hence $\mathcal{H}_2(M) = \{x^{-2}, x\}$. We cannot discard the generalized exponent $t^{-1}(1-4t)$ because it is not ruled out by the weak compatibility relations (4.5). Despite that, in this version we have a significantly shorter list of cA/B. ▲

4.4.3 Discussion

Due to the fact that each candidate for $\frac{A}{B}$ has a distinct type, this version has the following advantages over Version I:

- there are less candidates for $\frac{A}{B}$ in step BP1;
- no duplicate solutions will be produced.

However, as pointed out in Remark 61, if $cA/B \in \mathcal{H}_2$, where $\mathcal{H}_2$ is defined in Step BP2, leads to a solution $\text{hyp}(cA/B)P$, P may not have polynomial entries. We will see in the next version A, B can be chosen in a way to guarantee $P \in C[x]^n$.

4.5 Algorithm Version III

In Algorithm Version II we are able to greatly reduce the size of the list of $c\frac{A}{B}$ by avoiding repeated types. This section is devoted to the technical result that we can further improve the algorithm by choosing A/B in a way such that in Step BP3 we only need to search for polynomial solutions instead of rational ones. The idea is to make sure A/B satisfies the conditions in Theorem 46 and at the same time minimize $\text{slc}(A) - \text{slc}(B)$. A few notations are needed before stating the main result of this section.

Suppose $r_1, r_2 \in C(x)$ are of the same type. Say r_1 is *smaller* than r_2, denoted by $r_1 \preccurlyeq r_2$, if $\frac{\text{hyp}(r_2)}{\text{hyp}(r_1)} \in C[x]$. It is easy to see that $\preccurlyeq$ is a partial order (on the set of all non-zero rational functions or those that are of the same type). Theorem 46 states that every hypergeometric solution can be written as $\text{hyp}(c\frac{A}{B})P$ where $P \in C[x]^n$. If $\text{hyp}(c'\frac{A'}{B'})P' = \text{hyp}(c\frac{A}{B})P$ and $c'\frac{A'}{B'} \preccurlyeq c\frac{A}{B}$, then a consequence is $P' \in C[x]^n$. Thus, to achieve the goal, we want $\frac{A}{B}$ to be as small as possible. In general, there is no smallest rational function for a type, but with the restriction that $A \mid \text{denom}(M)$ and $B \mid \text{denom}(M^{-1})$, the smallest element does exist.

Recall that

$$\mathcal{S}(M) = \{\frac{A}{B} : A, B \in C[x] \text{ monic, } A \mid \text{denom}(M^{-1}), B \mid \text{denom}(M)\}$$

contains all potential $\frac{A}{B}$ in Algorithm Version I (section 4.3). Consider the partition $\mathcal{S}(M) = \bigcup_i \mathcal{T}_i(M)$ of $\mathcal{S}(M)$, where each $\mathcal{T}_i(M)$ consists of all elements of a particular type.

Theorem 63. *Suppose $s_i \in \mathcal{T}_i(M)$ has the smallest slc in $\mathcal{T}_i(M)$. Namely $\text{slc}(s_i) = \min\{\text{slc}(r) : r \in \mathcal{T}_i(M)\}$. Claim that s_i is the smallest element in $\mathcal{T}_i(M)$.*

Proof. All elements in $\mathcal{T}_i(M)$ have the same local types. Hence the local types of $\mathcal{T}_i(M)$ are well-defined. Suppose $\mathcal{T}_i$ at $[p_j]$ has non-zero local types $e_1, e_2, \ldots, e_N$ at $[p_1], [p_2], \ldots, [p_N]$, respectively, where $p_1, p_2, \ldots, p_N$ are mutually shift non-equivalent. We first consider the case where $\mathcal{T}_i(M)$ has only one single non-zero local type. Suppose p is a prime polynomial, at which the local type of $\mathcal{T}_i$ does not vanish.

Suppose p is a prime polynomial which divides either $\mathrm{denom}(M)$ or $\mathrm{denom}(M^{-1})$. Denote $p_j(x) = p(x+j)$ for any $j \in \mathbb{Z}$. There exists

Assume s_i is not the smallest element in $\mathcal{T}_i$. Then there exists $r \in \mathcal{T}_i$ such that $\mathrm{hyp}(\frac{r}{s_i}) \notin C[x]$. Therefore there exists an irreducible polynomial $p \in C[x]$ such that

$$\mathrm{hyp}(\frac{r}{s_i}) = \frac{f}{pg},$$

where $f, g \in C[x]$ and $\gcd(f, pg) = 1$. Then $s_i \frac{p}{\tau(p)} \in \mathcal{T}_i$and

$$\mathrm{slc}(s_i \frac{p}{\tau(p)}) = \mathrm{slc}(s_i) + \mathrm{slc}(p) - \mathrm{slc}(\tau(p)) = \mathrm{slc}(s_i) - \deg(p),$$

which contradicts to the assumption that s_i has the smallest slc among elements in $\mathcal{T}_i$.

Due to the anti-symmetry of $\preccurlyeq$, the smallest element is unique. □

We remark that if $ct^s(1+dt)$ is the unramified generalized exponent with the largest d that is compatible with $\mathrm{hyp}(c\frac{A}{B})P$ where $P \in C[x]^n$, then $d - \mathrm{slc}(A/B)$ is a degree bound for P due to the relation $d = \mathrm{slc}(A/B) + \deg(P)$. With a degree bound the problem of finding polynomial solutions reduces to solving a system of linear equations.

Example 64. Let M be the same as in Example 47. The unramified generalized exponents are:

$$t^2(1+2t), \quad t^{-1}(1-4t).$$

There are eight types of elements in $\mathcal{S}(M)$, two of them matching the generalized exponents, as was stated in Example 62. The smallest elements (in $\mathcal{S}(M)$) of these two types are

$$s_1 = \frac{x+1}{(x+2)^2(x+3)} \quad s_2 = \frac{(x+1)^2}{x+3}.$$

The set of candidates for $c\frac{A}{B}$ is $\{\frac{x+1}{(x+2)^2(x+3)}, \frac{(x+1)^2}{x+3}\}$. The algorithm in [2] finds the following polynomial solution for P in $\tau(P) = \mathrm{hyp}(s_1)MP$ (and no polynomial solution for $\tau(P) = \mathrm{hyp}(s_2)MP$):

$$\begin{pmatrix} x^3(x+2)(x+1)^4 \\ -x(x+2)(x+1)^3 \\ x(x+2)^2(x+1)^3 \\ -x(x+1)(x^4+4x^3+3x^2-1) \\ -x(x+2)(x+1) \\ x(x+3)(x+2)(x+1)) \end{pmatrix}$$

▲

4.6 Application: Beke-Bronstein Algorithm

The Beke-Bronstein algorithm ([10] and [5]) for factoring differential operators partially applies to polynomials and difference operators as well. In the recurrence case, the factorization problem converts into solving a recurrence system for hypergeometric solutions. With the algorithms introduced in this chapter, we have a complete algorithm for factoring recurrence operators. In the following we reformulate the Beke-Bronstein approach in the language of exterior algebra first for the polynomial case and then the difference case. We skip some details on the latter since it works almost the same for both cases.

4.6.1 Polynomial case

Let $f \in \mathbb{Q}[y]$ be the polynomial that is to be factored. Assume $y \nmid f$ throughout this section. If g is a factor of f then so is ag where $a \in \mathbb{Q}^*$. It is unnecessary to distinguish g and ag as factors of f. Thus we consider factors of f as elements in $\mathbb{P}(\mathbb{Q}[x])$. Introduce the notation $[g]$ for the element in $\mathbb{P}(\mathbb{Q}[x])$ with $g \in \mathbb{Q}[x] - \{0\}$ being its representative. Call $[g]$ the *projective class* of g and a *projective factor* of f if $g \mid f$. Let $\mathrm{Fact}_m(f) = \{[g] : g \mid f, \deg(g) = m\}$ for $m \in \mathbb{Z}_+$. Namely $\mathrm{Fact}_m(f)$ is the set of degree-m projective factors. It is worth noting that $\mathrm{Fact}_m(f)$ does not store the information of multiplicities of factors.

Let $M = \mathbb{Q}[y]/(f)$. Then M is a $\mathbb{Q}[y]$-module. For $g \mid f$ with $\deg(g) = m$, let $\mu(g) = g \wedge yg \wedge \cdots \wedge y^{n-m-1}g \in \bigwedge^{n-m} M$. Suppose $g = \sum_{i=0}^{m} b_i y^i$.

Lemma 65. *The coordinate of $\mu(g)$ with respect to $y^i \wedge y^{m+1} \wedge y^{m+2} \wedge \cdots \wedge y^{n-1}$ is $b_i(b_m)^{n-m-1}$.*

Proof. It is a straight-forward calculation. □

A direct consequence is, under the assumption that $y \nmid f$, $\mu(g) \neq 0$. For $a \in \mathbb{Q}^*$, we have $\mu(ag) = a^{n-m}\mu(g) \neq 0$. Thus

$$\mu([g]) := \mathbb{Q}\mu(g) \subset \bigwedge^{n-m} M,$$

is well-defined since $\mu([g])$ does not depend on the choice of representatives. Identify $\mu([g])$ with a point in $\mathbb{P}(\bigwedge^{n-m} M)$, since it is a one-dimensional subspace. Lemma 65 also implies once the expression of $\mu([g])$ under the standard basis is given then we are able to rebuild $[g]$ easily. In other words, there exists a map $\eta : \mathbb{P}(\bigwedge^{n-m} M) \to \mathbb{P}(\mathbb{Q}[y])$ induced by the projection map from $\bigwedge^{n-m} M$ to the subspace $\mathrm{span}_{\mathbb{Q}}(y^i \wedge y^{m+1} \wedge \cdots \wedge y^{n-1} : i = 0, \ldots, m)$ such that $\eta \circ \mu$ is the identity map on $\mathrm{Fact}_m(f)$.

Next we show that we are able to compute $\mu(\mathrm{Fact}_m(f))$.

Theorem 66. *Let*

$$E_m(f) := \{\textit{1-dimensional subspaces of } \bigwedge^{n-m} M \textit{ that are } \mathbb{Q}[y]\textit{-modules}\}.$$

Claim that

$$\mu(\mathrm{Fact}_m(f)) = E_m(f).$$

Proof. Show that

$$\mu_1 : \mathrm{Fact}_m(f) \to \{m\text{-dimensional subspace of } M \text{ that are } Q[y]\text{-module}\}$$

and

$$\mu_2 : \{(n-m)\text{-dimensional subspace of } M \text{ that are } Q[y]\text{-module}\} \to E_m(f)$$

are bijections and μ is the composition. □

By the proposition $\mu(\mathrm{Fact}_m(f))$ is in fact the collection of 1-dimensional eigensapces of y as a linear map over $\bigwedge^{n-m} M$. The algorithm of finding linear factors allows us to compute eigenvalues and consequently eigenspaces.

Example 67. Suppose $f = y^4 + 2y^3 + 3y^2 + 2y + 2$. Now the goal is to find factors of f of degree 2, since using the algorithm for computing linear factors we confirm that there is no first degree factor of f. In the module $M = \mathbb{Q}[y]/(f)$ the relation

$$y^4 = -2y^3 - 3y^2 - 2y - 2$$

holds. The action of y on $\bigwedge^2 M$ is given by

$$\begin{aligned}
1 \wedge y^3 &\mapsto y \wedge y^4 = y \wedge (-2y^3 - 3y^2 - 2y - 2)\\
y \wedge y^3 &\mapsto y^2 \wedge y^4 = y^2 \wedge (-2y^3 - 3y^2 - 2y - 2)\\
y^2 \wedge y^3 &\mapsto y^3 \wedge y^4 = y^3 \wedge (-2y^3 - 3y^2 - 2y - 2)\\
1 \wedge y &\mapsto y \wedge y^2\\
1 \wedge y^2 &\mapsto y \wedge y^3\\
y \wedge y^2 &\mapsto y^2 \wedge y^3
\end{aligned}$$

which can be represented by the matrix

$$\begin{pmatrix} 0 & -2 & 0 & 2 & 0 & -3 \\ 0 & 0 & -2 & 0 & 2 & 2 \\ 2 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

The eigenvalues and eigenvectors are

$$(2, \begin{pmatrix} -1 \\ 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix}), \quad (1, \begin{pmatrix} 1 \\ 1 \\ 0.5 \\ 2 \\ 2 \\ 1 \end{pmatrix}).$$

Next we verify whether the eigenvectors satisfy the Plücker relations. In this case the Plücker relations are one single polynomial

$$X_{0,1}X_{2,3} - X_{0,2}X_{1,3} + X_{0,3}X_{1,2},$$

where $X_{i,j}$ is the coordinate corresponding to the basis vector $y^i \wedge y^j$. A straight-forward calculation shows that both eigenvectors are solutions of the Plücker relations. To obtain factors of f, we take the components of the eigenvectors corresponding to $1 \wedge y^3, y \wedge y^3, y^2 \wedge y^3$:

$$-1 \cdot 1 \wedge y^3 - 1 \cdot y^2 \wedge y^3, \quad 1 \cdot 1 \wedge y^3 + 1 \cdot y \wedge y^3 + \frac{1}{2} y^2 \wedge y^3,$$

which lead to factors

$$-1 - y^2, 1 + y + \frac{1}{2} y^2.$$

▲

4.6.2 Difference case

Notice that $D = \mathbb{Q}(x)[\tau]$ is a vector space over $\mathbb{Q}(x)$. Let $[R] \in \mathbb{P}(D)$ be the equivalence class of R. Similar to the polynomial case, call $[R]$ the *projective class* of R and a *projective factor* of L if R is a right-hand factor of L. Let $\mathrm{Fact}_m(L) := \{[R] : R \text{ is an order-}m \text{ right-hand factor of } L\}$. Denote $M = D/DL$. Define

$$\mu : \mathrm{Fact}_m(f) \to \bigwedge^{n-m} M,$$

$$[R] \mapsto \mathbb{Q}(x)R \wedge \tau R \wedge \cdots \wedge \tau^{n-m-1} R.$$

Theorem 68. *Let*

$$E_m(L) := \{1\text{-}dimensional\ \mathbb{Q}(x)\text{-}subspaces\ of\ \bigwedge^{n-m} M\ that\ are\ D\text{-}modules\}.$$

Claim that $\mu(\mathrm{Fact}_m(L)) = E_m(L)$.

Proof. Similar to Theorem 66. □

Similar to the polynomial case, the problem of factorization converts to that of finding D-submodules of $\bigwedge^{n-m} M$ that are 1-dimensional over $\mathbb{Q}(x)$, This is in fact equivalent to solving a difference system for hypergeometric solutions.

Example 69. Let

$$\begin{aligned}L = \tau^4 + \frac{x^4+2x^3+6x^2-3x-18}{x^3+5x-6}\tau^3 - \frac{x^4+5x^3+14x^2+28x-12}{x^3+5x-6}\tau^2 \\ - \frac{x^5+3x^4+8x^3+3x^2-21x-18}{x^3+5x-6}\tau + \frac{3x^2(x^2+3x+8)}{x^3+5x-6}.\end{aligned}$$

To find order-2 factors of L, consider the D-module $\bigwedge^2 D/DL$. The action of τ on $\bigwedge^2 D/DL$ with respect to the basis

$$e_0 = 1 \wedge \tau^3, e_1 = \tau \wedge \tau^3, e_2 = \tau^2 \wedge \tau^3, e_3 = 1 \wedge \tau, e_4 = 1 \wedge \tau^2, e_5 = \tau \wedge \tau^2,$$

is given by

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ r_1 & 0 & r_2 & 0 & r_3 & 0 \\ 0 & r_4 & r_5 & 0 & 0 & r_6 \\ 0 & 0 & 0 & r_7 & r_8 & r_9 \end{pmatrix}$$

where

$$r_1 = \frac{3x^2(x^2+3x+8)}{x^3+5x-6}, \quad r_2 = \frac{x^4+5x^3+14x^2+28x-12}{x^3+5x-6}, \quad r_3 = -\frac{x^4+2x^3+6x^2-3x-18}{x^3+5x-6},$$

$$r_4 = \frac{3x^2(x^2+3x+8)}{x^3+5x-6}, \quad r_5 = -\frac{x^5+3x^4+8x^3+3x^2-21x-18}{x^3+5x-6},$$

$$r_6 = -\frac{x^4+2x^3+6x^2-3x-18}{x^3+5x-6}, \quad r_7 = \frac{3x^2(x^2+3x+8)}{x^3+5x-6},$$

$$r_8 = -\frac{x^5+3x^4+8x^3+3x^2-21x-18}{x^3+5x-6}, \quad r_9 = -\frac{x^4+5x^3+14x^2+28x-12}{x^3+5x-6}.$$

The hypergeometric vectors are

$$C_1(1, 0, -x, x+1, 0, x(x+1))^\intercal, \quad C_2(1, -x, -3, x^2+x+3, 3x+3, 9)^\intercal,$$

where C_1, C_2 are non-zero constants. They both satisfy Plücker relations so there are two essentially different factors of L:

$$\tau^2 - x, \quad \tau^2 - x\tau - 3.$$

▲

A difference system of order $\binom{n}{m}$ has to be solved in the course of computing order-m factors of an order-n operator. The size of the system can be large when m, n are small. Thus we want to have an algorithm that factors difference operators without solving a system of a big size.

CHAPTER 5

HEURISTIC FACTORIZER

In the Beke-Bronstein approach, when looking for order-m factors of an order-n operator, a differential/difference system of order $\binom{n}{m}$ needs to be solved. When m, n are not small, $\binom{n}{m}$ can be so large that solving a difference system of such an order is beyond the capacity for a normal PC. This motivates us to look for factorization algorithms without solving a system of a big size.

For a polynomial, the minimal polynomial of a root is either a non-trivial factor or the polynomial itself, which proves its irreducibility. In the difference case, however, the *minimal operators* (the difference analog of minimal polynomials) of most solutions of a recurrence operator are the given operator itself even when there are non-trivial factors. Hence just picking a random solution does not help. In this chapter the heuristic factorizer is presented, where certain solutions, which are likely to yield non-trivial factors, are constructed. The algorithm is fast since it avoids solving a system of a high rank.

In this entire chapter we mainly work with the difference field $\mathbb{C}(x)$. Let V be its universal extension (Definition 18) and $D = \mathbb{C}(x)[\tau]$ the ring of recurrence operators.

5.1 Heuristic Factorizer

Suppose f is a polynomial and a a root of f. Then the minimal polynomial of a is an irreducible factor of f. This fact is used in some algorithms for factoring polynomials. Similarly, in the difference case we can also define *minimal operator* of a solution: for $s \in V - \{0\}$, call R a minimal operator of s if $R(s) = 0$ and R has the smallest order among all non-zero annihilators of s. Equivalently, DR is the left ideal of annihilators of s. Solutions and minimal operators may yield algorithms for factoring difference operators without solving sizable systems. However, a major difficulty is that not all solutions lead to interesting results. Call s a *special solution* of L if L is an annihilator but not a minimal operator of s. Most solutions of a difference operator are not special. In this section we introduce the heuristic factorizer, in which certain solutions are constructed and it turns out the constructed solutions often contain special solutions.

5.1.1 Sequences and extension

Traditionally, a (infinite) sequence is a function whose domain is $\mathbb{N}$. Here we call a $\mathbb{C}$-valued function a *sequence* if its domain is $p+\mathbb{N}, p-\mathbb{N}$ or $p+\mathbb{Z}$ for some $p \in \mathbb{C}$. Furthermore, if the domain is $p+\mathbb{N}$ then call it a *right sequence*,if $p-\mathbb{N}$ then a *left sequence* and if $q+\mathbb{Z}$ then a *two-sided sequence*.

The sets $\mathbb{C}^{p+\mathbb{N}}, \mathbb{C}^{p-\mathbb{N}}, \mathbb{C}^{p+\mathbb{Z}}$ of sequences are $\mathbb{C}$-algebras, where addition and multiplication are defined pointwise.

Suppose $L = \sum_{i=0}^{n} a_i(x)\tau^i \in \mathbb{C}[x][\tau]$ where $a_n \neq 0$. Then L induces the following $\mathbb{C}$-linear maps for any $p \in \mathbb{C}$:

$$\begin{gathered} L : \mathbb{C}^{p+\mathbb{N}} \to \mathbb{C}^{p+\mathbb{N}}, \\ f(x) \mapsto \sum_{i=0}^{n} a_i(x) f(x+i); \\ L : \mathbb{C}^{p-\mathbb{N}} \to \mathbb{C}^{p-n-\mathbb{N}}, \\ f(x) \mapsto \sum_{i=0}^{n} a_i(x) f(x+i). \end{gathered}$$

If f is in the null space of L then say f is a *sequence solution* of L. If f is a solution of some non-zero operator then call f *recursive*.

Example 70 (Fibonacci Sequence)*.* The Fibonacci sequence

$$F(1) = 1, \quad F(2) = 1, \quad F(3) = 2, \quad \cdots$$

is recursive, since

$$(\tau^2 - \tau - 1)(F(x)) = F(x+2) - F(x+1) - F(x) = 0.$$

▲

If f is a sequence solution of L with $\mathrm{ord}(L) = n$ and n consecutive values of f are known, then in most cases (Example 71) we can *extend right* (extend towards ∞) and *extend left* (extend towards $-\infty$). However, there are cases (Example 72) where extending left or right cannot be done.

Example 71*.* Suppose $f \in \mathbb{C}^{\mathbb{Z}}$ is a solution of $L = 2\tau^2 - (2x+3)\tau + (2x-1)$ with $f(0) = 0, f(1) = 1$. Then f satisfies the equation

$$2f(x+2) - (2x+3)f(x+1) + (2x-1)f(x) = 0. \tag{5.1}$$

Plugging in $x = 0$ we see that

$$f(2) = \frac{1}{2}((2 \cdot 0 + 3)f(1) - (2 \cdot 0 - 1)f(0)) = \frac{3}{2}.$$

In fact, Equation 5.1 implies for integer $x > 1$, $f(x)$ is determined by $f(x-1), f(x-2)$. Then by induction we can compute $f(x)$ for all positive integer x. This process is right extension of f.

Similarly, $f(x)$ can be extended left (computed for all negative integer x). Here we give $f(-1)$ as an example and omit the rest:

$$f(-1) = \frac{-2f(1) + (-2+3)f(0)}{-2-1} = \frac{2}{3}.$$

▲

Example 72. Suppose $f \in \mathbb{C}^{1/2+\mathbb{Z}}$ is a solution of $L = 2\tau^2 - (2x+3)\tau + (2x-1)$ with $f(3/2) = 1, f(5/2) = 2$. To extend f to left we plug $x = 1/2$ into the equation

$$2f(x+2) - (2x+3)f(x+1) + (2x-1)f(x) = 0,$$

to obtain that

$$2 \cdot 2 - 4 \cdot 1 + 0 \cdot f(1/2) = 0,$$

which does not tell us anything about $f(1/2)$.

▲

5.1.2 Constructing solutions

Construction (Heuristic special solution algorithm). *Suppose $L = \sum_{i=0}^{n} a_i\tau^i$ where $a_n a_0 \neq 0$. Suppose q is the largest root of $a_n(x)$ in $q + \mathbb{Z}$. Construct a right sequence solution $u : q + \mathbb{Z} \to \mathbb{C}$ of L whose terms in $[q, q+n]$ are*

$$u(q) = 0, u(q+1) = 0, \ldots, u(q+n-1) = 0, u(q+n) = 1.$$

Clearly, u satisfies the recurrence relation $Lu = 0$ in the interval $[q, q+n]$. Since $q+n$ is the largest root of a_n in $q+\mathbb{Z}$, it can be extended into a right sequence solution of L. Call u a candidate-special solution.

Observation: if L is reducible, then the candidate-special solution is often a special solution of L.

Example 73. Let $L = (2x^2 + x - 1)\tau^2 + (-3x + 1)\tau + 1$. It corresponds to the recurrence relation

$$(2x^2 + x - 1)f(x + 2) + (-3x + 1)f(x + 1) + f(x) = 0.$$

Notice that $f(x + 2)$ is almost always determined by $f(x)$ and $f(x + 1)$. The only exceptions are the roots of $2x^2 + x - 1$. For instance, plugging $x = -1$ into the recurrence equation leads to the relation

$$0f(1) - 2f(0) + f(-1) = 0,$$

which implies that $f(1)$ is not determined by $f(0)$ and $f(-1)$.

Let $u : -1 + \mathbb{Z}_+ \to \mathbb{C}$ be a sequence such that $\tau(u) = 0$ and $u(x) = 0$ for $x < 1$. Without loss of generality, let $u(1) = 1$. Then the values of u for $x > 1$ are determined recursively.

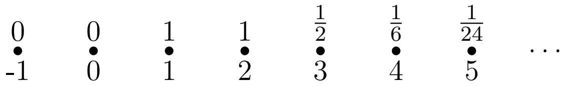

Figure 5.1: First few terms of u

Such defined u is a solution of the operator $x\tau - 1$, which turns out to be a right-hand factor of L. ▲

The construction does not always lead to all right-hand factors of the input operator. We use the theory of *valuation growths*, which is first introduced in [21], as a tool to study when it works and what kind of factors can be found this way.

5.2 Valuation Growths

5.2.1 Germs of sequences

Pre-universal extensions (Definition 18) of $\mathbb{C}(x)$ can be constructed out of spaces of sequences $\mathbb{C}^{p+\mathbb{N}}$ and $\mathbb{C}^{p-\mathbb{N}}$ where $p \in \mathbb{C}$. In the following we focus on $\mathbb{C}^{\mathbb{N}}$ and it works the same way in other cases.

Say $f, g \in \mathbb{C}^{\mathbb{N}}$ are *almost everywhere equal* if $f(x) = g(x)$ holds for all but finitely many x. Denote this by $f \overset{a.e.}{=\!=} g$. It is easy to see $\overset{a.e.}{=\!=}$ is an equivalence relation and the equivalence class containing 0 is an ideal. Denote by $[f]_{a.e.}$ the equivalence class of f and let V_0^+ be the quotient ring $\mathbb{C}^{\mathbb{N}}/[0]_{a.e.}$. Call it the *germ* of f. The action of τ on $\mathbb{C}^{\mathbb{N}}$ induces an action on V_0^+ by

$$\tau : [(a_0, a_1, a_2, \ldots)]_{a.e.} \mapsto [(a_1, a_2, \cdots)]_{a.e.}.$$

The inverse of τ on V_0^+ does exist and is given by

$$\tau^{-1} : [(a_0, a_1, a_2, \ldots)]_{a.e.} \mapsto [(a, a_0, a_1, \cdots)]_{a.e.}$$

where a can be any complex number. Then τ is an automorphism of V_0^+.

Claim that (V_0^+, τ) is a difference extension of $(\mathbb{C}(x), \tau)$ with the inclusion map

$$\mathbb{C}(x) \to V_0^+,$$

$$f(x) \mapsto [(f(0), f(1), f(2), \cdots)]_{a.e.},$$

where $[\cdot]_{a.e.}$ takes care of finitely many undefined values (or poles) of f. We need to show that $\frac{f_1}{g_1} \overset{a.e.}{=\!=} \frac{f_2}{g_2}$ for $f_1, g_1, f_2, g_2 \in \mathbb{C}[x]$ and $g_1, g_2 \neq 0$ implies $\frac{f_1}{g_1} = \frac{f_2}{g_2}$. In fact, when $\frac{f_1}{g_1} \overset{a.e.}{=\!=} \frac{f_2}{g_2}$, we have $f_1 g_2 \overset{a.e.}{=\!=} f_2 g_1$, and the almost everywhere equality can be replaced by an equality, since two polynomials in $\mathbb{C}[x]$ are equal if and only if they agree at infinitely many points. This proves (V_0^+, τ) is a difference extension of $(\mathbb{C}(x), \tau)$.

For $L \in \mathbb{C}(x)[\tau]$, denote by $V_0^+(L)$ the solution space of L in V_0^+. To verify that V_0^+ is a pre-universal extension, we need to show $\operatorname{ord}(L) = \dim_{\mathbb{C}}(V_0^+(L))$. In fact, for any $[u] \in V_0^+(L)$, there exists a sufficiently large integer N such that $L(u(x)) = 0$ for $x > N$, and hence $[u]$ is determined by its values at $N + 1, N + 2, \cdots, N + \operatorname{ord}(L)$.

We can also define V_q^+ and V_q^- (using sequences in $\mathbb{C}^{q-\mathbb{N}}$) for any $q \in \mathbb{C}$ in the same way. For $q_1, q_2 \in \mathbb{C}$ such that $q_1 - q_2 \in \mathbb{Z}$, we can identify $V_{q_1}^+$ with $V_{q_2}^+$, and $V_{q_1}^-$ with $V_{q_2}^-$. Hence V_p^+ and V_p^- are well-defined for $p \in \mathbb{C}/\mathbb{Z}$.

5.2.2 Shift singularities and valuation growths

As is seen in Example 72, a left solution of L cannot always be extended to a right solution. If the extension fails at the point q, we call q a *problem point* and $p = q + \mathbb{Z} \in \mathbb{C}/\mathbb{Z}$ a *shift singularity* of L.

Example 74. Let $L = (2x + 1)\tau^2 - 2x\tau + 1$ and $p = \frac{1}{2} + \mathbb{Z}$. Suppose u is a sequence with $u(1/2) = 1$ and $u(3/2) = 1$. A division by zero issue arises at $q = 2.5$ when extending u to right (Figure 5.2). The point $q = 2.5$ is a problem point, and $p = \frac{1}{2} + \mathbb{Z}$ is a shift singularity of L.

▲

In order to extend a left (right) sequence solution into a right (left) one, informally speaking, we can make a perturbation to the domain of a solution such that division by zero can be avoided.

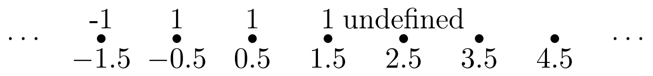

Figure 5.2: Extension fails due to division by zero

Let ϵ be a new constant that is transcendental over $\mathbb{C}(x)$ to obtain a difference extension $\mathbb{C}(x, \epsilon)$ of $\mathbb{C}(x)$.

Consider the map

$$\imath : D \to \mathbb{C}(x, \epsilon)[\tau],$$
$$\sum_{i=0}^{n} a_i(x)\tau^i \mapsto \sum_{i=0}^{n} a_i(x+\epsilon)\tau^i.$$

It preserves multiplication since

$$\tau \cdot (x+\epsilon) - (x+\epsilon)\tau = \tau$$

and hence is a ring homomorphism. In addition it is clearly injective and therefore an embedding.

Let D_ϵ be the image of D under $\imath$. Consider $\mathbb{C}(\epsilon)^{q+\mathbb{Z}}$, the space of $\mathbb{C}(\epsilon)$-valued sequences on $q + \mathbb{Z}$. The action of D_ϵ on $\mathbb{C}(\epsilon)^{q+\mathbb{Z}}$ is given by

$$x + \epsilon : u(x) \mapsto (x + \epsilon)u(x), \quad x \in q + \mathbb{Z}$$

$$\tau : u(x) \mapsto u(x + 1), \quad x \in q + \mathbb{Z}.$$

Different from the action of D on sequences, $\mathbb{C}(\epsilon)^{q+\mathbb{Z}}$ is a well-defined D_ϵ-module. Introducing ϵ eliminates the division by zero issue, since for any $a(x) \in \mathbb{C}(x)^*$ and $x \in \mathbb{C}$, $a(x+\epsilon)$ is always a non-zero rational function in ϵ. Therefore, if $u \in \mathbb{C}(\epsilon)^{q+\mathbb{Z}}$ is a solution of L_ϵ of order n, with its initial values at n consecutive points, we are able to compute any other term by the recurrence relation without any division by zero issue. For this reason the solution space of L_ϵ is n-dimensional over $\mathbb{C}(\epsilon)$. Denote by $V_q(L_\epsilon)$ $V_{q+\mathbb{Z}}(L_\epsilon)$ the solution space of L_ϵ in $\mathbb{C}(\epsilon)^{q+\mathbb{Z}}$. Denote by v_ϵ the ϵ-valuation on $\mathbb{C}(\epsilon)$.

Definition 75. *For a non-zero $\tilde{u} \in V_p(L_\epsilon)$, define its* left valuation *to be*

$$v_{\epsilon,l}(\tilde{u}) = \liminf_{m\to-\infty} v_\epsilon(\tilde{u}(m)),$$

and its right valuation

$$v_{\epsilon,r}(\tilde{u}) = \liminf_{m\to\infty} v_\epsilon(\tilde{u}(m)),$$

and the valuation growth

$$g_{p,\epsilon}(\tilde{u}) = v_{\epsilon,r}(\tilde{u}) - v_{\epsilon,l}(\tilde{u}).$$

For $L \in D$, define the set of valuation growths *of L at p to be the collection of valuation growths of solutions of L_ϵ, namely*

$$\overline{g}_p(L) = \{g_{p,\epsilon}(\tilde{u}) | \tilde{u} \in V_p(L_\epsilon), \tilde{u} \neq 0\}.$$

Let $g_{p,r}, g_{p,l}$ be the maximal and minimal valuation growths of L at p, respectively.

Denote

$$\mathcal{R}_{p,m} = \{\tilde{u} \in V_p(L_\epsilon) | v_{\epsilon,r}(\tilde{u}) \geqslant m\},$$

and

$$\mathcal{L}_{p,m} = \{\tilde{u} \in V_p(L_\epsilon) | v_{\epsilon,l}(\tilde{u}) \geqslant m\},$$

where m is an integer.

Lemma 76. *For each integer m, $\mathcal{L}_{p,m}$ is a subspace of $\mathcal{R}_{p,m+g_{p,r}}$ and $\mathcal{R}_{p,m}$ is a subspace of $\mathcal{L}_{p,m+g_{p,l}}$.*

Proof. Directly follows from the definitions. □

Lemma 77. *As $\mathbb{C}$-vector spaces, $\mathcal{R}_{p,m}(L)/\mathcal{R}_{p,m+1}(L) \cong V_p^+(L)$, $\mathcal{L}_{p,m}(L)/\mathcal{L}_{p,m+1}(L) \cong V_p^-(L)$.*

Proof. We prove $\mathcal{R}_{p,m}(L)/\mathcal{R}_{p,m+1}(L) \cong V_p^+(L)$ for the special case $p = \mathbb{Z}$. Proof for the general case can be done in the same way. For $\tilde{u} \in \mathcal{R}_{p,m}$, denote by $C(\tilde{u}, x, m)$ the coefficient of ϵ^m in the Laurent expansion of $\tilde{u}(x)$. Consider the map

$$\psi_m : \mathcal{R}_{p,m}(L) \to V_p^+(L),$$

$$\tilde{u} \mapsto [C(\tilde{u}, 1, m), C(\tilde{u}, 2, m), \cdots].$$

We verify the map is well-defined, that is, the image of $\tilde{u}$ is indeed in $V_p^+(L)$. Suppose $L = \sum_{i=0}^n a_i \tau^i$. Then the equation

$$\sum_{i=0}^{n} a_i(x+\epsilon)\tilde{u}(x+i) = 0 \tag{5.2}$$

holds for $x \in \mathbb{Z}$. By the definition of $\mathcal{R}_{p,m}$, there exists a large enough integer N such that for any integer $x > N$, the ϵ-adic valuation of $\tilde{u}(x)$ is greater than or equal to m, and therefore $C(\tilde{u}, x, m) = \epsilon^{-m}\tilde{u}(x) \mid_{\epsilon=0}$. Dividing both sides of (5.2) and plugging in $\epsilon = 0$, we have

$$\sum_{i=0}^{n} a_i(x) C(\tilde{u}, x+i, m) = 0$$

for $x > N$, which proves ψ_m is well-defined. It is routine to check ψ_m is $\mathbb{C}$-linear and its kernel is $\mathcal{R}_{p,m+1}(L)$. □

Lemma 78. *The map $E_{p,r}(L) : V_p^-(L) \to V_p^+(L)$ (resp. $E_{p,l}(L) : V_p^+(L) \to V_p^-(L)$) induced by the inclusion map $\mathcal{L}_{p,m} \to \mathcal{R}_{p,m+g_{p,r}}$ (resp. $\mathcal{R}_{p,m} \to \mathcal{L}_{p,m+g_{p,l}}$) is well-defined and independent on m.*

Proof. The fact that the image of $\mathcal{L}_{p,m+1}$ under the inclusion map $\mathcal{L}_{p,m} \to \mathcal{R}_{p,m+g_{p,r}}$ is contained in $\mathcal{R}_{p,m+1+g_{p,r}}$ validates the well-definedness of $E_{p,r}$. The independence of $E_{p,r}$ on m is due to Lemma 77. □

Definition 79. *We call $E_{p,r}(l)$ and $E_{p,l}(L)$ introduced in Lemma 78 extension maps.*

Lemma 80 ([21, Lemma 7])**.** *The compositions of extension maps $E_{p,r} \circ E_{p,l} : V_p^+ \to V_p^+$ and $E_{p,l} \circ E_{p,r} : V_p^- \to V_p^-$ are either identity maps or zero maps. To be more precise, when $g_{p,r} \neq g_{p,l}$, they are zero maps; when $g_{p,r} = g_{p,l}$, they are identity maps.*

Definition 81. *Suppose L is singular at $p \in \mathbb{C}/\mathbb{Z}$.*

- *If the set of valuation growths $\overline{g}_p(L) = \{0\}$, then p is called an* apparent shift singularity*.*
- *If $\overline{g}_p(L)$ has one single non-zero element, then p is called a* semi-apparent shift singularity*.*
- *If there are more than one elements in $\overline{g}_p(L)$, then p is called a* true shift singularity*.*

If p is a true shift singularity of L, then

$$0 \subsetneq Im(E_{p,r}) \subseteq \ker(E_{p,l}) \subsetneq V_{p,l}(L)$$

and

$$0 \subsetneq Im(E_{p,l}) \subseteq \ker(E_{p,r}) \subsetneq V_{p,r}(L).$$

With the notion of extension maps, some subspaces of $V(L)$ stand out, namely $\ker(E_{p,l})$ and $\ker(E_{p,r})$. The candidate-special solutions constructed in subsection 5.1.2 often lie in the kernels of extension maps.

With the theory of finite singularities, we can come up with some other special solution algorithms.

Suppose $p \in \mathbb{C}/\mathbb{Z}$. Let $\tilde{u}_1, \tilde{u}_2 \in V_p(L_\epsilon)$ be solutions with maximal and minimal valuation growths, respectively. The method to construct such solutions was given in [21, Section 4.2]. Suppose $v_{l,\epsilon}(\tilde{u}_1) = 0$ and $v_{r,\epsilon}(\tilde{u}_2) = 0$. Let $u_1 \in V_p^-(L)$, $u_2 \in V_p^+(L)$ be the sequence solutions induced by $\tilde{u}_1, \tilde{u}_2$ under the isomorphisms $\mathcal{L}_{p,0}(L)/\mathcal{L}_{p,1}(L) \cong V_p^-(L)$, $\mathcal{R}_{p,0}(L)/\mathcal{R}_{p,1}(L) \cong V_p^+(L)$, respectively. Claim that under some circumstances u_1 or u_2 is special.

Proposition 82. *When* $L = \mathrm{LCLM}(L_1, L_2)$ *where* $\mathrm{GCRD}(L_1, L_2) = 1$ *and* $g_p(L_1) \neq g_p(L_2)$*, at least one of the following statements is true:*

(i) u_1 *is a solution of either* L_1 *or* L_2*;*

(ii) u_2 *is a solution of either* L_1 *or* L_2*.*

Proof. When $g_p(L_1) \neq g_p(L_2)$, $\max(g_p(L_1)) \neq \max(g_p(L_2))$ or $\min(g_p(L_1)) \neq \min(g_p(L_2))$. We prove statement (i) for the case $\max(g_p(L_1)) \neq \max(g_p(L_2))$. The proof of statement (ii) when $\min(g_p(L_1)) \neq \min(g_p(L_2))$ is similar.

Without loss of generality, assume $\max(g_p(L_1)) > \max(g_p(L_2))$. Since $L = \mathrm{LCLM}(L_1, L_2)$, there exist $r_1 \in V_p(L_{1,\epsilon})$ and $s_1 \in V_p(L_{2,\epsilon})$ such that $\tilde{u}_1 = r_1 + s_1$. Since $\max(g_p(L_1)) > \max(g_p(L_2))$, we have

$$g_{p,\epsilon}(s_1) < \max(g_p(L_1)) \leqslant \max(g_p(L)) = g_{p,\epsilon}(\tilde{u}_1).$$

Due to the assumption that $\mathrm{GCRD}(L_1, L_2) = 1$,

$$v_{r,\epsilon}(\tilde{u}_1) = \min\{v_{r,\epsilon}(r_1), v_{r,\epsilon}(s_1)\},$$

since otherwise L_1 and L_2 have a common non-zero solution in V_p^+. Therefore,

$$v_{l,\epsilon}(s_1) = v_{r,\epsilon}(s_1) - g_{p,\epsilon}(s_1) > v_{r,\epsilon}(\tilde{u}_1) - g_{p,\epsilon}(\tilde{u}_1) = v_{l,\epsilon}(\tilde{u}_1).$$

Hence u_1 is in fact the image of r_1 so it is a solution of L_1.

□

In fact, when the assumption of Proposition 82 holds, the construction in subsection 5.1.2 almost always produces a special solution.

By combining the heuristic factorizer and the deterministic one from chapter 4, we obtain a program that is both fast and complete.

CHAPTER 6

DEGREE BOUND OF FACTORS

Let $D = F(x)[\tau]$ where F is a subfield of $\mathbb{C}$. Suppose u is a sequence solution of $L \in D$ and we want to decide if L is the minimal recurrence for u, or if u satisfies a lower order recurrence. A natural approach is to let $R = \sum_{i=0}^{n-1} \sum_{j=0}^{d} c_{ij} x^j \tau^i$ and solve the equations $Ru = 0$ for c_{ij}. This raises the question what d should be.

Hence we ask:

> Suppose $L \in D$ and R is a primitive right-hand factor of L, how can we bound the degrees of the coefficients of R?

In this chapter a complete solution is presented and with examples we demonstrate that our method yields quite sharp bound. The main tool is generalized exponents.

This chapter is organised in the following order: we first define generalized exponents of difference operators and then derive a relation between generalized exponents and the determinant; next we prove a theorem which connects determinant and degree bound for leading coefficients; then an algorithm for bounding the degree of the leading coefficients is presented and an example is given; finally we show how to bound the degree of other coefficients.

6.1 Generalized Exponents

We have defined generalized exponents for systems in subsection 4.3.2. In this section *generalized exponents* of operators will be introduced, following the approach in [6, Section 3.2]. Some background knowledge introduced in subsection 4.3.2 will not be repeated, such as the algebraic closure of $K = \mathbb{C}((t))$ where $t = 1/x$, the universal extension of $\overline{K}$ and the truncation map. Any notation or concept that lacks a definition in this chapter are defined in subsection 4.3.2.

6.1.1 Indicial equations

To solve a difference operator $L \in K_r[\tau]$, the first kind of solutions to consider are those in K_r. In order to find solutions in K_r, a straight-forward idea is to apply L on a power series in K_r with undetermined coefficients, set the result to be zero and solve the equation for coefficients. This naturally leads to the notion of *indicial equations*.

Let $\Delta = \tau - 1$. Then $\overline{K}[\Delta] = \overline{K}[\tau]$. The valuation of $\overline{K}$ extends to one on $\overline{K}[\Delta]$, also denoted by v:

$$v(\sum_i a_i \Delta^i) = \min\{v(a_i) + i\}.$$

Computations in Lemma 3.2.1 of [6] show that

$$v(L(t^\lambda)) = P(\lambda) t^{\lambda + v(L)} + o(t^\lambda). \tag{6.1}$$

where $P(\lambda) \in \mathbb{C}[\lambda]$.

Definition 83. *Call $P(\lambda) \in \mathbb{C}[\lambda]$ in the equation 6.1 the* indicial equation *of L. Denote it by* $\mathrm{Ind}(L, \lambda)$.

Lemma 84 ([6, Lemma 3.2.4])**.** *For $L \in K_r[\tau]$ where r is finite, there is a non-zero solution of L in K_r if and only if* $\mathrm{Ind}(L, \lambda)$ *has a root in $\frac{1}{r}\mathbb{Z}$. In particular, there exists a solution in K_r whose valuation is the largest root of* $\mathrm{Ind}(L, \lambda)$ *in $\frac{1}{r}\mathbb{Z}$.*

Proof. The case $r = 1$ is proved in [6], using the method of ansatz. The proof for a general r is no different.

□

Lemma 85. *Claim that* $\mathrm{Ind}(t^q \tau^0, \lambda) = 1$, *and* $\mathrm{Ind}(\Delta, \lambda) = -\lambda$.

Proof. The desired results follow immediately from

$$t^q t^\lambda = t^{q+\lambda}$$

and

$$\Delta(t^\lambda) = t^\lambda (1+t)^{-\lambda} - t^\lambda = -\lambda t^{\lambda+1} + o(t).$$

□

Next lemma will play a crucial role later on.

Lemma 86 ([6, Lemma 3.2.5])**.** *Suppose $L = L_1 L_2$. Then*

$$\mathrm{Ind}(L, \lambda) = \mathrm{Ind}(L_1, \lambda + v(L_2)) \cdot \mathrm{Ind}(L_2, \lambda).$$

Recall that in subsection 4.3.2 a new variable l which satisfies $\tau(l) - l = t$ is introduced.

Lemma 87 ([6, Theorem 3.2.10])**.** *Suppose $L \in K_r[\tau] \backslash \{0\}$. Then* $\dim_{\mathbb{C}}(\mathrm{Sol}(L, K_r[l]))$ *is the number of roots of* $\mathrm{Ind}(L)$ *in $\frac{1}{r}\mathbb{Z}$.*

6.1.2 A class of automorphisms

Following the classification of difference modules ([19, Chapter 6]) over $\overline{K}$, the solution space of $L \in \overline{K}[\tau]$ in $\Omega(\overline{K}) = \overline{K}\{\text{hyp}\}[l]$ has a $\mathbb{C}$-basis in $\overline{K}[l]_h$ (defined by (4.4)), that is,

$$\text{Sol}(L) = \text{Span}_{\mathbb{C}}\{\text{hyp}(a_1)p_1, \text{hyp}(a_2)p_2, \cdots, \text{hyp}(a_n)p_n\}$$

where $a_1, \ldots, a_n \in \overline{K}^*$ and $p_1, \ldots, p_n \in \overline{K}[l]$. We will construct a class of automorphisms $\phi_a : \overline{K}[\tau] \to \overline{K}[\tau]$, where $a \in \overline{K}^*$, such that solutions of L in $\text{hyp}(a)\overline{K}[l]$ correspond to those of $\phi_a(L)$ in $\overline{K}[l]$. We note that the construction is valid for any difference field.

Observe that

$$\tau \cdot x = (x+1)\tau, \quad (a\tau) \cdot x = (x+1)(a\tau),$$

where $a \in \overline{K}$. This suggests we can extend $\tau \mapsto a\tau, x \mapsto x$ to $\overline{K}[\tau]$ to obtain an endomorphism of $\overline{K}[\tau]$. The said endomorphism can be expressed by

$$\phi_a : \overline{K}[\tau] \to \overline{K}[\tau],$$
$$\sum_{i=0}^{n} a_i \tau^i \mapsto a_i (a\tau)^i.$$

It is routine to check it is indeed an endomorphism. When $a \in \overline{K}^*$, ϕ_a is an automorphism since $\phi_{a^{-1}}$ is its inverse.

Lemma 88. *Suppose* $L \in \overline{K}[\tau]$, $a \in \overline{K}^*$ *and* $y \in \Omega(\overline{K})$. *Then* $L(\text{hyp}(a)y) = \text{hyp}(a)(\phi_a(L))(y)$.

Proof. It follows from the fact that

$$\tau(\text{hyp}(a)y) = a\,\text{hyp}(a)\tau(y) = \text{hyp}(a)(a\tau)(y).$$

□

It follows immediately that $\text{hyp}(a)y$ is a solution of L if and only if y is a solution of $\phi_a(L)$.

Next we derive some results about how indicial equations behave under ϕ_a.

Lemma 89. *If* $\text{Trunc}(a) = 1$, *then* $\text{Ind}(\phi_a(L), \lambda) = \text{Ind}(L, \lambda)$.

Proof. By definition $a = 1 + o(t)$. Therefore

$$(\tau - a)(t^\lambda) = \Delta(t^\lambda) + (1-a)t^\lambda = (\lambda t^{\lambda+1} + o(t^{\lambda+1})) + o(t^{\lambda+1}).$$

By definition $\mathrm{Ind}(\tau - a) = \lambda$. Lemma 84 implies $\mathrm{hyp}(a) \in \overline{K}$ has valuation 0. Without loss of generality, assume $\mathrm{hyp}(a) = 1 + \sum_{i \in \frac{1}{r}\mathbb{Z}_+} h_i t^i \in K_r$.

Since

$$\mathrm{hyp}(a)\phi_a(L)(t^\lambda) = L(\mathrm{hyp}(a)t^\lambda),$$

a comparison of the leading terms of both sides shows that $\mathrm{Ind}(\phi_a(L), \lambda) = \mathrm{Ind}(L, \lambda)$. □

As a corollary, $\mathrm{Ind}(\phi_a(L), \lambda) = \mathrm{Ind}(\phi_{\mathrm{Trunc}(a)}(L), \lambda)$.

Lemma 90. *Suppose $L \in \overline{K}[\tau]$ and $c \in \mathbb{Q}$. Claim that*

$$\mathrm{Ind}(\phi_{1-ct}(L), \lambda) = \mathrm{Ind}(L, \lambda + c).$$

Proof. Notice that $(\frac{\tau(t^c)}{t^c}\tau)^n = (t^{-c}\tau t^c)^n = t^{-c}\tau^n t^c$, which means $\phi_{\frac{\tau(t^c)}{t^c}}(L) = t^{-c}Lt^c$. Apply Lemma 86 to $t^{-c}Lt^c$:

$$\mathrm{Ind}(t^{-c}Lt^c, \lambda) = 1 \cdot \mathrm{Ind}(L, \lambda + v(t^c)) \cdot 1 = \mathrm{Ind}(L, \lambda + c).$$

Since

$$\mathrm{Trunc}(\frac{\tau(t^c)}{t^c}) = 1 - ct,$$

we have

$$\mathrm{Ind}(\phi_{1-ct}(L), \lambda) = \mathrm{Ind}(t^{-c}Lt^c, \lambda) = \mathrm{Ind}(L, \lambda + c).$$

□

6.1.3 Multisets

Generalized exponents of an operator form a finite multiset. In this section we introduce necessary basics of finite multisets that will be used later. In the following a multiset will always be finite.

A multiset in the universe U is a collection of finite elements in U, where an element is allowed to appear multiple times. The number of ocurrences of an element in a multiset is called its *multiplicity*. A multiset can be denoted in the same way as a set by listing all its elements inside a $\{\cdot\}$.

Suppose $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$ are multisets. Then

$$A + B := \{a_1, \ldots, a_n, b_1, b_m\}.$$

For multisets A and C, if there is a multiset B such that $A + B = C$ then we say A is a subset of C, denoted by $A \subseteq C$. In this case we also write $B = C - A$. We will only subtract A from C when A is a subset of C.

Example 91. Let $A = \{1, 1, 2\}$, $B = \{1, 2, 2\}$, $C = \{1, 2\}$. Then

$$A + B = \{1, 1, 1, 2, 2, 2\}, \quad A + C = \{1, 1, 1, 2, 2\}, \quad B + C = \{1, 1, 2, 2, 2\}.$$

We have $C \subseteq A$ and $C \subseteq B$ because

$$C + \{1\} = A, \quad C + \{2\} = B.$$

▲

Example 92. The irreducible factors of a polynomial $p \in F[x]$ form a multiset, which we denote by $\mathrm{irr}(p)$. Then $\mathrm{irr}(pq) = \mathrm{irr}(p) + \mathrm{irr}(q)$. ▲

6.1.4 Generalized exponents

Definition 93. *Suppose $L \in \overline{K}[\tau]$ and $g \in \mathcal{G}_\infty$. If 0 is a root of $\mathrm{Ind}(\phi_g(L), n)$ with multiplicity m, call g a generalized exponent of L with multiplicity m. Denote by $\mathrm{gen}(L)$ the multiset of generalized exponents of L.*

Lemma 94. *For $a \in \overline{K}^*$, $\mathrm{Trunc}(a)$ is a generalized exponent of L with multiplicity m if and only if 0 is a root of $\mathrm{Ind}(\phi_a(L), n)$ with multiplicity m.*

Proof. By Lemma 89, $\mathrm{Ind}(\phi_a(L), n) = \mathrm{Ind}(\phi_{\mathrm{gen}(a)}(L), n)$. □

Lemma 95. *For $a \in \overline{K}^*$, $\tau - \mathrm{a} = \{\mathrm{Trunc}(a)\}$.*

Lemma 96. *If $L = L_1 L_2$, then $\mathrm{gen}(L_2) \subset \mathrm{gen}(L)$.*

Proof. This is an immediate consequence of Lemma 86 and the fact that ϕ_a is an automorphism of $\overline{K}[\tau]$ when $a \neq 0$. □

6.1.5 Generalized exponents and solutions

In this section we discuss the relation between generalized exponents and solutions of an operator and prove Theorem 102, which justifies Definition 51. Nothing in this section will be used for the major application of generalized exponents in this chapter (Theorem 106), so readers can safely skip this section if they want.

Extend the t-adic valuation to $\overline{K}[l]$ by setting

$$v(\sum_{i=0}^{d} a_i l^i) = \min\{v(a_i) : i = 0, 1, 2, \ldots, d\}.$$

Lemma 97. *For $L \in \overline{K}[\Delta]$ and $P \in \overline{K}[l]$, $v(L(P)) \geqslant v(L) + v(P)$. When $v(P)$ is not a solution of* $\mathrm{Ind}(L)$ *it is an equality.*

Proof. By the triangle inequality of valuation and additivity of $\Delta : \overline{K}[l] \to \overline{K}[l]$, it reduces to verifying

$$v(\Delta(al^n)) \geqslant 1 + v(a).$$

A straight-forward computation shows

$$\Delta(al^n) = \tau(a)(l+t)^n - al^n = (\tau(a) - a)l^n + \tau(a)\sum_{k=1}^{n} \binom{n}{k} t^k l^{n-k}.$$

Therefore its valuation is

$$v(\Delta(al^n)) = \min\{v(\tau(a) - a), v(\tau(a)t), v(\tau(a)t^2), \ldots, v(\tau(a)t^n)\} = \min\{v(\tau(a) - a), v(\tau(a)) + 1\}.$$

It remains to show that $v(\tau(a) - a), v(\tau(a)) + 1 \geqslant v(a) + 1$. These inequalities are true because

$$\frac{\tau(a)}{a} = 1 - v(a)t + o(t)$$

and hence

$$v(\tau(a)) = v(a),$$

$$v(\tau(a) - a) = v(a)v(\frac{\tau(a)}{a} - 1) = v(a)v(v(a)t) = v(a) + 1.$$

□

Denote

$$\tau_l : \overline{K}[l] \to \overline{K}[l],$$

$$P(l) \mapsto P(l+1)$$

and $\Delta_l = \tau_l - \mathrm{id}$. Then τ_l is an automorphism of $\overline{K}[l]$ that is commutative with τ. If $L(P) = 0$ for $L \in \overline{K}[\tau]$ and $P \in \overline{K}[l]$, then

$$L(\tau_l(P)) = L(\Delta_l(P)) = 0.$$

Lemma 98. *For $P \in \overline{K}[l]$, $v(\Delta_l(P)) \leqslant v(P)$.*

Proof. This reduces to the case $P = l^d$, which is obvious. □

Lemma 99. *Suppose $L \in \overline{K}[\tau]$ is the minimal operator for $P \in \overline{K}[l] - \{0\}$. Then*

$$\mathrm{Sol}(L) = \mathrm{Span}_{\mathbb{C}}\{\Delta_l^i(P) : i = 0, 1, \ldots, \deg_l(P)\}.$$

Proof. Clearly the right-hand side is a subset of the left-hand side. By the classification of difference modules over $\overline{K}$, the minimal operator of P must have an order greater than or equal to $\deg_l(P)+1$. □

Lemma 100. *Suppose $L \in \overline{K}[\tau]$ is the minimal operator for $P \in \overline{K}[l]-\{0\}$ where $v(P)=0$. Then 0 is a solution of the indicial equation of L.*

Proof. Suppose L_1 is the minimal operator of $\Delta_l(P)$. Then $L = L_2L_1$, where

$$L_2 = \Delta \cdot \frac{1}{L_1(P)}.$$

$$\mathrm{Ind}(L) = \mathrm{Ind}(L_2, n+v(L_1))\mathrm{Ind}(L_1,n) = (v+v(L_1)-v(L_1(P))\mathrm{Ind}(L_1,n).$$

If 0 is a root of $\mathrm{Ind}(L_1,n)$ then we are done; otherwise $v(L_1(P))-v(L_1)=v(P)=0$ is a root of $\mathrm{Ind}(L)$. □

Lemma 101. *Suppose* 0 *is a solution of* $\mathrm{Ind}(L)$. *Then there exists a solution P with valuation 0.*

Proof. Suppose $r_1 > r_2 > \cdots > r_k > 0$ are all positive integer roots of $\mathrm{Ind}(L)$. Using the technique in [6, Lemma 3.2.4] (there exists a solution in K with valuation being the largest integer root of the indicial equation), we can find L_2 with order k such that $\mathrm{Ind}(L_2) = (n-r_1)(n-r_2)\cdots(n-r_k)$ and $L = L_1L_2$. Let $s \in K$ be a solution of L_1 with the largest integer valuation and u a preimage of s under L_2. Then $v(s) = v(L_2(u)) \geqslant v(L_2)+v(u)$. Since $v(s)$ is the largest integer root of $\mathrm{Ind}(L_1)$, $v(s)-v(L_2)$ is the largest integer root of $\mathrm{Ind}(L_1, n+v(L_2))$, which is 0, and hence not a root of $\mathrm{Ind}(L_1,n)$. Therefore, $v(u) \leqslant v(s)-v(L_2)$ is not a root of $\mathrm{Ind}(L_1,n)$ and it follows $v(u) = v(s)-v(L_2) = 0$. □

Theorem 102. *An operator $L \in \overline{K}[\tau]$ has $g \in \mathcal{G}$ as a generalized exponent if and only if it has a solution in the form* $\mathrm{hyp}(e)f$, *where $f \in \overline{K}[l]$ has valuation 0.*

Proof. By definition g is a generalized exponent of L if and only if 0 is a solution of $\mathrm{Ind}(\phi_e(L),\lambda)$, if and only if $\phi_e(L)$ has a solution $f \in \overline{K}[l]$ with valuation 0 by Lemma 101, if and only if $\mathrm{hyp}(e)f$ is a solution of L. □

6.2 Generalized Exponents and Determinant

In this section Theorem 106, which connects generalized exponents of an operator with its *determinant* (Definition 105), is proved.

Definition 103. *For $a, b \in \overline{K}$, let*

$$\tilde{v}(a,b) = v(a\tau - b) = v(a\Delta + (a-b)) = \min\{v(a)+1, v(a-b)\}.$$

We note $\tilde{v}$ is symmetric although it does not seem so at first glance. When $v(a) = v(b)$, it is obvious; when $v(a) \neq v(b)$, by the definition of valuation,

$$\tilde{v}(a,b) = \min\{v(a)+1, v(a-b)\} = \min\{v(a)+1, v(a), v(b)\} = \min\{v(a), v(b)\}$$

and $\tilde{v}(b,a) = \min\{v(a), v(b)\}$ for the same reason. Therefore,

$$\tilde{v}(a,b) = \min\{v(a)+1, v(b)+1, v(a-b)\}.$$

Lemma 104. *Suppose $L = L_1(\tau - a)$. Claim that*

$$\mathrm{gen}(L) = \{g : g(1-\tilde{v}(g,a)t) \in \mathrm{gen}(L_1)\} + \{a\},$$

or equivalently,

$$\mathrm{gen}(L_1) = \{g(1-\tilde{v}(g,a)t) : g \in \mathrm{gen}(L) - \{a\}\}.$$

Proof. By Lemma 86,

$$\mathrm{Ind}(\phi_g(L), n) = \mathrm{Ind}(\phi_g(L_1), n + v(g\tau - a))\mathrm{Ind}(g\tau - a, n) = \mathrm{Ind}(\phi_g(L_1), n + \tilde{v}(g,a))\mathrm{Ind}(g\tau - a, n).$$

Lemma 90 says

$$\mathrm{Ind}(\phi_g(L_1), n + \tilde{v}(g,a)) = \mathrm{Ind}(\phi_{g(1-\tilde{v}(g,a)t)}(L_1), n).$$

Therefore,

$$\mathrm{Ind}(\Phi_g(L), n) = \mathrm{Ind}(\Phi_{g(1-\tilde{v}(g,a)t)}(L_1), n)\mathrm{Ind}(\Phi_g(\tau - a), n).$$

As a consequence,

$$\mathrm{gen}(L) = \{g \mid g(1-\tilde{v}(g,a)t) \in \mathrm{gen}(L_1)\} + \mathrm{gen}(\tau - a) = \{g \mid g(1-\tilde{v}(g,a)t) \in \mathrm{gen}(L_1)\} + \{a\}.$$

□

For any $L \in \overline{K}[\tau] \setminus \{0\}$ there exists a such that

$$L = L_1(\tau - a).$$

This can be seen by the classification of difference modules over $\overline{K}$ ([19, Chapter 6]). By induction it implies an order-n operator has exactly n generalized exponents counting multiplicity.

Definition 105. *For $L = \sum_{i=0}^{n} a_i\tau^i$ with $a_n a_0 \neq 0$, call* $\det(L) := (-1)^n \frac{a_0}{a_n}$ *the* determinant *of L.*

Notation: denote $a \overset{\mathcal{G}}{\sim} b$ for $a, b \in \overline{K}^*$ when $\mathrm{Trunc}(a) = \mathrm{Trunc}(b)$, in other words, the images of a and b in $\mathcal{G}$ are identical. Then $(1 - d_1 t)(1 - d_2 t) \overset{\mathcal{G}}{\sim} 1 - (d_1 + d_2)t$.

The following theorem shows the relation between generalized exponents and determinants.

Theorem 106. *Let $L = \sum_{i=0}^{n} a_i\tau^i \in \overline{K}[\tau]$ where $a_0 \neq 0$. Suppose* $\mathrm{gen}(L) = \{g_1, g_2, \ldots, g_n\}$*. Claim that*

$$\det(L) \overset{\mathcal{G}}{\sim} g_1 g_2 \cdots g_n (1 - \sum_{0<i<j\leqslant n} \tilde{v}(g_i, g_j)t).$$

Proof. Assume $a_n = 1$ without loss of generality. Then $\det(L) = (-1)^n a_0$.

We prove it by induction. The case $n = 1$ is proved in Lemma 95.

Assume the result holds for operators with order n. Consider $L' = L(\tau - f_0)$ where $\mathrm{ord}(L) = n$. Suppose $\mathrm{gen}(L') = \{g_0, g_1, g_2, \ldots, g_n\}$ where $g_0 \overset{\mathcal{G}}{\sim} f_0$. By Lemma 104, $\mathrm{gen}(L) = \{g_i(1 - \tilde{v}(g_0, g_i)t) \mid i = 1, 2, \ldots, n\}$.

Denote $g_i' = g_i(1 - \tilde{v}(g_0, g_i)t)$. We verify $\tilde{v}(g_i', g_j') = \tilde{v}(g_i, g_j)$. Definition of valuation implies

$$v(g_i') = v(g_i) + v(1 + O(t)) = v(g_i).$$

When $v(g_i' - g_j') < \min\{v(g_i) + 1, v(g_j) + 1\}$, by properties of valuation we know

$$v(g_i' - g_j') = v(g_i' - g_j' + g_i\tilde{v}(g_0, g_i)t - g_j\tilde{v}(g_0, g_j)t) = v(g_i - g_j).$$

As a result,

$$\begin{aligned} \tilde{v}(g_i', g_j') &= \min\{v(g_i') + 1, v(g_j') + 1, v(g_i' - g_j')\} \\ &= \min\{v(g_i) + 1, v(g_j) + 1, v(g_i - g_j)\} \\ &= \tilde{v}(g_i, g_j). \end{aligned}$$

Then we have

$$\begin{aligned}\det(L') &= f_0 \det(L)\\ &\overset{\mathcal{G}}{\sim} g_0 g_1' g_2' \cdots g_n' (1 - \sum_{0<i<j\leqslant n} \tilde{v}(g_i', g_j')t)\\ &= g_0 g_1 \cdots g_n \big(\prod_{i=1}^{n} (1 - \tilde{v}(g_i, g_0)t)\big)(1 - \sum_{0<i<j\leqslant n} \tilde{v}(g_i, g_j)t)\\ &= g_0 g_1 \cdots g_n (1 - \sum_{0\leqslant i<j\leqslant n} \tilde{v}(g_i, g_j)t).\end{aligned}$$

□

6.3 Determinant and Degree Bound for Leading Coefficient

In this section denote $\mathrm{lc}(\cdot)$ and $\mathrm{tc}(\cdot)$ the leading and trailing coefficients of an operator, respectively. Suppose $R \in F[x][\tau]$ is a right-hand factor of $L \in F[x][\tau]$ and both operators are primitive. According to subsection 3.2.1, after a shift, $\mathrm{lc}(R)$ is the product of the *essential part* and the *removable part*, both defined in Definition 24, and the essential part appears as a factor of $\mathrm{lc}(L)$ after a shift. Hence the essential part is naturally bounded by $\mathrm{lc}(L)$ and the problem is to bound the removable part.

In the following we define essential parts differently from Definition 24 so that they *are* factors of the leading coefficient without a shift. We also define essential parts for the trailing coefficient.

Definition 107. *Suppose $L \in F[x][\tau]$ is normal. The* leading (resp. trailing) essential part of L at order $k \in \mathbb{Z}_+ \cup \{\infty\}$ *is a monic polynomial $f \in F[x]$ such that*

(i) $\tau^{\mathrm{ord}(A)}(f) \mid \mathrm{lc}(AL)$ (resp. $f \mid \mathrm{tc}(AL)$) for any normal operator $A \in F(x)[\tau]$ with order $\leqslant k$ such that $AL \in F[x][\tau]$;

(ii) f is maximal in terms of divisibility among all polynomials satisfying (i).

Denote by $\mathrm{lc}_k(L)$ (resp. $\mathrm{tc}_k(L)$) the leading (resp. trailing) essential part of L at order k. Denote $\mathrm{lrp}_k(L) = \frac{\mathrm{lc}_{k-1}}{\mathrm{lc}_k}$ (resp. $\mathrm{trp}_k(L) = \frac{\mathrm{tc}_{k-1}}{\mathrm{tc}_k}$) for finite k and call it the leading (resp. trailing) removable part of L at level k.

Remark 108. The relation between lc_k and $\mathfrak{lc}_k$ (Definition 24) is $\mathrm{lc}_k(L) = \tau^{\deg_\tau(L)}(\mathfrak{lc}_k(L))$. By properties of $\mathfrak{lc}_k$ introduced in subsection 3.2.1 we know $\mathrm{lc}_i(L) \mid \mathrm{lc}_j(L)$ if $i > j$. Hence $\mathrm{lrp}_k(L) \in F[x]$. Trailing essential parts are not defined in subsection 3.2.1, but they are very similar to the leading essential parts. In particular, $\mathrm{trp}_k(L) \in F[x]$ and a trailing analog of Theorem 26 holds. In fact, $\mathrm{tc}(L)$ *is* the leading coefficient of L if we view L as a difference operator in τ^{-1}.

The following theorem is the main result of this section.

Theorem 109. *Suppose $L = \sum_{i=0}^{n} a_i \tau^i \in D$ and $R = \sum_{i=0}^{m} b_i \tau^i \in D$ are normal and primitive, and R is a right-hand factor of L. Let $B = \mathrm{lc}_\infty(R), A = \mathrm{tc}_\infty(R)$. There exist $c \in F, f \in F[x]$such that*

$$\frac{\tau(fb_m)}{fb_m} = c\frac{\tau(B)}{A}\det(R). \tag{6.2}$$

A straight-forward calculation shows that

$$\frac{\tau(fb_m)}{fb_m} \overset{g}{\sim} 1 + \deg(fb_m)t,$$

where $\deg(fb_m)$ is obviously a degree bound for b_m. On the other side, $\mathrm{Trunc}(c\frac{\tau(B)}{A}\det(R))$ is on a finite list because A, B are by definition monic factors of $a_0, \tau^{m-n}(a_n)$, respectively, and $\det(R)$ can be computed from $\mathrm{gen}(R) \subseteq \mathrm{gen}(L)$ by Theorem 106.

The rest of this section is devoted to the proof of Theorem 109.

The notion of *F-factors* is introduced to simplify the proof.

Definition 110. *Suppose E/F is a field extension. For a polynomial $f \in E[x]$, call a factor of f in $F[x]$ an* F-factor*. Denote by f^F the monic F-factor of f that is maximal in terms of divisibility.*

Clearly F-factors of f are closed under lcm. Hence f^F exists by taking the lcm of all F-factors.

Lemma 111. *Suppose E/F is a field extension and $f, g \in E[x]$. Then $f^F g^F = (fg)^F$.*

Proof. Clearly $f^F g^F$ is a monic F-factor of fg. We only need to show it is maximal in terms of divisibility. Suppose h is an F-factor of fg. Then there exist $h_1, h_2 \in F[x]$ such that $h = h_1 h_2$ and $h_1 \mid f, h_2 \mid g$. Hence $h \mid f^F g^F$. □

We will use the main result in [8], a reformulation of which is presented as Theorem 26. Here we give a second reformulation.

Theorem 112 (Reformulation of Theorem 6 in [8])**.** *Suppose $L \in F[x][\tau]$ is normal. Introduce new constants $c_1, c_2, \ldots, c_k$ that are algebraically independent over F. Let $A = (\tau - c_1)(\tau - c_2)\cdots(\tau - c_k)$ and $L' = \mathrm{Prim}(\mathrm{LCLM}(L, A)) \in F(c_1, \ldots, c_k)[x][\tau]$. Then $\mathrm{lc}(L')^F = \tau^k(\mathrm{lc}_k(L))$ and $\mathrm{tc}(L')^F = \mathrm{tc}_k(L)$.*

We explain why Theorem 26 and Theorem 112 are equivalent. Let c_i' denote the coefficient of τ^i in $A = (\tau - c_1)(\tau - c_2)\cdots(\tau - c_k)$. Then by the fundamental theorem of symmetric polynomials, $c_0', c_1', \ldots, c_{k-1}'$ are algebraically independent. Hence Theorem 26 applies. The result also holds for trailing coefficients because the original theorem ([8, Theorem 6]) is stated for any Ore operators, and viewed as an operator in τ^{-1}, the leading coefficient of L is $\mathrm{tc}(L)$.

Lemma 113. *For a normal and primitive operator $L \in F[x][\tau]$, $\tau(\mathrm{lrp}_1(L)) = \mathrm{trp}_1(L)$.*

Proof. Denote $L = \sum_{i=0}^{n} a_i \tau^i$.Let c be a new constant that is transcendental over F. In subsection 3.6.1 we calculated that

$$L' = \sum_{i=0}^{n+1} c^i (a_i \tau - \tau(a_{i-1}))L$$

is an LCLM of $\tau - c$ and L, where $a_i = 0$ for $i > n$ or $i < 0$. A further computation shows that

$$L' = (a\tau - c\tau(a))L,$$

where $a = \sum_{i=0}^{n} c^i a_i$. By assumption L is primitive, then so is a as a polynomial in c with coefficients in $F[x]$. Gauss's lemma implies a has no non-trivial factor in $F[x]$ as a polynomial in x. In other words, $a^F = 1$. By Theorem 112, $\mathrm{lc}(\mathrm{Prim}(L'))^F = \tau(\mathrm{lc}_1(L))$ and $\mathrm{tc}(\mathrm{Prim}(L'))^F = \mathrm{tc}_1(L)$. On the other hand,

$$\mathrm{lc}(\mathrm{Prim}(L')) = \frac{\mathrm{lc}(L')}{\mathrm{Cont}(L')} = \frac{a\tau(\mathrm{lc}(L))}{\mathrm{Cont}(L')}, \quad \mathrm{tc}(\mathrm{Prim}(L')) = \frac{\mathrm{tc}(L')}{\mathrm{Cont}(L')} = \frac{c\tau(a)\mathrm{tc}(L)}{\mathrm{Cont}(L')}.$$

Hence

$$\tau(\mathrm{lrp}_1(L)) = \frac{\mathrm{lc}(\mathrm{Prim}(L'))^F}{\tau(\mathrm{lc}(L))} = \frac{1}{\mathrm{Cont}(L')^F} = \frac{\mathrm{tc}(\mathrm{Prim}(L'))^F}{\mathrm{tc}(L)} = \mathrm{trp}_1(L).$$

□

Theorem 114. *For $L \in F[x][\tau]$, $\tau^k(\mathrm{lrp}_k(L)) = \mathrm{trp}_k(L)$.*

Proof. Let $c_1, c_2, \ldots, c_k$ be new constants that are algebraically independent over F and

$$L' = \mathrm{Prim}(\mathrm{LCLM}((\tau - c_1)(\tau - c_2)\cdots(\tau - c_{k-1}), L)) \in F(c_1, \ldots, c_{k-1})[x][\tau].$$

Apply Lemma 113 to L' to obtain

$$\tau(\mathrm{lrp}_1(L')) = \mathrm{trp}_1(L'). \tag{6.3}$$

Let

$$L'' = \mathrm{Prim}(\mathrm{LCLM}(\tau - c_k, L')).$$

As a result of Theorem 112, $\mathrm{lc}(L'')^F = \tau(\mathrm{lc}_1(L')), \mathrm{tc}(L'')^F = \mathrm{tc}_1(L')$. On the other side, by the definition of LCLM, we have

$$L'' = \mathrm{Prim}(\mathrm{LCLM}(\tau - c_k, (\tau - c_1)\cdots(\tau - c_{k-1}), L)) = \mathrm{Prim}(\mathrm{LCLM}((\tau - c_1)(\tau - c_2)\cdots(\tau - c_k), L)),$$

which implies $\tau(\mathrm{lc}_1(L'))^F = \mathrm{lc}(L'')^F = \tau^k(\mathrm{lc}_k(L)), \mathrm{tc}_1(L')^F = \mathrm{tc}(L'')^F = \mathrm{tc}_k(L)$. Hence by taking the maximal F-factors on both sides of (6.3) the desired result is proved.

□

Proof of Theorem 109. Let $B = \mathrm{lc}_\infty(R), A = \mathrm{tc}_\infty(R)$. There exists a sufficiently large $N \in \mathbb{Z}$ such that $B = \mathrm{lc}_N(R), A = \mathrm{tc}_N(R)$. Then

$$\mathrm{lc}(R) = c_1 \mathrm{lrp}_1(R)\mathrm{lrp}_2(R)\cdots\mathrm{lrp}_N(R)B,$$

$$\mathrm{tc}(R) = c_2 \mathrm{trp}_1(R)\mathrm{trp}_2(R)\cdots\mathrm{trp}_N(R)A,$$

where c_1, c_2 are leading coefficients of $\mathrm{lc}(R), \mathrm{tc}(R)$, respectively. Taking the quotient of the two equations, we obtain

$$\det(R) = c\frac{A}{B}\prod_{i=1}^{N} \mathrm{trp}_i(R)(\mathrm{lrp_i(R)})^{-1} = c\frac{A}{B}\prod_{i=1}^{N}\frac{\tau^i(\mathrm{lrp_i(R)})}{\mathrm{lrp_i(R)}},$$

where $c = (-1)^m\frac{c_2}{c_1}$. For a polynomial $p \in F[x]$, we have $\tau^i(p)/p = \frac{\tau(p\tau(p)\cdots\tau^{i-1}(p))}{p\tau(p)\cdots\tau^{i-1}(p)}$. This proves

$$\frac{\tau(fb_m/B)}{fb_m/B} = c\frac{B}{A}\det(R)$$

for some $f \in F[x]$. By the definition of essential parts, we know $\tau^{n-m}(B) \mid \mathrm{lc}_\infty(L), A \mid \mathrm{tc}_\infty(L)$. □

6.4 The Algorithm and an Example

Algorithm: degree bound for leading coefficients of factors.

Input: $L = \sum_{i=0}^n a_i\tau^i \in \mathbb{Q}(x)[\tau]$ where $a_0a_n \neq 0$ and positive integer $m < n$

Output: integer d, which bounds $\deg(\mathrm{lc}(R))$ for primitive right-hand factor R with $\mathrm{ord}(R) = m$

1 Compute $\mathrm{gen}(L)$ and list all its subsets of m elements.

2 Compute the truncations of all potential determinants of order m factors by using formula Theorem 106. Discard those that are not in the form $c(1 + dt)$ where $c \in \mathbb{Q}$ and $d \in \mathbb{Z}$. If there is none left then terminate and return "Order-m right-hand factor does not exist".

3 List all (A, B)-pairs where A, B are monic and $A \mid a_0$, $B \mid \tau^{m-n}(a_n)$. Compute $\mathrm{Trunc}(\frac{\tau(B)}{A}\det)$ for each (A, B)-pair and each det from the previous step. Only keep the ones that are in the form $c(1+dt)$, where $c \in \mathbb{Q}$, $d \in \mathbb{N}$ and $\deg(B) \leqslant d$. If there is none left, return "Order-m right-hand factor does not exist"; otherwise output the largest d.

Example 115. Let $L = \sum_{i=0}^{4} a_i\tau^i$ be the recurrence operator from [12], where

$$a_4 = 33x(3x-1)(3x-2),$$

$$a_3 = 11(2047x^3 - 10725x^2 + 17192x - 8520),$$

$$a_2 = 9(-4397x^3 - 10169x^2 + 110500x - 145368),$$

$$a_1 = -54(2x-5)(5353x^2 - 33313x + 53904),$$

$$a_0 = -115668(2x-5)(2x-7)(x-4).$$

Suppose $R = \sum_{j=0}^{2} b_j\tau^j \in \mathbb{Q}[x][\tau]$ is an order-2 primitive right-hand factor of L.

The generalized exponents of L are

$$g_1 = C_1(1-4t), g_2 = \overline{C_1}(1-4t),$$

$$g_3 = C_2(1-\frac{t}{2}), g_4 = \overline{C_2}(1-\frac{t}{2}),$$

where $C_1, \overline{C_1}$ are the solutions of the equation $11C^2 + 891C + 3213 = 0$, and $C_2, \overline{C_2}$ are the solutions of $27C^2 - 140C + 144 = 0$.

There are 6 condidates for genexp(R). Since we focus on finding factors in $\mathbb{Q}(x)[\tau]$, genexp(R) can only be $\{g_1, g_2\}$ or $\{g_3, g_4\}$. Using Theorem 106 we see that any other candidate leads to $\det(R)$ having irrational coefficients.

Let

$$\det{}_1 = \mathrm{Trunc}(g_1g_2(1 - v(g_1 - g_2)t)) = \frac{3213}{11}(1-8t);$$
$$\det{}_2 = \mathrm{Trunc}(g_3g_4(1 - v(g_3 - g_4)t)) = -\frac{16}{3}(1-t).$$

Next let (A, B) run over all the monic factors of $a_0, \tau^{-2}(a_4)$, respectively. There are two candidate for $\frac{\tau(B)}{A}$ that are compatible with at least one of potential determinants:

$$h_1 := \frac{x-1}{x-4}, \quad h_2 := \frac{(x-1)(x-4/3)(x-5/3)}{(x-5/2)(x-7/2)(x-4)}.$$

Then

$$\mathrm{Trunc}(h_1 \det{}_1) = \frac{3213}{11}(1-5t),$$

$$\begin{aligned}
\operatorname{Trunc}(h_2 \det_1) &= \frac{3213}{11}(1-2t),\\
\operatorname{Trunc}(h_1 \det_2) &= -\frac{16}{3}(1+2t),\\
\operatorname{Trunc}(h_2 \det_2) &= -\frac{16}{3}(1+5t).
\end{aligned}$$

By Theorem 109, $\deg(b_m)$ is either 2 or 5.

The heuristic special solution algorithm shows that one and likely the only one second-order right-hand factor is $R = \sum_{j=0}^{2} b_j \tau^j$, where

$$b_2 = 3(3x-8)(x-2)(3x-7)(221x^2 - 1607x + 2904),$$

$$b_1 = -2(2x-5)(7735x^4 - 94920x^3 + 432119x^2 - 864954x + 642312),$$

$$b_0 = -36(2x-5)(2x-7)(x-4)(221x^2 - 1165x + 1518).$$

In particular, $\deg(b_2) = 5$. ▲

6.5 Bounding Other Coefficients

A degree bound for leading coefficient easily yields one for other coefficients using the following proposition.

Proposition 116. *Suppose $L, R \in F[x][\tau]$ are both primitive and R is a right-hand factor of L. Then*

$$\deg_x(R) \leqslant \deg_x(L) - \deg(\mathrm{lc}(L)) + \deg(\mathrm{lc}(R)).$$

Proof. There exists a primitive operator $L_1 \in F[x][\tau]$ and a polynomial $f \in F[x]$ such that

$$fL = L_1 R.$$

Hence

$$\deg(f) + \deg_x(L) = \deg_x(L_1) + \deg_x(R)$$

and

$$\deg(f) + \deg(\mathrm{lc}(L)) = \deg(\mathrm{lc}(L_1)) + \deg(\mathrm{lc}(R)).$$

By subtracting the two equations we see that

$$\begin{aligned}
\deg_x(R) &= \deg_x(L) - \deg(\mathrm{lc}(L)) - (\deg_x(L_1) - \deg(\mathrm{lc}(L_1)) + \deg(\mathrm{lc}(R))\\
&\leqslant \deg_x(L) - \deg(\mathrm{lc}(L)) + \deg(\mathrm{lc}(R)).
\end{aligned}$$

□

A sharper bound is achievable by exploiting the Newton polygons (defined in [6, Page 20]) of L and R. In fact, Proposition 116 is a tacit application of Newton polygons where we only use the lowest vertices, which correspond to the terms with the largest x-degree.

BIBLIOGRAPHY

[1] S. A. Abramov, M. A. Barkatou, and M. van Hoeij. Apparent singularities of linear difference equations with polynomial coefficients. *Applicable Algebra in Engineering, Communication and Computing*, 17:117 – 133, 2006.

[2] Sergei Abramov. Eg-eliminations. *Journal of Difference Equations and Applications*, 5:393–433, 01 1999.

[3] Alin Bostan, Xavier Caruso, and Éric Schost. A fast algorithm for computing the characteristic polynomial of the p-curvature. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 59–66, New York, NY, USA, 2014. ACM.

[4] Manuel Bronstein. Personal communication.

[5] Manuel Bronstein. An improved algorithm for factoring linear ordinary differential operators. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC '94, pages 336–340, New York, NY, USA, 1994. ACM.

[6] Yongjae Cha. *Closed Form Solutions of Linear Difference Equations*. PhD thesis, Florida State University, 2010.

[7] Shaoshi Chen, Maximilian Jaroschek, Manuel Kauers, and Michael F. Singer. Desingularization explains order-degree curves for Ore operators. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, ISSAC '13, pages 157–164, New York, NY, USA, 2013. Association for Computing Machinery.

[8] Shaoshi Chen, Manuel Kauers, and Michael F. Singer. Desingularization of Ore operators. *Journal of Symbolic Computation*, 74:617 – 626, 2016.

[9] Thomas Cluzeau. Factorization of differential systems in characteristic p. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ISSAC '03, pages 58–65, New York, NY, USA, 2003. Association for Computing Machinery.

[10] Beke E. Die irreduzibilität der homogenen linearen differentialgleichungen. *Math. Ann.*, 45:278–294, 1894.

[11] OEIS Foundation Inc. Entry A002777 in the on-line encyclopedia of integer sequences. `http://oeis.org/A002777`.

[12] OEIS Foundation Inc. Entry A025184 in the on-line encyclopedia of integer sequences. `http://oeis.org/A025184`.

[13] OEIS Foundation Inc. Entry A151329 in the on-line encyclopedia of integer sequences. `http://oeis.org/A151329`.

[14] OEIS Foundation Inc. The on-line encyclopedia of integer sequences. Published electronically at `http://oeis.org`.

[15] Maximilian Jaroschek. *Removable Singularities of Ore Operators*. PhD thesis, RISC, Johannes Kepler University Linz, 2013.

[16] Raphaël Pagès. Computing characteristic polynomials of p-curvatures in average polynomial time. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*, ISSAC '21, pages 329–336, New York, NY, USA, 2021. Association for Computing Machinery.

[17] Marko Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 14(2):243–264, 1992. Symbolic Computation in Combinatorics.

[18] Harrison Tsai. Weyl closure of a linear differential operator. *J. SYMBOLIC COMPUT*, 29:4–5, 2000.

[19] M. van der Put and M.F. Singer. *Galois Theory of Linear Differential Equations*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2003.

[20] Marius van der Put. Reduction modulo p of differential equations. *Indagationes Mathematicae*, 7(3):367–387, 1996.

[21] Mark van Hoeij. Finite singularities and hypergeometric solutions of linear recurrence equations. *Journal of Pure and Applied Algebra*, 139(1):109 – 131, 1999.

[22] Mark van Hoeij. Implementation of `DEtools[Homomorphisms]`, added to maple in 2005. it uses `LCLM` to discard apparant singularities. `http://www.math.fsu.edu/~hoeij/files/Hom`, 2004.

[23] Robert J. Walker. *Algebraic Curves*. Springer-Verlag New York, 1978.

[24] Yi Zhang. Contraction of Ore ideals with applications. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 413–420, New York, NY, USA, 2016. Association for Computing Machinery.

[25] Yi Zhou. Implementations and examples. `http://www.math.fsu.edu/~yzhou/desing`, 2021.

BIOGRAPHICAL SKETCH

Yi Zhou was born in Jining, Shandong, China. He earned his Bachelor's degree in Mathematics at Beijing Normal University. After that, he started to pursue the doctoral degree in Pure Math at Florida State University under the supervision of Professor Mark van Hoeij. His research interests include computer algebra and differential/difference algebra.