Research Institute for Symbolic Computation
Johannes Kepler University Linz, Austria

Algebraic Extensions for Symbolic Summation

Burçin Eröcal

Doctoral Thesis

advised by
Priv.-Doz. Dr. Carsten Schneider
Univ.-Prof. Dr. Peter Paule

examined by
Priv.-Doz. Dr. Carsten Schneider
Prof. Dr. Marko Petkovšek

Defended February 14, 2011, Linz, Austria.

This work was partially supported by the projects P20347 and DK W1214 of the Austrian Science Foundation FWF.

There is no permanent place in this world for ugly mathematics.

G. H. Hardy

Abstract

The main result of this thesis is an effective method to extend Karr's symbolic summation framework to algebraic extensions. These arise, for example, when working with expressions involving $(-1)^n$. An implementation of this method, including a modernised version of Karr's algorithm is also presented.

Karr's algorithm is the summation analogue of the Risch algorithm for indefinite integration. In the summation case, towers of specialized difference fields called $\Pi\Sigma$-fields are used to model nested sums and products. This is similar to the way elementary functions involving nested logarithms and exponentials are represented in differential fields in the integration case.

In contrast to the integration framework, only transcendental extensions are allowed in Karr's construction. Algebraic extensions of $\Pi\Sigma$-fields can even be rings with zero divisors. Karr's methods rely heavily on the ability to solve first-order linear difference equations and they are no longer applicable over these rings.

Based on Bronstein's formulation of a method used by Singer for the solution of differential equations over algebraic extensions, we transform a first-order linear equation over an algebraic extension to a system of first-order equations over a purely transcendental extension field. However, this domain is not necessarily a $\Pi\Sigma$-field. Using a structure theorem by Singer and van der Put, we reduce this system to a single first-order equation over a $\Pi\Sigma$-field, which can be solved by Karr's algorithm. We also describe how to construct towers of difference ring extensions on an algebraic extension, where the same reduction methods can be used.

A common bottleneck for symbolic summation algorithms is the computation of nullspaces of matrices over rational function fields. We present a fast algorithm for matrices over $\mathbb{Q}(x)$ which uses fast arithmetic at the hardware level with calls to BLAS subroutines after modular reduction. This part is joint work with Arne Storjohann.

Keywords: Symbolic summation, difference rings, Karr's algorithm, algebraic extensions.

Zusammenfassung

Das Hauptresultat dieser Arbeit ist eine leistungsfähige Methode, die Karr's Algorithmus für symbolische Summation auf algebraische Erweiterungen ausdehnt. Diese Erweiterungen treten zum Beispiel bei der Arbeit mit Ausdrücken auf, die $(-1)^n$ enthalten. Eine Implementierung dieser Methode inklusive einer etwas weiterentwickelten Version des Algorithmus von Karr wird zusätzlich vorgestellt.

Karr's Algorithmus, das Summations-Analogon des Algorithmus von Risch zur indefiniten Integration, basiert auf Schachtelungen spezieller Differenzenkörper, sogennanter $\Pi\Sigma$-Körper. Diese Konstruktionen modellieren verschachtelte Summen und Produkte in einer Weise, wie im Integrationsfall elementare Funktionen, einschließlich verschachtelter Logarithmen und Exponentialfunktionen, dargestellt werden.

Karr's Konstruktion, deren Fundament die Lösbarkeit von Differenzengleichungen erster Ordnung in $\Pi\Sigma$-Körpern ist, erlaubt im Gegensatz zu den Rahmenbedingungen in der Integration nur transzendente Erweiterungen. Algebraische Erweiterungen von $\Pi\Sigma$-Körpern können zu Ringen mit Nullteiler führen, worauf Karr's Methoden nicht mehr anwendbar sind.

Basierend auf früheren Ergebnissen von Bronstein bezüglich leistungsfähiger Integrations-Methoden für algebraische Erweiterungen, wandeln wir eine lineare Gleichung erster Ordnung über einer algebraischen Erweiterung in ein System von Differenzengleichungen erster Ordnung über einem rein transzendenten Erweiterungskörper um. Allerdings ist dieser Bereich nicht notwendiger Weise ein $\Pi\Sigma$-Körper. Ein Struktursatz der Galois-Theorie für Differenzengleichungen ermöglicht die Reduktion dieses Systems zu einer einzigen Gleichung erster Ordnung über einem $\Pi\Sigma$-Körper, welche mit Hilfe von Karr's Algorithmus gelöst werden kann. Zusätzlich beschreiben wir die Konstruktion geschachtelter Erweiterungen von Differenzenringen über einer algebraischen Erweiterung, wobei dieselben Reduktionsmethoden angewandt werden können.

Ein Engpass aller symbolischen Summationsalgorithmen ist die Berechnung von Nullräumen von Matrizen über Körper rationaler Funktionen. Wir beschreiben einen schnellen Algorithmus für Matrizen über $\mathbb{Q}(x)$, welcher schnelle Arithmetik auf Hardware-Ebene verwendet, wobei nach modularer Reduktion BLAS Unterprogramme aufgerufen werden. Dieser Teil entstand in Zusammenarbeit mit Arne Storjohann.

Stichwörter: Symbolische Summation, Differenzenringe, Karr's Algorithmus, algebraische Erweiterung

Contents

1. Introduction

Symbolic summation methods aim to simplify expressions involving sums by finding *closed form* representations if they exist. Similarly, symbolic integration methods find closed form solutions of integrals.

These methods can be broadly classified based on the type of functions they work with. Sums or integrals involving hypergeometric or hyperexponential functions can be simplified by reducing the problem to one involving only rational functions then polynomials [8, 10, 14, 29, 33, 53, 55–57, 81]. Holonomic functions are handled with Gröbner bases in D-modules [22,47,49,77]. Liouvillian functions are modeled in towers of differential fields and their summation analogues, indefinite sums and products are represented in towers of difference fields.

The latter class dates back to Liouville's work to show which integrals could be expressed in terms of a well defined set of functions, called *elementary functions*. The algebraic framework for this theory was provided by Ritt [60] and Rosenlicht [61]. Risch used this framework to give an algorithm for indefinite integration of elementary functions [58, 59]. A historical account of these developments can be found in [43].

This framework for integration using towers of differential fields was transferred to the difference setting by Karr [41, 42] to model nested indefinite sums and products in towers of difference fields. Even though Risch's algorithm could not be carried over easily, Karr also provided an algorithm to find closed form solutions of indefinite sums and products that could be represented within this construction.

In contrast to the integration case, Karr covers only transcendental extensions in the tower of difference fields. Algebraic extensions of difference fields are not as well behaved as those of differential fields. Even in the simple case of extending a difference field with an indeterminate representing $(-1)^n$, the result is a ring with zero divisors.

This thesis presents effective algorithms to handle algebraic extensions in the towers of difference fields, or rings as they turn out to be, in Karr's symbolic summation framework. To allow algebraic extensions in this framework, we describe how first-order linear equations can be solved over these rings and further transcendental extensions of them.

An algebraic extension of a difference field is not necessarily a field, but it can be viewed as a direct sum of fields. With this point of view, a first-order linear difference equation over the algebraic extension ring becomes a system of first-order equations over the component fields. Algorithms to solve such systems [5, 13], or an equivalent higher order equation [3, 4] over rational function fields exist, but they are computationally demanding. Fortunately, this potentially costly path can be avoided. Using extra knowledge about the structure of the extension, this system can be uncoupled by hand and transformed to an equation which is still within the range of Karr's algorithm. Example 3.26 presents a concrete instance, while the general solution is described in Section 3.4.

Based on a decomposition of the splitting ring of the solutions of a difference equation [78, Corollary 1.16] into components invariant under σ^d where σ is the associated difference automorphism and d is the degree of the algebraic extension, we also show that transcendental extensions built on top of an algebraic extension have the same structure, a finite-dimensional algebra over a tower of transcendental difference field extensions. This fact allows us to use the same method of reducing a system to a single equation to solve difference equations over towers that contain algebraic extensions.

Before presenting a brief overview of the contents, we shortly mention some recent results related to symbolic summation and algorithms for difference equations.

It was noted in [62] that Karr's algorithm is capable of solving creative telescoping equations that arise in Zeilberger's definite summation framework [83]. Other methods from hypergeometric summation such as solving higher order difference equations [3,4, 6,14,17] or finding hypergeometric solutions of higher order ordinary linear operators [19,56] were partially adapted to the setting of difference fields [62,68,69]. The efficient implementation of these methods in Mathematica [64] led to various applications [2, 15,51,54,72].

In [70] an algorithm to construct a depth optimal tower of difference fields for a given sum is presented. This gives a more practical definition of simplification for sums by providing a measure of complexity based on an embedding into the ring of sequences.

Extensions of Karr's algorithm to handle various new classes of input are provided in [44–46,66]. Radical expressions such as $\sqrt[d]{k}$ are covered in [46], using the recipe described in [44,45]. This involves solving a set of subproblems that come up during the application of Karr's algorithm. Once these subproblems are answered, Karr's algorithm can be applied in its original form over a new domain.

The approach taken in Chapter 3 is to reduce the problem of working with a tower including an algebraic extension to a domain where Karr's algorithm was originally designed for. With the procedure described in Section 3.4, only a first-order linear difference equation needs to be solved over a $\Pi\Sigma$-field, as described by Karr.

In [66], in order to solve a first-order linear difference equation over a tower with a higher order relation at the top, a system of first-order equations is considered. This is similar to the approach used in Section 3.1, adopted from the treatment of the algebraic case of differential equations [16,73]. The problem of extending the fields described in [66] is still open.

Theoretical results on the classification and solution of q-difference equations when q is a root of unity are presented in [39,40,78]. Since we consider a canonical choice for extensions representing indefinite products of roots of unity, such as $(-1)^k$, we can avoid problems with the uniqueness of the splitting ring of the solutions [39, Chapter 6] and provide an efficient algorithm to solve these equations.

1.1. A brief overview

The framework used by Risch's algorithm for indefinite integration represents the so-called *elementary functions* in towers of differential fields [18]. Elementary functions are those that can be formed from the rational functions in the integration variable x by repeatedly adjoining a finite number of nested logarithms, exponentials and algebraic functions. For a differential field extension $F(t), \delta$ of F, δ, we say that t is *logarithmic* over F if $\delta(t) = \frac{\delta(u)}{u}$ and *exponential* over F if $\delta(t) = \delta(u)t$ for some $u \in F$. Intuitively, t represents $\log(u)$ if t is logarithmic and $\exp(u)$ if it is exponential. If there exists a polynomial $p \in F[z]$ such that $p(t) = 0$, we call t algebraic over F.

For example, to apply Risch's algorithm to the integral

$$\int \frac{xe^{\sqrt{x^2+2}}}{\sqrt{x^2+2}} dx,$$

the integrand is modeled in $\mathbb{Q}(x, t, u)$ where $t = \sqrt{x^2+2}$ and $u = e^t$. In this case, t is algebraic over $\mathbb{Q}(x)$ and u is exponential over $\mathbb{Q}(x, t)$.

A similar construction is used for summation, where nested indefinite sums and products are modeled in towers of difference fields. In this case, at each step a difference field extension $F(t), \sigma$ of F, σ is formed. If $\sigma(t) = t + \beta$ for some nonzero $\beta \in F$, then t represents an indefinite sum over β, that is $t = \sum \beta$. If $\sigma(t) = \alpha t$ for some nonzero $\alpha \in F$, then it models an indefinite product $\prod \alpha$.

For example, to simplify the sum

$$\sum_{k=a}^{b} \binom{n}{k} H_k,$$

where H_k denotes the harmonic numbers $\sum_{j=1}^{k} \frac{1}{j}$, the summand is represented in the difference field $\mathbb{Q}(k, h, b)$ with $\sigma(k) = k + 1$, $\sigma(h) = h + \frac{1}{k+1}$, and $\sigma(b) = \frac{n-k}{k+1} b$. Here, h represents the harmonic number H_k and b is the binomial $\binom{n}{k}$.

Karr's framework prohibits algebraic extensions in this construction, by introducing constraints on the coefficients α and β. For instance, α cannot be a root of unity in Karr's setting, so the extension $\sigma(t) = -t$ which models $(-1)^k$ is not allowed. A detailed account of the specialized difference fields allowed in Karr's construction is presented in Chapter 2.

The key component of Karr's algorithm is the ability to find an explicit solution or algorithmically decide that no solution exists for a first-order linear difference equation over a specialized difference field. Namely, when α, β are given, Karr describes an algorithm to find g in the same domain such that

$$\sigma(g) - \alpha g = \beta.$$

This procedure is used both in simplifying a given input sum by finding an *antidifference* and verifying at each step of the construction that the newly formed difference field extension is still within the range of the algorithm.

An algebraic difference field extension can lead to objects ranging from rings with zero divisors to polynomial rings with infinitely many variables. For example, the expression $\sqrt[d]{k}$, where $d > 1$ is an integer and $\sigma(k) = k + 1$, does not satisfy any recurrence with polynomial coefficients [32]. In order to form a difference ring which includes $\sqrt[d]{k}$, all possible shifts of this expression have to be added as new variables. A strategy to solve summation problems over such rings with infinitely many variables is presented in [46]. This case is not considered in this thesis.

We consider extensions where the new variable t satisfies the difference equation $\sigma(t) = \alpha t$ where α is a primitive root of unity. Difference rings formed this way are finitely generated and contain zero divisors. For example, the ring $\mathbb{Q}[t]/\langle t^2 - 1\rangle$ where $\sigma(t) = -t$ is constructed by adding an element representing $(-1)^k$ to the constant field $\mathbb{Q}$. Since $\left((-1)^k\right)^2 = 1$, we have the relation $t^2 - 1 = 0$. This ring has zero divisors since the polynomial $t^2 - 1$ factors as $(t - 1)(t + 1)$ over $\mathbb{Q}[t]$.

To extend Karr's framework to such expressions, two problems need to be addressed. We need an algorithm to solve first-order linear difference equations over these rings and we need to specify how to build further difference ring extensions based on them where first-order linear difference equations can still be solved effectively.

An algebraic extension on a tower of transcendental extensions can be viewed as a finite-dimensional algebra over the transcendental tower. This reduces the problem of solving a linear difference equation to solving a system of equations over the transcendental extension. In Section 3.1, we generalize Bronstein's formulation [16] of a method Singer used [73] to handle the algebraic case for differential equations with Liouvillian coefficients to include the difference case. This leads to an explicit formulation of this system when a basis is specified for the finite-dimensional algebra corresponding to the algebraic extension ring. This system can be reduced to a single first-order linear difference equation over a $\Pi\Sigma$-field using the formulation in Section 3.4.

In [78, Corollary 1.16], Singer gives a decomposition of the splitting ring of the solutions of a difference equation as a direct sum of integral domains. In our case, these domains are just Karr's $\Pi\Sigma$-fields. These components are invariant under σ^d, where d is the degree of the algebraic extension. This makes it possible to view transcendental extensions on an algebraic extension as transcendental extensions of the individual components, as detailed in Section 3.2. Then towers involving an algebraic extension at any step can be viewed as a finite-dimensional algebra over a purely transcendental tower. Karr's framework already provides algorithms to solve first-order linear difference equations over this purely transcendental tower. Hence, transforming such equations to this purely transcendental setting provides an algorithm to solve them.

Additional material

Many symbolic summation algorithms rely on a method to find the nullspace of a matrix with entries from a rational function field. Since arithmetic with rational functions is quite expensive and classical linear algebra methods are prone to intermediate expression swell, better algorithms are necessary for even moderate input. Chapter 4 presents two algorithms to attack this problem, with a performance comparison on matrices obtained from the implementation of WZ-summation in Wegschaider's Mathematica package [80]. This is joint work with Arne Storjohann.

One of the algorithms described is an application of common computer algebra tools, such as reducing the computation to word size primes using the Chinese remainder theorem, with a core optimized to use BLAS operations for x-adic lifting. With early termination strategies, this method outperforms other implementations on our examples, including stock methods from common computer algebra systems and a specialized implementation developed solely for summation problems. The second algorithm implemented has a better complexity, but does not lend itself to optimizations easily such as the ability to use fast machine arithmetic at the core.

An implementation of a modern version of Karr's algorithm in the open source computer mathematics system Sage [28, 75] is described in the appendix. While being well suited for experimentation, especially with the availability of a diverse selection of algebraic data structures and algorithms in Sage, this implementation is not ready for use in applications since it lacks a friendly user interface. Schneider's Sigma package [69] for Mathematica also implements Karr's algorithms with several enhancements and optimizations as well as an accessible interface. It was used in large applications related to the computation of Feynman integrals [15, 51]. Gärtner's thesis [30] describing an implementation in the Reduce system is also a good reference for pseudo-code of Karr's methods.

1.2. Notation and conventions

Preliminary notions and specialized notation will be explained at the beginning of each chapter. The following notation and conventions are used throughout the thesis.

The set of natural numbers $\{0, 1, \dots\}$ is denoted by $\mathbb{N}$, the ring of integers by $\mathbb{Z}$, and the field of rational numbers by $\mathbb{Q}$.

All rings and fields are of characteristic 0 and ideals are two-sided ideals. Rings are not necessarily commutative, but all integral domains and fields are commutative. Given a ring R, its group of units will be denoted by R^*. We denote the total quotient ring of R with $Q(R)$. For an element $p \in R$, the ideal generated by p will be denoted by $\langle p \rangle$.

A column vector with components a_i will be written $(a_0, \dots, a_n)^T$ to help typesetting, where the superscript T alludes to a transpose.

For most sums, the bounds will be shown under the summation signs in the notation $a \leq k < b$, where the lower bound is included in the range, but the upper bound is not. This notation from [41] fits the telescoping paradigm, since if we have an antidifference g for a summand f, that is, $g(k+1) - g(k) = f(k)$, then we can write $\sum_{a \leq k < b} f(k) = g(b) - g(a)$. It is also useful from a computer science and implementation point of view [24].

The variable i is used as a summation variable and not the complex unit such that

$i^2 = -1$, unless specified otherwise.

2. Summation in finite terms

This chapter presents a brief overview of algorithms for symbolic summation and introduces the basic machinery behind Karr's algorithm. Only the details of concepts which will be required later are provided.

For a more complete exposition of Karr's algorithm, see [41] and [42]. Hypergeometric summation methods, presented at the end of the introduction to this chapter, are explained in [55].

The problem

Algorithms for symbolic summation aim to find a *simpler* formula for a given expression by eliminating the summation quantifiers. For example, consider the well known identity for the sum of integers from 1 to n,

$$\sum_{1 \leq i < n} i = \frac{(n-1)n}{2}.$$

Given the sum on the left hand side, an algorithm would output the rational function on the right hand side. The numerical evaluation of the rational function is much simpler than that of the sum expression, at the cost of 3 arithmetic operations over the base field versus $n - 2$.

A slightly more formal statement of this task would be: Given a sum

$$\sum_{a \leq i < n} f(i) \tag{2.1}$$

over an explicitly specified univariate function f, find a function s such that

$$s(n) = \sum_{a \leq i < n} f(i).$$

In this statement, the properties of the function f and the result s are left open, since the range of expressions each algorithm can work with differs. To specify the range of summation problems that Karr's algorithm can handle, we need to introduce the algebraic framework behind it.

A solution

In order to simplify a sum expression such as (2.1), we can look for an antidifference g for the given function f:

$$g(i+1) - g(i) = f(i)$$

Once such a function is found, we only need to evaluate it at the bounds of the sum to get the desired result, assuming that the summand is well defined in the summation range:

$$s(n) = g(n) - g(a) = \sum_{a \leq i < n} f(i)$$

Note that this approach is similar to solving a definite integration problem by finding a primitive function.

Looking for an antidifference transforms the problem statement so that the summation quantifier does not appear. If we consider the function f to be a rational function in $\mathbb{Q}(i)$ and take σ to be the automorphism of $\mathbb{Q}(i)$ such that $\sigma : c \mapsto c$ for all $c \in \mathbb{Q}$ and $\sigma : i \mapsto i + 1$, the equation can be written as

$$\sigma(g) - g = f.$$

This point of view leads to a generalization which allows us to work with more complicated objects. If an indefinite sum $\sum_{1 \leq j < i} f(j)$ cannot be simplified further, it can be added to the current domain as an indeterminate which satisfies the equality $\sigma(g) - g = f$. For example, the harmonic numbers $H_i = \sum_{1 \leq j < i+1} \frac{1}{j}$ cannot be represented as a rational function in $\mathbb{Q}(i)$. Then, we can form the extension field $\mathbb{Q}(i, h)$ such that $\sigma(h) = h + \frac{1}{i+1}$. In this setting, to simplify the sum $\sum_{1 \leq i < n} iH_i$, we would solve the equation $\sigma(g) - g = ih$ over $\mathbb{Q}(i, h)$.

A more general form of the equality $\sigma(g) - g = f$ extends this framework to model products as well. An indefinite product, $\prod_{1 \leq j < i} f(j)$, satisfies the recurrence $g(i + 1) - f(i)g(i) = 0$. Considering the equality

$$\sigma(g) - ag = f$$

instead would also cover this case. For example, the factorial function, $i! = \prod_{1 \leq j < i+1} j$, satisfies the recurrence $g(i+1) - (i+1)g(i) = 0$, which can be written as $\sigma(g) - (i+1)g = 0$ in our setting. In order to represent $i!$ in an extension of $\mathbb{Q}(i)$, we would add an indeterminate t which satisfies the equality $\sigma(t) - (i + 1)t = 0$. To simplify the sum $\sum_{1 \leq i < n} ii!$ over this domain, we would solve the first-order linear difference equation $\sigma(g) - g = it$ over $\mathbb{Q}(i, t)$

With the ability to solve the general equation $\sigma(g) - ag = f$ for g when a and f are given in a difference field, we also gain the ability to simplify products. Similar to the sum case, where we solved $\sigma(g) - g = f$ when given a sum $\sum_{a \leq i < n} f(i)$, now we consider the equation $\sigma(g) - fg = 0$ when a product $\prod_{a \leq i < n} f(i)$ is given. Thus, with the same algorithm we can simplify both product and sum expressions.

Karr's algorithm

Karr's algorithm uses the ideas described above to build a tower of difference fields containing the given summand or, in the case of a product, a multiplicand. The problem of simplifying the input sum or product is reduced to solving a first-order linear difference equation

$$\sigma(g) - \alpha g = \beta,$$

for g in this tower, where α and β are chosen based on the input. If this equation has a solution, all that is left is to evaluate the solution at the bounds of the given sum or product. Otherwise, simplifying the given expression is beyond the capabilities of Karr's algorithm, though for sums, other methods such as Zeilberger's creative telescoping could still be applied.

Starting from a sum $\sum_{a\leq i<n} f(i)$ or a product $\prod_{a\leq i<n} f(i)$, the steps in the execution of Karr's algorithm are roughly the following:

(i) construct a field K containing all the constants appearing in f,

(ii) then the rational function field $K(i)$ with the automorphism $\sigma(i) = i+1$,

(iii) build a tower $K(i, t_1, \ldots, t_m)$ where the t_j for $1 \leq j \leq m$ are the indefinite sums or products needed to express f.

(iv) Solve the first-order linear difference equation

$$\sigma(g) - g = f$$

for a sum, or

$$\sigma(g) - fg = 0,$$

if the input is a product over $K(i, t_1, \ldots, t_m)$.

(v) If there is such a g, return $g(n) - g(a)$ for a sum, or $g(n)/g(a)$ for a product. Otherwise, return `NO SOLUTION`.

The following example demonstrates these steps more concretely.

Example 2.1. Consider the following sum,

$$\sum_{1\leq i<n} H_i^2,$$

where H_i is the harmonic number defined as $H_i := \sum_{1\leq j<i+1} \frac{1}{j}$.

The constants appearing in this sum are just the rational numbers, so we take $\mathbb{Q}$ as the constant field K.

Starting from the rational function field $K(i) = \mathbb{Q}(i)$, we try to model H_i^2. Karr provides algorithmic criteria to decide that H_i cannot be represented as a rational function in $K(i)$. Then, we extend the field with a new indeterminate which models H_i. The difference field we work with is $K(i, h)$ with the automorphishm σ such that

$$\begin{aligned} \sigma(c) &= c \text{ for all } c \in K, \\ \sigma(i) &= i+1, \\ \sigma(h) &= h + \tfrac{1}{i+1}. \end{aligned}$$

Using Karr's algorithm, we solve the equation $\sigma(g) - g = h^2$ in $K(i, h)$ to get

$$g = h^2 i - 2hi - h + 2i.$$

Now, the result is

$$\sum_{1\leq i<n} H_i^2 = g(n) - g(1) = H_n^2 n - 2H_n n - H_n + 2n.$$

We briefly mention other approaches to symbolic summation before presenting the details of this mathematical framework. The treatment of towers of difference fields that Karr's algorithm can work with is continued in Section 2.1.

Other approaches

These alternative algorithms mainly work with proper hypergeometric summands, which means the term ratio of the summand is a rational function where the denominator can be factored into linear terms. For a precise definition, see [55, page 143]. There are excellent descriptions of these algorithms in the books [55], [79, Chapter 23] and [34, Section 5.8].

Gosper's algorithm:

In analogy to the question of indefinite integration of rational functions, Gosper considered indefinite summation in closed from for hypergeometric summands. The algorithm designed by Gosper finds a hypergeometric solution for the equation $\sigma(g) - g = f$ or proves that such a solution cannot exist. This leads to a hypergeometric closed form representation for the sum

$$g(n) - g(0) = \sum_{0\leq k<n} f(k)$$

with hypergeometric summand f free of the variable n. In this case f is called *Gosper-summable*.

The problem can be simply rewritten in terms of finding a rational solution y for the equivalent equation $r\sigma(y) - y = 1$ where r is the rational function given by the shift quotient $\frac{\sigma(f)}{f}$. Then by denominator bounding this becomes equivalent to searching for a polynomial solution. The polynomial solution is determined by first finding an upper bound for its degree and then comparing coefficients from both sides of the equation.

Zeilberger's algorithm:

Gosper's algorithm works for indefinite hypergeometric sums, where the summand does not depend on the limits of summation. Such sums satisfy a first-order linear recurrence of the form $\sigma(g) - g = f$, where f is the summand. Unfortunately, this is not the case for definite sums such as $\sum_{k=0}^{n} \binom{n}{k}$.

Zeilberger's algorithm [83] finds recurrences for definite sums

$$s(n) = \sum_{k=an+b}^{un+v} f(n,k)$$

where the summation limits depend linearly on n. This is done by considering a more general form of the telescoping equation $\sigma(g) - g = f$, where the right side has more terms obtained by shifting the summand f in the variable n. This approach, called *creative telescoping*, solves the equation

$$\sigma(g) - g = c_1 f_1 + \cdots c_n f_n$$

for g and coefficients $c_1, \ldots, c_n$, where $f_1, \ldots, f_n$ are given and $c_1, \ldots, c_n$ are constant under σ, though they might contain the variable n.

Wilf and Zeilberger proved in [81, Section 2] the existence of the creative telescoping equation for *proper* hypergeometric summands f. The homogeneous or inhomogeneous recurrence satisfied by the sum $s(n)$ is obtained by summing this *certificate recurrence* over the given range.

Karr's algorithm can solve creative telescoping equations which occur in the context of Zeilberger's algorithm for definite summation. This was first observed in [62, Section 1.3 and 4.3]. An equivalent of the fundamental theorem of hypergeometric summation, which states that proper hypergeometric terms satisfy recurrence relations of this type, is not available in the context of $\Pi\Sigma$-fields. Thus, termination of a definite summation algorithm in this context is still an open problem.

Petkovšek's algorithm [7,56], can find hypergeometric solutions, if they exist, for a linear recurrence with polynomial coefficients like the ones returned by Zeilberger's algorithm. A more general class of solutions for linear difference equations with polynomial coefficients, called *d'Alembertian* solutions, is returned by a generalization of this approach [9].

Together with Zeilberger'a algorithm and the more general WZ-summation methods [81], these provide a set of tools to determine closed forms representations for definite sums $\sum_k f(n, k)$ with proper hypergeometric summand f in both its arguments.

2.1. Difference fields and summation

This is a brief overview of the algebraic domains used in Karr's algorithm to represent nested indefinite sums and products. The exposition is based on Karr's treatment of the topic in [41, 42] with minor additions from [17].

Karr's algorithm is the summation analogue of the Risch algorithm [59] for indefinite integration, which builds on the theory of differential field extensions to work with elementary functions in an algebraic setting. Liouville's investigation of the question "Can the indefinite integral of an explicitly given function of one variable always be expressed *explicitly*?" led to the definition of elementary functions. These are functions which are built in a finite number of steps by the application of exponentials, logarithms, algebraic functions and arithmetic operations. In the same spirit, Karr's algorithm builds on towers of difference fields to model nested sums and products.

We will be working with the following definition of difference fields [23].

Definition 2.2. A difference field is a field F together with an automorphism σ of F. The constant field $\text{const}_\sigma(F) \subset F$ is the fixed field of σ.

Note that we assume all fields to have characteristic 0.

Definition 2.3. We say that $E, \tilde{\sigma}$ is a difference field extension of F, σ if and only if F is a subfield of E and $\tilde{\sigma}(a) = \sigma(a)$ for all $a \in F$.

When there is no ambiguity the same symbol will be used for the difference automorphisms of extension fields. For example, we will say E, σ is a difference field extension of F, σ.

2.1.1. Towers of difference fields

Starting from the constants appearing in a given summand f, we will successively add new indeterminates to a difference field in order to create a domain where the summand can be modeled algebraically. Once we have the final tower of difference fields, we need to solve the difference equation $\sigma(g) - g = f$ for an antidifference g from the tower. If it exists, this solution g will lead to a simpler representation of the sum.

The algorithm to solve this difference equation relies on two important aspects in the construction of the tower of difference fields. At each step, the field of constants should remain the same and the new indeterminate added in the current extension should be transcendental over the existing tower. To make sure that these two conditions are satisfied, we investigate the different types of extensions that occur in the construction and provide algorithmically verifiable criteria for these conditions.

In order to model nested indefinite sums and products by adding a new indeterminate to a difference field K, the indeterminate should satisfy a first-order linear difference equation of the form $\sigma(t) = \alpha t + \beta$ where t is the new indeterminate and α and β are elements of the base field K.

Example 2.4. Let K be a field. Consider $K(x), \sigma$ where $\sigma(x) = x + 1$.

(i) (Example 6 from [41]) Let $\beta(x) \in K(x)$. We can extend $K(x)$ by an indefinite sum over $\beta(x)$. Let $t(x) = \sum_{0 \le i < x} \beta(i)$, then

$$\sigma(t(x)) = \sum_{0 \le i < x+1} \beta(i) = \sum_{0 \le i < x} \beta(i) + \beta(x) = t(x) + \beta(x).$$

Hence, an indefinite sum can be represented by the first-order difference equation $\sigma(t) = t + \beta$ for β in the base field $K(x)$.

(ii) (Example 5 from [41]) Let $\alpha(x) \in K(x)$. It is possible to do the same for an indefinite product over $\alpha(x)$. Let $t(x) = \prod_{0 \le i < x} \alpha(i)$, then

$$\sigma(t(x)) = \prod_{0 \le i < x+1} \alpha(i) = \alpha(x) \prod_{0 \le i < x} \alpha(i) = \alpha(x)t(x).$$

The first-order difference equation $\sigma(t) = \alpha t$ with $\alpha \in K(x)$ models an indefinite product.

2.1.2. First order linear extensions

To simplify expression with nested sums and products, it is enough to consider only extensions of difference fields where the automorphism acts on the new indeterminate t as $\sigma(t) = \alpha t + \beta$. These extensions are called *first-order linear* extensions when they also satisfy the two properties mentioned in Section 2.1.1, namely, not introducing new constants and being transcendental over the existing tower.

Definition 2.5. Let $F(t), \sigma$ be a difference field extension of F, σ. This extension is *first-order linear* if and only if

(a) $\sigma(t) = \alpha t + \beta$ where $\alpha \in F^*$ and $\beta \in F$,

(b) t is transcendental over F and

(c) $\text{const}_\sigma(F(t)) = \text{const}_\sigma(F)$.

Note that in [17], *unimonomial extensions* where the constants are not extended correspond to first-order linear extensions.

In a first-order linear extension, if $\beta = 0$, the new indeterminate t models an indefinite product. If $\alpha = 1$ and $\beta \neq 0$, then it models an indefinite sum.

Example 2.6. Let K be a field and σ be the identity map on K.

(i) Take $K(x), \sigma$ with $\sigma(x) = x + 1$. Then x models $\sum_{i=1}^{x} 1$.

(ii) Let $K(x, t), \sigma$ with $\sigma(x) = x + 1$ and $\sigma(t) = 2t$. Then t models the indefinite product $\prod_{i=1}^{x} 2 = 2^x$.

(iii) Let $K(x, t), \sigma$ with $\sigma(x) = x+1$ and $\sigma(t) = (x+1)t$. Then t models the factorial function $\prod_{i=1}^{x} i = x!$.

To see that the extension in the first item is transcendental, assume that $p(x) = 0$ for a nonzero polynomial $p \in K[x]$. Then $\sigma^j(p(x)) = 0$ for all $j \in \mathbb{N}$. Since elements of K are constants, $\sigma^j(p(x)) = p(\sigma^j(x)) = p(x+j) = 0$ for all $j \in \mathbb{N}$. As K has characteristic 0, this means that p has infinitely many distinct zeros, which is impossible. To show that the constants are not extended, assume that $\sigma(p(x)/q(x)) = p(x)/q(x)$ where p, q are coprime polynomials in $K[x]$. Then $p(x + 1)/q(x + 1) = p(x)/q(x)$, which implies that $p(x + 1) = cp(x)$ and $q(x + 1) = cq(x)$ for some $c \in K^*$. Comparing leading coefficients, we see that $c = 1$. Without loss of generality, we can assume $\deg(p) \geq 1$. Write $p(x) = ax^d + bx^{d-1} + \cdots$ where $a \neq 0$. By comparing coefficients of x^{d-1} in the equality $p(x+1) = p(x)$, we see that $ad+b = b$. Hence, $d = 0$, which is a contradiction. So $p(x)/q(x) \in K$ and no new constants were introduced with this extension.

As we can see from this example, verifying that a given extension is transcendental and does not introduce any new constants can be nontrivial even in the simplest cases.

In the following discussion, following Karr's work [41] we will outline the algebraic framework underlying the algorithmic criteria to decide if each layer of a tower of difference field extensions is first-order linear.

The following lemma will be used in the proofs of the main theorems below.

Lemma 2.7. *Let $F(t), \sigma$ be a first-order linear extension of F, σ and $f, g \in F[t]$. If $\sigma(\gcd(f, g)) \in F$ then $\gcd(\sigma(f), \sigma(g)) \in F$.*

Proof. This follows from the fact that σ is also an automorphism of $F[t]$ and it commutes with division in $F[t]$. For completeness, we still give the full proof.

To show that σ is an automorphism of $F[t]$, it is enough to show that the image of a polynomial is a polynomial under σ. Consider a polynomial $p \in F[t]$, which can be written as $p(t) = \sum_{i=0}^{m} p_i t^i$ with $\deg(p) = m$. Then $\sigma(p(t)) = \sum_{i=0}^{m} \sigma(p_i)(\alpha t + \beta)^i$, which is still a polynomial in t.

Let $d = \gcd(f, g)$ and $s = \gcd(\sigma(f), \sigma(g))$. We will show that $\sigma(d)$ and s are associates in $F[t]$. We know that $d \mid f$ and $d \mid g$, so $\sigma(d) \mid \sigma(f)$ and $\sigma(d) \mid \sigma(g)$. Hence, $\sigma(d) \mid s$. Similarly, let $r \in F[t]$ such that $\sigma(r) = s$. Then we have $s \mid \sigma(f)$ and $s \mid \sigma(g)$, so $r \mid f$ and $r \mid g$. It follows that $r \mid d$ and $s \mid \sigma(d)$. □

2.1.3. Homogeneous and inhomogeneous extensions

Given a difference field extension E, σ over F, σ, we call an element $t \in E$ homogeneous, if it satisfies a homogeneous first-order linear equation $\sigma(t) - \alpha t = 0$ with $\alpha \in F$. Applying this concept to extensions is not so straightforward, since even if the extension is defined by an inhomogeneous equation, it can still contain homogeneous elements.

Example 2.8. Take the difference field $\mathbb{Q}, \sigma$ where $\sigma(c) = c$ for all $c \in \mathbb{Q}$. Consider an extension of this field with a new indeterminate t where $\sigma(t) = 2t + 2$. Then $t + 2$ satisfies a homogeneous difference equation over $\mathbb{Q}$. Namely, $\sigma(t + 2) - 2(t + 2) = 0$.

To make the property of being homogeneous inherent to an extension, independent of the way it is constructed, we will call an extension homogeneous if it contains a homogeneous element.

Definition 2.9. Let E, σ be a difference field extension of F, σ. We say that $g \in E$ is *homogeneous over* F if and only if $g \notin F$, but $\sigma(g)/g \in F$. The extension E, σ is called *homogeneous* if and only if there exists $g \in E$ which is homogeneous over F.

The following theorem, which can also be found in [41, Theorem 1] and [42, Theorem 2.1], characterizes when a difference field extension is homogeneous.

Theorem 2.10. *Let $F(t), \sigma$ be a nontrivial difference field extension of F, σ in which $\sigma(t) = \alpha t + \beta$ where $\alpha \in F^*$ and $\beta \in F$. Then the following are equivalent:*

(a) the extension is homogeneous,

(b) there exists $g \in F[t] \setminus F$ with $\sigma(g)/g \in F$,

(c) the equation $\sigma(w) - \alpha w = \beta$ can be solved for $w \in F$.

We follow the proof of Theorem 2.1 in [42].

Proof. We first show the implication $(a) \Rightarrow (b)$. Assume $F(t)$ is homogeneous and take $g_0 \in F(t) \setminus F$ such that $\sigma(g_0)/g_0 \in F$. If t is algebraic, $F(t) = F[t]$, so we have (b) by letting $g = g_0$. If t is transcendental, write $g_0 = g_1/g_2$ with $g_1, g_2 \in F[t]$ and $\gcd(g_1, g_2) = 1$. Now, we have

$$\frac{\sigma(g_0)}{g_0} = \frac{\sigma(g_1)}{\sigma(g_2)} \frac{g_2}{g_1} \in F.$$

Then $\sigma(g_2) \mid \sigma(g_1)g_2$ and $g_1 \mid \sigma(g_1)g_2$. We know that $\gcd(\sigma(g_1), \sigma(g_2)) \in F$ from Lemma 2.7. Therefore $g_1 \mid \sigma(g_1)$ and $\sigma(g_2) \mid g_2$. Hence, $\sigma(g_1)/g_1 \in F$ and $\sigma(g_2)/g_2 \in F$. Since $g_0 \notin F$, at least one of g_1 or g_2 is not in F. Taking g as this element completes the proof of this part.

To show $(b) \Rightarrow (c)$, take $g \in F[t] \setminus F$ with $\sigma(g)/g \in F$. Let $m = \deg(g)$ and write $g = \sum_{i=0}^{m} w_i t^i$. Let $u = \sigma(g)/g \in F$. Comparing coefficients of t^m and t^{m-1} in $\sigma(g) = ug$, we get

$$\begin{aligned} uw_m &= \alpha^m \sigma(w_m) \qquad \text{and} \\ uw_{m-1} &= m\alpha^{m-1}\beta\sigma(w_m) + \alpha^{m-1}\sigma(w_{m-1}). \end{aligned}$$

Combining these two equations, we have

$$\sigma\left(\frac{w_{m-1}}{w_m}\right) - \alpha\frac{w_{m-1}}{w_m} = -m\beta.$$

Now $w = -\frac{w_{m-1}}{mw_m}$ is the solution in F we are looking for.

For the last step, $(c) \Rightarrow (a)$, let $w \in F$ such that $\sigma(w) - \alpha w = \beta$. Then we have

$$\begin{aligned} \sigma(t) - \alpha t &= \beta \\ \sigma(w) - \alpha w &= \beta. \end{aligned}$$

Therefore $\sigma(t - w) - \alpha(t - w) = 0$, so $\sigma(t - w)/(t - w) \in F$. Since $t - w \notin F$, the extension is homogeneous. □

Note that the last part of the proof shows how to change the basis so that any homogeneous extension where $\sigma(t) = \alpha t + \beta$ can be transformed to one where $\beta = 0$. Using this trick, we will only work with homogeneous extensions with $\sigma(t) = \alpha t$.

Example 2.11. Continuing Example 2.8, the equation $\sigma(w) - 2w = 2$ is satisfied by $-2 \in \mathbb{Q}$. Then the construction in the last part of the proof gives us the homogeneous element $t + 2$.

Homogeneous extensions that are also first-order linear are called Π-extensions since they model products such as $n!$ or 2^n.

Definition 2.12 (Π-extension)**.** A difference field extension $F(t), \sigma$ of F, σ is called a Π-extension if and only if

(a) the extension is first-order linear,

(b) $\sigma(t) = \alpha t$ for some $\alpha \in F^*$.

When is a homogeneous extension first-order linear? In other words, when is the constant field not extended and the extension is transcendental? The following definition is needed to answer this question.

Definition 2.13. Let F, σ be a difference field. An element $a \in F$ is called a *σ-radical over F* if and only if $\sigma(z) = a^n z$ for some $z \in F^*$ and a positive integer n.

Karr uses the *homogeneous group* [41, Definition 8], defined as

$$H(F, \sigma) = \{\, \sigma(g)/g \mid 0 \neq g \in F \,\},$$

to state the criteria for a homogeneous extension to be transcendental without introducing new constants. This concept also appears in further investigations of properties required for Karr's algorithm such as Π and Σ-regularity. Since we will not delve into these technical details, the definition of a σ-factorial will suffice for our purposes. Note that if a is a σ-radical over F, then $a^n \in H(F, \sigma)$ for some positive integer $n \in \mathbb{N}$.

Example 2.14. Consider the difference field $\mathbb{Q}(t), \sigma$ where $\sigma(t) = 4t$. Note that t models 4^n in this case. Now, 2 is a σ-radical over $\mathbb{Q}(t)$, since $\sigma(t) = 2^2 t$.

Now, we can answer the question above. This theorem can be found in [41, Theorem 2] and [42, Theorem 2.2].

Theorem 2.15. *Let $F(t), \sigma$ be a nontrivial difference field extension of F, σ and $\sigma(t) = \alpha t$ for $\alpha \in F^*$. This extension is first-order linear if and only if α is not a σ-radical over F.*

We follow Karr's proof from [42].

Proof. Let $F(t), \sigma$ be a first-order linear extension of F, σ. Suppose for a contradiction that α is a σ-radical over F. Let $w \in F$ be such that $\sigma(w)/w = \alpha^n$ for a positive integer n. Then we have

$$\sigma\left(\frac{t^n}{w}\right) = \frac{\alpha^n t^n}{\sigma(w)} = \frac{t^n}{w}.$$

Therefore $t^n/w \in \mathrm{const}_\sigma(F(t))$. Hence, either $t^n/w \in \mathrm{const}_\sigma(F) \subset F$ or the constant field is extended. If $t^n/w \in F$, then t is algebraic over F. So we have a contradiction in both cases and α cannot be a σ-radical over F.

For a contradiction in the other direction, assume that the extension is not first-order linear. Suppose that t is algebraic over F, with minimal polynomial $g(z) = \sum_{i=0}^{m} w_i z^i \in F[z]$ of degree m. Since $g(t) = 0$, we can write

$$0 = \sigma(g(t)) = \sum_{i=0}^{m} \sigma(w_i) \alpha^i t^i.$$

Let $h(z) = \sum_{i=0}^{m}(\sigma(w_i)\alpha^i)z^i$. Note that $h(t) = 0$. Then h must be a multiple of g, because g is the minimal polynomial of t. Since g is monic, we conclude that $g = \alpha^m h$. By comparing coefficients of z^i in this identity, we get

$$\alpha^m w_i = \sigma(w_i)\alpha^i \text{ for } 0 \leq i \leq m.$$

We distinguish two cases, either $g(z) = z$, or there is some $i < m$ for which $w_i \neq 0$. In the first case, the extension is trivial. In the latter, $\alpha^{m-i} = \sigma(w_i)/w_i$, so α is a σ-radical over F and we have a contradiction.

Now suppose that t is transcendental over F, but the constant field is extended. Then there exists $g_0 \in F(t) \setminus F$ such that $\sigma(g_0) = g_0$. Write $g_0 = \frac{g_1}{g_2}$ with $g_1, g_2 \in F[t]$ and $\gcd(g_1, g_2) = 1$. Then

$$\frac{\sigma(g_0)}{g_0} = \frac{\sigma(g_1)g_2}{\sigma(g_2)g_1} = 1 \in F. \tag{2.2}$$

Since $\gcd(\sigma(g_1), \sigma(g_2)) \in F$ by Lemma 2.7, $\sigma(g_1)/g_1$ and $\sigma(g_2)/g_2$ are in F. Without loss of generality, we can assume that g_1 is monic and $\deg(g_1) \geq 1$. If g_1 has more than one non-zero coefficient we can argue as above to show that α is a σ-radical over F. Otherwise, $g_1 = t^m$ for some $m > 0$. Since $\gcd(g_1, g_2) = 1$, the constant term w of g_2 is non-zero. From Equation (2.2), we know that

$$\alpha^m = \frac{\sigma(t^m)}{t^m} = \frac{\sigma(g_1)}{g_1} = \frac{\sigma(g_2)}{g_2}.$$

From $\sigma(g_2) = \alpha^m g_2$, we have $\sigma(w) = \alpha^m w$ and α is a σ-radical over F. □

Example 2.16. Consider a difference ring extension $\mathbb{Q}[t], \sigma$ of the constant difference field $\mathbb{Q}, \sigma$, with the action of σ defined as $\sigma(t) = -t$. Note that -1 is a σ-radical over $\mathbb{Q}$ since $\sigma(c) = (-1)^2 c$ for all $c \in \mathbb{Q}$. This extension is not first-order linear by Theorem 2.15. It is either algebraic or new constants are introduced in $\mathbb{Q}[t]$. If we assume that t is transcendental, t^2 is indeed a new constant. But t can be algebraic over $\mathbb{Q}$, for example $t = \sqrt{2}$. Then σ is the automorphism of the number field $\mathbb{Q}(\sqrt{2})$ which maps $\sqrt{2}$ to its conjugate $-\sqrt{2}$, and t^2 is a constant – but not a new one.

When t models $(-1)^n$, using the fact that $((-1)^n)^2 = 1$, this new constant can be eliminated by considering the extension to be the quotient ring $\mathbb{Q}[t] \simeq \mathbb{Q}[x]/\langle x^2 - 1\rangle$. This way the constant field is not extended, but the extension is algebraic. How to extend Karr's algorithm to work with such rings is treated in Chapter 3.

Karr provides an algorithm to test if an element a in a certain difference field F, σ is a σ-radical. A short summary of this algorithm can be found in Section 2.2.2. This algorithm is also a crucial step in determining the minimal polynomial of the algebraic extension when α is a σ-radical. Details of this procedure can be found in Section 3.2.

It is much easier to decide if inhomogeneous extensions are first-order linear. The following theorem also appears in [41, Theorem 3] and [42, Theorem 2.3].

Theorem 2.17. *Let $F(t), \sigma$ be an inhomogeneous extension of F, σ in which $\sigma(t) = \alpha t + \beta$ where $\alpha, \beta \in F^*$. Then the extension is first-order linear.*

We follow the proof of Theorem 2.3 in [42].

Proof. We need to show that the constant field is not extended and t is not algebraic over F. If the constant field is extended, there exists $g \in F(t) \setminus F$ such that $\sigma(g) = g$, that is, $\sigma(g)/g = 1 \in F$. Hence the extension is homogeneous. If t is algebraic over F, let $g(z) = z^m + \sum_{i=0}^{m-1} w_i z^i \in F[z]$ be its minimal polynomial. Then $g(t) = 0$ and we have

$$0 = \sigma(g(t)) = (\alpha t + \beta)^m + \sum_{i=0}^{m-1} \sigma(w_i)(\alpha t + \beta)^i.$$

If we write $h(z) = (\alpha z + \beta)^m + \sum_{i=0}^{m-1} \sigma(w_i)(\alpha z + \beta)^i$, then $h(t) = 0$. So, h must be a multiple of g. Since g is monic and $\deg(g) = \deg(h)$, we know that $h = \alpha^m g$. In this last identity, matching coefficients of degree $m-1$ gives

$$\alpha^m w_{m-1} = \sigma(w_{m-1})\alpha^{m-1} + m\alpha^{m-1}\beta.$$

This can be rewritten to $\sigma(w_{m-1}) = \alpha w_{m-1} - m\beta$. Then we find that

$$\sigma\left(t + \frac{w_{m-1}}{m}\right) = \alpha t + \beta + \alpha\frac{w_{m-1}}{m} - \beta = \alpha\left(t + \frac{w_{m-1}}{m}\right)$$

and the extension is homogeneous.

□

The extensions that are used in Karr's algorithm are a subset of inhomogeneous extensions. In particular, α is not allowed to be a σ-radical even though the inhomogeneous extension is first-order linear.

Definition 2.18 (Σ-extension)**.** A difference field extension $F(t), \sigma$ of F, σ is called a Σ-extension if and only if

(a) the extension is inhomogeneous and

(b) if α is a σ-radical over F, then there exists $f \in F$ such that $\sigma(f)/f = \alpha$.

While most homogeneous extensions can also be handled by the hypergeometric summation algorithms mentioned in Section 2, expressions modeled by inhomogeneous extensions are beyond the reach of these methods. For example, Karr's algorithm can work with the harmonic numbers, $H_i = \sum_{1 \le j < i+1} \frac{1}{j}$.

Example 2.19. Starting from the constant difference field $\mathbb{Q}, \sigma$ where $\sigma(c) = c$ for all $c \in \mathbb{Q}$, we can construct the following tower of inhomogeneous extensions:

(i) extend $\mathbb{Q}, \sigma$ with a new indeterminate i such that $\sigma(i) = i + 1$

(ii) add a new indeterminate h to $\mathbb{Q}(i), \sigma$ where $\sigma(h) = h + \frac{1}{i+1}$.

Then $h \in \mathbb{Q}(i, h)$ models the harmonic numbers H_i.

$$\sigma\left(\sum_{1 \le j < i+1} \frac{1}{j}\right) = \sum_{1 \le j < i+2} \frac{1}{j} = \left(\sum_{1 \le j < i+1} \frac{1}{j}\right) + \frac{1}{i+1}$$

Finally, we can define the domain where Karr's algorithms work.

Definition 2.20 ($\Pi\Sigma$-extension)**.** A difference field extension $F(t), \sigma$ over F, σ is called a $\Pi\Sigma$-extension if and only if it is a Π-extension or a Σ-extension. Given a constant field K, we say that F, σ is a $\Pi\Sigma$-field over K if and only if there is a tower of fields $K = F_0 \subset \cdots \subset F_n = F$ in which F_i, σ is a $\Pi\Sigma$-extension of F_{i-1}, σ for $i = 1, \ldots, n$.

The following theorem from [41, Theorem 9 (a)] and [42, Lemma 3.5] will be needed to solve difference equations over algebraic extensions later on.

Theorem 2.21. *Let F, σ be a $\Pi\Sigma$-field over a constant field K. Then F, σ^k is a $\Pi\Sigma$-field over K whenever $k \neq 0$.*

The proof relies on technical properties of $\Pi\Sigma$-fields which are beyond the scope of this simple summary. For details, please refer to [42, Lemma 3.5].

2.2. Algorithms for σ-radicals

In this section we summarize the algorithm to test if an element α in a $\Pi\Sigma$-field F, σ is a σ-radical over F. This test depends on the concept of a σ-factorization, which is also briefly described.

Even though we only need to find out if there exists a nonzero $n \in \mathbb{N}$, and $g \in F$ such that $\sigma(g) = \alpha^n g$, our investigation will lead to a more general case of this equation. Namely, given $\alpha_1, \ldots, \alpha_k \in F^*$, find the set of $(n_1, \ldots, n_k) \in \mathbb{Z}^k$ such that

$$\sigma(g) = \alpha_1^{n_1} \cdots \alpha_k^{n_k} g \text{ for some } g \in F^*.$$

This equality can be considered to be the product analogue of the creative telescoping identity

$$\sigma(g) - g = c_1 f_1 + \cdots + c_k f_k,$$

where $f_1, \ldots, f_k \in F$ are given and we find the set of solutions $g \in F$ and $c_1, \ldots, c_k \in \operatorname{const}_\sigma(F)$. Similarly, for summation using Karr's algorithm, this general identity needed to be considered instead of the simple telescoping identity $\sigma(g) - g = f$.

2.2.1. σ-factorization

In [41], Karr describes algorithms to decide if an element f in a $\Pi\Sigma$-extension $F(t), \sigma$ of F, σ can be transformed to $g \in F(t)$ using the automorphism σ. More explicitly, it is possible to decide algorithmically if there is an $n \in \mathbb{N}$ such that

$$\sigma^n(f) = \alpha g \text{ for some } \alpha \in F.$$

In this case, f is said to be *shift equivalent* to g over F and n is called the *specification* of the equivalence.

The details of this algorithm can be found in [41, Theorem 4]. Karr's proof, which also extends to [42], is very technical. For our purposes, it is enough to treat this as a black box and omit the details. Note that an implementation of this method can be found and inspected in the package described in the appendix, as the function call `spec` on $\Pi\Sigma$-field elements. Example A.1 shows how this function can be called.

Example 2.22. Consider the $\Pi\Sigma$-field $\mathbb{Q}(x,t),\sigma$ over $\mathbb{Q}$, where $\sigma(x) = x+1$ and $\sigma(t) = 2t$. Take $f = xt + 1$, and $g = 8tx^2 + 24tx + x$. Now, we have

$$\sigma^3(f) = \frac{1}{x}g.$$

The *shift equivalence* algorithm can be used to compute a factorization of a given element $f \in F(t)$ with the factors separated into equivalence classes under σ. This decomposition is called the *σ-factorization.*

Definition 2.23. Let $F(t),\sigma$ be a $\Pi\Sigma$-extension of F,σ and $f_1,\dots,f_n \in F(t)$. Suppose that

$$f_i = u_i t^{e_i} \prod_{j,k} \sigma^k(g_j^{e_{i,j,k}}), \text{ for } i = 1,\dots,n,$$

where $e_{i,j,k} \in \mathbb{Z}$, $u_i \in F$ and $g_j \in F[t] \setminus F$. We say that $((g_j)_j, (u_i, e_i, (e_{i,j,k})_{j,k})_i)$ is a σ-factorization of $f_1,\dots,f_k$ if and only if

(i) for all $j \in \mathbb{N}$, g_j is monic, irreducible and $\deg(g_j) > 0$,

(ii) for all $j \in \mathbb{N}$, there exists some i,k such that $e_{i,j,k} \neq 0$,

(iii) if the extension is homogeneous, $t \neq g_j$ for any $j \in \mathbb{N}$,

(iv) if the extension is inhomogeneous, $e_i = 0$ for all $i = 0,\dots,n$,

(v) for $i \neq j$, g_i is not shift equivalent to g_j.

Example 2.24. Consider the $\Pi\Sigma$-field $\mathbb{Q}(x,t),\sigma$ from the previous example. Then the σ-factorization for $f = xt(t+x^2)\sigma(xt(t+x^2))$ is

$$f = 2x(x+1)t^2(t+x^2)\sigma(t+x^2),$$

where $u = 2x(x+1)$, $e = 2$, and we have $g_0 = t + x^2$, $e_{0,0,0} = 1$, $e_{0,0,1} = 1$.

The following theorem appears as Theorem 7 in [41] and states the existence and uniqueness of σ-factorizations of elements in $\Pi\Sigma$-fields.

Theorem 2.25. *Let F,σ be a $\Pi\Sigma$-field and $F(t),\sigma$ a $\Pi\Sigma$-extension of F,σ and $k \in \mathbb{N}$ a positive integer. For every $(f_1,\dots,f_k) \in F(t)^k$, there exists a σ-factorization of the form $((g_j)_j, (u_i, e_i, (e_{i,j,k})_{j,k})_i)$, where g_j, u_i, e_i and $e_{i,j}$ are defined as in Definition 2.23. This σ-factorization is unique up to the u_i, permutation of the g_j, and translation of the last index of $e_{i,j,k}$.*

Once an algorithm for deciding shift equivalence is known, the σ-factorization can be computed over any domain where factorization of elements is possible. First, a regular factorization of each f_i is obtained. This factorization provides the u_i and e_i. The remaining factors become the g_j. Then, the g_j are put into equivalence classes with respect to the shift equivalence relation. This determines the power k of σ, that corresponds to each pair i, j.

Note that this is the only point in Karr's algorithm that polynomial factorization is used. Whereas the Risch integration algorithm was improved to eliminate calls to this possibly expensive procedure, the question of eliminating factorization in Karr's algorithm is still open. Other approaches to providing such a decomposition, especially in the rational difference field $\mathbb{Q}(x), \sigma$ with $\sigma(x) = x + 1$ can be found in [31, 50, 53], though these have not been extended to elements of $\Pi\Sigma$-fields.

2.2.2. Tests for σ-radicals

Given an element α of a $\Pi\Sigma$-field F, σ, the σ-factorization algorithm can be used to decide if α is a σ-radical over F. This procedure will actually answer a more general problem, which comes up when going through the levels of a $\Pi\Sigma$-tower recursively. Given $\alpha_1, \ldots, \alpha_k$, we will describe the set of $n_1, \ldots, n_k \in \mathbb{Z}^k$ such that $\alpha_1^{n_1} \cdots \alpha_k^{n_k} = \sigma(g)/g$ for some $g \in F^*$.

Applying this procedure for $k = 1$, we get the set of all $n \in \mathbb{Z}$ such that $\sigma(g) = \alpha^n g$ for some $g \in F^*$, even though we only wanted to know if there were any such n. This extra information will be useful in the construction in Section 3.2 to find the minimal polynomial of an algebraic element.

Definition 2.26. Let F, σ be a $\Pi\Sigma$-field and $\alpha_1, \ldots, \alpha_k \in F^*$. We define the set of k-tuples

$$M((\alpha_1, \ldots, \alpha_k), F) = \left\{ (n_1, \ldots, n_k) \in \mathbb{Z}^k \,\middle|\, \exists g \in F^* \text{ such that } \sigma(g) = \alpha_1^{n_1} \cdots \alpha_k^{n_k} g \right\}.$$

This set is a $\mathbb{Z}$-module, so we have access to a large collection of algorithms to compute canonical bases, intersections and unions of subspaces.

Lemma 2.27. *Let F be a $\Pi\Sigma$-field and $\alpha_1, \ldots, \alpha_k \in F^*$. Then $M((\alpha_1, \ldots, \alpha_k), F)$ forms a $\mathbb{Z}$-submodule of $\mathbb{Z}^k$.*

Example 2.28. Consider the $\Pi\Sigma$-field $\mathbb{Q}(x), \sigma$ where $\sigma(x) = x+1$. Let $\alpha_1 = (x+1)(x+2)$, $\alpha_2 = (x + 2)(x + 3)$, and $\alpha_3 = x(x + 1)$. Then a basis for $M((\alpha_1, \alpha_2, \alpha_3), \mathbb{Q}(x))$ is given by

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

The following theorem [41, Theorem 8] reduces the problem of computing the module $M((\alpha_1, \ldots, \alpha_k), F(t))$ over a $\Pi\Sigma$-extension $F(t)$ to instances of this problem over the base field F. We use $Ann((n_1, \ldots, n_k))$ to denote the annihilator of the vector $(n_1, \ldots, n_k) \in \mathbb{Z}^k$.

Theorem 2.29. *Let $F(t), \sigma$ be a $\Pi\Sigma$-extension of F, σ and let $(\alpha_1, \dots, \alpha_k) \in F(t)^k$ with $\alpha_l \neq 0$ for $0 \leq l < k$ have a σ-factorization $((g_j)_j, (u_i, e_i, (e_{i,j,k})_{j,k})_i)$ with $e_{i,j,k} \in \mathbb{Z}$, $u_i \in F$ and $g_j \in F[t]$. Then*

$$M((\alpha_1, \dots, \alpha_k), F(t)) = M_1 \cap M_2 \cap M_3,$$

where

(a) (i) $M_1 = M((u_1, \dots, u_k), F)$ for a Σ-extension,

(ii) $M_1 = \{\, (n_1, \dots, n_k) \,|\, (n_1, \dots, n_k, d) \in M((u_1, \dots, u_k, 1/\alpha), F) \,\}$ for a Π-extension with $\sigma(t) = \alpha t$,

(b) $M_2 = Ann((e_1, \dots, e_k))$,

(c) $M_3 = \cap_j Ann((\sum_l e_{1,j,l}, \dots, \sum_l e_{k,j,l}))$.

Note that in the case of a Π-extension for condition (a) above, we add $1/\alpha$ to the input tuple $(u_1, \dots, u_k)$ and consider a similar problem over the base field, albeit with dimension increased by 1.

This theorem can be applied recursively to reduce the problem to the constant field. An algorithm to find $M((\alpha_1, \dots, \alpha_k), K)$ where K is a rational function field over an algebraic number field with $\sigma(k) = k$ for all $k \in K$ is described in [67, Proposition 3.1 and Theorem 3.2].

Example 2.30 (Example 8 in [41]). Consider the extension of $\mathbb{Q}(x), \sigma$ where $\sigma(x) = x+1$ with $x!$. Then $\sigma(x!) = (x+1)x!$. In order to show that this extension is indeed transcendental and $(x+1)$ is not a σ-radical, we apply the previous theorem with $\alpha_1 = (x+1)$. The σ-factorization of $(x+1)$ gives $f_1 = x+1$, $u_1 = 1$, $e_1 = 0$, $e_{1,1,0} = 1$. From condition (c) of Theorem 2.29, we have $M_3 = Ann(e_{1,1,0}) = Ann(1) = \{0\}$. Then $M(x+1, \mathbb{Q}(x)) = M_1 \cap M_2 \cap \{0\} = \{0\}$. Hence, there is no nonzero $n \in \mathbb{N}$ such that $\sigma(g) = \alpha^n g$ for some $g \in \mathbb{Q}(x)$ and $(x+1)$ is not a σ-radical.

Example 2.31. Consider the $\Pi\Sigma$-field $\mathbb{Q}(t)$ where $\sigma(t) = 4t$. Here, t models 4^n. Trying to extend this field with 2^n, which has the shift behavior $\sigma(2^n) = 2(2^n)$, we check if 2 is a σ-radical over $\mathbb{Q}(t)$. From condition (a) of Theorem 2.29, we have

$$M_1 = \{\, n \,|\, (n, d) \in M((2, 1/4), \mathbb{Q}) \,\} = \{\, n \,|\, (n, d) \in \{c(2, 1) \text{ for } c \in \mathbb{Z}\} \,\} = 2\mathbb{Z},$$

and $M_2 = M_3 = \mathbb{Z}$. Then $M((2), \mathbb{Q}(t)) = 2\mathbb{Z} \cap \mathbb{Z} \cap \mathbb{Z} = 2\mathbb{Z}$. Hence for $n = 2$, $\sigma(g) = 2^n g$ for some $g \in \mathbb{Q}(t)$, namely, for $g = t$.

This function is called `hom_group_exponents` in the implementation described in the appendix. Example A.2 demonstrates a call to this function.

3. Algebraic extensions for summation in finite terms

This chapter presents an extension of Karr's symbolic summation framework to allow algebraic extensions in the tower of difference fields.

The problem

In Section 2.1, we have seen that while building a tower of difference fields to represent a summand, Karr's framework only allows transcendental extensions. This limits the set of expressions which can be modeled by these extensions. For example, $(-1)^n$, a common component of combinatorial identities, leads to an algebraic extension since $((-1)^n)^2 = 1$. Any summand which contains $(-1)^n$ is beyond the reach of Karr's method.

When a $\Pi\Sigma$-field is extended with a term like $(-1)^n$, we either get an algebraic extension or introduce new constants. This violates the basic assumptions of Karr's construction, so his algorithm can no longer be used to solve first-order linear difference equations over these domains.

In this case, the problem is twofold. A method to solve difference equations over an algebraic extension as well as a way to build towers on this extension is needed. To be able to extend algebraic extensions, in turn, requires decision procedures to check if the resulting structure is within the scope of our algorithms.

Going back to the $(-1)^n$ example, the action of the shift map σ on an indeterminate t modeling $(-1)^n$ is $\sigma(t) = -t$. However, the coefficient -1 is a σ-radical in every extension, since $\sigma(c) = (-1)^2 c$ for every $c \in \mathbb{Q}$. Then, by Theorem 2.15, we cannot form a homogeneous extension to model $(-1)^n$ within Karr's framework. Using the algebraic relation $t^2 = 1$, we can form the extension $R = \mathbb{Q}[t]/\langle t^2 - 1\rangle$. This ring, R, contains zero divisors, because the minimal polynomial of this extension, $t^2 - 1$ is not irreducible over $\mathbb{Q}[t]$.

In order to simplify sums in this case, we need to solve difference equations over these rings with zero divisors and further transcendental or algebraic extensions of these rings.

A solution

A generalization of the formulation in [16] for solving differential equations over an algebraic extension presented in [73] is the first step towards a solution. Viewing the algebraic extension on top of the tower as a finite-dimensional algebra, the problem of solving a first-order linear difference/differential equation over this extension can be reduced to solving a system of first-order equations over the last transcendental extension in the tower.

This system can be transformed to an n-th order difference equation, where n is the degree of the algebraic extension. Even though Karr's algorithm is only designed to handle first-order linear equations over $\Pi\Sigma$-fields, the special form of this equation falls within its range as well. Hence, the problem is reduced to one we know how to solve.

To work with the algebraic extension as a finite-dimensional algebra, a basis should be chosen. A natural option is to use the power basis $1, z, z^2, \ldots, z^{n-1}$ where z is the generator of the algebraic extension. However, since this ring extension has zero divisors, an orthogonal basis of idempotents can also be used. This basis provides a decomposition of the ring into integral domains.

For example, the ring $R = \mathbb{Q}[t]/\langle t^2 - 1\rangle$ can be decomposed into two integral domains so that $R \simeq R_0 \oplus R_1$. The components R_0 and R_1 are generated by the elements $t_0 = \frac{t+1}{2}$ and $t_1 = \frac{-t+1}{2}$ respectively. Note that $t_0 t_1 = 0$ in R.

It turns out that the shift automorphism σ permutes this basis and σ^n leaves the components invariant [78, Corollary 1.16], where n is the degree of the algebraic extension. Each of these components is a $\Pi\Sigma$-field, so Karr's machinery works without modification over them. This structure can be used to build towers on the algebraic extension. In particular, by building a $\Pi\Sigma$-extension on each component in the case of a transcendental extension.

Continuing the previous example, consider the extension $R[s], \sigma$ of R, σ where $\sigma(s) = s - \frac{t}{n+1}$. The indeterminate s models the shift behavior of the expression $\sum_{1 \leq i < n+1} \frac{(-1)^i}{i}$. Then, $R[s]$ can be decomposed in a similar way to R. Namely, $R[s] \simeq R_0[h] \oplus R_1[h]$, where $R_i[h], \sigma$ is a difference ring extension of R_i with $\sigma(h) = -h + \frac{1}{n+1}$ for $i = 0, 1$.

Other approaches

While the problem of algebraic extensions for Karr's algorithm has not been addressed directly, there are improvements to work with unspecified sequences [44,45] and radical expressions [46]. Note that the radical expressions such as $\sqrt{k}$ covered in [46] lead to infinitely generated difference rings, not algebraic extensions as they would in the integration case.

These approaches follow the recipe laid out in [45], by calling a domain *σ-computable* [45, Definition 1] if certain subproblems in Karr's framework can be answered algorithmically. Providing solutions to these basic subproblems, such as shift equivalence and the orbit problem, makes it possible to use Karr's algorithm in its original form. The open problems listed in [62, page 240] are also an effort in this direction. In contrast, our strategy is to reduce the task of solving a linear difference equation to one in a $\Pi\Sigma$-field where Karr's algorithm already works.

In Sections 3.1 and 3.4, we present a method adapted from [16] to solve difference equations over algebraic extensions. This method is similar to the approach used in [66] to solve difference equations over a $\Pi\Sigma$-field extended by a term satisfying a higher order recurrence. We provide an explicit formulation of the system that is formed and show that solving a single first-order linear difference equation over a $\Pi\Sigma$-field is sufficient in our case.

Adjoining $(-1)^n$ to a difference field F, σ where $\sigma(n) = n + 1$, is similar to taking $q = -1$ in a q-difference equation. Chapter 6 of [39] studies the case of q-difference equations over $\mathbb{C}(x)$ where q is an n-th root of unity. In this case, the Picard-Vessiot ring, the splitting ring of solutions of a q-difference equation, is not unique. This presents a problem defining the Galois group of the equation, which is used to classify its solutions. Being closely tied to concrete applications, we are able to chose a canonical construction for our problem and give an algorithm for solving difference equations involving primitive roots of unity over $\Pi\Sigma$-fields. Note that restricting to a primitive root of order k makes the Galois group cyclic of order k, greatly simplifying our task.

Liouvillian sequence solutions of a difference equation over $\mathbb{C}(x)$ are investigated in [40]. Liouvillian sequences are constructed iteratively using steps similar to the construction of Π and Σ extensions, and additionally interlacing of sequences. Interlacing sequences also encapsulate algebraic extensions. For example, the sequence obtained from $(-1)^n$ is $(1, -1, 1, -1, \dots)$, which is the 2-interlacing of the constant 1 sequence and the constant -1 sequence. An algorithm to find the Liouvillian solutions of an n-th order difference equation over $\mathbb{C}(x)$ is also presented in [40]. Our main interest is first-order equations over the more general setting of $\Pi\Sigma$-fields. Though, the reduction described in Section 3.2 can be used to reduce an n-th order equation to an equivalent one over a tower with only transcendental extensions.

3.0. Preliminaries

Before starting with the treatment of algebraic extensions in Karr's symbolic summation framework, this section briefly presents preliminary facts that will be used in the rest of this chapter.

3.0.1. Commutative algebra

The following facts from commutative algebra will be needed for the proof of Theorem 3.33. We refrain from providing proofs since they can easily be found in textbooks, for instance, see [11, 27, 82]. We assume all rings to be commutative in this section.

Definition 3.1. Let R be a ring. An ideal I of R is called *prime* if for all $x, y \in R$, when $xy \in I$ either $x \in I$ or $y \in I$. It is called *radical* if for all $x \in R$, when $x^n \in I$ for some $n \in \mathbb{N}$ then $x \in I$.

Lemma 3.2. *Let R be a ring and I a prime ideal of R. Then the quotient ring R/I is a domain.*

Definition 3.3. Let R be a ring. An element $x \in R$ is *nilpotent* if there exists an integer $n > 0$ such that $x^n = 0$. The set of all nilpotent elements in a ring R is called the *nilradical* of R.

Definition 3.4. Let R be a ring and I, J ideals of R. We say that I and J are *comaximal* if $I + J = R$.

Theorem 3.5. [82, Theorem 31, Ch. III] *Let R be a ring with identity, and let $I_1, \ldots, I_n$ be ideals in R. The I_i are pairwise comaximal if and only if their radicals are. If an ideal B is comaximal with each I_i, then it is comaximal with $I_1 \cap \cdots \cap I_n$ and $I_1 \cdots I_n$. If $I_1, \ldots, I_n$ are pairwise comaximal, then*

$$I_1 \cap \cdots \cap I_n = I_1 \cdots I_n.$$

If, moreover, $b_1, \ldots, b_n$ are elements of R, then there exists an element b in R such that

$$b \equiv b_i \pmod{I_i}, \quad i = 1, \ldots, n.$$

Here, the notation $a \equiv b \pmod{I}$ means that $a - b \in I$.

Definition 3.6. Let R be a ring. We say that an ideal I of R is *primary* if for all $x, y \in R$ when $xy \in I$ either $x \in I$ or $y^n \in I$ for some $n \in \mathbb{N}$.

Note that the radical of a primary ideal is a prime ideal.

Theorem 3.7. [82, Theorem 4, Ch. IV] *In a ring R with ascending chain condition every ideal admits an irredundant representation as finite intersection of primary ideals.*

Definition 3.8. If J is a primary ideal, then its radical $I = \sqrt{J}$ is called the *associated prime ideal* of J, and we say that J *is primary for* I.

Definition 3.9. Let R be a ring and I, J ideals of R. The *ideal quotient*, denoted $I : J$ is defined as the set $\{ r \in R \,|\, rJ \subset I \}$.

Theorem 3.10. [82, Theorem 6, Ch. IV] *Let R be an arbitrary ring and I an ideal of R admitting an irredundant primary representation $\cap_i J_i$ and let $P_i = \sqrt{J_i}$. For a prime ideal P of R to be equal to some P_i it is necessary and sufficient that there exist an element c of R not contained in I and such that the ideal $I : \langle c \rangle$ is primary for P. The prime ideals P_i are therefore uniquely determined by I.*

3.0.2. D-rings

Even though the structure of difference and differential algebras are quite different, methods to solve difference and differential equations share many common aspects. Our main focus is on solving difference equations. Though generalizations of algorithms for differential equations, such as the one described in Section 3.1, play an important role. We will use the framework introduced in [19] to study a common setting for differential and difference equations [17].

Definition 3.11. Let R be a commutative ring (resp. field), σ an endomorphism of R. A σ-derivation on R is a map $\delta : R \to R$ such that

$$\delta(a+b) = \delta(a) + \delta(b) \quad \text{and} \quad \delta(ab) = \sigma(a)\delta(b) + \delta(a)b$$

for any $a, b \in R$. The triple (R, σ, δ) is called a *D-ring (resp. D-field).*

When talking about difference (resp. differential) rings, we will often drop the redundant map δ (resp. σ).

Example 3.12. D-rings model both difference and differential rings, as well as a hybrid structure.

(i) If σ is the identity map on R, then δ is a usual derivation on R, which satisfies the Leibniz rule. In this case, (R, δ) is called a differential ring.

(ii) For any endomorphism σ of R, $\delta = 0$ is a σ-derivation. In this case, (R, σ) is called a difference ring.

(iii) If R is commutative, σ an endomorphism of R, and $\alpha \in R$, the map $\delta_\alpha = \alpha(\sigma - 1_R)$ given by $\delta_\alpha(a) = \alpha(\sigma(a) - a)$ is a σ-derivation.

The three examples above exhaust all possible σ-derivations over a commutative ring [17, Lemma 2]. In particular, the framework of D-rings do not allow working with the usual derivation and shift with respect to the same variable.

Example 3.13. Take $R = \mathbb{Q}(x)$, with the maps $\sigma : x \mapsto x + 1$ and $\delta : x \mapsto 1$. Then,

$$\delta(x^2) = \delta(xx) = \sigma(x)\delta(x) + \delta(x)x = 2\delta(x)x + \delta(x) = 2x + 1.$$

For the usual derivation, we would expect the result to be $2x$.

Extensions of D-rings are defined in a similar way to difference field extensions from Definition 2.3.

Definition 3.14. Let (R, σ, δ) and (R', σ', δ') be D-rings. We say that (R', σ', δ') is a D-ring extension of (R, σ, δ) if R is a subring of R' where $\sigma'(a) = \sigma(a)$ and $\delta'(a) = \delta(a)$ for any $a \in R$.

To keep the notation simple, we will often use the same symbols for the endomorphisms and associated derivations on R' and R.

Definition 3.15. An ideal I of a D-ring (R, σ, δ) is called a *D-ideal* if it is closed under σ and δ. A D-ring R is called *simple* if it has no D-ideals other than the zero ideal and R itself.

If R, σ, δ is a D-ring, the quotient R/I is also a D-ring if and only if the ideal I is a D-ideal.

Definition 3.16. Let (R, σ, δ) be a D-ring (resp. D-field). An element $a \in R$ is called *invariant* if $\sigma(a) = a$. The set

$$\mathrm{const}_{\sigma,\delta}(R) = \{\, a \in R \,|\, \sigma(a) = a \text{ and } \delta(a) = 0 \,\}$$

is called the *constant subring (resp. subfield)* of R with respect to σ and δ.

In Sections 3.1 and 3.4, the ring of operators will be mentioned briefly. We adapt the definition of a skew polynomial ring introduced by Ore [52] to the setting of D-rings.

Definition 3.17. Let (R, σ, δ) be a D-ring and X an indeterminate over R. The *left skew polynomial ring over* R, denoted $R[X; \sigma, \delta]$ is the ring of polynomials in X over R with the usual polynomial addition and multiplication given by

$$Xa = \sigma(a)X + \delta(a) \text{ for any } a \in R.$$

The multiplication in a skew polynomial ring can be uniquely extended to multiplication of monomials by

$$(aX^n)(bX^m) = (aX^{n-1})(Xb)X^m = (aX^{n-1})(\sigma(b)X^{m+1} + \delta(b)X^m) \text{ for } m, n > 0$$

and to arbitrary polynomials by distributivity.

Example 3.18. (i) For a difference ring R, σ, the skew polynomial ring $R[S; \sigma, 0]$ is the ring of linear ordinary recurrence operators, where S denotes the shift operator.

(ii) Let R, δ be a differential ring, then $R[D; 1_R, \delta]$ is the ring of linear ordinary differential operators where D is the differential operator.

3.1. Difference/differential equations over algebraic extensions

This section presents a generalization of the method of [16], which in turn is based on [73], to solve the Risch differential equation over algebraic extensions to the setting of D-rings. As a result, we get a method to solve difference equations over algebraic extensions.

Definition 3.19 (Definition 3 in [17]). Let (R, σ, δ) be a D-ring and M a left R-module. A map $\theta : M \to M$ is called R-pseudo-linear (with respect to σ and δ) if

$$\theta(u + v) = \theta(u) + \theta(v) \qquad \text{and} \qquad \theta(au) = \sigma(a)\theta(u) + \delta(a)u$$

for any $a \in R$ and $u, v \in M$. We write $End_{R,\sigma,\delta}(M)$ for the set of all R-pseudo-linear maps of M.

A characterization of all R-pseudo-linear maps over a commutative ring R is provided in [17, Lemma 5]. For our purposes θ can be either a shift automorphism or the differential δ. This framework allows us to use the same theoretical basis for both difference and differential equations.

Let (F, σ, δ) be a D-field of characteristic 0 and (E, σ, δ) an algebraic extension of (F, σ, δ). Let $\theta : E \to E$ be an F-pseudo-linear map in $End_{F,\sigma,\delta}(E)$. For indefinite summation or integration, we are interested in solving the equation

$$\theta(z) + fz = g$$

for $z \in E$, where $f, g \in E$ are given.

Example 3.20. In order to find a simpler expression for $\sum_{1\leq k<n}(-1)^k k$, we need to solve the equation

$$\sigma(z) - z = (-1)^k k$$

in the difference ring $\mathbb{Q}(k)[(-1)^k]$, with $\sigma(k) = k + 1$ and $\sigma((-1)^k) = -(-1)^k$.

The algebraic extension E can be viewed as a finite-dimensional algebra over F. Let $\mathbf{b} = (b_0, \ldots, b_{n-1})$ be a basis of E over F and denote the column vector of the coordinates of $u \in E$ in the basis $\mathbf{b}$ with $u_{\mathbf{b}} \in F^n$. Now, the equation we want to solve becomes

$$(\theta(z) + fz)_{\mathbf{b}} = g_{\mathbf{b}}.$$

We know how to write elements of E in the chosen basis $\mathbf{b}$. In particular, we can express g as a vector in the given basis $\mathbf{b}$.

Example 3.21. Let $\mathbf{b} = ((-1)^k, 1)$. We have $g = (-1)^k k$, then

$$g_{\mathbf{b}} = \begin{pmatrix} k \\ 0 \end{pmatrix}.$$

The left hand side of the equation, $\theta(z) + fz$, still needs to be expressed as a vector with the basis $\mathbf{b}$. We start with $\theta(z)$. Let $z \in E$. Write $z = z_0b_0 + z_1b_1 + \cdots + z_{n-1}b_{n-1}$ where $z_i \in F$ and $\mathbf{b} = (b_0, \ldots, b_{n-1})$ is a basis of E over F as above. Using the fact that θ is a pseudo linear map, we have

$$\begin{aligned}\theta(z) &= \theta(z_0b_0) + \cdots + \theta(z_{n-1}b_{n-1})\\ &= \sigma(z_0)\theta(b_0) + \delta(z_0)b_0 + \cdots + \sigma(z_{n-1})\theta(b_{n-1}) + \delta(z_{n-1})b_{n-1}\\ &= \sum_{0\leq i<n} \sigma(z_i)\theta(b_i) + \sum_{0\leq i<n} \delta(z_i)b_i.\end{aligned}$$

The second component of this sum is already in the basis $\mathbf{b}$. To express the first part, note that it is written out in the basis $\mathbf{b}$ transformed by θ.

For an F-pseudo-linear map θ, define the matrix $\theta_{\mathbf{b}}$ as follows

$$\theta_{\mathbf{b}} = \begin{pmatrix} | & | & \cdots & | \\ (\theta(b_0))_{\mathbf{b}} & (\theta(b_1))_{\mathbf{b}} & \cdots & (\theta(b_{n-1}))_{\mathbf{b}} \\ | & | & \cdots & | \end{pmatrix}.$$

Example 3.22. For our example, $\mathbb{Q}(k)[(-1)^k], \sigma$, we have

$$\sigma_{\mathbf{b}} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Using the notation σI, respectively δI, for the application of σ, respectively δ, to each component of a vector, we have

$$\begin{aligned}(\theta(z))_{\mathbf{b}} &= \theta_{\mathbf{b}}(\sigma I)(z)_{\mathbf{b}} + (\delta I)(z)_{\mathbf{b}} \\ &= \left(\theta_{\mathbf{b}}(\sigma I) + (\delta I)\right)(z)_{\mathbf{b}}\end{aligned}$$

The only part left is to express fz in the basis $\mathbf{b}$. Multiplication by any fixed element $f \in E$ can be viewed as a linear map. We denote the matrix corresponding to this map with $M_{\mathbf{b}}(f)$.

For any $f \in E$, define $M_{\mathbf{b}}(f)$ to be

$$M_{\mathbf{b}}(f) = \begin{pmatrix} | & | & \cdots & | \\ (fb_0)_{\mathbf{b}} & (fb_1)_{\mathbf{b}} & \cdots & (fb_{n-1})_{\mathbf{b}} \\ | & | & \cdots & | \end{pmatrix}.$$

Then for any $f, g \in E$, we obtain

$$(fg)_{\mathbf{b}} = M_{\mathbf{b}}(f)(g)_{\mathbf{b}}.$$

Example 3.23. In our example $f = -1$, so the corresponding matrix is

$$M_{\mathbf{b}}(-1) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Putting all the parts together, we get

$$\begin{aligned}g_{\mathbf{b}} &= (\theta(z) + fz)_{\mathbf{b}} \\ &= \left(\theta_{\mathbf{b}}(\sigma I) + (\delta I)\right)(z)_{\mathbf{b}} + M_{\mathbf{b}}(f)(z)_{\mathbf{b}} \\ &= \left(\theta_{\mathbf{b}}(\sigma I) + (\delta I) + M_{\mathbf{b}}(f)\right)(z)_{\mathbf{b}}\end{aligned}$$

In the differential case, we have $\sigma = 1_F$ and $\theta = \delta$, which leads to

$$\left(\delta_{\mathbf{b}} + (\delta I) + M_{\mathbf{b}}(f)\right)(z)_{\mathbf{b}} = g_{\mathbf{b}}.$$

Similarly, for the difference case, $\delta = 0_F$ and $\theta = \sigma$, so we have

$$\left(\sigma_{\mathbf{b}}(\sigma I) + M_{\mathbf{b}}(f)\right)(z)_{\mathbf{b}} = g_{\mathbf{b}}.$$

We have shown the following result.

Theorem 3.24. *Let R, σ, δ be a D-ring, M a left R-module, and $\mathbf{b}$ a basis of M over R. Given $f, g \in M$ and $\theta : M \to M$ an R-pseudo-linear map, solving $\theta(z) + fz = g$ for $z \in M$ is equivalent to solving the system of equations*

$$(\theta_{\mathbf{b}}(\sigma I) + (\delta I) + M_{\mathbf{b}}(f))z_{\mathbf{b}} = g_{\mathbf{b}}.$$

This method is not limited to first-order equations or a single term on the right hand side. With more than one term on the right hand side, this difference equation is also known as the creative telescoping problem, which was mentioned in Chapter 2 when describing Zeilberger's algorithm. How to handle creative telescoping problems with this method to find $c_0, \dots, c_{n-1} \in \operatorname{const}_{\sigma,\delta}(R)$ and $g \in M$ such that

$$\sigma(g) - ag = c_0 f_0 + \cdots + c_{n-1} f_{n-1}$$

when $f_0, \dots, f_{n-1}, a \in M$ are given, will be shown in Section 3.4.

Example 3.25. In order to simplify $\sum_{1 \leq k < n} (-1)^k k$, we need to solve the system of first-order linear equations given by $(\sigma_{\mathbf{b}}(\sigma I) + M_{\mathbf{b}}(f))\,(z)_{\mathbf{b}} = g_{\mathbf{b}}$ with

$$\sigma_{\mathbf{b}} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } M_{\mathbf{b}}(f) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

as shown in Example 3.22 and 3.23. Then, we have the system

$$\begin{pmatrix} -\sigma - 1 & 0 \\ 0 & \sigma - 1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} = \begin{pmatrix} k \\ 0 \end{pmatrix},$$

which gives the two equations $-\sigma(z_0) - z_0 = k$ and $\sigma(z_1) - z_1 = 0$. Solving these, we get the result

$$z_0 = -\frac{2k-1}{4} \text{ and } z_1 = c,$$

for any $c \in \mathbb{Q}$. The final answer is

$$z = (-1)^{k+1} \frac{2k-1}{4} + c$$

After checking initial values, we determine that $c = -1/4$ and we can write

$$\sum_{1 \leq k < n} (-1)^k k = (-1)^{n+1} \frac{2n-1}{4} - \frac{1}{4}.$$

Example 3.26 (Exercise 6.53 in [34]). For a more realistic example, consider

$$\sum_{1 \leq k < n+1} \frac{(-1)^k H_k}{\binom{n}{k}}.$$

The summand without $(-1)^k$ can be modeled in the ΠΣ-field $\mathbb{Q}(k, b, h), \sigma$ with $\sigma(k) = k + 1$, $\sigma(b) = \frac{n-k}{k+1} b$, and $\sigma(h) = h + \frac{1}{k+1}$, where b is the binomial $\binom{n}{k}$ and h is the harmonic number H_k. We extend this field with t such that $\sigma(t) = -t$ and $t^2 = 1$ to get the ring $R = \mathbb{Q}(k, b, h)[t]/\langle t^2 - 1 \rangle$. Then, our summand can be represented by the element $\frac{th}{b} \in R$.

In order to solve the equation $\sigma(z) - z = \frac{th}{b}$, we will use a different basis, namely, $\mathbf{b} = \left(\frac{t+1}{2}, \frac{-t+1}{2}\right)$. This basis will play a central role in Section 3.2 and we postpone the discussion of its properties till then. For $a \in \mathbb{Q}(k, b, h)$ the representation of a in the basis $\mathbf{b}$ is $(a, a)^T$and that of ta is $(a, -a)^T$.

Now, in the basis $\mathbf{b}$, we have the matrices

$$\sigma_{\mathbf{b}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } M_{\mathbf{b}}(f) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

This leads to the following system

$$\begin{pmatrix} -1 & \sigma \\ \sigma & -1 \end{pmatrix} \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} = \begin{pmatrix} \frac{h}{b} \\ -\frac{h}{b} \end{pmatrix},$$

which gives the equations $-z_0 + \sigma(z_1) = \frac{h}{b}$ and $\sigma(z_0) - z_1 = -\frac{h}{b}$. By isolating z_0 in the first equation, and plugging it into the second, we get

$$\sigma^2(z_1) - z_1 = \sigma\left(\frac{h}{b}\right) - \frac{h}{b}.$$

This equation needs to be solved over $\mathbb{Q}(k, b, h), \sigma$. Even though this is a second order equation, Karr's algorithm can still be applied by considering σ^2 to be the difference automorphism instead of σ. Note that Theorem 2.21 states that $\mathbb{Q}(k, b, h), \sigma^2$ is still a $\Pi\Sigma$-field.

We get the result

$$z_1 = -\frac{(k - n - 1)(nh + 2h - 1)}{b(n + 2)^2} \in \mathbb{Q}(k, b, h).$$

Putting this value in the equation $z_0 = -\sigma(z_1) + \frac{h}{b}$, we get the answer,

$$z = (-1)^k \frac{(k - n - 1)(nH_k + 2H_k - 1)}{\binom{n}{k}(n + 2)^2}.$$

The commands required to perform these computations are given in Example A.3.

3.2. Structure of algebraic extensions

In this section we investigate the structure of algebraic extensions of $\Pi\Sigma$-fields. Especially structural properties which will be useful in solving difference equations over these extensions will be of interest.

Example 3.27. Let us start by looking at a few examples of algebraic extensions. In the examples below, even though it is not stated explicitly, we take $\sigma(n) = n + 1$. Hence, an expression like 4^n has the shift $\sigma(4^n) = 4^{n+1} = 4(4^n)$. The symbol i stands for the complex imaginary unit, not a summation variable.

(i) We begin with the usual example $\mathbb{Q}[(-1)^n]$. This is isomorphic to $\mathbb{Q}[x]/\langle x^2 - 1\rangle$ since $((-1)^n)^2 = 1$. The polynomial $x^2 - 1$ factors over $\mathbb{Q}[x]$, so $\mathbb{Q}[(-1)^n]$ has zero divisors.

(ii) Extending $\mathbb{Q}$ with $i(-1)^n$, which has the same shift behavior $\sigma(i(-1)^n) = -i(-1)^n$, forms a completely different structure. Now, we have $\mathbb{Q}[i(-1)^n] \simeq \mathbb{Q}[x]/\langle x^2+1\rangle$ since $(i(-1)^n)^2 = -1$. The minimal polynomial x^2+1 is irreducible over $\mathbb{Q}[x]$ and $\mathbb{Q}[i(-1)^n]$ is indeed a field.

(iii) In the previous example, if the field of constants is $\mathbb{C}$ instead of $\mathbb{Q}$, then x^2+1 splits into $(x-i)(x+i)$ over $\mathbb{C}[x]$, and we end up in the familiar situation of a ring with zero divisors.

(iv) An algebraic extension may also occur without a root of unity being involved. We have already seen that extending by 2^n when 4^n is already in the field leads to an algebraic extension. In this case, $\mathbb{Q}(4^n)[2^n] \simeq \mathbb{Q}(4^n)[x]/\langle x^2 - 4^n\rangle$ and the minimal polynomial x^2-4^n is irreducible over $\mathbb{Q}[4^n]$.

(v) The following examples show that the constant term of the minimal polynomial does not need to be a generator of a previous extension in the tower. When faced with an algebraic extension, that is, when we detect that α in the action of $\sigma(t) = \alpha t$ is a σ-radical, we still need to determine the minimal polynomial correctly.

(a) $\mathbb{Q}(8^n)[4^n] \simeq \mathbb{Q}(8^n)[x]/\langle x^3 - (8^n)^2\rangle$
$x^3-(8^n)^2$ is irreducible over $\mathbb{Q}(8^n)[x]$.

(b) $\mathbb{Q}(4^n, 9^n)[6^n] \simeq \mathbb{Q}(4^n, 9^n)[x]/\langle x^2 - 4^n 9^n\rangle$
$x^2 - 4^n9^n$ is irreducible over $\mathbb{Q}(4^n, 9^n)[x]$.

(c) $\mathbb{Q}(18^n, 2^n)[3^n] \simeq \mathbb{Q}(18^n, 2^n)[x]/\langle x^2 - 18^n/2^n\rangle$
$x^2 - 18^n/2^n$ is irreducible over $\mathbb{Q}(18^n, 2^n)[x]$.

Note that the minimal polynomials in all examples were of the form $x^n - c$ for some c in the base field. This is not merely due to a lack of imagination. In fact, in an algebraic extension with the new indeterminate satisfying a first-order recurrence, this is the only type of minimal polynomial that occurs.

For a D-ring R and $a, b \in R$, we introduce the notation

$$V_{a,b}(R) = \{w \in R \text{ such that } \sigma(w) = aw + b\}$$

to denote the solutions of the first-order linear difference equation $\sigma(w) = aw + b$ Recall from Definition 2.13 that, we call an element $a \in R$ a *σ-radical over R* if there exists $z \in R^*$ such that $\sigma(z) = a^n z$ for an integer $n > 0$.

The following theorem is based on a proposition of Schneider [71] regarding the form of the minimal polynomial. We show that algebraic extensions are simple radical extensions by carrying out the analysis of [17, Lemma 13] further.

Theorem 3.28. *Let (R, σ, δ) be a D-ring with σ an automorphism of R. Let E be a D-ring extension of R, and $t \in E^*$ be algebraic over R such that $\sigma(t) = at + b$ with $a \in R^*, b \in R$. If $V_{a,b}(R)$ has no nonzero elements and* $\mathrm{const}_{\sigma,\delta}(R) = \mathrm{const}_{\sigma,\delta}(E)$*, then* *(a)* $b = 0$,

(b) *a is a σ-radical over R and*

(c) *the minimal polynomial $p \in R[x]$ of t is of the form $x^d - c$ for some $c \in R$.*

Proof. Following the proof of Theorem 2.15 or [42, Theorem 2.3], let $p(x) = x^d + \sum_{0 \le i < d} p_i x^i$ be the minimal polynomial of t over k where $d > 0$. We have

$$0 = \sigma(p(t)) = (at+b)^d + \sum_{0 \le i < d} \sigma(p_i)(at+b)^i.$$

Let $h(x) = \sigma(p(x))$. Since t is a root of $h(x)$, $p(x)$ divides $h(x)$. Both $p(x)$ and $h(x)$ have degree d. Therefore, $h(x) = a^d p(x)$ and we have

$$a^d \sum_{0 \le i < d} p_i x^i = \sum_{0 \le i < d} \binom{d}{i} (ax)^i b^{d-i} + \sum_{0 \le i < d} \sigma(p_i)(ax+b)^i. \tag{3.1}$$

Comparing coefficients for x^{d-1}, we get the equality

$$a^d p_{d-1} = d a^{d-1} b + \sigma(p_{d-1}) a^{d-1}. \tag{3.2}$$

Note that $d \in \mathbb{N} \subset \mathrm{const}_{\sigma,\delta}(R)$, so $w = -p_{d-1}/d \in V_{a,b}(R)$. Since $V_{a,b}(R)$ has no nonzero elements, $p_{d-1} = 0$. Replacing p_{d-1} by 0 in (3.2) , shows that b is also 0.

Looking back at equation (3.1) and comparing coefficients for x^i, we get

$$\sigma(p_i) = a^{d-i} p_i, \quad \text{for } 0 \le i < d.$$

Since t is nonzero, $p_j \ne 0$ for some $j < d$. For $j > 0$, $\sigma(t^{d-j}/p_j) = t^{d-j}/p_j$, so t^{d-j}/p_j is a new constant in E. Hence, for $0 < i < d$, $p_i = 0$. The equality $\sigma(p_0) = a^d p_0$ shows that a is a σ-radical over R.

□

This theorem does not give any information about the purely differential case, where $\sigma = 1_R$, since $V_{1,0}(R) = R$ and the statement does not apply.

Knowing that the minimal polynomial of an algebraic extension E of R is of the form $x^d - c$, we only need to determine d and c to construct the extension. The procedure to check if an element is a σ-radical also returns $d \in \mathbb{N}$, as explained in Section 2.2.2. Then $\sigma(c) = a^d c$ for some $c \in R$. Since this is a first-order linear difference equation, we can solve for $c \in R$ and get both components of the minimal polynomial.

Example 3.29. Consider the extension of $\mathbb{Q}(4^n, 9^n), \sigma$, where $\sigma(4^n) = 4(4^n)$ and $\sigma(9^n) = 9(9^n)$ with 6^n. We have $\sigma(6^n) = 6(6^n)$, so $a = 6$. Using the method from Section 2.2.2, we find that the set of $d \in \mathbb{Z}$ such that $\sigma(c) = a^d c$ for some $c \in \mathbb{Q}(4^n, 9^n)$ is $2\mathbb{Z}$. Now, we need to solve the equation $\sigma(c) = 36c$ to find the constant term of the minimal polynomial. The solution is $4^n 9^n$. Hence, the minimal polynomial is $x^2 - 4^n 9^n$.

The code required for these computations is presented in Example A.4.

Proposition 3.30. *Let R, σ be a difference field and $E = R[x]/\langle x^d - c\rangle$ an algebraic extension of R such that $\sigma(x) = ax$ where $V_{a,0}(R)$ has no nonzero elements and* $\mathrm{const}_\sigma(R) = \mathrm{const}_\sigma(E)$*. Then E is a simple difference ring.*

Proof. We need to show that $\langle x^d - c\rangle$ is a maximal difference ideal in $R[x]$. Let I be a proper difference ideal of $R[x]$ which contains $\langle x^d - c\rangle$. Since $R[x]$ is a principal ideal domain, $I = \langle p\rangle$ for some $p \in R[x]$. Write $p = x^k + \sum_{0\leq i<k} p_i x^i$ where $k = \deg(p)$. Now,

$$\sigma(p(x)) = a^k x^k + \sum_{0\leq i<k} \sigma(p_i) a^i x^i.$$

is also in I, so $p(x) \mid \sigma(p(x))$. The polynomials $p(x)$ and $\sigma(p(x))$ have the same degree. Matching the coefficients of the degree k terms, we get $a^k p(x) = \sigma(p(x))$. Comparing coefficients of terms with degree i for $i = 0, \ldots, k-1$, we have the set of equations

$$\sigma(p_i) = a^{k-i} p_i.$$

If $p_i \neq 0$ for some $1 \leq i < k$, then $\sigma(\frac{x^{k-i}}{p_i}) = \frac{x^{k-i}}{p_i}$ and we have a new constant. Since $\mathrm{const}_\sigma(E) = \mathrm{const}_\sigma(R)$, we conclude that the coefficients p_i are zero for $i = 1, \ldots, k-1$. Then $p = x^k - p_0$, for some $p_0 \in R$ where $\sigma(p_0) = a^k p_0$. By Theorem 3.28, a is a σ-radical over R where $\sigma(c) = a^d c$. This d is minimal, since if there was an integer $e < d$ such that $\sigma(w) = a^e w$ for some $w \in R$, then x^e/w would be a new constant in E. Recall from Lemma 2.27 that the exponents $n \in \mathbb{Z}$ for which a satisfies the first-order linear homogeneous equation $\sigma(g) = a^n g$ for $g \in R$ form a $\mathbb{Z}$-module. Then $d \mid k$ and $p_0 = \lambda c^{k/d}$ for some constant λ. Now, we have $x^d - c \mid x^k - p_0$. Hence $I = \langle x^d - c\rangle$, proving the claim that $\langle x^d - c\rangle$ is a maximal difference ideal in $R[x]$. □

A ring R with an idempotent $e \in R$ can be decomposed as $R \simeq Re \oplus R(1-e)$. The extensions we are interested in have such a decomposition.

Example 3.31. Take $R = \mathbb{Q}(n)[e]$ with $\sigma(n) = n + 1$ and $\sigma(e) = -e$, where e behaves like $(-1)^n$. Since $((-1)^n)^2 = 1$, the indeterminate e satisfies the polynomial $x^2 - 1 \in \mathbb{Q}(n)[x]$, which factors into linear factors $x - 1$ and $x + 1$ over $\mathbb{Q}(n)$. Then, we have

$$R \simeq \mathbb{Q}(n)[x]/\langle x^2 - 1\rangle \simeq \mathbb{Q}(n)[x]/\langle x - 1\rangle \oplus \mathbb{Q}(n)[x]/\langle x + 1\rangle.$$

Note that $e_0 = \frac{e+1}{2}$ and $e_1 = \frac{-e+1}{2}$ are idempotents in R. Since $1 = e_0 + e_1$ and $e_0 e_1 = 0$, we also have the decomposition $R \simeq \mathbb{Q}(n)e_0 \oplus \mathbb{Q}(n)e_1$.

The decomposition given by idempotents interacts with the difference automorphism σ to create a very useful structure. The following lemma is needed for the proof of the structure theorem.

Lemma 3.32. *A simple difference ring has no nilpotent elements.*

Proof. Let I be the ideal of all nilpotent elements of a simple difference ring R, σ. Let $r \in I$, then there exists $k \in \mathbb{N}$ such that $r^k = 0$. Since $0 = \sigma(0) = \sigma(r^k) = \sigma(r)^k$, it follows that $\sigma(r)$ is also nilpotent. Hence, I is a difference ideal. Then I is either R or $\langle 0\rangle$. It cannot be R because R contains 1, so I is $\langle 0\rangle$. □

The following theorem gives a decomposition of a difference ring R into integral domains. This decomposition plays a central role in the reduction of the problem of solving linear difference equations over R to domains where Karr's algorithms can be applied.

Theorem 3.33. [78, Corollary 1.16] and [36, Lemma 6.8]

Let (K, σ, δ) be a D-field with σ an automorphism of K, R be a finitely generated simple difference ring and a D-ring extension of K. Then, there exist idempotents $e_0, \ldots, e_{d-1} \in R$ for $d > 0$ such that

(a) $R = R_0 \oplus \cdots \oplus R_{d-1}$ where $R_i = e_i R$,

(b) $\sigma(e_i) = e_{i+1 \pmod{d}}$ so σ maps R_i isomorphically onto $R_{i+1 \pmod{d}}$ and σ^d leaves each R_i invariant.

(c) For each i, R_i is a domain, a simple difference ring and a D-ring extension of $e_i K$ with respect to σ^d.

The theorem can be proven with an argument identical to that of Proof 2 of Corollary 1.16 from [78, page 12]. The main difference is that we only require R to be a simple difference ring, not a Picard-Vessiot ring and the field of constants is not algebraically closed. In Corollary 1.16 of [78], the latter condition is used only to show that R_i is a Picard-Vessiot extension of $e_i K$. Hence, this condition is not relevant for our purpose.

Proof. The ring R has no nilpotent elements by Lemma 3.32 and is finitely generated. We can write $\langle 0 \rangle = \cap_{0 \le i < d} I_i$, where I_i are prime ideals of R. Since the I_i are distinct and $I_i \not\supset \cap_{j \ne i} I_j$ for $0 \le i < d$, this is a minimal representation. By Theorem 3.10 this decomposition is unique.

For each i, $\cap_{j \ge 0} \sigma^j(I_i)$ is a difference ideal. Since R is simple, this intersection must be $\langle 0 \rangle$. We have $\left\{ I_i, \sigma(I_i), \ldots, \sigma^{d-1}(I_i) \right\} = \{I_0, I_1, \ldots, I_{d-1}\}$ by the uniqueness of the minimal decomposition. We can assume that $\sigma(I_i) = I_{i+1 \pmod{d}}$ after renumbering the indices.

Let $S_i = R/I_i$. We want to show that S_i is a difference ring with respect to σ^d. Let $J_i = \left\{ r \in R \,\middle|\, \sigma^d(r) \in I_i \right\}$ for $0 \le i < d$. For each i, J_i is a prime ideal. To see this take $rs \in J_i$. Then $\sigma^d(rs) = \sigma^d(r)\sigma^d(s) \in I_i$. Since I_i is prime, either $\sigma^d(r) \in I_i$ or $\sigma^d(s) \in I_i$. Which implies either $r \in J_i$ or $s \in J_i$. Note that $J_i \supset I_i$ for each i. Now $\cap_{0 \le i < d} J_i$ is a proper difference ideal of R, hence it must be $\langle 0 \rangle$. By the uniqueness of the minimal decomposition, $J_i = I_i$ for each i. Hence $r \in I_i$ if and only if $\sigma^d(r) \in I_i$, so $S_i = R/I_i$ is a difference ring with respect to σ^d.

Let $\pi_i : R \to S_i$ be the canonical homomorphism. Note that σ induces an isomorphism $\sigma_i : S_i \to S_{i+1}$.

In order to show that S_i is a simple difference ring for each i with respect to σ^d, we claim that there is no proper σ^d-invariant difference ideal J_i of R properly containing I_i. Let J_i be such an ideal and consider $\cap_{0 \le i < d} \sigma^i(J_i)$. This is a proper difference ideal of R with respect to σ, hence it must be $\langle 0 \rangle$. Then we also have $\cap_{0 \le i < d} \sigma^i(J_i) \subset I_i$. Since the I_j are prime we must have that for some $0 \le k < d$, $\sigma^k(I_i) \subset \sigma^k(J_i) \subset I_i$.

This implies $I_{i+k \pmod d} = I_i$, which can only happen if $k = 0$. Then $J_i = I_i$, which is a contradiction. Hence S_i is a simple difference ring with respect to σ^d.

Now we need to show that R is a direct sum of S_i's for $0 \leq i < d$. For $i \neq j$, $I_i + I_j$ is a σ^d invariant difference ideal. Since there is no proper σ^d-invariant difference ideal of R properly containing I_i as shown above, $I_i + I_j$ must be all of R. Then I_i are pairwise comaximal, that is, $I_i + I_j = R$ for $i \neq j$ and Theorem 3.5 implies that the map $\pi : R \to \oplus_{0 \leq i < d} S_i$ given by $\pi(r) = (\pi_0(r), \pi_1(r), \ldots, \pi_{d-1}(r))$ is a ring isomorphism. The ring $\oplus_{0 \leq i < d} S_i$ is also a difference ring with the automorphism $\tilde{\sigma}(r_0, \ldots, r_{d-1}) = (\sigma_{d-1}(r_{d-1}), \sigma_0(r_0), \ldots, \sigma_{d-2}(r_{d-2}))$. This makes π a K-isomorphism of difference rings. Letting $R_i = \pi^{-1}(S_i)$ and $e_i = \pi^{-1}(1_i)$ where 1_i is the identity of S_i, we get the conclusions of the theorem. □

This structure carries over to the ring of quotients of R.

Corollary 3.34 (Corollary 6.9 in [36])**.** *Let K, R, R_i and e_i be as in Theorem 3.33 and E the ring of quotients of R. Then $E = E_0 \oplus \cdots \oplus E_{d-1}$ where E_i is the quotient field of R_i.*

An important special case is when a $\Pi\Sigma$-field F, σ is extended by an indeterminate t with $\sigma(t) = \zeta t$ where ζ is an n-th root of unity.

Corollary 3.35. *Let F, σ be a $\Pi\Sigma$-field over a constant field K, $k \in F$ such that $\sigma(k) = k + 1$, $\zeta \in K$ a primitive n-th root of unity and $F[t], \sigma$ an extension of F, σ where t models ζ^k. Then there are idempotents $e_i \in F[t]$ such that*

$$e_i = \prod_{\substack{0 \leq j < n \\ j \not\equiv -i \pmod n}} \frac{x - \zeta^j}{\zeta^{n-i} - \zeta^j}$$

and

$$F[t] = \bigoplus_{0 \leq j < n} F_i, \text{ where } F_i = e_i F \simeq F[X]/\langle X - \zeta^{n-j} \rangle \simeq F.$$

Moreover, $\sigma(e_i) = e_{i+1 \pmod n}$ so σ maps F_i isomorphically to $F_{i+1 \pmod n}$ and σ^n leaves each F_i invariant.

Proof. Since t represents ζ^k, we have $\sigma(t) = \zeta t$ and $t^n = (\zeta^k)^n = 1$. Then the minimal polynomial of the extension is $X^n - 1 \in F[X]$. This polynomial splits into linear factors because $\zeta \in K$. Now $X^n - 1 = \prod_{0 \leq j < n}(X - \zeta^j)$ and we have the isomorphism

$$\begin{aligned} \pi : F[X] &\to \bigoplus_{0 \leq j < n} F[X]/\langle X - \zeta^{n-j} \rangle \\ p &\mapsto (p(1), p(\zeta^{n-1}), p(\zeta^{n-2}), \ldots, p(\zeta)) \end{aligned}$$

Note that e_i is simply a Lagrange interpolant and $\pi(e_j)$ is $1 \in F_i$ if $j = i$ and $0 \in F_j$ for $j \neq i$. The e_i are ordered to satisfy the cyclic permutation of Theorem 3.33.

Namely,

$$\sigma(e_i) = \prod_{\substack{0\le j<n \\ j\not\equiv -i \pmod n}} \frac{\zeta x - \zeta^j}{\zeta^{n-i} - \zeta^j} = \prod_{\substack{0\le j<n \\ j\not\equiv -i \pmod n}} \frac{x - \zeta^{j-1}}{\zeta^{n-(i+1)} - \zeta^{j-1}} = e_{i+1}.$$

From the definition of σ it is easy to see that $\sigma^n(F_i) = F_i$.

□

Example 3.36. Take $R = \mathbb{Q}(\zeta)(k)[e]$ with $\sigma(k) = k+1$ and $\sigma(e) = \zeta e$, where e behaves like ζ^k with ζ a 3rd root of unity. Then $\left(\zeta^k\right)^3 = 1$ and e satisfies the polynomial $x^3 - 1 \in \mathbb{Q}(\zeta)(k)[x]$. Now, we have

$$\begin{aligned} R &\simeq \mathbb{Q}(\zeta)(k)[x]/\langle x^3-1\rangle, \\ &\simeq \mathbb{Q}(\zeta)(k)[x]/\langle x-1\rangle \oplus \mathbb{Q}(\zeta)(k)[x]/\langle x-\zeta^2\rangle \oplus \mathbb{Q}(\zeta)(k)[x]/\langle x-\zeta\rangle. \end{aligned}$$

The idempotents e_i are given by

$$e_0 = \frac{e^2+e+1}{3}, e_1 = \frac{\zeta^2 e^2 + \zeta e + 1}{3} \text{ and } e_2 = \frac{\zeta e^2 + \zeta^2 e + 1}{3}.$$

This decomposition, while providing the basis to build towers on an algebraic extension, also gives a description of the operator matrices that arise when solving the equation $\theta(z) - fz = g$. Recall that the operator matrix on the left hand side was given by $\sigma_{\mathbf{b}}(\sigma I) + M_{\mathbf{b}}(f)$. If the basis is chosen as above, σ_b is a simple permutation matrix that cyclically shifts each coordinate to right. For example,

$$\sigma_{\mathbf{b}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ for } n=2 \text{ and } \sigma_{\mathbf{b}} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{ for } n=3.$$

Let $f_{\mathbf{b}} = (f_0, \ldots, f_{n-1})^T$. Since the basis e_i is orthogonal, that is $e_i e_j = 0$ when $i \neq j$, the matrix $M_{\mathbf{b}}(f)$ is a diagonal matrix where the entries on the diagonal are $f_0, \ldots, f_n$. Hence, the matrix $A_{n,f} = \sigma_{\mathbf{b}}(\sigma I) + M_{\mathbf{b}}(f)$ has only two nonzero entries per row. It can easily be computed explicitly. For instance,

$$\begin{pmatrix} -f_0 & \sigma \\ \sigma & -f_1 \end{pmatrix} \text{ for } n=2 \text{ and } \begin{pmatrix} -f_0 & 0 & \sigma \\ \sigma & -f_1 & 0 \\ 0 & \sigma & -f_2 \end{pmatrix} \text{ for } n=3.$$

The general form for $f \in E$ and $n \in \mathbb{N}$ is given by $A_{n,f} = (a_{i,j})_{0\le i,j<n}$ where

$$\begin{aligned} &a_{i,i} = -f_i, \\ &a_{i+1,i \pmod n} = \sigma \text{ for } i = 0,\ldots,n-1, \text{ and} \\ &a_{i,j} = 0 \text{ otherwise.} \end{aligned} \tag{3.3}$$

The system obtained from this matrix will have two variables in each equation. Recall from Example 3.26, with $f = -1$ and $n = 2$, the equations were

$$\begin{aligned} -z_0 + \sigma(z_1) &= g_0, \\ \sigma(z_0) - z_1 &= g_1. \end{aligned}$$

In order to solve this system, it should be transformed to have a single variable in each equation. In other words, it should be uncoupled.

Using the structure of $A_{n,f}$, we can write a transformation matrix to convert the system to a more suitable form. For the operator matrix from Example 3.26, we have

$$\begin{pmatrix} 1 & \sigma \\ \sigma & 1 \end{pmatrix} \begin{pmatrix} -1 & \sigma \\ \sigma & -1 \end{pmatrix} = \begin{pmatrix} \sigma^2 - 1 & 0 \\ 0 & \sigma^2 - 1 \end{pmatrix}.$$

When $f = -1$, like all indefinite summation problems, this transform can be used. However, for general f in the extension ring E and $n = 2$, we need the following transform

$$\begin{pmatrix} \sigma(f_1) & \sigma \\ \sigma & \sigma(f_0) \end{pmatrix} \begin{pmatrix} -f_0 & \sigma \\ \sigma & -f_1 \end{pmatrix} = \begin{pmatrix} \sigma^2 - f_0\sigma(f_1) & 0 \\ 0 & \sigma^2 - \sigma(f_0)f_1 \end{pmatrix}.$$

For $n = 3$, we have the following

$$\begin{pmatrix} \sigma^2(f_1)\sigma(f_2) & \sigma^2 & \sigma^2(f_1)\sigma \\ \sigma^2(f_2)\sigma & \sigma(f_0)\sigma^2(f_2) & \sigma^2 \\ \sigma^2 & \sigma^2(f_0)\sigma & \sigma^2(f_0)\sigma(f_1) \end{pmatrix} \begin{pmatrix} -f_0 & 0 & \sigma \\ \sigma & -f_1 & 0 \\ 0 & \sigma & -f_2 \end{pmatrix} =$$
$$\begin{pmatrix} \sigma^3 - f_0\sigma^2(f_1)\sigma(f_2) & 0 & \\ 0 & \sigma^3 - \sigma(f_0)f_1\sigma^2(f_2) & 0 \\ 0 & 0 & \sigma^3 - \sigma^2(f_0)\sigma(f_1)f_2 \end{pmatrix}.$$

For arbitrary $n \in \mathbb{N}$ we can write $T_{n,f} = (t_{i,j})_{0 \leq i,j < n}$ where

$$t_{i,j} = \left(\prod_{0 \leq k < j-i-1 \pmod{n}} \sigma^{n-k-1}\left(f_{i+1+k \pmod{n}}\right) \right) \sigma^{i-j \pmod{n}}.$$

The uncoupled matrix $T_{n,f}A_{n,f}$ has the entries

$$\sigma^n - \prod_{0 \leq k < n-1} \sigma^{n-k-1}\left(f_{i+1+k \pmod{n}}\right),$$

on the i-th place of the diagonal and zeroes elsewhere.

Hence, the systems of first-order linear equations obtained by reducing the problem to a transcendental extension can be solved without the need for a general uncoupling algorithm.

3.3. Extensions of algebraic extensions

In many applications, building further extensions on top of an algebraic extension is necessary. For example, to simplify the sum

$$\sum_{i=1}^{n} \frac{(-1)^i}{i} \sum_{j=1}^{i} \frac{(-1)^j}{j},$$

we need to model the summand $\frac{(-1)^i}{i} \sum_{j=1}^{i} \frac{(-1)^j}{j}$ in a difference field. The inner sum $\sum_{j=1}^{i} \frac{(-1)^j}{j}$ satisfies the first-order equation $\sigma(t) = t - \frac{(-1)^i}{i+1}$. In this case, keeping the algebraic extension which represents $(-1)^i$ on top of the tower is not possible. The equation modeling this sum depends on $(-1)^i$ and the corresponding extension for this sum must come after the one for $(-1)^i$.

Recall from Theorem 2.17 that an inhomogeneous extension of a difference field is first-order linear. That is, the extension is transcendental and does not introduce any new constants. This theorem holds even when the base of the extension is a difference ring with zero divisors. Theorem 2.10 provides a way to check if a certain extension is inhomogeneous by looking for a solution in the base ring. This theorem also holds for difference extensions over rings. Hence, we have an algorithmic method to verify if an extension on a tower containing an algebraic extension is first-order linear as well.

Example 3.37. For the summand $\frac{(-1)^i}{i} \sum_{j=1}^{i} \frac{(-1)^j}{j}$, we construct the difference ring $R = \mathbb{Q}(i)[e]/\langle e^2 - 1\rangle$ where $\sigma(i) = i+1$ and $\sigma(e) = -e$. The difference equation

$$\sigma(g) - g = -\frac{e}{i+1}$$

does not have a solution in R. Hence, the extension $R[s], \sigma$ of R, σ where $\sigma(s) = s - \frac{e}{i+1}$ is first-order linear.

The code to perform this computation is available in Example A.5.

Now we investigate the structure of a transcendental extension over a difference ring with the decomposition described in the previous section. Our aim is to keep this decomposition intact and represent this extension as a transcendental extension on each component.

Example 3.38. We wish to extend R from the previous example with a new indeterminate s such that $\sigma(s) = s - \frac{e}{i+1}$, where s represents the sum $\sum_{j=1}^{n} \frac{(-1)^j}{j}$. Take (e_0, e_1) to be the basis for R described in Corollary 3.35. Writing $s = s_0 e_0 + s_1 e_1$ and $e = e_0 - e_1$, we have

$$\begin{aligned}\sigma(s_0 e_0 + s_1 e_1) &= \sigma(s_0) e_1 + \sigma(s_1) e_0 \\ &= s_0 e_0 + s_1 e_1 + \frac{-e_0}{i+1} + \frac{e_1}{i+1}.\end{aligned}$$

Comparing coefficients of e_i to isolate the shift behavior of each component of s, we find

$$\begin{aligned}\sigma(s_0) &= s_1 + \frac{1}{i+1},\\ \sigma(s_1) &= s_0 - \frac{1}{i+1}.\end{aligned}$$

Since the action of σ on the components of the new indeterminate s is not compatible with this view of the new extension, difference equations over this extension cannot be solved using the method described in Section 3.1. However, it is possible to find a new extension which contains an element h with the shift behavior $\sigma(h) = -h + \frac{1}{i+1}$ and the same action of σ on the indeterminates added to the component fields.

Example 3.39. Take $\sigma(h) = -h + \frac{1}{i+1}$, $s_0 = h$ and $s_1 = -h$. Then

$$\begin{aligned}\sigma(s_0) &= \sigma(h) = -h + \frac{1}{i+1},\\ \sigma(s_1) &= \sigma(-h) = h - \frac{1}{i+1}.\end{aligned}$$

The extension $R[h], \sigma$ with $\sigma(h) = -h + \frac{1}{i+1}$ contains the element $s = he_0 - he_1 = eh$, which models the sum $\sum_{j=1}^{i} \frac{(-1)^j}{j}$.

The following lemma describes how to find the action of σ on the new variable h.

Lemma 3.40. *Let F, σ be a $\Pi\Sigma$-field over a constant field K such that there exists $k \in F$ where $\sigma(k) = k + 1$. Let $\zeta \in K$ be primitive n-th root of unity and $F[e], \sigma$ a difference ring extension of F, σ where e models ζ^k. If $F[e][t], \sigma$ is a difference ring extension of $F[e], \sigma$ where $\sigma(t) = t + be^i$ with $b \in F^*$ and $0 \leq i < n$, then*

$$(F[e][t], \sigma) \simeq (F[t][x]/\langle x^n - 1\rangle, \bar{\sigma})$$

where $\bar{\sigma}(t) = \zeta^{-i}(t + b)$.

Proof. Using Corollary 3.35, we can view $F[e]$ as $F[x]/\langle x^n - 1\rangle$.

As rings, $(F[x]/\langle x^n - 1\rangle)[t] \simeq F[t][x]/\langle x^n - 1\rangle$. Let

$$\varphi : (F[x]/\langle x^n - 1\rangle)[t] \to F[t][x]/\langle x^n - 1\rangle$$

be the ring isomorphism where $\varphi(x) = x$ and $\varphi(t) = x^i t$. We need to show that $\varphi(\sigma(f)) = \bar{\sigma}(\varphi(f))$ for $f \in (F[x]/\langle x^n - 1\rangle)[t]$. Since f is a polynomial in t, and σ is an automorphism on the polynomial ring $(F[x]/\langle x^n - 1\rangle)[t]$ it is enough to consider $f = t$. Now, we have

$$\begin{aligned}\varphi(\sigma(t)) = \varphi(t + bx^i) &= tx^i + bx^i\\ &= x^i(t + b) = \zeta^i x^i \zeta^{-i}(t + b) = \bar{\sigma}(x^i t) = \bar{\sigma}(\varphi(t)).\end{aligned}$$

Hence φ is a difference ring isomorphism. □

This can be extended to the ring of quotients in the usual way to get an isomorphism between $(Q(F[e][t]),\sigma)$ and $(F(t)[x]/\langle x^n-1\rangle,\bar{\sigma})$. Note that $Q(F(t)[x]/\langle x^n-1\rangle) = F(t)[x]\langle x^n-1\rangle$ by Corollary 3.34.

Example 3.41. Matching the previous example to the statement of the lemma, we have $\zeta=-1$, $i=1$ and $b=\frac{1}{k+1}$. Then $\bar{\sigma}(h)=-1(h+\frac{1}{k+1})$.

Note that Lemma 3.40 makes no claims on the extension being transcendental or introducing new constants. If the equation $\sigma(t)=t+be^i$ already has a solution in $F[e]$, then $F[t]$ will not be a first-order linear extension.

Theorem 3.42. *Let F,σ be a $\Pi\Sigma$-field over a constant field K such that there exists $k\in F$ where $\sigma(k)=k+1$. Let $\zeta\in K$ be a primitive n-th root of unity and $F[e],\sigma$ a difference ring extension of F,σ where e models ζ^k. If $F[e][t],\sigma$ is a difference ring extension of $F[e],\sigma$ where $\sigma(t)=t+\beta$ with $\beta=\sum_{1\leq i<n} b_ie^i\in F[e]$, then*

$$(F[e][t],\sigma)\hookrightarrow(F[t_1,\ldots,t_{n-1}][x]/\langle x^n-1\rangle,\bar{\sigma})$$

where $\bar{\sigma}(t_i)=\zeta^{-i}(t_i+b_i)$ for $1\leq i<n$.

Proof. Following Lemma 3.40, we have the difference ring isomorphisms

$$\psi_i:(F[e][t_i],\sigma)\to(F[t_i][x]/\langle x^n-1\rangle,\bar{\sigma})$$

where $\sigma(t_i)=t_i+b_ix^i$ and $\bar{\sigma}(t_i)=\zeta^{-i}(t_i+b_i)$ with $\psi_i(t_i)=x^it_i$. Let ψ be the difference ring isomorphism

$$\psi:(F[e][t_1,\ldots,t_{n-1}],\sigma)\to(F[t_1,\ldots,t_{n-1}][x]/\langle x^n-1\rangle,\bar{\sigma})$$

defined by $\psi(t_i)=\psi_i(t_i)$. Take π to be the embedding

$$\pi:(F[e][t],\sigma)\hookrightarrow(F(e)[t_1,\ldots,t_{n-1}],\sigma)$$

such that $\pi:t\mapsto t_1+\cdots+t_{n-1}$. Note that

$$\sigma(\pi(t))=\sigma(\sum_{1\leq i<n}t_i)=\sum_{1\leq i<n}(t_i+b_ix^i)=\pi(t+\beta)=\pi(\sigma(t)).$$

Therefore π is a difference ring embedding. Then $\varphi=\psi\circ\pi$ is the required difference ring embedding. □

This structure can be used to solve first-order difference equations.

Proposition 3.43. *Let F,σ be a $\Pi\Sigma$-field over a constant field K such that there exists $k\in F$ where $\sigma(k)=k+1$. Let $\zeta\in K$ be a primitive n-th root of unity and $F[e],\sigma$ a difference ring extension of F,σ where e models ζ^k. Let $F[e][t],\sigma$ be a difference ring extension of $F[e],\sigma$ where $\sigma(t)=t+\beta$ with $\beta=\sum_{1\leq i<n}b_ix^i\in F[e]$ and $J=\{\,i\in\mathbb{N}\,|\,b_i\neq 0\,\}=\{j_1,j_2,\ldots,j_m\}$. Suppose that*

$$\sigma^n(w)-w=\sum_{0\leq l<n}\frac{\sigma^l(b_{j_d})}{\zeta^{j_d(l+1)}}\tag{3.4}$$

has no solution in $F(t_{j_1},\dots,t_{j_{d-1}}),\sigma$ *where* $\sigma(t_{j_d}) = \zeta^{-j_d}(t_{j_d}+b_{j_d})$ *for any* $d = 1,\dots,m$. *Then there is an algorithm to find* $g \in Q(F[e][t])$ *such that*

$$\sigma(g) - \alpha g = f$$

for any $\alpha \in Q(F[e][t])^*$ *and* $f \in Q(F[e][t])$.

Proof. By Theorem 3.42 and Corollary 3.34, we can view $Q(F[e][t])$ as a finite-dimensional algebra over $F(t_{j_1},\dots,t_{j_m})$. We can transform $\sigma(g)-\alpha g = f$ to the system of difference equations

$$(\theta_{\mathbf{b}}(\sigma I) + M_{\mathbf{b}}(f))z_{\mathbf{b}} = g_{\mathbf{b}}$$

using Theorem 3.24. When this matrix is uncoupled the equation we need, to solve becomes

$$\sigma^n(g_0) - \prod_{0\le k<n-1} \sigma^{n-k-1}\left(\alpha_{1+k \pmod n}\right) g_0 = \\ \sum_{0\le l<n-1}\left(\prod_{0\le k<j-i-1 \pmod n} \sigma^{n-k-1}\left(\alpha_{1+k \pmod n}\right)\right)\sigma^{n-j} f_l. \quad (3.5)$$

Since at each step $d = 1,\dots,m$, Equation (3.4) for $i = j_{m+1}$ does not have a solution in $F(t_{j_1},\dots,t_{j_{d-1}})$, this extension is inhomogeneous. By Theorem 2.10, this is a Σ extension with respect to σ^n. Then we can solve Equation (3.5) over $F(t_{j_1},\dots,t_{j_m})$ and reconstruct the solutions in $Q(F[e][t])$. □

Example 3.44. We try to find a simpler form for the sum

$$g = \sum_{i=1}^{n} \frac{(-1)^i}{i} \sum_{j=1}^{i} \frac{(-1)^j}{j}$$

which arises in physics applications [1]. In Example 3.39, we saw that

$$(\mathbb{Q}(i)[e][s],\sigma) \simeq (\mathbb{Q}(i)[h][e],\bar{\sigma}),$$

where $\sigma(s) = s - \frac{e}{i+1}$ and $\bar{\sigma}(h) = -h - \frac{1}{i+1}$. Now, we need to solve

$$\sigma(g) - g = \frac{e}{i}s$$

over $Q(\mathbb{Q}(i)[e][s])$. This is equivalent to solving

$$\sigma^2(g_0) - g_0 = \frac{h}{i} + \sigma\left(\frac{h}{i}\right)$$

over $\mathbb{Q}(i)[h][e]$.

Code available in Example A.7.

3.4. Creative telescoping over algebraic extensions

The creative telescoping method, briefly explained at the end of the introduction to Chapter 2, to simplify definite sums was introduced by Zeilberger [83]. The core of this approach is to solve the equation

$$\sigma(g) - g = c_0 f_0 + \cdots + c_{k-1} f_{k-1} \tag{3.6}$$

for $c_1, \ldots, c_{k-1}$ and g when $f_1, \ldots, f_{k-1}$ are given. If the f_i are in a $\Pi\Sigma$-field F, Karr's algorithm already provides a method to find $c_i \in \operatorname{const}_\sigma(F)$ and $g \in F$. This was noted first by Schneider [62, Section 1.3 and 4.3].

Karr stumbled upon this problem because going down a $\Pi\Sigma$ tower recursively to solve $\sigma(g) - g = f$ leads to more terms on the right hand side, similar to the way a more general form of the multiplicative analogue had to be treated in Section 2.2.2. In this section, we will describe how to solve the equation (3.6) over an algebraic extension of a $\Pi\Sigma$-field. Our motivation is to solve definite summation problems only, since indefinite summation problems in towers built on algebraic extensions can still be solved using the method of the previous section. This case is explained in more detail in Section 3.3.

Let E, σ be an algebraic extension of a difference ring R, σ of order n. Take $\mathbf{b}$ to be the basis of E as an R-module formed by the idempotents described in Section 3.2. Given $f_1, \ldots, f_{k-1}, a \in E$, we want to find $g \in S$ and $c_1, \ldots, c_{k-1} \in \operatorname{const}_\sigma(S)$ such that Equation (3.6) is satisfied. Writing this equation in the basis $\mathbf{b}$, we get

$$(\sigma_{\mathbf{b}}(\sigma I) - I)\, g_{\mathbf{b}} = M_{\mathbf{b}}(c_0)(f_0)_{\mathbf{b}} + \cdots + M_{\mathbf{b}}(c_{k-1})(f_{k-1})_{\mathbf{b}}.$$

The coefficients c_i correspond to the column vector $(c_i, \ldots, c_i)^T$ in the basis $\mathbf{b}$, since $c_i \in R$. Then the matrices $M_{\mathbf{b}}(c_i)$, representing multiplication by c_i are diagonal matrices with all entries on the diagonal equal to c_i. Now, the equation in basis $\mathbf{b}$ becomes

$$(\sigma_{\mathbf{b}}(\sigma I) - I)\, g_{\mathbf{b}} = c_0(f_0)_{\mathbf{b}} + \cdots + c_{k-1}(f_{k-1})_{\mathbf{b}}. \tag{3.7}$$

In order to apply Karr's algorithm to this problem, the left side still needs to be uncoupled. Applying the transformation matrix $T_{n,1}$ from Section 3.2, we get

$$\begin{pmatrix} \sigma^n - 1 & 0 & \ldots & 0 \\ 0 & \sigma^n - 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma^n - 1 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix} =$$

$$\sum_{0 \le i < k} c_i \begin{pmatrix} f_{i,0} + \sum_{1 \le j < n} \sigma^{n-j}(f_{i,j}) \\ f_{i,1} + \sum_{1 \le j < n} \sigma^{n-j}(f_{i,j+1 \pmod n}) \\ \vdots \\ f_{i,n-1} + \sum_{1 \le j < n} \sigma^{n-j}(f_{i,j-1 \pmod n}) \end{pmatrix}.$$

To find the solutions $g \in E$ and $c_0, \ldots, c_{k-1} \in \mathrm{const}_\sigma(E)$, it is enough to solve only one equation from this system. Once one coordinate of g is known, the others can be obtained by substituting this value in the original equations obtained from (3.7). The right hand side of these equations are explicitly stated in (3.3). For example, if g_0 is known, g_1 can be obtained from

$$g_1 = \sigma(g_0) - \sum_{0 \le i < k} c_i f_{i,0}.$$

Example 3.45. Consider the sum

$$\sum_{0 \le i < n} (-1)^i H_i^{(2)} \binom{n}{i},$$

where $H_i^{(2)} = \sum_{1 \le j < i+1} \frac{1}{j^2}$ denotes the harmonic numbers of second order. The summand can be represented in the difference ring $E = \mathbb{Q}(i, b, h)[t]/\langle t^2 - 1 \rangle$, where $\sigma(i) = i + 1$, $\sigma(b) = \frac{n-i}{i+1} b$, and $\sigma(h) = h + \frac{1}{(i+1)^2}$.

The equation $\sigma(g) - g = thb$ does not have a solution in E. Hence, there is no antidifference for this summand. Though, the creative telescoping method can be used by adding a second term to the right hand side, the summand shifted by 1 with respect to n. Then, the equation becomes

$$\sigma(g) - g = c_0 thb + c_1 \left(\frac{n+1}{n+1-i} thb \right).$$

When uncoupled, this is transformed to the system

$$\sigma^2(g_0) - g_0 = c_0 \left(hb - \sigma(hb) \right) + c_1 \left(\frac{n+1}{n+1-i} hb - \sigma\left(\frac{n+1}{n+1-i} hb\right) \right).$$

We get the nontrivial solution

$$g_0 = -\frac{bhi^2}{i-n-1} - \frac{b}{n+1}, \quad c_0 = n, \text{ and } c_1 = -n-1.$$

Then, we have g_1 by

$$\begin{aligned} g_1 &= \sigma(g_0) + c_0(hb) + c_1\left(\frac{n+1}{n+1-i} hb\right) \\ &= \sigma\left(-\frac{bhi^2}{i-n-1} - \frac{b}{n+1} \right) + nhb - \frac{(n+1)^2}{n+1-i} hb \\ &= \frac{bhi^2}{i-n-1} + \frac{b}{n+1} \end{aligned}$$

Hence,

$$g = (-1)^{i+1} \binom{n}{i} \left(\frac{H_i i^2}{i-n-1} + \frac{1}{n+1} \right)$$

The commands used for the computations in this example are listed in Example A.6.

4. Nullspace computation over rational function fields

This chapter, based on joint work with Arne Storjohann, presents two algorithms for computation of nullspaces of matrices over $\mathbb{Q}(x)$. This is a common bottleneck in symbolic summation algorithms, especially hypergeometric summation methods as described briefly in the introduction to Chapter 2. Experiments based on implementations of these in the Sage computer algebra system suggest that a combination of homomorphic imaging and early termination strategies outperforms previous implementations present in established computer algebra systems or specialized packages for symbolic summation.

The problem

Symbolic summation methods, either based on the WZ-Fasenmyer paradigm [29,80,81] or Karr's algorithm [41], reduce the given summation problem to finding vectors in the nullspace of a matrix over a rational function field with one or more variables. For small examples, this reformulation leads to matrices that can be solved easily with classical methods. However, even moderately large summation problems can get stuck at this phase due to intermediate expression swell.

For example, we show how the problem of finding a recurrence for a hypergeometric function

$$F(k,i,j,X) = \binom{k}{j}\binom{j}{i} x^i y^{j-i} x^{k-j} = (x+y+z)^k$$

with $X = (x,y,z)$, can be transformed to a nullspace computation over $\mathbb{Q}(x,y,z)$ using Fasenmyer's method. We first make an Ansatz for the structure set

$$S = \{(0,0,0),(1,0,0),(1,0,1),(1,1,1)\}$$

and write

$$\sum_{(a,b,c)\in S} p_{(a,b,c)}(k,i,j)F(k-a,i-b,j-c,X) = 0 \tag{4.1}$$

for unknown $p(k,i,j) \in \mathbb{Q}[k,i,j]$. Since F is hypergeometric in all variables, dividing this equation by $F(k,i,j,X)$ yields

$$\sum_{(a,b,c)\in S} p_{(a,b,c)}(k,i,j) r_{(a,b,c)}(k,i,j,X) = 0$$

with $r_{(a,b,c)}(k,i,j,X) \in \mathbb{Q}(k,i,j,X)$. Clearing denominators and comparing coefficients of monomials in X leads to the system of equations described by the matrix

$$M = \begin{pmatrix} z & 1 & 0 & 0 \\ 0 & -y & z & 0 \\ 0 & 0 & -x & y \end{pmatrix}.$$

Any vector in the nullspace of M gives a set of coefficients $p_{(a,b,c)}(k,i,j)$ for (a,b,c) in the structure set S such that Equation (4.1) holds.

Note that linearly independent vectors in the nullspace do not necessarily lead to independent solutions of this equation, since this reformulation of the problem does not take the shift operators into account.

In this case, the nullspace has dimension one. It is generated by the vector

$$N = \begin{pmatrix} 1 & -z & -y & -x \end{pmatrix}^T.$$

Substituting the values of the coefficients $p_{(a,b,c)}$ with the corresponding entries of N, we obtain the recurrence

$$F(k,i,j) - zF(k-1,i,j) - yF(k-1,i,j-1) - xF(k-1,i-1,j-1) = 0.$$

With this approach free variables in the recurrence (4.1), the variables which are not shifted, end up in the matrix formed by coefficient comparison. Generally, for other summation problems as well, the matrix M contains the parameters which are not summation variables.

We concentrate on the case where the matrix M depends only on one variable with the aim of using this as a base case for multivariate problems. The algorithms we describe take a matrix $M \in \mathbb{Z}[x]^{n\times(n+m)}$ of rank n as input and return $N \in \mathbb{Q}(x)^{(n+m)\times m}$, a right nullspace of M, as output. In particular, we are interested in matrices where n and m are typically large and the degree of M, that is, the maximum degree of its entries, is low. Since these matrices are generated by summation problems, they have an inherent structure and the degree of the entries in the nullspace N is low.

For example, while computing a recurrence in n for the expression

$$\frac{(-1)^{k+l}\binom{j}{k}\binom{-j+n-2}{l}\Gamma(l+1)\Gamma(n)\Gamma(k+l+s+2)s!}{(k+l+1)\Gamma(n+1)\Gamma(l+s+2)\Gamma(k+l+s+3)}$$

using Wegschaider's Mathematica package [80], we obtain a matrix M with $n = 112$, $m = 19$, degree 3, and coefficients with maximum 32 bits. Entries in the nullspace N for this matrix have degree 15 again with maximum 32-bit coefficients.

Other approaches

One of the classical approaches to this problem is Gaussian elimination with heuristic pivoting strategies such as Markowitz pivoting. It is well known that intermediate expressions obtained in the process of Gaussian elimination grow rapidly. Coupled with the complexity of rational function arithmetic, this becomes an important obstacle. Choosing pivots that reduce the fill-in at each step, for example the row with the least number of nonzero entries, or minimize coefficient growth can help control intermediate expression swell. Indeed, this is the method used by Axel Riese's Mathematica implementation.

Another classical method is to use fraction-free elimination [12] in order to avoid rational function arithmetic. However, this elimination strategy has shortcomings when the input matrix has many zero entries [48] and the degrees of the intermediate polynomials still grow large.

In this chapter we report on implementations of two algorithms based on modern computer algebra methods to avoid intermediate expression swell. These approaches make essential use of many asymptotically fast methods for integers and polynomials, including not only multiplication, but also radix conversion, interpolation, rational function and number reconstruction. The single problem of computing a nullspace over $\mathbb{Q}(x)$ thus provides a good test of, and should motivate the further development of, highly optimized libraries such as GMP [35], FFLAS [26], FLINT [37] and zn_poly [38].

4.1. Outline of approach

Both approaches we present share a common framework to reduce the problem of computing a right nullspace of an arbitrary rank and shape matrix M over $\mathbb{Q}(x)$ to that of computing the nullspace of a full row rank matrix $[\,A\,|\,B\,] \in \mathbb{Q}(x)^{n\times(n+m)}$, with $A \in \mathbb{Q}(x)^{n\times n}$, $B \in \mathbb{Q}(x)^{n\times m}$, and A nonsingular. We also assume that the entries of A and B are actually over $\mathbb{Z}[x]$. This step can be performed in a straightforward manner using a Monte Carlo algorithm. To find a set of rows and columns which give a full rank square minor A, we first clear denominators in the matrix and reduce the coefficients modulo a random word-size prime p. By evaluating these entries we obtain a matrix $\bar{A}$ over $\mathbb{Z}_p$. The rows and columns which give a full rank minor of $\bar{A}$ will lead to a full rank minor of M with high probability.

The problem is further reduced to compute a nullspace over $\mathbb{Z}_p(x)$ by reducing the coefficients in the entries of the matrix $[\,A\,|\,B\,]$ modulo a word-size prime p. The modular images for the result are lifted to $\mathbb{Q}[x]$.

To use homomorphic imaging we need to ensure consistent images after the reduction to prime modulus. This preprocessing step ensures that the matrix $[\,A\,|\,B\,] \in \mathbb{Z}[x]^{n\times(n+m)}$ has a canonical nullspace basis

$$\left[\begin{array}{c} sA^{-1}B \\ \hline -sI \end{array}\right] \in \mathbb{Q}[x]^{(n+m)\times m},$$

where the scaling polynomial $s \in \mathbb{Q}[x]$, a factor of $\det A$, is used to clear denominators from $A^{-1}B \in \mathbb{Q}(x)$. The pair $(s, sA^{-1}B)$ is computed modulo various word-size primes p and the final result over $\mathbb{Q}[x]$ is recovered using Chinese remaindering and, if needed, rational number reconstruction.

Bound for Chinese remaindering over $\mathbb{Z}[x]$

To make sure the result obtained from homomorphic imaging is correct, we need a bound on the size of the coefficients for the polynomials that occur in the answer.

If $A = (a_{i,j})$ is an $n \times n$ matrix over $\mathbb{Z}$, Hadamard's inequality states that

$$|\det A| \leq \left(\prod_{i=1}^{n} \sum_{j=1}^{n} |a_{i,j}|^2 \right)^{\frac{1}{2}} \equiv H(A).$$

Now let $A(x) = (A_{i,j}(x))$ be a matrix over $\mathbb{Z}[x]$. Let $a_0, a_1, \ldots,$ be the coefficients of the polynomial representation of $\det A(x)$. If $W = (w_{i,j})$, where $w_{i,j}$ denotes the sum of the absolute values of the coefficients of $A_{i,j}(x)$, then

$$\left(\sum |a_k|^2 \right)^{\frac{1}{2}} \leq H(W).$$

Therefore, $H(W)$ provides a bound for the coefficients of the determinant s and the matrix $sA^{-1}B$.

Note that it is possible to use early termination strategies at this step. For example, the strategy described in [20] can be adapted to the polynomial setting.

Two approaches to compute a canonical nullspace

We have implemented two different core computational routines using this framework to compute a suitable tuple $(s, sA^{-1}B)$. The first one based on x-adic Dixon lifting to utilize BLAS based optimization is described in Section 4.2, and the second one using the outer product adjoint formula [76] in Section 4.3. We briefly summarize these two approaches here.

Output sensitive x-adic lifting

> We compute $(s, sA^{-1}B) \bmod p \in \mathbb{Z}_p[x]$ using x-adic lifting, with an output sensitive approach that performs lifting up to $\max(\deg s, \deg sA^{-1}B)$ instead of the a priori degree bound nd. Here, s will be a monic divisor, possibly proper, of $\det A$, and the final recovery of the result over $\mathbb{Q}[x]$ requires rational number reconstruction as well as Chinese remaindering. The lifting can be reduced to calling the FFLAS [26] library to perform multiply-add operations, which are implemented efficiently using hardware floating point arithmetic. The asymptotic cost of computing each image is $O\tilde{}(n^3md)$ operations modulo p. Due to the structure of the input matrices, using an output sensitive approach both for the x-adic lifting and Chinese remaindering greatly improves the performance of this method.

Nullspace via outer product adjoint formula

> We compute $(\det A, (\det A)A^{-1}B) \bmod p \in \mathbb{Z}_p[x]$ using the outer product adjoint formula approach of [76]. The cost of computing each image is $O\tilde{}(n^3d +$

n^2md) operations modulo p. The outer product formula requires us to work with a preconditioned matrix whose adjoint has degrees $(n-1)d$. This hides the structure of the inputs and prevents effective use of an early termination strategy.

4.2. BLAS optimized output sensitive x-adic lifting

We will describe a variation of linear x-adic lifting [21,25] that reduces almost all the computation to BLAS matrix operations to solve a linear system $Av = B$ over $\mathbb{Z}_p[x]$ where p is a word size prime. Recall that $A \in \mathbb{Z}_p^{n\times n}$ and $B \in \mathbb{Z}_p^{n\times m}$. The solution matrix v will have rational function entries in $\mathbb{Z}_p(x)$.

For this procedure we need the condition $x \perp \det(A)$. In case this is not true, we consider $A(x+a)$, where $a \in \mathbb{Z}_p^*$ is chosen randomly. This transformation can be reversed at the end of the computation by evaluating at $x-a$.

The x-adic lifting algorithm first determines the inverse C of the matrix A modulo x using classical methods. That is, we find $CA \equiv I \pmod{x}$ with matrix inversion over $\mathbb{Z}_p$. This can be done in $O(n^3)$ steps. Next, an x-adic approximation, $z \in \mathbb{Z}_p[[x]]$, for v is computed incrementally. Consider the matrices b_i and z_i where $b_0 = B$ and $z_0 \equiv Cb_0 \pmod{x}$. The solution $z = z_0 + z_1x + \dots$ is lifted at each step by computing

$$\begin{aligned} b_{i+1} &= x^{-1}(b_i - Az_i), \text{ and} \\ z_{i+1} &\equiv Cb_{i+1} \pmod{x}. \end{aligned}$$

Note that $(b_i - Az_i) \equiv A(Cb_i - z_i) \equiv 0 \pmod{x}$, so the entries of b_{i+1} are polynomials. Now, let $z^{(k)} = \sum_{0\le i<k} z_i x^i$. We have

$$Az^{(k)} = \sum_{0\le i<k} x^i A z_i = \sum_{0\le i<k} x^i(z_i - xz_{i+1}) = b_0 - x^k b_k.$$

Hence, $Az^{(k)} \equiv B \pmod{x^k}$.

Let $d = \deg(A)$ and $e = \deg(B)$. In the equality $Av = B$, the numerators of the entries of v have degree bounded by $N = (n-1)d + e$ and the degrees of the denominators are bounded by $D = nd$. In order to recover the matrix v correctly from the approximation $z^{(k)}$ using rational function reconstruction, k should satisfy $k > N + D$.

Now we describe the core algorithm which reduces the lifting step to multiply-add operations implemented in BLAS libraries. We assume that p satisfies $n(p-1)^2 + p \le 2^{53} - 1$, thus meeting the precondition for use of BLAS routines for matrix operations over $\mathbb{Z}_p$.

We can write A and B, and the linear system $Av = B$, as matrix polynomials

$$\overbrace{(A_0 + A_1x + A_2x^2 + \dots + A_dx^dv)}^{A} = \overbrace{(B_0 + B_1x + \dots + B_ex^e)}^{B}. \tag{4.2}$$

By computing the inverse of $A_0 \in \mathbb{Z}_p^{n\times n}$, and multiplying both sides of (4.2) by this inverse, we may assume, without loss of generality, that the constant coefficient A_0 of A is equal to I_n. Also, for convenience, and without loss of generality, by negating $A_1, A_2, \ldots, A_d$, we will write the equation with the $+$'s in the left hand side replaced by $-$'s:

$$\overbrace{\left(I - A_1x - A_2x^2 - \cdots - A_dx^d\right)}^{A} v = \overbrace{\left(B_0 + B_1x + \cdots + B_ex^e\right)}^{b}. \tag{4.3}$$

We will compute the x-adic expansion of the solution v of (4.3) up to order x^{N+D+1}. Let $z_0, \ldots, z_{N+D} \in \mathbb{Z}_p^{n\times m}$ such that $z_i = B_i$ for $i = 0, \ldots, e$ and $z_i = 0$ for $i > e$. Note that initially the first $e+1$ vectors $z_1, z_2, \ldots, z_e$ correspond to the right hand side of (4.3). The following code modifies the z_i in-place.

```
$N := (n-1)d + e$
$D := nd$
for $i = 0$ to $N + D$
        do for $j = 1$ to $d$
                do if $i + j \leq N + D$
                        then $z_{i+j} := z_{i+j} + A_j z_i$
```

The inner `for` loop for the variable `j` performs the lifting step $b_{i+1} = x^{-1}(b_i - Az_i)$. Viewing A as a matrix polynomial, we formulate the polynomial multiplication as multiplication of the coefficient matrices.

On termination we have

$$v \equiv z_0 + z_1x + \cdots + z_{N+D} \mod x^{N+D+1}.$$

The cost of this operation is less than $2nd^2 + de$ matrix-vector products. The rational function presentation of the solution can now be obtained via rational function reconstruction.

The bound $N + D$ for the degree of the series solution is often too pessimistic. Assume that we only lifted up to $k < N + D$. It is possible to verify the correctness of the solution $z^{(k)}$. The rational function reconstruction step fails with an error if there is no unique rational function representation for the entries of $z^{(k)}$. In this case, we increase k and continue with the lifting. Note that the lifting loop allows to reuse the data that is already computed, so we do not have to start from the beginning. Once all the entries of $z^{(k)}$ have been reconstructed, we can verify that the result is correct with the following lemma adapted from [20].

Lemma 4.1. *Let $u \in \mathbb{Z}_p[x]$ be the least common multiple of the denominators obtained from rational function reconstruction of the entries of $z^{(k)}$ and $t \equiv uz^{(k)} \pmod{x^k}$. If $\deg(u)\deg(B) < k$ and $n\deg(A)\deg(t) < k$ then $At = Bd$.*

Proof. From the lifting process we know that $Az^{(k)} \equiv B \pmod{x^k}$. Multiplying this equivalence with the denominator u we get $At - Bd \equiv 0 \pmod{x^k}$. The degree of the left hand side is less than k, so we can conclude that $At - Bd = 0$. □

4.3. Nullspace via outer product adjoint formula

We compute the canonical nullspace $(\det(A), \det(A)A^{-1}B)$, based on the outer product adjoint formula algorithm described in [76].

Recall that the adjoint of an $n \times n$ nonsingular matrix A, denoted $A^{\mathtt{adj}}$, satisfies $A^{\mathtt{adj}} = \det(A)A^{-1}$. The algorithm presented in [76] computes an efficient representation of the adjoint, called the outer product adjoint formula. This representation can be used to compute $A^{\mathtt{adj}}v$ for a vector v without having to explicitly write down the entries of $A^{\mathtt{adj}}$.

The Smith form of a nonsingular matrix $A \in \mathbb{Z}_p[x]^{n\times n}$ is given by $S = UAV = \mathrm{Diag}(s_1, s_2, \ldots, s_n)$, where U and V are unimodular matrices and $s_i \mid s_{i+1}$ for $1 \leq i < n$. Let v_i and u_i be column i and row i of V and U respectively, for $1 \leq i \leq n$. Inverting both sides of the equation $S = UAV$, multiplying by s_n gives

$$s_nA^{-1} = V(s_nS^{-1})U = \frac{s_n}{s_n}v_nu_n + \frac{s_n}{s_{n-1}}v_{n-1}u_{n-1} + \cdots + \frac{s_n}{s_1}v_1u_1.$$

Here v_iu_i, where v_i is a column vector and u_i is a row vector, denotes the outer product. Since each outer product v_iu_i is scaled by s_n/s_i, this equation will still hold modulo s_n if the entries of v_i and u_i are reduced modulo s_i for $1 \leq i \leq n$.

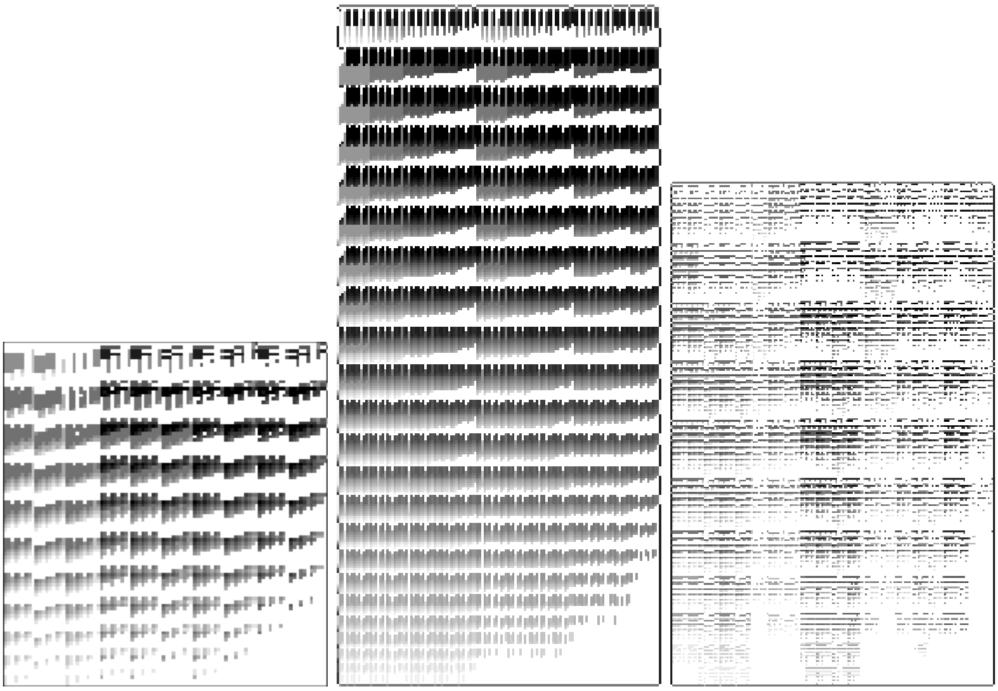
An outer product adjoint formula of a nonsingular $A \in \mathbb{Z}_p[x]^{n\times n}$ is a set of tuples $(s_{n-i}, v_{n-i}, u_{n-i})_{0\leq i<k}$ such that

- the Smith form of A is $\mathrm{Diag}(1, 1, \ldots, 1, s_{n-k+1}, s_{n-k+2}, \ldots, s_n)$ with $s_{n-k+1} \neq 1$,
- $v_{n-i+1} \in (\mathbb{Z}_p[x]/\langle s_{n-i+1}\rangle)^{n\times 1}$ and $u_{n-i+1} \in (\mathbb{Z}_p[x]/\langle s_{n-i+1}\rangle)^{1\times n}$ for $0 \leq i < k$,
- $s_nA^{-1} \equiv \frac{s_n}{s_n}v_nu_n + \frac{s_n}{s_{n-1}}v_{n-1}u_{n-1} + \cdots + \frac{s_n}{s_1}v_1u_1 \pmod{s_n}$.

The algorithm `OuterProductAdjoint` from [76] computes an outer product adjoint formula for a nonsingular matrix A. This formula can be applied to a vector v using the algorithm `OPM` in [76] to obtain $A^{\mathtt{adj}}v \pmod{s_n}$. Now the canonical nullspace modulo s_n can be computed by applying the `OPM` method to each column of B.

The polynomial s_n will be the determinant of A in our case. Since we can only compute a result with maximum degree given by $\deg(\det(A))$, we apply a preprocessing step to ensure $\deg(A) = \deg(\det(A))$. This can be done by making sure that $x \nmid \det(A)$, by shifting the polynomials in A by a random $a \in \mathbb{Z}_p$ if necessary, then reversing the coefficients of the entries of A, that is considering $x^dA(1/x)$ where $d = \deg(A)$. The coefficients matrix for degree d of the resulting matrix in this case will have full rank, which will produce a determinant with degree d.

Note that the computational core of this approach relies on fast polynomial arithmetic over $\mathbb{Z}_p[x]$. We use the FLINT library [37] which provides implementations of asymptotically fast algorithms for arithmetic in $\mathbb{Z}[x]$ and $\mathbb{Z}_p[x]$. Unfortunately, these routines cannot match the performance of the BLAS library used for x-adic lifting. It may be possible to optimize this algorithm faster by reducing more steps to only coefficient arithmetic over $\mathbb{Z}_p$.



(a) $C_{5,2k}$

dim:	100 x 94
degree:	8
rank:	68

(b) $C_{6,2k}$

dim:	340 x 161
degree:	16
rank:	160

(c) $C_{7,2k}$

dim:	853 x 545
degree:	10
rank:	401

darker dots indicate a higher degree polynomial

Figure 4.1.: Dataset used for timings

4.4. Performance comparison

We compare the performance of the output sensitive x-adic lifting implementation in the open source computer algebra system Sage to a custom Mathematica implementation by Axel Riese and the `linalg[nullspace]` function of Maple 12. These are indicated by the labels lifting, `EANullSpace`, and Maple respectively in the following table.

The input matrices are generated by the Mathematica package `MultiSum` [80] for the problems $C_{5,2k}, C_{6,2k}$ and $C_{7,2k}$ from [74], using the input provided therein.

	lifting	`EANullSpace`	Maple
$C_{5,2k}$	17 s	20 s	13 s
$C_{6,2k}$	42 s	$>$ 1 hour	$>$ 1 hour
$C_{7,2k}$	3227 s	-	-

The following table provides timings to compute a canonical nullspace for a matrix with random entries of dimension 500 x 600 with the given degree, using x-adic lifting and the outer product adjoint formula algorithms.

degree	lifting	OPAF
10	5341 s	5267 s
20	20461 s	20599 s
30	45879 s	45030 s

These computations were done on a 2.66GHz Intel(R) Xeon(R) X7460 CPU.

Conclusions

The x-adic lifting method is especially well suited to symbolic summation applications since it allows the use of output sensitive lifting. Instead of the a priori degree bound nd, lifting is performed up to $\max(\deg s, \deg sA^{-1}B)$. Empirical evidence suggests that this degree is considerably lower than the expected degree of the solution matrix. Moreover, the lifting process can take advantage of various optimizations, including parallelization, of the underlying BLAS library.

The complexity of the outer product adjoint formula algorithm, $O\tilde{}(n^3d + n^2md)$, is better than x-adic lifting. Yet, the outer product formula works with a preconditioned matrix whose adjoint has degrees $(n-1)d$. This hides the structure of the input and prevents use of an early termination strategy. Even though the performance of this method on random matrices is asymptotically faster than x-adic lifting, for symbolic summation problems it is not preferable.

A. Implementation

We present an implementation of Karr's symbolic summation framework in the open source computer algebra system Sage [28, 75]. Due to the nature of open source software, this allows direct experimentation with the algorithms and structures involved while taking advantage of the state of the art primitives provided by Sage. Even though these methods are used behind the scenes in the summation package Sigma [64] and they were previously implemented in [30], this is the first open source implementation.

Design Issues

Sage provides an algebraic type hierarchy similar to that of Magma. Each algebraic structure in Sage is represented by a corresponding *parent* and an *element* class. This implementation defines new parent and element classes inheriting from `FractionField` and `FractionFieldElement` respectively to represent ΠΣ-fields and their elements.

These classes define methods corresponding to the subproblems in the algorithms or mathematical structure at hand. For example, the element methods can compute σ-factorials:

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F = PiSigmaField(QQ, 1, 1,'x')
sage: x = F.gen()
sage: (x+1)._sfactorial(-3)
(-x + 1)/(x^2 - 2*x)
```

ΠΣ-fields can determine the structure of an algebraic extension.

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F.<k> = PiSigmaField(QQ, 1, 1)
sage: E.<e> = F.extension((-1,0))
sage: E
Difference Ring with base Quotient of Multivariate Polynomial Ring\
        in e, k over Rational Field by the ideal (e^2 - 1) and \
        homomorphism Ring morphism:
  From: Multivariate Polynomial Ring in e, k over Rational Field
  To:   Fraction Field of Multivariate Polynomial Ring in e, k \
                  over Rational Field
  Defn: e |--> -e
        k |--> k + 1
```

Recent additions to the Sage library also allow the specification of a collection of methods which can be used for mathematical objects satisfying certain properties. These *category* definitions, inspired by those of Aldor or MuPAD, provide a more flexible approach to attach the specialized methods expected from elements of ΠΣ-fields to any element class provided by Sage.

Construction of difference fields

In order to construct ΠΣ-fields directly, without going through the verification process to check if the given extension is homogeneous or first-order linear, the `PiSigmaExtension` function can be used. This function takes 3 arguments: the base field, and the coefficients α and β where $\sigma(t) = \alpha t + \beta$.

```
sage: from karr.pi_sigma_field import PiSigmaExtension
sage: F.<n> = PiSigmaExtension(QQ, 1, 1)
sage: sigma = F.sigma(); sigma
Ring endomorphism of PiSigmaField with constant field Rational\
    Field and tower:
[('n', 1, 1)]
  Defn: n |--> n + 1
sage: sigma(n)
n + 1
sage: E.<b> = PiSigmaExtension(F, 2, 0)
sage: sigma = E.sigma(); sigma
Ring endomorphism of PiSigmaField with constant field Rational\
    Field and tower:
[('n', 1, 1),
 ('b', 2, 0)]
  Defn: b |--> 2*b
        n |--> n + 1
```

In the following examples, this construct is used extensively.

Element methods

The class defining elements of ΠΣ-fields defines methods to compute properties such as *specification of the equivalence*, Π and Σ-regularity [41, Definition 16-17].

Example A.1. Continuing from the previous example, we have:

```
sage: (n+1).spec(n+3)
2
sage: n.spec(n^2) is None
True
sage: t = n*b+1
sage: t.spec( (sigma^3)(t) )
3


sage: (n+1).pi_regularity(n^2 + 3*n + 2)
2
sage: (n*b).pi_regularity(2*n^2*b^2 + 2*n*b^2)
2
sage: (n+1).sigma_regularity(n^2 + 4*n + 4)
```

```
3
# ._sfactorial() computes the sigma factorial
sage: (n*b).sigma_regularity((n*b)._sfactorial(4))
4
```

For a difference field F, σ, let $H(F) = \left\{ \frac{\sigma(g)}{g} \,\middle|\, g \in F^* \right\}$. Given $f_1, \ldots, f_k \in F$, we can compute the set of $(n_1, \ldots, n_k) \in \mathbb{Z}^k$ such that $f_1^{n_1} f_2^{n_2} \cdots f_k^{n_k} \in H(F)$ using the algorithm described in [41, Theorem 8].

Example A.2. We compute the result from Example 2.28.

```
sage: from karr.orbit import homogeneous_group_exponents
sage: f1 = (n+1)*(n+2)
sage: f2 = (n+2)*(n+3)
sage: f3 = n*(n+1)
sage: homogeneous_group_exponents(f1, f2, f3, parent=F)
[ 1  0 -1]
[ 0  1 -1]
```

We can also compute denominator bounds [17,63] and degree bounds [41,65] which allow us to use the algorithm described in [68] to solve first-order linear difference equations and find telescopers over $\Pi\Sigma$-fields. More precisely, given a $\Pi\Sigma$-field (F, σ) with $\mathrm{const}_\sigma(F) = K$ and $a_0, a_1, f_1, \ldots, f_n \in F$, $c_1, \ldots, c_n \in K$, we can find $g \in F$ and $c_1, \ldots, c_n \in K$ satisfying $a_1\sigma(g) + a_0 g = c_1 f_1 + \cdots + c_n f_n$. Note that the solutions form a vector space $V \subset K^n \times F$.

Difference equations

In the following example, we consider the sum $\sum_{i=1}^{n-1} H_i^2$ where H_i denotes the harmonic sum $\sum_{i=1}^{n} \frac{1}{i}$ and find an equivalent expression containing only single sums. By extending the previously defined difference field `F` $= (\mathbb{Q}(n), \sigma)$ with an element h satisfying $\sigma(h) = h + \frac{1}{n+1}$, we construct the $\Pi\Sigma$-field $\mathbb{Q}(n, h)$ containing our summand h^2 .

```
sage: H.<h> = PiSigmaExtension(F, 1, 1/(n+1))
sage: h, n = H.gens()
```

We call the `solve_plde()` function, to find $g \in H$ such that $\sigma(g) - g = h^2$. The first argument is the field we work in, `H` in this case. The second argument is a tuple with the coefficients (a_0, a_1). In this case we have $(a_1, a_0) = (1, -1)$, which we convert to be elements of H using the Python command `map(H, (1, -1))`. The last argument is a list containing the f_i's, where we have only the summand h^2.

```
sage: from karr.plde import solve_plde
sage: solve_plde(H, map(H, (1, -1)), [h^2])
(
[1]  [h^2*n - 2*h*n - h + 2*n]
[0], [                      1]
)
```

The answer has two matrices, in $K^{2\times 1}$ and $F^{2\times 1}$ respectively. A row of the first matrix contains the c_i component of a solution, while the corresponding row in the second matrix gives g. We have $c_1 = 1, g = h^2 n - 2hn - h + 2n$ in the first row and the trivial solution $c_1 = 0, g = 1$ in the second row. From the first row we can deduce that our sum $\sum_{i=1}^{n-1} H_i^2$ is equal to $g(n) - g(1) = H_n^2 n - 2H_n n - H_n + 2n$.

Next example reproduces the computations in Example 2.3 of [68], which were used in the proof of the identity

$$\sum_{k=0}^{n}(1 - 3(n-2k)H_k)\binom{n}{k}^3 = (-1)^n$$

appearing in [54].

```
sage: K.<n> = FractionField(QQ['n'])
sage: F.<k> = PiSigmaExtension(K, 1, 1)
sage: E.<b> = PiSigmaExtension(F, ((n-k)/(k+1))^3, 0)
sage: H.<h> = PiSigmaExtension(E, 1, 1/(k+1))
sage: f = (b*(1 + h*(-6*k + 3*n)), \
        (b*(1 + n)^3*(1 + h*(3 - 6*k + 3*n))) / (1 - k + n)^3, \
        (b*(1 + n)^3*(2 + n)^3*(1 + h*(6 - 6*k + 3*n))) / \
                (2 + k^2 + k*(-3 -2*n) + 3*n + n^2)^3)
sage: C, g = solve_plde(H, map(H, (1, -1)), f)
sage: C.row(0)
(6*n + 6, 12*n + 18, 6*n + 12)
sage: g.row(0).factor()
(6) * (k - n - 2)^-3 * (k - n - 1)^-3 * (n + 1) * b * k^2 * \
        (3*h*k^5 - 15*h*k^4*n - 24*h*k^4 + 27*h*k^3*n^2 + \
        90*h*k^3*n + 72*h*k^3 - 24*h*k^2*n^3 - 120*h*k^2*n^2 -\
        195*h*k^2*n - 102*h*k^2 + 12*h*k*n^4 + 78*h*k*n^3 + \
        186*h*k*n^2 + 192*h*k*n + 72*h*k - 2*k^4 + 12*k^3*n + \
        18*k^3 - 27*k^2*n^2 - 84*k^2*n - 63*k^2 + 28*k*n^3 + \
        134*k*n^2 + 208*k*n + 104*k - 12*n^4 - 78*n^3 - \
        186*n^2 - 192*n - 72)
```

Note that this is an application of the creative telescoping method and `f` contains 3 components, $S(n)$, $S(n+1)$ and $S(n+2)$ where $S(n) = \sum_{k=0}^{n}(1 - 3(n-2k)H_k)\binom{n}{k}^3$.

Examples

This section includes listing corresponding to the examples from the text.

Example A.3. Code from Example 3.26

```
# (-1)^n H_k / binom(n,k)
sage: from karr.pi_sigma_field import PiSigmaField
sage: K = Frac(QQ['n'])
sage: F.<k> = PiSigmaField(K, 1, 1)
sage: F.inject_variables()
sage: B.<b> = F.extension(( (n-k)/(k+1), 0 ))
sage: H.<h> = B.extension((1, 1/(k+1)))
```

```
sage: from karr.plde import solve_plde
sage: C, g = solve_plde(H, map(H,(1,0,-1)), [h/b-H.sigma()(h/b)])
sage: C
[0]
[1]
sage: g[0,0]
1
sage: g[0,1]
(h*k*n + 2*h*k - h*n^2 - 3*h*n - 2*h - k + n + 1)/\
         (b*n^2 + 4*b*n + 4*b)
```

Example A.4. Code from Example 3.29.

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F.<n> = PiSigmaField(QQ, 1, 1)
sage: E1.<e1> = PiSigmaField(F, 4, 0)
sage: E2.<e2> = PiSigmaField(E1, 9, 0)
sage: from karr.orbit import hom_group_exponents
sage: hom_group_exponents(E2(6), parent=E2)
[2]
sage: from karr.plde import solve_plde
sage: solve_plde(E2, map(E2,(1,-6)), [E2(0)])
(
[1]  [    0]
[0], [e1*e2]
)
```

Example A.5. Code from Example 3.37.

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F.<k> = PiSigmaField(QQ, 1, 1)
sage: E.<e> = F.extension((-1,0))
sage: from karr.plde import solve_plde
sage: solve_plde(E, map(E, (1,-1)), [e/(k+1)])
(
[0], [1]
)
```

Example A.6. Code from Example 3.45.

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: K = Frac(QQ['n'])
sage: F.<k> = PiSigmaField(K, 1, 1)
sage: F.inject_variables()
Defining k, n
sage: B1.<b1> = PiSigmaField(F, (n-k)/(k+1), 0) # binomial(n, k)
sage: H.<h> = PiSigmaField(B1, 1, 1/(k+1)^2) # H^(2)_{k}
sage: H.inject_variables()
Defining h, b1, k, n
sage: sigma = H.sigma()
sage: f0 = b1*h
```

```
sage: f1 = (n+1)/(n+1-k)*f1
sage: from karr.plde import solve_plde
sage: C,g = solve_plde(H, map(H, (1, 0, -1)), \
          map(lambda x: x - sigma(x), [f1, f2]))
sage: C
[      n -n - 1]
[      0      0]
sage: g.column(0)
((h*b1*k^2*n + h*b1*k^2 + b1*k - b1*n - b1)/\
          (-k*n - k + n^2 + 2*n + 1), 1)
sage: sigma(g[0,0]) + n*f0 - (n+1)*f1
(-h*b1*k^2*n - h*b1*k^2 - b1*k + b1*n + b1)/\
          (-k*n - k + n^2 + 2*n + 1)
```

Example A.7. 3.44

```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F.<n> = PiSigmaField(QQ, 1, 1)
sage: E.<h2> = PiSigmaField(F, 1, 1/(n+1)^2)
sage: GG.<hh> = PiSigmaField(E, -1, -1/(E(n)+1))
sage: sigma = GG.sigma()
sage: GG.inject_variables()
Defining hh, h2, n
sage: f = sigma(hh/(n+1)) + hh/(n+1) + \
          sigma(1/(n+1)^2) + 1/(n+1)^2; f
(hh*n^2 + 3*hh*n + 2*hh + n^2 + 3*n + 3)/\
          (n^4 + 6*n^3 + 13*n^2 + 12*n + 4)
sage: from karr.plde import solve_plde
sage: C, g = solve_plde(GG, map(GG, (1,0,-1)), [f]); C, g
(
[1]  [1/2*hh^2 + 1/2*h2]
[0], [                1]
)
```

Example A.8.
```
sage: from karr.pi_sigma_field import PiSigmaField
sage: F.<k> = PiSigmaField(QQ, 1, 1)
sage: H.<h> = PiSigmaField(F, 1, 1/(k+1))
sage: GG.<hh> = PiSigmaField(H, -1, -1/(k+1))
sage: GG.inject_variables()
Defining hh, h, k
sage: from karr.plde import solve_plde
sage: C, g = solve_plde(GG, map(GG, (1,0,-1)), \
          [h-GG.sigma()(h)]); C, g
(
[1]  [-1/2*hh - 1/2*h]
[0], [              1]
)
```

Notation and symbols

$\mathbb{N} = \{0, 1, 2, \ldots\}$	-	the set of natural numbers
$\mathbb{Z}, \mathbb{Q}, \mathbb{C}$	-	the sets of integers, rational and complex numbers
H_n	-	harmonic number, $H_n = \sum_{j=1}^{n} \frac{1}{j}$
R^*	-	the group of units for a ring R
$\langle p \rangle$	-	the ideal generated by an element p from a ring R
$(a_0, \ldots, a_n)^T$	-	the column vector with components a_i
$Q(R[t])$	-	the total quotient ring of the ring R extended by t
F, σ	-	a difference field
$\mathrm{const}_\sigma(F)$	-	the constant field of a difference field F, σ
(R, σ, δ)	-	a D-ring
$\mathrm{const}_{\sigma,\delta}(R)$	-	the constant subring of the D-ring (R, σ, δ)
$End_{R,\sigma,\delta}(M)$	-	the set of all R-pseudo-linear maps of M, where (R, σ, δ) is a D-ring and M a left R-module
$R[X; \sigma, \delta]$	-	the left skew polynomial ring over the D-ring (R, σ, δ)
$u_{\mathbf{b}}$	-	the column vector of the coordinates of u in the basis $\mathbf{b}$
$M_{\mathbf{b}}(f)$	-	the matrix of the multiplication map by f in the basis $\mathbf{b}$
$V_{a,b}(R)$	-	solutions of the equation $\sigma(w) = aw + b$ in R, where $a, b \in R$ and R, σ a difference field
$\sigma I, \delta I$	-	the application of δ and σ to each component of a vector
$[\,A \,\vert\, B\,]$	-	the matrix A augmented by the matrix B
$\left[\begin{array}{c} A \\ \hline B \end{array}\right]$	-	the matrix A stacked on the matrix B
$A^{\mathtt{adj}}$	-	the adjoint of the matrix A

Bibliography

[1] Jakob Ablinger. A computer algebra toolbox for harmonic sums related to particle physics. Master's thesis, RISC, Johannes Kepler University, February 2009.

[2] Jakob Ablinger, Johannes Bluemlein, Sebastian Klein, Carsten Schneider, and F. Wissbrock. The $O(\alpha_s^3)$ massive operator matrix elements of $O(n_f)$ for the structure function $F_2(x, Q^2)$ and transversity. *Nucl. Phys. B*, 844:26–54, 2011.

[3] Sergei A. Abramov. Rational solutions of linear differential and difference equations with polynomial coefficients. *U.S.S.R. Comput. Math. Math. Phys.*, 29(6):7–12, 1989.

[4] Sergei A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. In T. Levelt, editor, *Proc. ISSAC 1995*, pages 285–289. ACM Press, 1995.

[5] Sergei A. Abramov and Moulay A. Barkatou. Rational solutions of first order linear difference systems. In *Proc. ISSAC 1998*, pages 124–131, New York, 1998. ACM.

[6] Sergei A. Abramov and Manuel Bronstein. Hypergeometric dispersion and the orbit problem. In C. Traverso, editor, *Proc. ISSAC 2000*. ACM Press, 2000.

[7] Sergei A. Abramov, Manuel Bronstein, and Marko Petkovšek. On polynomial solutions of linear operator equations. In *Proc. ISSAC 1995*. ACM Press, 1995.

[8] Sergei A. Abramov, Peter Paule, and Marko Petkovšek. q-Hypergeometric solutions of q-difference equations. *Discrete Math.*, 180(1-3):3–22, 1998.

[9] Sergei A. Abramov and Marko Petkovšek. D'Alembertian solutions of linear differential and difference equations. In *Proc. ISSAC 1994*. ACM Press, 1994.

[10] Gert Almkvist and Doron Zeilberger. The method of differentiating under the integral sign. *J. Symb. Comput.*, 10:571–591, 1990.

[11] Michael F. Atiyah and Ian G. Macdonald. *Introduction to Commutative Algebra.* Addison-Wesley Publishing Co., Reading, 1969.

[12] Erwin H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Math. Comp.*, 22:565–578, 1968.

[13] Moulay A. Barkatou. Rational solutions of matrix difference equations: the problem of equivalence and factorization. In *Proc. ISSAC 1999*, pages 277–282, New York, 1999. ACM.

[14] Andrej Bauer and Marko Petkovšek. Multibasic and mixed hypergeometric Gosper-type algorithms. *J. Symb. Comput.*, 28(4–5):711–736, 1999.

[15] Johannes Bluemlein, Sebastian Klein, Carsten Schneider, and Flavia Stan. A symbolic summation approach to Feynman integral calculus. Technical Report DESY 10-185, Deutsches Elektronen-Synchrotron, 2010.

[16] Manuel Bronstein. The Risch differential equation on an algebraic curve. In *Proc. ISSAC 1991*, pages 241–246. ACM Press, New York, USA, 1991.

[17] Manuel Bronstein. On solutions of linear ordinary difference equations in their coefficient field. *J. Symb. Comput.*, 29(6):841 – 877, 2000.

[18] Manuel Bronstein. *Symbolic Integration. I*, volume 1 of *Algorithms and Computation in Mathematics*. Springer, second edition, 2005.

[19] Manuel Bronstein and Marko Petkovšek. An introduction to pseudo-linear algebra. *Theor. Comput. Sci.*, 157(1):3–33, 1996.

[20] Stanley Cabay. Exact solution of linear equations. In *Proc. SYMSAC '71*, pages 392–398. ACM, 1971.

[21] Zhuliang Chen and Arne Storjohann. A BLAS based c library for exact linear algebra on integer matrices. In *Proc. ISSAC 2005*, pages 92–99. ACM, 2005.

[22] Frédéric Chyzak. Gröbner bases, symbolic summation and symbolic integration. In *Gröbner Bases and Applications*, volume 251, page 32–60. Cambridge University Press, 1998. Lecture Notes Series of the LMS.

[23] Richard M. Cohn. *Difference algebra*. Interscience Publishers John Wiley & Sons, New York-London-Sydeny, 1965.

[24] Edsger W. Dijkstra. Why numbering should start at zero. http://www.cs.utexas.edu/users/EWD/ewd08xx/EWD831.PDF, August 1982.

[25] John D. Dixon. Exact solution of linear equations using p-adic expansions. *Numerische Mathematik*, 40:137–141, 1982. 10.1007/BF01459082.

[26] Jean Guillaume Dumas, Thierry Gautier, and Clément Pernet. Finite field linear algebra subroutines. In *Proc. ISSAC 2002*, pages 63–74. ACM Press, New York, USA, 2002.

[27] David Eisenbud. *Commutative Algebra : with a View Toward Algebraic Geometry (Graduate Texts in Mathematics)*. Springer, February 1999.

[28] Burçin Eröcal and William Stein. The Sage Project: Unifying free mathematical software to create a viable alternative to Magma, Maple, Mathematica and MATLAB. In *Mathematical Software – ICMS 2010*, volume 6327 of *LNCS*, pages 12–27. Springer, 2010.

[29] Mary Celine Fasenmyer. *Some generalized hypergeometric polynomials.* PhD thesis, University of Michigan, November 1945.

[30] Johannes Oswald Gärtner. Summation in finite terms - presentation and implementation of M. Karr's algorithm. Master's thesis, RISC, Johannes Kepler University, Linz, May 1986.

[31] Jürgen Gerhard, Mark Giesbrecht, Arne Storjohann, and Evgueni V. Zima. Shiftless decomposition and polynomial-time rational summation. In *Proc. ISSAC 2003*, pages 119–126. ACM, 2003.

[32] Stefan Gerhold. On some non-holonomic sequences. *Electron. J. Combin.*, 11(1):Research Paper 87, 8 pp. (electronic), 2004.

[33] R. William Gosper, Jr. Decision procedure for indefinite hypergeometric summation. *Proc. Nat. Acad. Sci. U.S.A.*, 75(1):40–42, 1978.

[34] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley, Boston, MA, USA, 2nd edition, 1994.

[35] Torbjörn Granlund. *GMP: The GNU Multiple Precision arithmetic library*, 2004. Edition 4.1.4. http://www.swox.com/gmp.

[36] Charlotte Hardouin and Michael Singer. Differential Galois theory of linear difference equations. *Mathematische Annalen*, 342:333–377, 2008. 10.1007/s00208-008-0238-z.

[37] Bill Hart and David Harvey. *FLINT: Fast Library for Number Theory.* http://www.flintlib.org/.

[38] David Harvey. Faster polynomial multiplication via multipoint kronecker substitution. *J. Symb. Comput.*, 44(10):1502 – 1510, 2009.

[39] Peter A. Hendriks. *Algebraic Aspects of Linear Differential and Difference Equations.* PhD thesis, University of Groningen, November 1996.

[40] Peter A. Hendriks and Michael F. Singer. Solving difference equations in finite terms. *J. Symb. Comput.*, 27:239–259, March 1999.

[41] Michael Karr. Summation in finite terms. *J. ACM*, 28(2):305–350, 1981.

[42] Michael Karr. Theory of summation in finite terms. *J. Symb. Comput.*, 1(3):303 – 315, 1985.

[43] Toni Kasper. Integration in finite terms: the Liouville theory. *SIGSAM Bull.*, 14:2–8, November 1980.

[44] Manuel Kauers and Carsten Schneider. Application of unspecified sequences in symbolic summation. In *ISSAC 2006*, pages 177–183. ACM, New York, 2006.

[45] Manuel Kauers and Carsten Schneider. Indefinite summation with unspecified summands. *Discrete Math.*, 306(17):2073–2083, 2006.

[46] Manuel Kauers and Carsten Schneider. Symbolic summation with radical expressions. In *ISSAC 2007*, pages 219–226. ACM, New York, 2007.

[47] Christoph Koutschan. *Advanced Applications of the Holonomic Systems Approach*. PhD thesis, RISC, Johannes Kepler University, September 2009.

[48] Hong R. Lee and B. David Saunders. Fraction free Gaussian elimination for sparse matrices. *J. Symb. Comput.*, 19(5):393 – 402, 1995.

[49] Christian Mallinger. Algorithmic manipulations and transformations of univariate holonomic functions and sequences. Master's thesis, RISC, Johannes Kepler University, August 1996.

[50] Yiu-Kwong Man and Francis J. Wright. Fast polynomial dispersion computation and its application to indefinite summation. In *Proc. ISSAC 1994*, pages 175–180. ACM, New York, NY, USA, 1994.

[51] Sven-Olaf Moch and Carsten Schneider. Feynman integrals and difference equations. In , editor, *Proc. ACAT 2007*, volume PoS(ACAT)083, pages 1–11, 2007.

[52] Oystein Ore. Theory of non-commutative polynomials. *Ann. of Math. (2)*, 34(3):480–508, 1933.

[53] Peter Paule. Greatest factorial factorization and symbolic summation. *J. Symb. Comput.*, 20(3):235–268, 1995.

[54] Peter Paule and Carsten Schneider. Computer proofs of a new family of harmonic number identities. *Adv. in Appl. Math.*, 31(2):359–378, 2003.

[55] Marko Petkovšek, Herbert S. Wilf, and Doron Zeilberger. $A = B$. A K Peters, 1996.

[56] Marko Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *J. Symb. Comput.*, 14:243–264, 1992.

[57] Axel Riese. qMultiSum - A Package for Proving q-Hypergeometric Multiple Summation Identities. *J. Symb. Comput.*, 35:349–376, 2003.

[58] Robert H. Risch. The problem of integration in finite terms. *Trans. Amer. Math. Soc.*, 139:167–189, 1969.

[59] Robert H. Risch. The solution of the problem of integration in finite terms. *Bull. Amer. Math. Soc.*, 76:605–608, 1970.

[60] Joseph F. Ritt. *Integration in Finite Terms: Liouville's Theory of Elementary Models.* Columbia Univ. Press, New York, 1948.

[61] Maxwell Rosenlicht. Liouville's theorem on functions with elementary integrals. *Pacific J. Math.*, 24(1):153–161, 1968.

[62] Carsten Schneider. *Symbolic Summation in Difference Fields.* PhD thesis, RISC, Johannes Kepler University, Linz, May 2001.

[63] Carsten Schneider. A collection of denominator bounds to solve parameterized linear difference equations in ΠΣ-extensions. *An. Univ. Timisoara Ser. Mat.-Inform.*, 42(2):163–179, 2004.

[64] Carsten Schneider. The summation package Sigma: Underlying principles and a rhombus tiling application. *Discrete Math. Theor. Comput. Sci.*, 6(2):365–386, 2004.

[65] Carsten Schneider. Degree bounds to find polynomial solutions of parameterized linear difference equations in ΠΣ-fields. *Appl. Algebra Engrg. Comm. Comput.*, 16(1):1–32, 2005.

[66] Carsten Schneider. A new Sigma approach to multi-summation. *Adv. in Appl. Math.*, 34(4):740–767, 2005.

[67] Carsten Schneider. Product representations in ΠΣ-fields. *Ann. Comb.*, 9(1):75–99, 2005.

[68] Carsten Schneider. Solving parameterized linear difference equations in terms of indefinite nested sums and products. *J. Difference Equ. Appl.*, 11(9):799–821, 2005.

[69] Carsten Schneider. Symbolic summation assists combinatorics. *Sém. Lothar. Combin.*, 56:1–36, 2007.

[70] Carsten Schneider. A refined difference field theory for symbolic summation. *J. Symb. Comput.*, 43(9):611–644, 2008.

[71] Carsten Schneider. private communication, 2009.

[72] Carsten Schneider. A symbolic summation approach to find optimal nested sum representations. In *Motives, Quantum Field Theory, and Pseudodifferential Operators*, volume 12 of *Clay Math. Proc.*, pages 285–308. AMS, 2010.

[73] Michael F. Singer. Liouvillian solutions of linear differential equations with liouvillian coefficients. *J. Symb. Comput.*, 11(3):251–273, 1991.

[74] Flavia Stan. On recurrences for Ising integrals. *Adv. in Appl. Math.*, 45(3):334–345, 2010.

[75] Willam A. Stein et al. *Sage Mathematics Software.* The Sage Development Team. http://www.sagemath.org.

[76] Arne Storjohann. On the complexity of inverting integer and polynomial matrices. *Computational Complexity*, 2008. To appear.

[77] Nobuki Takayama. An approach to the zero recognition problem by Buchberger algorithm. *J. Symb. Comput.*, 14(2-3):256–282, 1992.

[78] Marius van der Put and Michael F. Singer. *Galois theory of difference equations*, volume 1666 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, 1997.

[79] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, New York, NY, USA, 1999.

[80] Kurt Wegschaider. Computer generated proofs of binomial multi-sum identities. Master's thesis, RISC, Johannes Kepler University, Linz, May 1997.

[81] Herbert S. Wilf and Doron Zeilberger. An algorithmic proof theory for hypergeometric (ordinary and "q") multisum/integral identities. *Invent. Math.*, 108(3):575–633, 1992.

[82] Oscar Zariski and Pierre Samuel. *Commutative Algebra, Volume I.* D. Van Nostrand Company, Inc., Princeton, New Jersey, 1958.

[83] Doron Zeilberger. A fast algorithm for proving terminating hypergeometric identities. *Discrete Mathematics*, 80(2):207–211, 1990.

Acknowledgements

I am grateful to many people for making the past four years I spent at RISC so enjoyable. The following list is most definitely incomplete. If I've missed your name, I hope you'll excuse me. Even extended deadlines are sometimes hard to keep and there is never enough time for the important things.

I would like to thank . . .

Peter Paule and Carsten Schneider for giving me this problem, with the chance and time to start in a new field;

Carsten Schneider for challenging me with alternative views and hiring me in his physics project;

Peter Paule for advice in all areas, mathematical and social, as well as for all the positive energy he generously shares;

Bruno Buchberger and Franz Winkler for the opportunity to join RISC;

Arne Storjohann for patiently answering countless questions I sent his way, showing me how computer algebra tricks fit together and being a great host;

Marko Petkovšek for his time and understanding;

Michael Singer and Felix Ulmer for reminding me at crucial moments that mathematics is wonderful and showing me a path in the fog;

Ralf Hemmecke for our long chats on life the universe and everything;

William Stein for leading by example, all his time and work on Sage, and for making it fun to work together;

the Sage team for their dedicated effort to provide an open-source alternative; especially Martin Albrecht, Michael Abshoff, Robert Bradshaw, Clément Pernet, Mike Hansen, Michael Brickenstein, Jason Grout and Karl-Dieter Crisman;

Gert-Martin Greuel and Wolfram Decker for welcoming me in the Singular team;

Alexander Dreyer and Michael Brickenstein for making me feel right at home in Kaiserlautern;

Zaf, Ionela, Veronika, Jakob, Hamid, Cevo, Mădălina and Clemens for making Linz a better place to live, in particular Zaf for his cooking skills and Ionela for the cake;

Ali Nesin, İlhan İkeda and İsmail Güloğlu for introducing me to research;

Flavia for sharing life and mathematics and my parents for their support, patience and sacrifice.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, daß ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, December 2010 Burçin Eröcal

Curriculum Vitae

Personal data

Name	Burçin Eröcal
e-mail	burcin@erocal.org

Affiliation

Research Institute for Symbolic Computation (RISC)
Johannes Kepler Universität Linz
Altenbergerstraße 69
4040 Linz, AUSTRIA

Education

06/1999	High school degree (dir. natural sciences) Özel Amerikan Robert Lisesi, Istanbul, Turkey
09/1999–07/2003	Undergraduate Studies Mathematics and Computer Science Department Istanbul Bilgi University, Istanbul, Turkey. *Diploma thesis:* “AKS algorithm and primality tests” *Thesis advisor:* Assoc.Prof.Dr. İlhan İkeda
09/2003–07/2006	Master Studies in Cryptography Institute of Applied Mathematics Middle East Technical University, Ankara, Turkey. *M.Sc. thesis:* “Some sequence synthesis algorithms” *Thesis advisor:* Prof.Dr. Ferruh Özbudak
11/2006–01/2011	Doctoral studies Research Institute for Symbolic Computation Johannes Kepler University, Linz, Austria.