Universitat Autònoma de Barcelona

Facultat de Ciències

Departament de Matemàtiques

Triangular bases of integral closures

Thesis submitted by **Hayden Duncan Stainsby** for the degree of Philosophiae Doctor by the Universitat Autònoma de Barcelona, under the supervision of Prof. Jesús Montes and Prof. Enric Nart

Barcelona, October 2014

Triangular bases of integral closures

Thesis submitted by **Hayden Duncan Stainsby** for the degree of Philosophiae Doctor by the Universitat Autònoma de Barcelona, under the supervision of Prof. Jesús Montes and Prof. Enric Nart, in the Department of Mathematics.

Barcelona, October 2014

Author

Hayden Duncan Stainsby

Supervisors

Prof. Jesús Montes

Prof. Enric Nart

Abstract

In this work, we consider the problem of computing triangular bases of integral closures of one-dimensional local rings.

"MaxMin" is presented, an efficient algorithm which employs OM representations of prime ideals to compute local bases of fractional ideals of number fields and function fields. The proposed algorithm generates bases which are guaranteed to be *reduced* and triangular. In this way, it avoids the application of triangularisation routines, such as the Hermite Normal Form, which are slow for fields of large degree.

We show that this algorithm has the same asymptotic computational complexity as existing methods based on OM representations.

MaxMin has been developed and included as part of the +Ideals package for arithmetic in large fields. This implementation is almost always faster than existing OM-based routines. It is also considerably faster than the routines currently found in standard computer algebra systems, excepting some cases involving very small field extensions.

Resumen

En este trabajo, consideramos el problema de computar bases triangulares de clausuras enteras de anillos locales unidimensionales.

Se presenta "MaxMin", un algoritmo eficiente que emplea representaciones OM de ideales primos para computar bases locales de ideales fraccionarios de cuerpos de números y cuerpos de funciones. MaxMin garantiza que las bases generadas son *reducidas* y triangulares. De este modo, se evita la aplicación de rutinas de triangularización, como el paso a la forma normal de Hermite, que son lentas para cuerpos de grado alto.

Mostramos que este algoritmo tiene la misma complejidad computacional asintótica que los métodos ya existentes basados en representaciones OM.

MaxMin ha sido desarrollado e incluido en +Ideals, un paquete diseñado para trabajar cuestiones aritméticas en cuerpos grandes. La implementación casi siempre es más rápida que las otras rutinas basadas en representaciones OM. Respecto a las rutinas que se encuentran actualmente en los sistemas de álgebra computacional estándard, nuestra implementación de MaxMin es de nuevo considerablemente más rápida, exceptuando casos concretos de extensiones de cuerpos muy pequeñas.

Resum

En aquest treball, considerem el problema de computar bases triangulars de clausures enteres d'anells locals unidimensionals.

Es presenta "MaxMin", un algoritme eficient que empra representacions OM d'ideals primers per computar bases locals d'ideals fraccionaris de cossos de nombres i cossos de funcions. MaxMin garanteix que les bases generades són *reduïdes* i triangulars. D'aquesta manera, s'evita l'aplicació de rutines de triangularització, com ara el pas a la forma normal d'Hermite, que són lentes per a cossos de grau alt.

Mostrem que aquest algoritme té la mateixa complexitat computacional asimptòtica que els mètodes ja existents basats en representacions OM.

MaxMin ha estat desenvolupat i inclòs en el paquet +Ideals, dissenyat per treballar qüestions aritmètiques en cossos grans. La implementació quasi sempre és més ràpida que la de les altres rutines basades en representacions OM. Respecte a les rutines que es troben actualment als sistemes d'àlgebra computacional estàndard, la nostra implementació de MaxMin és també considerablement més ràpida, exceptuant casos concrets d'extensions de cossos molt petites.

Acknowledgements

First, I must thank my advisors Jesús Montes and Enric Nart. I would like to thank Jesús for the original idea behind my thesis and his encouragement. It is clear that I wouldn't be here without the support and assistance of Enric. In accepting me to be his student, I think he was more conscious of the way ahead us than I was. Thanks to his advice, support, and mentorship, today I can venture to call myself a mathematician.

I would like thank Jürgen Klüners for hosting me in Paderborn and providing invaluable advice and a fantastic experience while I was there. The company and friendship of Inga, Thorsten, and Friedrich made my visit so much more enjoyable, I thank you all!

To my friends and workmates at the UAB, the last four years have been an incredible experience, and it's been wonderful to share it with all of you, especially Jens Bauch, my fellow "student of the Montes algorithm".

I want to thank everyone in the Barcelona number theory group. Especially to my fellow students, Carlos, Elisa, Iago, Jens, Nuno, and Piermarco, and to my fellow seminar organisers Montse and Piermarco.

A la familia Martínez-Trujillo-Zaguirre-Heras, mi familia aquí en el hemisferio del norte. Me habéis aceptado como uno más de la familia con vuestro amor y apoyo. Quiero agradecer a Mari Carmen y Manolo, quienes especialmente han hecho de Granada mi segunda casa.

To Thea and Bruce, you are my parents, my mentors, and my friends. Thank you for the unconditional support you've given me, growing up and then in the nearly ten years since I packed up and left to live on the other side of the world. To my brother Evan, and to Steph, it's always fantastic to see you guys and catch up on our lives apart. Let's see if we can spend a few years living on the same continent at some point!

I have, of course, left the most important person to last. To Andrea who, through the entirety of our respective theses, has been my constant companion, partner, friend, and more. There are too few words in any language to describe what you mean to me. I would not have made it here without your love! Thank you for everything!

To Dijkstra, Lamport, Schneier, and Wallace.

Although we've never met,
it's all your fault that I started this.

Contents

List of Figures

List of Algorithms

Introduction

"It's a rare gift, to know where you need to be, before you've been to all the places you don't need to be."

– Ursula K. Le Guin, *Tales from Earthsea*

This work deals with the computation of triangular bases of integral closures of one-dimensional local rings.

The theory of ideals of the ring of integers of a number field dates back to R. Dedekind and E.E. Kummer in the mid 19^{th} century. The theory of the existence and the representation of these ideals was the objective of work by K. Hensel, Ø. Ore, and S. MacLane which extended into the first half of the 20^{th} century.

In his 1999 Ph.D. thesis, J. Montes extended the ideas of Ore and MacLane and implemented an algorithm that Ore had envisioned, to com-

pute a representation of prime ideals by way of factoring the defining polynomial over the ring of p-adic numbers. This "Montes algorithm" coupled with work by K. Okutsu on constructing explicit integral bases of local fields, gave rise to OM[1] representations of prime ideals.

All these results extend in a well-known way to the computation of bases of integral closures in function fields.

Traditionally, there are two methods of representing fractional ideals of a number field or a function field in a computer system; either as a basis as a free module over a certain base ring or as a pair of generators [PZ89]. While the basis representation needs more space than the generators, it has the advantage of requiring less complex arithmetic.

In current computer algebra systems, most of the methods used to compute integral bases are variants of the Round-2 and Round-4 routines by Zassenhaus and Ford [Coh93][FPR02][Hal01][PZ89][Poh93][vH94].

Two OM-based routines have also been developed. The first of these routines [GMN13] is based on an existing technique which produces a local basis as the union of bases of prime ideals multiplied by a certain "multiplier" in each case. The advantage of starting from OM representations of prime ideals is that all the requisite polynomials are fabricated from data present in these OM representations.

The second is the "method of the quotients" [GMN], which constructs a basis using the quotients of certain divisions with remainder performed as part of the Montes algorithm, which produces OM representations.

Contribution

This memoir presents the MaxMin algorithm, a method of computing triangular local bases of fractional ideals of number fields and function fields directly from the OM representations of prime ideals.

Triangular bases have an advantage in that, given appropriate local triangular bases, we may construct a global basis using the Chinese Remainder Theorem, without having to first convert the local bases into a normal form,

[1] OM stands indistinctly for *Ore-MacLane* or *Okutsu-Montes*.

a process which is often computationally expensive. Triangular bases also often simplify arithmetic on the ideals they represent.

The proposed algorithm presents the same computational complexity as the previous OM-based routines for computing bases, and in practice is almost always faster. It is also considerably faster than the Round-2 and Round-4 based routines present in current software packages for all but the smallest fields.

As a product of the work presented in this document, the +Ideals package [GMN10a] has been extended to support this new method of computing bases of fractional ideals. The latest version of the package can be downloaded from `https://github.com/MontesProject/plus-ideals`.

Structure of this memoir

In Chapter 1, we present the necessary algebraic background as well as a discussion of two normal forms of bases of fractional ideals, focussing on the local case. We conclude the chapter with a description of the standard technique used to construct a global basis from the necessary local bases.

Chapter 2 describes the basic tools used in this work, the OM representations of prime ideals. We discuss the "types" which represent prime ideals, as well as an efficient computational method for computing these objects, the Montes algorithm.

Our own results begin in Chapter 3 with a discussion of optimal polynomials, the primary ingredient in our method for computing local triangular bases. Here, we concern ourselves with reducing the space in which we must search to find such optimal polynomials.

In Chapter 4, we present the main results of this thesis. MaxMin is an efficient, and extremely simple algorithm for constructing local triangular bases of the integral closure of a discrete valuation ring in a finite extension of its field of fractions, directly from OM representations of the prime ideals of the integral closure. The main theorem of this chapter shows that the MaxMin algorithm performs this task.

Chapter 5 describes an adapted version of the MaxMin algorithm, which can compute local triangular bases of fractional ideals using the same input

polynomials as in the "maximal order version". The bases produced by the MaxMin algorithm are always *reduced*. This has some advantages in certain applications, such as the computation of bases of the Riemann-Roch spaces attached to divisors of curves.

A detailed complexity analysis of the MaxMin algorithm is presented in Chapter 6. The computational complexity is given for the entire process required to compute a local triangular basis. An analysis of the space complexity of the MaxMin algorithm is also presented.

In Chapter 7, the performance of an implementation of the MaxMin algorithm is compared to two other OM-based routines as well as the routine used internally by the Magma Computational Algebra System. We present results in number fields as well as function fields over finite fields.

1

Algebraic background

"I'd take the awe of understanding over the awe of ignorance any day."

– Douglas Adams, *The Salmon of Doubt*

1.1 Localisation and completion

Let A be a commutative ring with unity and let $\mathrm{Max}(A) \subseteq \mathrm{Spec}(A)$ denote the maximal spectrum and the spectrum of A, respectively; that is, $\mathrm{Max}(A)$ is the set of maximal ideals of A, and $\mathrm{Spec}(A)$ the set of prime ideals.

For every $\mathbf{p} \in \mathrm{Spec}(A)$ we consider the local ring $A_{\mathbf{p}} := A[(A\backslash\mathbf{p})^{-1}]$, obtained from A by localisation at $\mathbf{p}$. Also, if M is an A-module, we may consider the $A_{\mathbf{p}}$-module $M_{\mathbf{p}} := M[(A\backslash\mathbf{p})^{-1}]$, obtained by localisation at $\mathbf{p}$.

The elements of $M_{\mathbf{p}}$ are formal quotients x/a, where $x \in M$, $a \in A\backslash\mathbf{p}$, and they satisfy:

$$\frac{x}{a} = \frac{x'}{a'} \quad \Longleftrightarrow \quad \exists\, b \in A\backslash\mathbf{p} \text{ such that } b(a'x - ax') = 0.$$

The localisation comes equipped with a natural map of A-modules

$$\begin{aligned} M &\longrightarrow M_{\mathbf{p}}, \\ x &\mapsto x/1. \end{aligned}$$

By the above identity, $x/1$ vanishes in $M_{\mathbf{p}}$ if and only if x is annihilated in M by some element in $A\backslash\mathbf{p}$.

The assignment $M \mapsto M_{\mathbf{p}}$ determines an exact functor from the category of A-modules to the category of $A_{\mathbf{p}}$-modules. Since the module $M_{\mathbf{p}}$ may be identified to $M \otimes_A A_{\mathbf{p}}$, the exactness of the functor shows that $A_{\mathbf{p}}$ is a flat A-algebra.

The following result shows that certain properties may be deduced locally.

Lemma 1.1. *Let M be an A-module, $N \subset M$ an A-submodule and $x \in M$.*

- *$N = M$ if and only if $N_{\mathfrak{m}} = M_{\mathfrak{m}}$, for all $\mathfrak{m} \in \operatorname{Max}(A)$.*
- *$x \in N$ if and only if $x/1 \in N_{\mathfrak{m}}$, for all $\mathfrak{m} \in \operatorname{Max}(A)$.*

Suppose A is a noetherian ring and let $\mathfrak{a}$ be an ideal of A. The *$\mathfrak{a}$-adic topology* on A is determined by taking the subsets $\{a + \mathfrak{a}^n\}_{n\geqslant 0}$ as a fundamental system of neighbourhoods of any $a \in A$. With this topology, A becomes a topological ring; that is, the operations of addition and multiplication are represented by continuous maps. Since A is noetherian, we have $\bigcap_{n\geqslant 0} \mathfrak{a}^n = \{0\}$, and A is a Hausdorff topological space.

Any A-module M inherits a similar topology by taking $\{x + \mathfrak{a}^n M\}_{n\geqslant 0}$ as a fundamental system of neighbourhoods of any $x \in M$.

The ring A, or the module M, are said to be *complete* with respect to the $\mathfrak{a}$-adic topology if any Cauchy sequence is convergent. It is possible to

construct the *completion* of A, or M, as the inverse limit:

$$\hat{A} := \varprojlim A/\mathfrak{a}^n,$$
$$\hat{M} := \varprojlim M/\mathfrak{a}^n M.$$

The ring $\hat{A}$ is a complete topological ring, and we have a canonical continuous ring monomorphism, $A \hookrightarrow \hat{A}$. Any continuous ring homomorphism, $A \to B$, from A to a complete topological ring B, extends in a unique way to a continuous ring homomorphism $\hat{A} \to B$.

The assignment $M \mapsto \hat{M}$ determines an exact functor from the category of finitely generated A-modules to the category of finitely generated $\hat{A}$-modules. Moreover, if M is finitely generated, the canonical map $M \otimes_A \hat{A} \longrightarrow \hat{M}$ is an isomorphism of $\hat{A}$-modules. Thus, $\hat{A}$ is also a flat A-algebra.

1.2 Finite extensions of Dedekind domains

Let A be a Dedekind domain; that is, A is a noetherian, integrally closed domain of dimension 1. Every nonzero prime ideal of A is maximal; in other words, $\mathrm{Spec}(A) = \mathrm{Max}(A) \cup \{0A\}$.

Every nonzero ideal of A decomposes in a unique way as a product of nonzero prime ideals. This is the essential property of Dedekind domains.

Let K be the field of fractions of A. A *fractional ideal* of A is a finitely generated A-submodule $I \subset K$. The set $\mathcal{I}_A$ of nonzero fractional ideals has the structure of a commutative group with respect to the operation of multiplication of fractional ideals. By the unique factorisation property, $\mathcal{I}_A$ is a free abelian group over the set of nonzero prime ideals:

$$\mathcal{I}_A = \bigoplus_{\mathfrak{m} \in \mathrm{Max}(A)} \mathfrak{m}^{\mathbb{Z}}.$$

The *class group* of A is the quotient group $\mathrm{Cl}(A) := \mathcal{I}_A / \mathrm{Pr}_A$, where

$$\mathrm{Pr}_A := \{xA : x \in K^*\} \subset \mathcal{I}_A,$$

is the subgroup of principal nonzero fractional ideals.

Let $\mathfrak{m}$ be a non-zero prime ideal of A and consider the map

$$v_{\mathfrak{m}} : \mathcal{I}_A \longrightarrow \mathbb{Z},$$

determined by:

$$I = \prod_{\mathfrak{m} \in \mathrm{Max}(A)} \mathfrak{m}^{v_{\mathfrak{m}}(I)}.$$

For any $I, J \in \mathcal{I}_A$, we say that $I \mid J$ if any of the following equivalent conditions are satisfied :

1. $J \subset I$,
2. there exists an ideal $\mathfrak{a} \subset A$ such that $J = \mathfrak{a}I$,
3. $v_{\mathfrak{m}}(I) \leqslant v_{\mathfrak{m}}(J)$ for all $\mathfrak{m} \in \mathrm{Max}(A)$.

The induced map

$$\begin{aligned} v_{\mathfrak{m}} : K^* &\longrightarrow \mathbb{Z}, \\ x &\mapsto v_{\mathfrak{m}}(x) := v_{\mathfrak{m}}(xA), \end{aligned}$$

with the extended value $v_{\mathfrak{m}}(0) := \infty$, is a discrete valuation of K. The local ring $A_{\mathfrak{m}}$ may be identified with the valuation ring of $v_{\mathfrak{m}}$:

$$A_{\mathfrak{m}} = \{x \in K : v_{\mathfrak{m}}(x) \geqslant 0\} \subset K.$$

In particular, $A_{\mathfrak{m}}$ is a principal domain.

Let $f \in A[x]$ be a monic and irreducible polynomial of degree $n > 1$. Let $\theta \in \overline{K}$ be a root of f and $L = K(\theta)$ the finite extension of K generated by θ. The integral closure B of A in L is a Dedekind domain. We assume throughout that the following hypothesis is satisfied.

Hypothesis. *B is finitely generated as an A-module.*

This condition holds under very natural assumptions; for instance, if L/K is separable, or A is complete with respect to some discrete valuation, or A is a finitely generated algebra over a field [Ser68, I, §4].

Consider the factorisation of $\mathfrak{m}B$ into a product of prime ideals in L:

$$\mathfrak{m}B = \mathfrak{p}_1^{e(\mathfrak{p}_1/\mathfrak{m})} \cdots \mathfrak{p}_g^{e(\mathfrak{p}_g/\mathfrak{m})}.$$

Let $K_\mathfrak{m}$, $L_\mathfrak{p}$, be the completions of K and L with respect to the $\mathfrak{m}$-adic and $\mathfrak{p}$-adic topology, respectively. Denote the ring of integers of these fields by:

$$\begin{aligned} &\hat{A}_\mathfrak{m} \subset K_\mathfrak{m}, \\ &\hat{B}_\mathfrak{p} \subset L_\mathfrak{p}, \qquad\qquad\qquad \forall\ \mathfrak{p} \mid \mathfrak{m}. \end{aligned}$$

Finally, we denote by $n_\mathfrak{p} := [L_\mathfrak{p} : K_\mathfrak{m}] = e(\mathfrak{p}/\mathfrak{m}) f(\mathfrak{p}/\mathfrak{m})$ the local degrees, where $f(\mathfrak{p}/\mathfrak{m}) := [B/\mathfrak{p} : A/\mathfrak{m}]$ are the residual degrees.

The natural homomorphisms $B \longrightarrow \hat{B}_\mathfrak{p}$ induce a canonical isomorphism of $\hat{A}_\mathfrak{m}$-algebras [Ser68, II, Prop. 4]:

$$B \otimes_A \hat{A}_\mathfrak{m} \xrightarrow{\sim} \bigoplus\nolimits_{\mathfrak{p} \mid \mathfrak{m}} \hat{B}_\mathfrak{p}. \tag{1.1}$$

By a classical theorem of Hensel [Hen08], the prime ideals $\mathfrak{p} \mid \mathfrak{m}$ are in 1-1 correspondence with the different monic irreducible factors of $f(x)$ in $\hat{A}_\mathfrak{m}[x]$.

Definition 1.2. *For each prime ideal $\mathfrak{p} \mid \mathfrak{m}$, let us fix a continuous embedding, $i_\mathfrak{p} : L \subset L_\mathfrak{p} \hookrightarrow \overline{K}_\mathfrak{m}$, with respect to the $\mathfrak{p}$-adic topology. Then $\theta_\mathfrak{p} := i_\mathfrak{p}(\theta)$ is the root of a monic irreducible factor (say) $F_\mathfrak{p}(x)$ of $f(x)$ in $\hat{A}_\mathfrak{m}[x]$. Also, we denote:*

$$w_\mathfrak{p} := e(\mathfrak{p}/\mathfrak{m})^{-1} v_\mathfrak{p} : L^* \longrightarrow e(\mathfrak{p}/\mathfrak{m})^{-1}\mathbb{Z},$$

where $v_\mathfrak{p}$ is the discrete valuation of L attached to $\mathfrak{p}$.

Clearly, $w_\mathfrak{p}(\alpha) = v(i_\mathfrak{p}(\alpha))$ for all $\alpha \in L$, where $v := v_\mathfrak{m}$ is the canonical extension of $v_\mathfrak{m}$ to $\overline{K}_\mathfrak{m}$. Thus, for any polynomial $g(x) \in A[x]$,

$$w_\mathfrak{p}(g(\theta)) = v(g(\theta_\mathfrak{p})).$$

This identity will be implicitly used throughout the memoir without further mention, when we apply local results to a global situation.

The semilocal ring $B_{\mathfrak{m}} = B[(A\backslash\mathfrak{m})^{-1}]$ may be identified to the integral closure of $A_{\mathfrak{m}}$ in L; that is, to the subring of $\mathfrak{m}$-integral elements of L:

$$B_{\mathfrak{m}} = \{\alpha \in L : v_{\mathfrak{p}}(\alpha) \geqslant 0,\ \forall\ \mathfrak{p} \in \operatorname{Spec}(B),\ \mathfrak{p} \mid \mathfrak{m}\} \subset L.$$

The ring $B_{\mathfrak{m}}$ is a torsion-free finitely generated $A_{\mathfrak{m}}$-module. Since $A_{\mathfrak{m}}$ is a PID, this implies that $B_{\mathfrak{m}}$ is a free $A_{\mathfrak{m}}$-module. Also, since $B_{\mathfrak{m}}$ contains a K-basis of L, it is a free $A_{\mathfrak{m}}$-module of rank n.

Given any nonzero fractional ideal $I \in \mathcal{I}_B$, there exists $a \in A$ such that $aI \subset B$. Therefore, I is finitely generated as an A-module. As we argued for the A-module B itself, the localised module $I_{\mathfrak{m}}$ is also a free $A_{\mathfrak{m}}$-module of rank n.

Definition 1.3. *An $\mathfrak{m}$-integral basis of I is an $A_{\mathfrak{m}}$-basis of $I_{\mathfrak{m}}$.*

Definition 1.4. *Let $I \in \mathcal{I}_B$ be a fractional ideal of B. Consider the following mapping:*

$$\begin{aligned} w := w_{\mathfrak{m},I} : L &\longrightarrow \mathbb{Q} \cup \{\infty\}, \\ \alpha &\longmapsto w(\alpha) = \min\left\{(v_{\mathfrak{p}}(\alpha) - v_{\mathfrak{p}}(I))/e(\mathfrak{p}/\mathfrak{m})\right\}_{\mathfrak{p}\,|\,\mathfrak{m}}. \end{aligned}$$

The map w does not behave well with respect to multiplication, but it has some of the typical properties of a valuation.

Lemma 1.5. *Let $I \in \mathcal{I}_B$, $a \in K$, and $\alpha, \beta \in L$.*

1. $w(a\alpha) = v_{\mathfrak{m}}(a) + w(\alpha)$.
2. $w(\alpha + \beta) \geqslant \min\{w(\alpha), w(\beta)\}$, *and equality holds if* $w(\alpha) \neq w(\beta)$.

Proof. The first item is an immediate consequence of $w_{\mathfrak{p}}(a) = v_{\mathfrak{m}}(a)$.

Let us prove the second item. Denote $a_{\mathfrak{p}} := v_{\mathfrak{p}}(I)/e(\mathfrak{p}/\mathfrak{m})$, for all $\mathfrak{p} \mid \mathfrak{m}$. Suppose $w(\beta) \leqslant w(\alpha)$, and let $\mathfrak{q} \mid \mathfrak{m}$ such that

$$w(\beta) = w_{\mathfrak{q}}(\beta) - a_{\mathfrak{q}} \leqslant \min\left\{w_{\mathfrak{p}}(\alpha) - a_{\mathfrak{p}}\right\}_{\mathfrak{p}\,|\,\mathfrak{m}}.$$

Take $\mathfrak{p}' \mid \mathfrak{m}$ such that $w(\alpha+\beta) = w_{\mathfrak{p}'}(\alpha+\beta) - a_{\mathfrak{p}'}$. Clearly,

$$w(\alpha+\beta) = w_{\mathfrak{p}'}(\alpha+\beta) - a_{\mathfrak{p}'} \geqslant \min\left\{w_{\mathfrak{p}'}(\beta), w_{\mathfrak{p}'}(\alpha)\right\} - a_{\mathfrak{p}'} \geqslant w(\beta).$$

If $w(\beta) < w(\alpha)$, we have $w_{\mathfrak{q}}(\beta) - a_{\mathfrak{q}} < w_{\mathfrak{p}}(\alpha) - a_{\mathfrak{p}}$, for all $\mathfrak{p} \mid \mathfrak{m}$. Hence,

$$w(\beta) = w_{\mathfrak{q}}(\beta) - a_{\mathfrak{q}} = w_{\mathfrak{q}}(\alpha+\beta) - a_{\mathfrak{q}} \geqslant w(\alpha+\beta) \geqslant w(\beta),$$

so that all inequalities must be equalities. In particular, $w(\alpha+\beta) = w(\beta)$. □

This map $w = w_{\mathfrak{m},I}$ is useful to detect what elements in L belong to $I_{\mathfrak{m}}$. Clearly,

$$I_{\mathfrak{m}} = \{x \in L : w_{\mathfrak{m},I}(x) \geqslant 0\} \subset L. \tag{1.2}$$

1.3 Indices of lattices over principal ideal domains

Let A be a PID, with field of fractions K. Let

$$\mathcal{I}_A = \mathrm{Pr}_A = \{xA : x \in K^*\} \simeq K^*/A^*,$$

be the group of fractional ideals of A, that coincides with the subgroup of nonzero principal ideals.

In this section we fix a K-vector space V of finite dimension n. Let us be precise about the way we consider transition matrices between two bases of V.

Definition 1.6. *Let $\mathcal{B} = (\alpha_1, \ldots, \alpha_n) \in V^n$, $\mathcal{B}' = (\alpha'_1, \ldots, \alpha'_n) \in V^n$ be two bases of V. The transition matrix from B to B' is the matrix $T = T(\mathcal{B}' \leftarrow \mathcal{B}) \in \mathrm{GL}_n(K)$ determined by:*

$$(\alpha'_1 \ \cdots \ \alpha'_n)T = (\alpha_1 \ \cdots \ \alpha_n).$$

Note that the j-th column of T collects the coordinates of α_j with respect to the basis B'.

Definition 1.7. *An A-lattice of V is a finitely generated A-submodule $M \subset V$, containing a set of generators of V as a K-vector space.*

Since our base ring A is a PID, our lattices will be free A-modules of rank n. Thus, a lattice M is determined by an arbitrary basis $\mathcal{B} = (\alpha_1, \ldots, \alpha_n)$ of V as a K-vector space, by taking $M = \langle \alpha_1, \ldots, \alpha_n \rangle_A$.

Definition 1.8. *Let $M, N \subset V$ be two lattices of V. The* index $[M : N] \in \mathcal{I}_A$ *is defined to be the fractional ideal generated by the determinant of the transition matrix from any A-basis of N to any A-basis of M.*

The choice of different A-bases of N and M leads to transition matrices $T, T' \in \mathrm{GL}_n(K)$ related by:

$$T' = PTQ, \qquad P, Q \in \mathrm{GL}_n(A).$$

Thus, $\det(T') = u \det(T)$, for some unit $u \in A^*$, so that $\det(T)$ and $\det(T')$ generate the same principal ideal. Therefore, the index $[M : N]$ is well-defined.

Lemma 1.9. *Let $L, M, N \subset V$ be lattices of V.*

1. $[L : N] = [L : M][M : N]$.
2. $[M : N] = [N : M]^{-1}$.
3. $[xM : xN] = [M : N]$, *for all* $x \in K^*$.
4. *If* $N \subset M$, *then* $[M : N] = (a_1 \cdots a_n)A$, *where* $a_1, \ldots, a_n \in A \backslash \{0\}$ *satisfy*
$$M/N \simeq (A/a_1A) \times \cdots \times (A/a_nA).$$
5. $[M : N]_{\mathfrak{m}} = [M_{\mathfrak{m}} : N_{\mathfrak{m}}]$, *for all* $\mathfrak{m} \in \mathrm{Max}(A)$.
6. $[M : N] \otimes_A \hat{A}_{\mathfrak{m}} = [M \otimes_A \hat{A}_{\mathfrak{m}} : N \otimes_A \hat{A}_{\mathfrak{m}}]$, *for all* $\mathfrak{m} \in \mathrm{Max}(A)$.

Proof. The three first items are an immediate consequence of well-known properties of the transition matrices.

The fourth item follows from the theory of elementary divisors. There exist an A-basis $(\alpha_1, \ldots, \alpha_n)$ of M, and nonzero elements $a_1, \ldots, a_n \in A$ such that $(a_1\alpha_1, \ldots, a_n\alpha_n)$ is an A-basis of N.

Let us prove (5). Consider the transition matrix T from an A-basis $\mathcal{B}_N$ of N to an A-basis $\mathcal{B}_M$ of M; then, T is the transition matrix from the $A_\mathfrak{m}$-basis $\mathcal{B}_N$ of $N_\mathfrak{m}$ to the $A_\mathfrak{m}$-basis $\mathcal{B}_M$ of $M_\mathfrak{m}$. Hence, $[M : N]_\mathfrak{m} = \det(T)A_\mathfrak{m} = [M_\mathfrak{m} : N_\mathfrak{m}]$.

The same argument proves (6). □

The next result follows immediately from these properties.

Lemma 1.10. *Let $N \subset M$ be two lattices of V and let $\mathfrak{m} \in \mathrm{Max}(A)$.*

1. *$N = M$ if and only if $[M : N] = A$.*
2. *$N_\mathfrak{m} = M_\mathfrak{m}$ if and only if $\mathfrak{m} \nmid [M : N]$.*

Remark 1.11. This index of lattices is a particular instance of a more general invariant $\chi(M, N)$ introduced by J. P. Serre for an arbitrary Dedekind ring A [Ser68].

1.4 Normal forms of bases of fractional ideals

Let A be a PID, with field of fractions K, and let L/K be a finite field extension of degree n as in Section 1.2. By assumption, the integral closure B of A in L is a finitely generated A-module. Since A a PID, B is a free A-module of rank n.

The fractional ideals $I \in \mathcal{I}_B$ are lattices of the K-vector space $V = L$. As such they are free A-modules of rank n, and we are interested in the computation of A-bases for them.

Proposition 1.12 ([Ser68])**.** *For any fractional ideal $I \in \mathcal{I}_B$, we have $[B : I] = N_{L/K}(I)$.*

From a computational perspective, we consider the elements of L as K-linear combinations of the powers $1, \theta, \ldots, \theta^{n-1}$ of the root θ of the given

polynomial $f(x) \in A[x]$, defining the extension L/K. Consider the chain of K-subspaces,

$$0 = V_0 \subset V_1 = K \subset V_2 \subset \cdots \subset V_n = L, \tag{1.3}$$

where for $1 \leqslant i \leqslant n$, V_i is the subspace generated by $1, \theta, \ldots, \theta^{i-1}$.

The A-bases of fractional ideals are more easily handled in practice if they are given in adequate normal forms.

1.4.1 Triangular bases

Let $\mathbb{P}$ be a complete set of non-associate prime elements in A. Thus, every prime element $q \in A$ is written as $q = up$, for unique $u \in A^*$, $p \in \mathbb{P}$. Let $\mathbb{P}^{\mathbb{Z}} \subset K$ be the subset of elements which are finite products of powers of primes in $\mathbb{P}$, with integer exponents. Thus, for every $x \in K^*$, there exist unique $u \in A^*$, $y \in \mathbb{P}^{\mathbb{Z}}$ such that $x = uy$. Finally, we denote $\mathbb{P}^{\mathbb{N}} := \mathbb{P}^{\mathbb{Z}} \cap A$; clearly, $\mathbb{P}^{\mathbb{N}}$ is a complete set of non-associate elements in $A\backslash\{0\}$.

Definition 1.13. *Let $(\alpha_0, \ldots, \alpha_{n-1}) \in L^n$ be an A-basis of a nonzero fractional ideal $I \in \mathcal{I}_B$. We say that the basis is* triangular *(with respect to the choice of $f(x)$ as a defining polynomial of L/K) if it satisfies the following two properties:*

1. *For every $0 \leqslant j < n$, $\alpha_j = d_j g_j(\theta)$, where $d_j \in \mathbb{P}^{\mathbb{Z}}$ and*

$$g_j(x) = x^j + a_{j-1,j}x^{j-1} + \cdots + a_{1,j}x + a_{0,j} \in A[x]$$

   is a monic polynomial of degree j. We take $a_{j,j} := 1$ by convention.

2. *$d_0 A \subset d_1 A \subset \cdots \subset d_{n-1} A$.*

Our first aim is to show that every fractional ideal admits a triangular basis. This is a specific property of fractional ideals, since not every lattice of L admits a triangular basis. For instance, if $A = \mathbb{Z}$, $K = \mathbb{Q}$, and $L = \mathbb{Q}(\theta)$ is an arbitrary quadratic extension, then the lattice $M = \langle 2,\ 2\theta + 1 \rangle_{\mathbb{Z}}$ does not admit a triangular basis.

Lemma 1.14. *Let $I \in \mathcal{I}_B$ be a nonzero fractional ideal. For every integer $0 \leqslant m < n$, denote*

$$I_m := \{d \in K : d\theta^m \in I + V_m\},$$

for the chain $V_0 \subset \cdots \subset V_{n-1}$ of K-subspaces of L defined in (1.3)*. Then,*

1. *$I_m \in \mathcal{I}_A$ is a nonzero fractional ideal of A. In particular, there exists a unique $d_m \in \mathbb{P}^{\mathbb{Z}}$ such that $I_m = d_m A$.*
2. *$I_0 = I \cap K$.*
3. *$I_0 \subset I_1 \subset \cdots \subset I_{n-1}$.*
4. *$I \cap V_{m+1} \subset I_m \cdot V_{m+1}$.*

Proof. Clearly, $I_m \subset K$ is an A-submodule. In order to prove (1) we need only to check that it is nonzero and finitely generated as an A-module.

Let $a, b \in A$ be nonzero elements such that $aI \subset A[\theta]$ and $bA[\theta] \subset I$. Clearly, $b \in I_m$ for all m; thus I_m is nonzero. Also, $aI_m \subset A$, so that I_m is finitely generated.

The second item is an immediate consequence of the definition of I_0. The third item follows from the fact that I is stable by multiplication by θ.

Let us prove (4) by induction on m. For $m = 0$ the statement is a consequence of (2). Suppose that $0 < m < n$ and (4) holds for all indices less than m. By item 3, we have $d_{m-1} = d_m a$, for some $a \in A$. If $\alpha := c_0 + c_1\theta + \cdots + c_m\theta^m \in I$, then $c_m \in I_m$ by definition; write $c_m = d_m b$, for some $b \in A$. Consider now any $\beta := e_0 + e_1\theta + \cdots + e_{m-1}\theta^{m-1} \in I$, with $e_{m-1} = d_{m-1}$. By the induction hypothesis, $e_0, \ldots, e_{m-1} \in I_{m-1}$. Now, the element $a\alpha - b\beta\theta \in I$ is a polynomial in θ of degree $m-1$; by the induction hypothesis, all its coefficients $ac_i - be_{i-1}$ belong to I_{m-1}. Since $be_{i-1} \in I_{m-1}$, we deduce that $ac_i \in I_{m-1}$, for all $0 \leqslant i < m$. Finally, $ac_i \in I_{m-1} = d_{m-1}A$ is equivalent to $c_i \in (d_{m-1}/a)A = d_m A$. □

Definition 1.15. *These elements $d_0, \ldots, d_{n-1} \in K^*$ are canonical invariants of I (they depend only on the choice of $f(x) \in A[x]$ as a defining polynomial of L/K). We allow an abuse of language and we say that they are the* elementary divisors *of I.*

The next result shows the existence of triangular bases and the essential property of these bases that may be used to construct them.

Theorem 1.16. *Let $I \in \mathcal{I}_B$ be a nonzero fractional ideal. For every integer $0 \leqslant m < n$, consider a pair of elements $d_m \in K^*$ and $\beta_m = b_{0,m} + b_{1,m}\theta + \cdots + b_{m-1,m}\theta^{m-1} + \theta^m \in A[\theta]$ satisfying:*

1. *$d_m\beta_m \in I$,*
2. *$d_m A$ is maximal (with respect to the inclusion of fractional ideals) with this property, for all possible choices of β_m.*

Then, $d_0\beta_0, \ldots, d_{n-1}\beta_{n-1}$ is a triangular basis of I.

Proof. By Lemma 1.14, $d_0, \ldots, d_{n-1}$ are the elementary divisors of I. Thus, we need only to show that $d_0\beta_0, \ldots, d_{n-1}\beta_{n-1}$ is an A-basis of I.

Let $\alpha = c_0 + c_1\theta + \cdots + c_{n-1}\theta^{n-1} \in I$, for some $c_0, \ldots, c_{n-1} \in K$. By definition, $c_{n-1} \in I_{n-1}$, so that there exists $a \in A$ such that $c_{n-1} = ad_{n-1}$; hence, $\alpha - ad_{n-1}\beta_{n-1} \in I$ is a polynomial in θ of degree less than $n-1$. An iteration of this argument shows that α is an A-linear combination of $d_0\beta_0, \ldots, d_{n-1}\beta_{n-1}$. □

Corollary 1.17. *If $I \in \mathcal{I}_B$ has elementary divisors $d_0, \ldots, d_{n-1}$, then:*

1. *$[A[\theta] : I] = d_0 \cdots d_{n-1}A$.*
2. *$N_{L/K}(I) = [B : A[\theta]] \cdot (d_0 \cdots d_{n-1}A)$.*

Proof. Clearly, $\beta_0, \ldots, \beta_{n-1}$ is an A-basis of $A[\theta]$; this proves the first item. The second item is a consequence of Proposition 1.12 and the transitivity of the index (first item of Lemma 1.9). □

Remark 1.18. The maximality of $d_m A$ is equivalent to the minimality of d_m under the following partial ordering of $\mathbb{P}^{\mathbb{Z}} \simeq K^*/A^*$:

$$x \leqslant y \iff x \mid y \iff \text{there exists } a \in A \text{ such that } y = ax.$$

For $I = B$, the elements d_m are of the form $1/u_m$, with $u_m \in A$. In this case, the minimality of d_m is equivalent to the maximality of $u_m \in A$, under the same partial ordering.

1.4.2 Hermitian bases

Definition 1.19. *Let $I \in \mathcal{I}_B$ and let $(\alpha_0, \ldots, \alpha_{n-1}) \in L^n$ be a triangular A-basis of I; that is, conditions (1) and (2) of Definition 1.13 are satisfied.*

We say that $(\alpha_0, \ldots, \alpha_{n-1})$ is a Hermitian basis *of I if it also satisfies:*

3. *For all $0 \leqslant i < j < n$, the element $a_{i,j}$ belongs to a fixed subset of representatives of $A/(d_i/d_j)A$.*

This condition is equivalent to the fact that a certain nonsingular square matrix over A is in Hermite normal form (HNF).

Definition 1.20. *Let $H = (b_{i,j}) \in A^{n\times n}$ be a nonsingular matrix, with rows and columns indexed by $0 \leqslant i, j < n$. We say that H is in* Hermite normal form *(under column transformations) over A if*

1. *It is an upper triangular matrix.*
2. *$b_{i,i} \in \mathbb{P}^{\mathbb{N}}$, for $0 \leqslant i < n$.*
3. *$b_{i,j}$ belongs to a fixed subset of representatives of $A/b_{i,i}A$, for all $0 \leqslant i < j < n$.*

Every nonsingular square matrix $M \in A^{n\times n}$ can be transformed into a unique matrix in HNF by elementary column transformations with coefficients in A. In other words,

1. There exists $Q \in \mathrm{GL}_n(A)$ such that MQ is in HNF.
2. If $H, H' \in A^{n\times n}$ are matrices in HNF and $H' = HQ$ for some $Q \in \mathrm{GL}_n(A)$, then $H = H'$.

The next result is a straightforward consequence of Lemma 1.14.

Lemma 1.21. *Let $I \in \mathcal{I}_B$ be a fractional ideal of B and then let $\mathcal{B} = (\alpha_0, \ldots, \alpha_{n-1}) \in L^n$ be an A-basis of I. Let $\mathcal{B}_\theta = (1, \theta, \ldots, \theta^{n-1})$ be the standard A-basis of $A[\theta]$, and consider the transition matrix $T = T(\mathcal{B}_\theta \leftarrow \mathcal{B}) \in \mathrm{GL}_n(K)$. Let $d \in A$ be any element such that $dT \in A^{n\times n}$. Then, the basis $\mathcal{B}$ is Hermitian if and only if the matrix dT is in HNF over A.*

Corollary 1.22. *Every $I \in \mathcal{I}_B$ admits a unique Hermite basis.*

As usual, the Hermite basis of I depends on the choice of the defining polynomial $f(x) \in A[x]$ of the extension L/K. So, the "uniqueness" statement implicitly assumes that this defining polynomial is fixed.

In practice, we may find the Hermite basis of I by the following procedure:

1. Compute a triangular basis $\mathcal{B}$ of I. Let $d_0, \ldots, d_{n-1}$ be the elementary divisors of I.
2. Compute the transition matrix $T = T(\mathcal{B}_\theta \leftarrow \mathcal{B}) \in \mathrm{GL}_n(K)$.
3. Apply the HNF routine over A to the upper triangular matrix $T' := d_{n-1}^{-1} T \in A^{n \times n}$. Let H be the HNF of T'.
4. The coordinates of the elements in the Hermite basis of I, with respect to the K-basis 1, θ, $\ldots$, θ^{n-1} of L, are the columns of the matrix $d_{n-1} H$.

The crucial point is the computation of the triangular basis. The rest of the steps are trivial and reasonably efficient. In step (3) the efficiency relies on the fact that the input matrix to the HNF routine is already upper triangular.

1.5 Local triangular bases

Local bases exist for arbitrary Dedekind domains A, as we saw in Section 1.2. For any fractional ideal $I \in \mathcal{I}_B$ and any $\mathfrak{m} \in \mathrm{Max}(A)$ the localised ideal $I_\mathfrak{m}$ is a free $A_\mathfrak{m}$-module of rank n and an $A_\mathfrak{m}$-basis of $I_\mathfrak{m}$ is called an $\mathfrak{m}$-integral basis of I (Definition 1.3). The local ring $A_\mathfrak{m}$ is a PID with

$$\mathbb{P}(A_\mathfrak{m}) = \{\pi\}\,,$$
$$\mathbb{P}(A_\mathfrak{m})^\mathbb{Z} = \{\pi^\nu : \nu \in \mathbb{Z}\}\,,$$

where $\pi \in \mathfrak{m}$ is a local generator.

By the results of Section 1.4, $I_\mathfrak{m}$ admits triangular $A_\mathfrak{m}$-bases, which may be called $\mathfrak{m}$-*triangular bases* of I.

Theorem 1.23. *Let $I \in \mathcal{I}_B$ be a nonzero fractional ideal. For every integer $0 \leqslant m < n$, consider a pair of elements $\nu_m \in \mathbb{Z}$ and $\beta_m = b_{0,m} + b_{1,m}\theta + \cdots + b_{m-1,m}\theta^{m-1} + \theta^m \in A[\theta]$ satisfying:*

1. *$\pi^{\nu_m}\beta_m \in I$,*
2. *ν_m is minimal with this property, for all possible choices of β_m.*

Then, $\pi^{\nu_0}\beta_0, \ldots, \pi^{\nu_{n-1}}\beta_{n-1}$ is an $\mathfrak{m}$-triangular basis of I.

Proof. This is a particular instance of Theorem 1.16 applied to the ring $A_{\mathfrak{m}}$, except for the fact that we require the β_m's to be polynomials in θ with coefficients in A, and not merely in $A_{\mathfrak{m}}$. However, if $\pi^{\nu_m}\beta'_m \in I_{\mathfrak{m}}$, for some $\beta'_m = b'_{0,m} + b'_{1,m}\theta + \cdots + b'_{m-1,m}\theta^{m-1} + \theta^m \in A_{\mathfrak{m}}[\theta]$, then we also have $\pi^{\nu_m}\beta_m \in I_{\mathfrak{m}}$ for $\beta_m = b_{0,m} + b_{1,m}\theta + \cdots + b_{m-1,m}\theta^{m-1} + \theta^m \in A[\theta]$, if all b_m are sufficiently close to b'_m in the $\mathfrak{m}$-adic topology. □

We may be more precise about the link between ν_m and β_m.

Corollary 1.24. *With the above notation, $\nu_m = \lceil -w_{\mathfrak{m},I}(\beta_m) \rceil$, for all $0 \leqslant m < n$. In particular, $\nu_0 = \lceil \max_{\mathfrak{p}\,|\,\mathfrak{m}}\{v_{\mathfrak{p}}(I)/e(\mathfrak{p}/\mathfrak{m})\} \rceil$.*

Proof. Fix one index $0 \leqslant m < n$. By (1.2),

$$\pi^{\nu_m}\beta_m \in I_{\mathfrak{m}} \Longleftrightarrow w_{\mathfrak{m},I}(\pi^{\nu_m}\beta_m) \geqslant 0 \Longleftrightarrow w_{\mathfrak{m},I}(\beta_m) \geqslant -\nu_m.$$

By the minimality of ν_m, we have necessarily $\nu_m = \lceil -w_{\mathfrak{m},I}(\beta_m) \rceil$.
The statement about ν_0 follows from $\beta_0 = 1$. □

Condition (2) of Theorem 1.23 says that the integer $\lfloor w_{\mathfrak{m},I}(\beta_m) \rfloor$ is maximal, for $\beta_m = h_m(\theta)$ with $h_m \in A[x]$ running on all monic polynomials of degree m. If we require the stronger property, that the rational number $w_{\mathfrak{m},I}(\beta_m)$ is maximal, we obtain $\mathfrak{m}$-*reduced* bases.

Definition 1.25. *A family $\alpha_1, \ldots, \alpha_r \in B$ is called* $\mathfrak{m}$-reduced *if for any family $a_1, \ldots, a_r \in A_{\mathfrak{m}}$:*

$$w_{\mathfrak{m},I}\left(\sum\nolimits_{1 \leqslant i \leqslant r} a_i\alpha_i\right) = \min\left\{w_{\mathfrak{m},I}(a_i\alpha_i) : 1 \leqslant i \leqslant n\right\}.$$

We may strengthen Theorem 1.23 in the following way.

Theorem 1.26. *Let $I \in \mathcal{I}_B$ be a nonzero fractional ideal. For every integer $0 \leqslant m < n$, consider $g_m \in A[x]$ a monic polynomial of degree m such that $w_{\mathfrak{m},I}(g_m(\theta))$ is maximal among all possible choices of g_m. Denote $\nu_m := \lceil -w_{\mathfrak{m},I}(g_m(\theta)) \rceil$. Then, $\pi^{\nu_0} g_0(\theta), \ldots, \pi^{\nu_{n-1}} g_{n-1}(\theta)$ is an $\mathfrak{m}$-triangular $\mathfrak{m}$-reduced basis of I.*

Proof. By Theorem 1.23, $\pi^{\nu_0} g_0(\theta), \ldots, \pi^{\nu_{n-1}} g_{n-1}(\theta)$ is an $\mathfrak{m}$-triangular basis of I, so it only remains to show that it is also $\mathfrak{m}$-reduced.

Denote $\alpha_i := \pi^{\nu_i} g_i(\theta)$.

Given a family $a_0, \ldots, a_{n-1} \in A_{\mathfrak{m}}$, let $\delta = \min\{w_{\mathfrak{m},I}(a_i\alpha_i) : 0 \leqslant i < n\}$. By Lemma 1.5,

$$w_{\mathfrak{m},I}\left(\sum_{0\leqslant i<n} a_i\alpha_i\right) \geqslant \min\{w_{\mathfrak{m},I}(a_i\alpha_i) : 1 \leqslant i < n\} = \delta.$$

Hence, it suffices to show that $w_{\mathfrak{m},I}\left(\sum_{0\leqslant i<n} a_i\alpha_i\right) \leqslant \delta$.

Take $\mathbb{i} = \{i : w_{\mathfrak{m},I}(a_i\alpha_i) = \delta\}$. By Lemma 1.5,

$$w_{\mathfrak{m},I}\left(\sum_{0\leqslant i<n} a_i\alpha_i\right) = w_{\mathfrak{m},I}\left(\sum_{i\in\mathbb{i}} a_i\alpha_i\right).$$

Since $0 \leqslant w_{\mathfrak{m},I}(\alpha_i) < 1$ for all i, the values $v_{\mathfrak{m}}(a_i) \in \mathbb{Z}_{\geqslant 0}$ are all constant for $i \in \mathbb{i}$. Dividing by an adequate π-power we may assume $v_{\mathfrak{m}}(a_i) = 0$ for all $i \in \mathbb{i}$. If $i_0 = \max(\mathbb{i})$, we may divide everything by a_{i_0} (which is now a unit in $A_{\mathfrak{m}}$) so that we may assume that $a_{i_0} = 1$. Now,

$$\sum_{i\in\mathbb{i}} a_i\alpha_i = \pi^{\nu_{i_0}} h(\theta),$$

for a certain monic polynomial $h \in A_{\mathfrak{m}}[\theta]$ of degree i_0. By the maximality of $g_{i_0}(\theta)$ we must have

$$w_{\mathfrak{m},I}\left(\sum_{i\in\mathbb{i}} a_i\alpha_i\right) \leqslant w_{\mathfrak{m},I}(\pi^{\nu_{i_0}} g_{i_0}(\theta)) = w_{\mathfrak{m},I}(\alpha_{i_0}) = \delta.$$

□

1.6 Global triangular bases

If A is a PID, there is a standard procedure to patch triangular $\mathfrak{m}$-integral bases of I into a triangular A-basis of I, by means of the CRT.

Notation. *Denote by* $\mathrm{Up_n(A)}$ *the subgroup of* $\mathrm{GL}_n(A)$ *formed by the upper triangular matrices such that all elements in the principal diagonal are equal to* 1.

Theorem 1.27. *Let* $I \in \mathcal{I}_B$ *be a nonzero fractional ideal of* B*, and let*

$$\mathcal{P}_I := \mathrm{Supp}(I) \cup \mathrm{Supp}([B : A[\theta]]) \subset \mathrm{Max}(A),$$

be the set of all $\mathfrak{m} \in \mathrm{Max}(A)$ *such that* $\mathfrak{m}$ *divides some* $\mathfrak{p} \in \mathrm{Spec}(B)$ *with* $v_{\mathfrak{p}}(I) \neq 0$*, or* $\mathfrak{m}$ *divides* $[B : A[\theta]]$*.*

Suppose that for each $\mathfrak{m} = \pi_{\mathfrak{m}} A \in \mathcal{P}_I$ *we have an* $\mathfrak{m}$*-triangular basis of* $I_{\mathfrak{m}}$*:*

$$\pi_{\mathfrak{m}}^{\nu_{0,\mathfrak{m}}}\beta_{0,\mathfrak{m}}, \ldots, \pi_{\mathfrak{m}}^{\nu_{n-1,\mathfrak{m}}}\beta_{n-1,\mathfrak{m}}, \qquad \nu_{j,\mathfrak{m}} = \lceil -w_{\mathfrak{m},I}(\beta_{j,\mathfrak{m}}) \rceil,$$

where $\beta_{0,\mathfrak{m}}, \ldots, \beta_{n-1,\mathfrak{m}}$ *is a triangular* A*-basis of* $A[\theta]$*, given by a transition matrix* $U_{\mathfrak{m}} \in \mathrm{Up_n(A)}$*:*

$$(1 \ \theta \ \cdots \ \theta^{n-1})U_{\mathfrak{m}} = (\beta_{0,\mathfrak{m}} \ \cdots \ \beta_{n-1,\mathfrak{m}}).$$

Take $U \in \mathrm{Up_n(A)}$ *whose columns* U^j *are the solution of the following CRT problem:*

$$U^j \equiv (U_{\mathfrak{m}})^j \pmod{\pi_{\mathfrak{m}}^{\nu_{0,\mathfrak{m}} - \nu_{j,\mathfrak{m}}}}, \qquad \forall\ \mathfrak{m} \in \mathcal{P}_I,$$

and consider the triangular A*-basis* $\beta_0, \ldots, \beta_{n-1}$ *of* $A[\theta]$ *determined by:*

$$(1 \ \theta \ \cdots \ \theta^{n-1})U = (\beta_0 \ \cdots \ \beta_{n-1}).$$

Then, $d_0\beta_0, \ldots, d_{n-1}\beta_{n-1}$ *is a triangular* A*-basis of* I*, where we denote* $d_j := \prod_{\mathfrak{m} \in \mathcal{P}_I} \pi_{\mathfrak{m}}^{\nu_{j,\mathfrak{m}}}$*.*

Proof. Let us check first that $d_m\beta_m$ belongs to I for all $0 \leqslant m < n$. It is sufficient to check that $d_m\beta_m \in I_{\mathfrak{m}}$, for all $\mathfrak{m} \in \mathrm{Max}(A)$ (cf. Section 1.1).

If $\mathfrak{m} \notin \mathcal{P}_I$, we have $I_{\mathfrak{m}} = A[\theta]_{\mathfrak{m}}$. In fact, $I_{\mathfrak{m}} = B_{\mathfrak{m}}$, because $\mathfrak{m} \notin \mathrm{Supp}(I)$, and $B_{\mathfrak{m}} = A[\theta]_{\mathfrak{m}}$ by Lemma 1.10, because $\mathfrak{m} \nmid [B : A[\theta]]$. Now, since $d_m \in A_{\mathfrak{m}}$ and $\beta_m \in A[\theta]$, clearly $d_m\beta_m \in A[\theta]_{\mathfrak{m}} = I_{\mathfrak{m}}$.

If $\mathfrak{m} \in \mathcal{P}_I$, then by Corollary 1.24 we have

$$w_{\mathfrak{p}}\left(\beta_m - \beta_{m,\mathfrak{m}}\right) \geqslant \nu_{0,\mathfrak{m}} - \nu_{m,\mathfrak{m}} \geqslant \frac{v_{\mathfrak{p}}\left(I\right)}{e(\mathfrak{p}/\mathfrak{m})} - \nu_{m,\mathfrak{m}}, \qquad \forall\ \mathfrak{p} \mid \mathfrak{m}.$$

Hence, $w_{\mathfrak{m},I}\left(\beta_m - \beta_{m,\mathfrak{m}}\right) \geqslant -\nu_{m,\mathfrak{m}}$. Since $w_{\mathfrak{m},I}\left(\beta_{m,\mathfrak{m}}\right) \geqslant -\nu_{m,\mathfrak{m}}$, Lemma 1.5 shows that $w_{\mathfrak{m},I}\left(\beta_m\right) \geqslant -\nu_{m,\mathfrak{m}}$ and $w_{\mathfrak{m},I}\left(d_m\beta_m\right) = \nu_{m,\mathfrak{m}} + w_{\mathfrak{m},I}\left(\beta_m\right) \geqslant 0$. Thus, $d_m\beta_m$ belongs to $I_{\mathfrak{m}}$, by (1.2).

Finally, it is clear that d_mA is maximal amongst all ideals dA, for $d \in K^*$ satisfying $d\beta'_m \in I$ for some β'_m a monic polynomial in θ of degree m with coefficients in A. In fact, if $d_mA \subsetneq dA$, then there must be a prime ideal $\mathfrak{m}$, with $v_{\mathfrak{m}}\left(d_m\right) > v_{\mathfrak{m}}\left(d\right)$. If $\mathfrak{m} \notin \mathcal{P}_I$, we get $0 > v_{\mathfrak{m}}\left(d\right)$, and this contradicts the equality $I_{\mathfrak{m}} = A[\theta]_{\mathfrak{m}} = A_{\mathfrak{m}}[\theta]$. If $\mathfrak{m} \in \mathcal{P}_I$, we get $\nu_{m,\mathfrak{m}} > v_{\mathfrak{m}}\left(d\right)$, and this contradicts the minimality of $\nu_{m,\mathfrak{m}}$, or equivalently, the fact that $\pi^{\nu_{m,\mathfrak{m}}}$ is the m-th elementary divisor of $I_{\mathfrak{m}}$.

By Theorem 1.16, $d_0\beta_0, \ldots, d_{n-1}\beta_{n-1}$ is a triangular basis of I. □

1.7 Aim of this memoir

The fastest methods to construct $\mathfrak{m}$-integral bases of B are the OM method given in [GMN13], and the *method of the quotients* developed in [GMN]. The second method is more efficient, but it has the disadvantage that it only applies to finding $\mathfrak{m}$-integral bases of B, while the first method (based on the construction of certain multipliers) is able to yield $\mathfrak{m}$-integral bases of arbitrary fractional ideals.

These methods yield non-triangular bases. However, in many applications, such as the computation of global A-bases of fractional ideals when A is a PID, we need local triangular bases, so that it is necessary to apply a triangularisation routine to certain nonsingular matrices in $A^{n\times n}$.

The main aim of this memoir is to find a direct construction of $\mathfrak{m}$-

triangular bases of fractional ideals, which works as fast as the aforementioned methods, and avoids the triangularisation routine.

Even if we want to construct a Hermitian basis, the HNF routine is much more efficient when it is applied to a matrix which is already triangular.

Our method for the constructions of $\mathfrak{m}$-triangular bases is called the MaxMin algorithm and it is discussed in detail in Chapter 4. The $\mathfrak{m}$-triangular bases computed by the MaxMin algorithm have a maximal $w_{\mathfrak{m},I}$-value; hence they are $\mathfrak{m}$-reduced too, as shown by Theorem 1.26. This is crucial for some applications to arithmetic properties of function fields (see Section 5.4).

Since we are only interested in local bases, we shall work in a purely local context. Our base ring will be an arbitrary discrete valuation ring $\mathcal{O}$ with field of fractions K. We shall consider a finite field extension L/K and our aim will be the computation of triangular $\mathcal{O}$-bases of the integral closure $\mathcal{O}_L$ of $\mathcal{O}$ in L.

The MaxMin algorithm is also an OM method. It requires an initial application of the Montes algorithm to compute OM representations of the prime ideals of $\mathcal{O}_L$. For the convenience of the reader, we will review the necessary background material in Chapter 2.

2

OM representations of prime ideals

"One should never mistake pattern for meaning."

– Iain M. Banks, *The Hydrogen Sonata*

Let (K, v) be a discrete valued field with valuation ring $\mathcal{O}$. Let $\mathfrak{m}$ be the maximal ideal, $\pi \in \mathfrak{m}$ a generator of $\mathfrak{m}$ and $\mathbb{F} = \mathcal{O}/\mathfrak{m}$ the residue class field.

Let K_v be the completion of K, and retain $v : \overline{K}_v^* \to \mathbb{Q}$ the canonical extension of v to a fixed algebraic closure of K_v. Let $\mathcal{O}_v$ be the valuation ring of K_v and $\mathfrak{m}_v$ its maximal ideal.

Let $f \in \mathcal{O}[x]$ be a monic, irreducible polynomial of degree n and fix a root $\theta \in \overline{K}$ in the algebraic closure of K. Let $L = K(\theta)$ be the finite extension of K defined by f and let $\mathcal{O}_L$ be the integral closure of $\mathcal{O}$ in L,

which is a Dedekind domain. We denote the set of prime ideals of $\mathcal{O}_L$ by $\mathcal{P}$.

We suppose that $\mathcal{O}_L$ is finitely generated as an $\mathcal{O}$-module. This condition holds under very natural assumptions; for instance, if L/K is separable, or (K, v) is complete, or $\mathcal{O}$ is a finitely generated algebra over a field [Ser68, I, §4].

By a theorem of Hensel [Hen08], the prime ideals of $\mathcal{O}_L$ are in 1-to-1 correspondence with the prime factors of f in $\mathcal{O}_v[x]$. The construction of "OM representations" of each prime factor of f yields computational data about the prime ideals of $\mathcal{O}_L$, encoding relevant arithmetic information about these ideals.

For this reason, we are initially interested in the representation of monic irreducible polynomials $F \in \mathcal{O}_v[x]$.

2.1 Okutsu equivalence of prime polynomials

Definition 2.1. *A prime polynomial with respect to v is a monic irreducible polynomial with coefficients in $\mathcal{O}_v[x]$. Let us denote by*

$$\mathbb{P} := \mathbb{P}(\mathcal{O}_v[x]) := \{F \in \mathcal{O}_v[x] : F \text{ monic, irreducible}\},$$

the set of all prime polynomials.

Let $F \in \mathbb{P}$ be a prime polynomial and fix $\theta \in \overline{K}_v$ a root of F. Let $K_F = K_v(\theta)$ be the finite extension of K_v generated by θ.

Definition 2.2. *The* Okutsu bound *of $F \in \mathbb{P}$ is defined as,*

$$\delta_0(F) := \deg(F) \max\{v(g(\theta))/\deg g : g \in \mathcal{O}[x], g \text{ monic}, \deg g < \deg F\}.$$

Definition 2.3. *Let $F, G \in \mathbb{P}$ be two prime polynomials of the same degree, and let $\theta \in \overline{K}_v$ be a root of F. We say that F and G are* Okutsu equivalent*, and we write $F \approx G$, if*

$$v(G(\theta)) > \delta_0(F).$$

We denote by $[F] \subseteq \mathbb{P}$ the *Okutsu class* of F; that is, the set of all prime polynomials which are Okutsu equivalent to F. The idea behind this concept is that all members of $[F]$ share certain discrete invariants, which are further described in the following section.

2.2 Types over (K, v)

A type $\mathbb{t}$ is a computational object consisting of discrete data, structured into levels:

$$\mathbb{t} = (\psi_0; (\phi_1, \lambda_1, \psi_1); \cdots ; (\phi_r, \lambda_r, \psi_r)). \tag{2.1}$$

The number r of levels is called the *order* of the type.

A type $\mathbb{t} = (\psi_0)$ of order 0 is determined by the choice of an arbitrary monic irreducible polynomial $\psi_0 \in \mathbb{F}[y]$. It supports the following data at level 0:

- Numerical data: $e_0 = m_0 = 1$, $\lambda_0 = h_0 = 0$.
- A discrete valuation $v_0 : K(x)^* \longrightarrow \mathbb{Z}$, determined by the following action on polynomials:

$$v_0 \left(\sum_{i>0} a_i x^i \right) := \min \{ v_0 (a_i) : i > 0 \} .$$

- $\psi_0 \in \mathbb{F}_0[y]$ a monic irreducible polynomial.
- $\mathbb{F}_1 = \mathbb{F}_0[y]/(\psi_0)$ a finite extension of $\mathbb{F}$ of degree $f_0 := \deg \psi_0$.
- $z_0 \in \mathbb{F}_1$ the class of y. Hence, $\mathbb{F}_1 = \mathbb{F}_0[z_0]$ and ψ_0 is the minimal polynomial of z_0 over $\mathbb{F}_0$.
- The residual polynomial operator $\mathrm{R}_0 : K[x] \to \mathbb{F}_0[y]$, where $\mathbb{F}_0 = \mathbb{F}$. It is defined as $\mathrm{R}_0(g) = \overline{g(y)/\pi^{v_0(g)}}$ for any non-zero $g \in K[x]$.

If $\mathbb{t}_0 = (\psi_0; (\phi_1, \lambda_1, \psi_1); \ldots ; (\phi_{i-1}, \lambda_{i-1}, \psi_{i-1}))$ is a type of order $i-1 \geqslant 0$, then a type $\mathbb{t} = (\mathbb{t}_0; (\phi_i, \lambda_i, \psi_i))$ of order i may be obtained by adding the following data at the i-th level:

- A *representative* ϕ_i of $\mathfrak{t}_0$. That is, a monic polynomial $\phi_i \in \mathcal{O}[x]$ of degree $m_i := e_{i-1}f_{i-1}m_{i-1}$ such that $\mathrm{R}_{i-1}(\phi_i) = \psi_{i-1}$.
- The value $V_i := v_{i-1}(\phi_i) \in \mathbb{Z}_{\geqslant 0}$.
- A Newton polygon operator $\mathrm{N}_i := \mathrm{N}_{v_{i-1},\phi_i}$.
- A positive rational number $\lambda_i = h_i/e_i$, with h_i, e_i positive coprime integers. We say that λ_i is the *slope* of $\mathfrak{t}$ at level i.
- The Bézout identity $\ell_i h_i + \ell_i' e_i = 1$, $0 \leqslant \ell_i < e_i$.
- A normalised augmented valuation v_i of $K(x)$.
- $\psi_i \in \mathbb{F}_i[y]$ a monic irreducible polynomial, $\psi_i \neq y$.
- $\mathbb{F}_{i+1} = \mathbb{F}_i[y]/(\psi_i)$ a finite extension of $\mathbb{F}_i$ of degree $f_i := \deg \psi_i$.
- $z_i \in \mathbb{F}_{i+1}$ the class of y. Hence, $\mathbb{F}_{i+1} = \mathbb{F}_i[z_r]$ and ψ_i is the minimal polynomial of z_i over $\mathbb{F}_i$.
- A residual polynomial operator $\mathrm{R}_i := \mathrm{R}_{v_{i-1},\phi_i,\lambda_i}$.

The polynomials $\phi_1, \ldots, \phi_i$ are prime polynomials with coefficients in $\mathcal{O}$. That is, they are irreducible over $\mathcal{O}_v[x]$.

We will now describe the i-th level operators N_i, v_i, and R_i in further detail.

Newton polygon

At each level a type $\mathfrak{t}$ defines a Newton polygon,

$$\mathrm{N}_i := \mathrm{N}_{v_{i-1},\phi_i} : K[x] \longrightarrow 2^{\mathbb{R}^2}, \qquad 1 \leqslant i \leqslant r,$$

where $2^{\mathbb{R}^2}$ is the set of subsets of the Euclidean plane. Any non-zero polynomial $g \in K_v[x]$ has a canonical ϕ_i-development:

$$g = \sum_{0 \leqslant s} a_s \phi_i^s, \qquad \deg a_s < m_i,$$

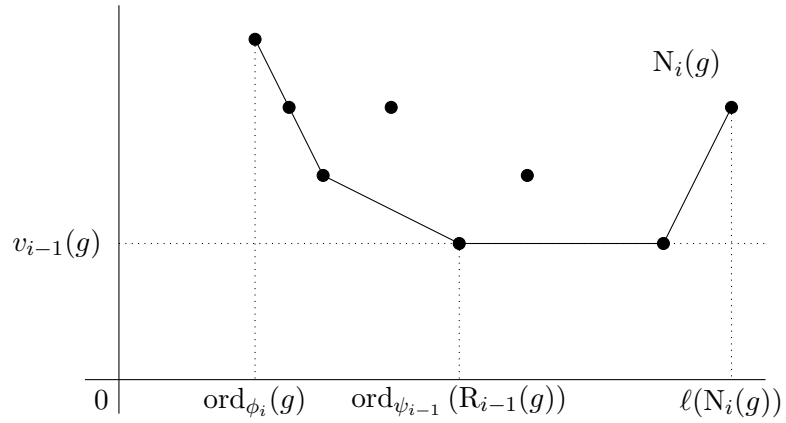


Figure 2.1: Newton polygon of a polynomial $g \in K[x]$.

and the polygon $\mathrm{N}_i(g)$ is the lower convex hull of the cloud of all the points $(s, v_{i-1}(a_s\phi_i^s))$. Figure 2.1 shows the typical shape of $\mathrm{N}_i(g)$.

If the Newton polygon $N = \mathrm{N}_i(g)$ is not a single point, we formally write $N = S_1 + \cdots + S_k$, where S_i are the sides of N, ordered by their increasing slopes. The left and right end-points of N and the points joining two sides of different slopes are called the *vertices* of N.

Usually, we are only interested in the *principal Newton polygon* $\mathrm{N}_i^-(g) \subseteq \mathrm{N}_i(g)$ formed by the sides of negative slope. If there are no sides of negative slope, then $\mathrm{N}_i^-(g)$ is the left end-point of $\mathrm{N}_i(g)$.

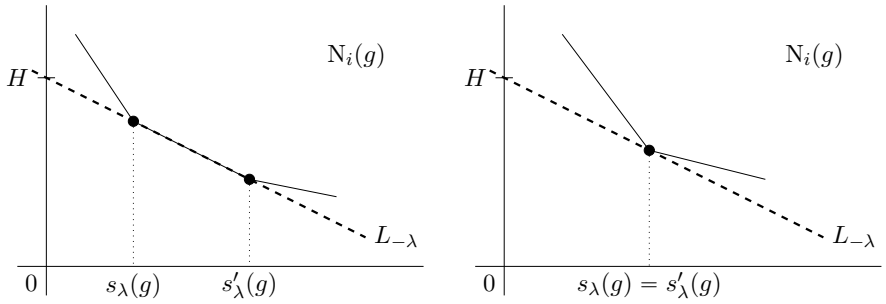
The length $\ell(N)$ of a Newton polygon N is the abscissa of its right end-point. For every non-zero polynomial $g \in K[x]$, we have

$$\ell(\mathrm{N}_i^-(g)) = \mathrm{ord}_{\psi_{i-1}}(\mathrm{R}_{i-1}(g)), \qquad \text{in } \mathbb{F}_{i-1}[y]. \tag{2.2}$$

Let $\lambda \in \mathbb{Q}_{>0}$ be a positive rational number and let $L_{-\lambda}$ be the line of slope $-\lambda$ which first touches the polygon $\mathrm{N}_i(g)$ from below. We define the λ-*component* of $N = \mathrm{N}_i(g)$ as the segment

$$S_\lambda := \{(x, y) \in N : y + \lambda x \text{ is minimal}\} = N \cap L_{-\lambda},$$

and we denote by $s_\lambda(g) \leqslant s'_\lambda(g)$ the abscissas of the end-points of $S_\lambda(g)$. If N has a side S of slope $-\lambda$, then $S_\lambda = S$ and $s_\lambda(g) < s'_\lambda(g)$, otherwise $S_\lambda(g)$ is a vertex of N and $s_\lambda(g) = s'_\lambda(g)$ (see Figure 2.2).



S_λ describes a side of $\mathrm{N}_i(g)$ S_λ describes a vertex of $\mathrm{N}_i(g)$

Figure 2.2: The λ-component of $\mathrm{N}_i(g)$.

Normalised augmented valuation

The Newton polygon operator N_i together with the slope λ_i define the normalised augmented valuation v_i of the field $K(x)$.

Given a polynomial $g \in K[x]$, let $L = L_{-\lambda_i}$ be the line of slope $-\lambda_i$ first touching $\mathrm{N}_i(g)$ from below. Let $(0, H)$ be the point where L crossed the vertical axis (see Figure 2.2). Then the v_i-valuation of g is defined as,

$$v_i(g) = He_i. \tag{2.3}$$

Residual polynomial

A type $\mathtt{t}$ also has a residual polynomial operator at each level,

$$\mathrm{R}_i := \mathrm{R}_{v_{i-1},\phi_i,\lambda_i} : K[x] \longrightarrow \mathbb{F}_i[y], \qquad 1 \leqslant i \leqslant r.$$

The operator R_i maps 0 to 0. For a non-zero $g \in K[x]$ with ϕ_i-expansion $g = \sum_{0 \leqslant s} a_s \phi_i^s$, let us denote by $s_i(g) \leqslant s'_i(g)$ the abscissas of the end points of the λ_i-component $S = S_{\lambda_i}(\mathrm{N}_i(g))$ of $\mathrm{N}_i(g)$.

Let $d = (s'_i(g) - s_i(g))/e_i$ be the *degree* of S. There are $d+1$ points of integer coordinates $P_0, \ldots, P_d$ lying on S, with abscissas $s_j := s_i(g) + je_i$ for $0 \leqslant j \leqslant d$ (see Figure 2.3). Denote by $Q_j = (s_j, v_{i-1}(a_{s_j}\phi_i^{s_j}))$ the point of abscissa s_j in the cloud of points which is used to compute the Newton polygon $\mathrm{N}_i(g)$.

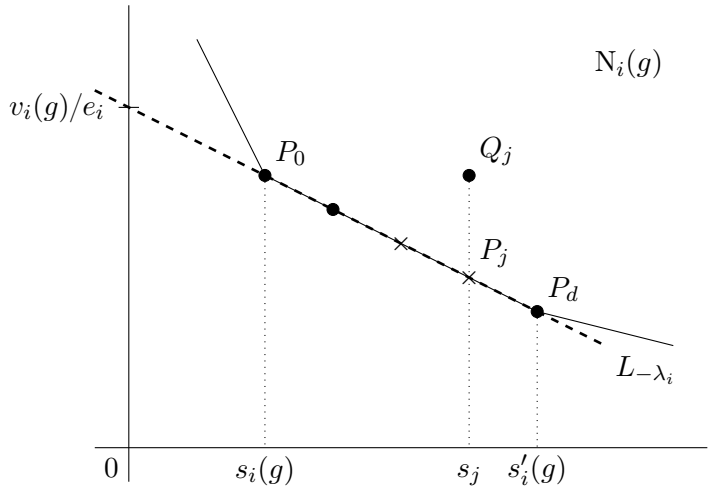


Figure 2.3: Computation of $\mathrm{R}_i(g)$ for a non-zero polynomial $g \in K[x]$.

Consider the following residual coefficient:

$$c_j := \begin{cases} 0, & \text{if } Q_j \text{ lies above } \mathrm{N}_i(g), \\ z_{i-1}^{t_{i-1}(a_{s_j})}\, \mathrm{R}_{i-1}(a_{s_j})(z_{i-1}) \in \mathbb{F}_i^*, & \text{if } Q_j \text{ lies on } \mathrm{N}_i(g), \end{cases} \tag{2.4}$$

where for any $a \in K[x]$ we define $t_0(a) = 0$ and $t_k(a) = (s_k(a) - \ell_k v_k(a))/e_k$ if $k > 0$. Then, we define

$$\mathrm{R}_i(g)(y) := R_{v_{i-1},\phi_i,\lambda_i}(g)(y) = c_0 + c_1 y + \cdots + c_d y^d \in \mathbb{F}_i[y].$$

Since $c_0 c_d \neq 0$, the polynomial $\mathrm{R}_i(g)$ has degree d and it is never divisible by y.

Remark. Note that a type $\mathtt{t}$ of order i determines the numerical values

$$\begin{aligned} m_{i+1} &:= e_i f_i m_i, \\ V_{i+1} &:= e_i f_i (e_i V_i + h_i), \end{aligned}$$

of any enlargement of $\mathtt{t}$ to a type of order $i+1$.

In fact, any representative ϕ of $\mathtt{t}$ has degree $m_{i+1} := e_i f_i m_i$ by definition. Also, Theorem 2.6 shows that $\mathrm{N}_i(\phi)$ is one-sided of slope $-\lambda_i$ and length $e_i f_i = m_{i+1}/m_i$. Hence, $V_{i+1} := v_i(\phi) = e_i(e_i f_i V_i + f_i h_i)$ by (2.3).

Definition 2.4. *Let $\mathfrak{t}$ be a type of order r over (K, v).*

The truncation $\mathrm{Trunc}_j(\mathfrak{t})$ *of $\mathfrak{t}$ at level j, is the type of order j obtained from $\mathfrak{t}$ by dropping all levels higher than j.*

Definition 2.5. *Let $\mathfrak{t}$ be a type of order r. For any $g \in K[x]$ we specify*

$$\mathrm{ord}_{\mathfrak{t}}(g) := \mathrm{ord}_{\psi_r} \mathrm{R}_r(g) \text{ in } \mathbb{F}_r[y].$$

This function is multiplicative: $\mathrm{ord}_{\mathfrak{t}}(gh) = \mathrm{ord}_{\mathfrak{t}}(g) + \mathrm{ord}_{\mathfrak{t}}(h)$ *for all $g, h \in K[x]$.*

If $\mathrm{ord}_{\mathfrak{t}}(g) > 0$*, we say that* $\mathfrak{t}$ divides g*, and we write* $\mathfrak{t} \mid g$.

Let $F \in \mathbb{P}$ be a prime polynomial with respect to v as in Definition 2.1 and $\theta \in \overline{K}_v$ a root of F. The next result explains the use of the term "type". All prime polynomials divisible by a type $\mathfrak{t}$ share certain common features described by the parameters supported by $\mathfrak{t}$. It therefore makes sense to say that these polynomials "are of type $\mathfrak{t}$".

Theorem 2.6. *Let $\mathfrak{t}$ be a type of order r such that $\mathfrak{t} \mid F$. Then, for all $1 \leqslant i \leqslant r$:*

- $\mathrm{N}_i(F)$ *is one-sided of slope $-\lambda_i$ and* $v\left(\phi_i(\theta)\right) = \dfrac{V_i + \lambda_i}{e_1 \dots e_{i-1}}$
- $\deg(F) = \deg(\phi_i) \cdot \ell(\mathrm{N}_i(F))$
- $\mathrm{R}_i(F) = \psi_i^a$, $a = \ell(\mathrm{N}_i(F))/e_i f_i$

Moreover, if $\phi \in \mathrm{Rep}(\mathfrak{t})$ is a representative of $\mathfrak{t}$ such that $\phi \neq F$, then

- $\mathrm{N}_{v_r,\phi}(F)$ *is one-sided of slope $-\lambda$, such that* $v\left(\phi(\theta)\right) = \dfrac{V_{r+1} + \lambda}{e_1 \dots e_r}$
- $\deg(F) = \deg(\phi) \cdot \ell(\mathrm{N}_{v_r,\phi}(F))$
- $\mathrm{R}_{v_r,\phi,\lambda}(F) = \psi^a$, *for $\psi \in \mathbb{F}_{r+1}[y]$, monic irreducible, and*

$$a = \ell(\mathrm{N}_{v_r,\phi}(F))/(e_\lambda \deg(\psi)),$$

where e_λ is the least positive denominator of λ.

Additionally, the type $\mathfrak{t}' = (\mathfrak{t}; (\phi, \lambda, \psi))$ divides F.

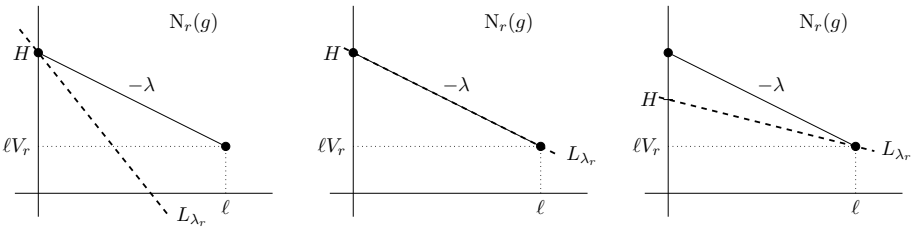


Figure 2.4: Three possible positions for line L_{λ_r}.

Theorem 2.7. *Let $\mathfrak{t}$ be a type of order r such that $\mathfrak{t} \mid F$ and select any $g \in K_v[x]$. Let L_{λ_r} be the line of slope $-\lambda_r$ that first touches $\mathrm{N}_r(g)$ from below. Then,*

$$v\left(g(\theta)\right) \geqslant \frac{H}{e_1 \cdots e_{r-1}},$$

where $(0, H)$ is the point where L_{λ_r} crosses the vertical axis. Equality occurs if and only if $\mathfrak{t} \nmid g$.

We can apply Theorem 2.7 to some prime polynomial $g \in \mathbb{P}$, such that $\mathrm{Trunc}_{r-1}(\mathfrak{t}) \mid g$, but $\mathfrak{t} \nmid g$. Then, Theorem 2.6 and (2.2) show that $\mathrm{R}_{r-1}(g) = \psi_{r-1}^{\ell}$ with $\ell = \ell(\mathrm{N}_r^-(g)) = \deg g / \deg \phi_r$. Hence, a look at Figure 2.4 shows that,

$$v\left(g(\theta)\right) = \frac{H}{e_1 \cdots e_{r-1}} = \frac{\deg g}{\deg \phi_r} \cdot \frac{V_r + \min\{\lambda, \lambda_r\}}{e_1 \cdots e_{r-1}}, \tag{2.5}$$

where $-\lambda$ is the slope of $\mathrm{N}_r^-(g)$, according to Theorem 2.6.

Construction of types

In this paragraph, we recall the existence of a concrete procedure to construct a representative of a type.

Proposition 2.8. *Let $\mathfrak{t}$ be a type of order $r \geqslant 1$. Let $\varphi \in \mathbb{F}_r[y]$ be a non-zero polynomial of degree less than f_r and let $b \geqslant V_{r+1}$ be an integer. Then,*

we may construct a polynomial $g \in \mathcal{O}[x]$ *such that*

$$\deg g < m_{r+1}, \qquad v_r(g) = b, \qquad y^{\lfloor s_r(g)/e_r \rfloor} \mathrm{R}_r(g) = \varphi.$$

In order to construct a representative ϕ of $\mathbb{t}$ we may apply the procedure given in Proposition 2.8 to construct a polynomial $g \in \mathcal{O}[x]$ such that $\mathrm{R}_r(g) = \psi_r - y^{f_r}$, and take $\phi = \phi_r^{e_r f_r} + g$. In this way, we may *efficiently* construct representatives of types.

We denote by $\mathrm{Rep}(\mathbb{t})$ the set of all representatives of a type $\mathbb{t}$. By Proposition 2.8 this is a nonempty subset of $\mathbb{P}$.

Since the level data λ_i, ψ_i are arbitrarily chosen, we may construct types of prescribed order r and prescribed numerical data h_i, e_i, f_i for $1 \leqslant i \leqslant r$. In other words, we may construct local extensions of K_v with prescribed arithmetic properties.

2.3 Types parameterise Okutsu classes of prime polynomials

2.3.1 Equivalence of types

Definition 2.9. *Let* $\mathbb{t}, \mathbb{t}'$ *be two types of order* r, r'*, respectively. We say that* $\mathbb{t}$ *and* $\mathbb{t}'$ *are "equivalent" if they have the same set of representatives:* $\mathrm{Rep}(\mathbb{t}) = \mathrm{Rep}(\mathbb{t}')$*. We write* $\mathbb{t} \equiv \mathbb{t}'$ *in this case.*

This is clearly an equivalence relation. The properties shared by equivalent types are studied in full detail in [Nar14]. Let us just mention that

$$\mathbb{t} \equiv \mathbb{t}' \implies v_{\mathbb{t}} = v_{\mathbb{t}'} \quad \text{and} \quad \mathrm{ord}_{\mathbb{t}} = \mathrm{ord}_{\mathbb{t}'},$$

where $v_{\mathbb{t}}, v_{\mathbb{t}'}$ are the valuations of the last level of the respective types.

Definition 2.10. *Let* $\mathbb{t}$ *be a type of order* $r \geqslant 0$*. We say that* $\mathbb{t}$ *is* optimal *if* $m_1 < \cdots < m_r$*. We say that* $\mathbb{t}$ *is* strongly optimal *if in addition to being optimal,* $m_r < m_{r+1}$*.*

We agree that a type of order zero is strongly optimal.

Denote by $\mathcal{T}$ the set of all types over (K, v) and let $\mathcal{T}^{\mathrm{str}} \subseteq \mathcal{T}$ be the subset of all strongly optimal types.

Proposition 2.11. *Consider two strongly optimal types:*

$$\begin{aligned} \mathbb{t} &= (\psi_0; (\phi_1, \lambda_1, \psi_1); \ldots; (\phi_r, \lambda_r, \psi_r)), \\ \mathbb{t}^* &= (\psi_0^*; (\phi_1^*, \lambda_1^*, \psi_1^*); \ldots; (\phi_{r^*}^*, \lambda_{r^*}^*, \psi_{r^*}^*)). \end{aligned}$$

Then $\mathbb{t}$ and $\mathbb{t}^*$ are equivalent if and only if they satisfy the following conditions:

(i) $r = r^*$.

(ii) $\phi_i^* = \phi_i + a_i$, $\deg a_i < m_i$, $v_i(a_i) \geqslant v_i(\phi_i)$, *for all* $1 \leqslant i \leqslant r$.

(iii) $\lambda_i^* = \lambda_i$ *for all* $1 \leqslant i \leqslant r$.

(iv) $\psi_i^*(y) = \psi_i(y - \eta_i)$ *for all* $0 \leqslant i \leqslant r$ *with* η_i *defined as follows:*

$$\eta_i = \begin{cases} 0, & \text{if } v_i(a_i) > v_i(\phi_i), \\ \mathrm{R}_i(a_i) \in \mathbb{F}_i^*, & \text{if } v_i(a_i) = v_i(\phi_i). \end{cases} \tag{2.6}$$

We denote by $\mathbb{T} = \mathcal{T}^{\mathrm{str}}/\equiv$ the quotient set and we write $[\mathbb{t}] \subseteq \mathcal{T}^{\mathrm{str}}$ for the class of all types equivalent to $\mathbb{t}$.

We can link these classes of equivalent types to the Okutsu classes of prime polynomials presented in Section 2.1.

Theorem 2.12. *There is a canonical bijection between the set of equivalence classes of strongly optimal types and the set of Okutsu classes of prime polynomials:*

$$\mathbb{T} \longrightarrow (\mathbb{P}/\approx),$$

that sends the equivalence class $[\mathbb{t}]$ of a strongly optimal type $\mathbb{t}$ to the Okutsu class of any representative of $\mathbb{t}$.

We denote by $\mathrm{om} : (\mathbb{P}/\approx) \to \mathbb{T}$ the inverse of the above map.

2.3.2 MacLane-Okutsu invariants of prime polynomials

Let $F \in \mathbb{P}$ be a prime polynomial. Let $\mathtt{t}$ be any strongly optimal type in the class $\mathrm{om}([F])$.

The order r of $\mathtt{t}$ is called the *Okutsu depth* of F.

The *basic MacLane-Okutsu invariants* of F are the following positive integers supported by $\mathtt{t}$:

$$e_1, \dots, e_r, \quad h_1, \dots, h_r, \quad f_0, \dots, f_r. \tag{2.7}$$

Recall that h_i, e_i are coprime such that $h_i/e_i = \lambda_i$ for all $1 \leqslant i \leqslant r$, and $f_i = \deg \psi_i$ for all $0 \leqslant i \leqslant r$.

By Proposition 2.11, these invariants are shared by all strongly optimal types in the equivalence class of $\mathtt{t}$.

There are a number of further invariants of F, which can be constructed from these invariants.

Definition 2.13. *A* MacLane-Okutsu invariant *of F is a rational number that depends only on the basic invariants presented in* (2.7).

For simplicity we shall refer to these invariants as *OM invariants* of F. Some examples of OM invariants are,

$$\begin{aligned}
m_i &:= \deg \phi_i = e_{i-1} f_{i-1} m_{i-1}, \\
V_i &:= v_{i-1}(\phi_i) = e_{i-1} f_{i-1} (e_{i-1} V_{i-1} + h_{i-1}). \\
e(F) &:= e(K_F/K_v) = e_1 \cdots e_r, \text{ the ramification index of } K_F/K_v, \\
f(F) &:= f(K_F/K_v) = f_0, \dots, f_r, \text{ the residual degree of } K_F/K_v, \\
\delta_0(F) &:= V_{r+1}/e(F), \\
\mathrm{cap}(F) &:= \max\{v(g(\theta)) : g \in \mathcal{O}[x] \text{ monic}, \deg g < \deg F\} \\
&= \delta_0(F) - \sum_{j=1}^{r} h_j/(e_1 \cdots e_j), \\
\mathrm{ind}(F) &:= \mathrm{length}_{\mathcal{O}_v}(\mathcal{O}_F/\mathcal{O}_v[\theta]) = n\left(\mathrm{cap}(F) - 1 + e(F)^{-1}\right)/2, \\
\mathfrak{f}(F) &:= \min\left\{\delta \in \mathbb{Z}_{\geqslant 0} : (\mathfrak{m}_F)^\delta \subset \mathcal{O}_v[\theta]\right\} = 2\,\mathrm{ind}(F)/f(F), \\
\exp(F) &= \min\left\{\delta \in \mathbb{Z}_{\geqslant 0} : \mathfrak{m}^\delta \mathcal{O}_F \subset \mathcal{O}_v[\theta]\right\} = \lfloor \mathrm{cap}(F) \rfloor.
\end{aligned}$$

The Okutsu bound $\delta_0(F)$ was defined in Definition 2.2. The final four operators are the *capacity*, *index*, *conductor*, and *exponent* of F respectively.

2.3.3 Tree structure on the set of types

Let us introduce a tree structure on the set $\mathcal{T}$ of types. Given two types $\mathbb{t}, \mathbb{t}' \in \mathcal{T}$, there is an oriented edge $\mathbb{t}' \to \mathbb{t}$ if and only if $\mathbb{t}' = \mathrm{Trunc}_{r-1}(\mathbb{t})$, where r is the order of $\mathbb{t}$. Thus, we have a unique path of length equal to the order of $\mathbb{t}$:

$$\mathrm{Trunc}_0(\mathbb{t}) \longrightarrow \mathrm{Trunc}_1(\mathbb{t}) \longrightarrow \cdots \longrightarrow \mathrm{Trunc}_{r-1}(\mathbb{t}) \longrightarrow \mathbb{t}. \tag{2.8}$$

The root nodes are the types of order zero. Thus, the connected components of $\mathcal{T}$ are the subtrees $\mathcal{T}_\varphi$ of all types $\mathbb{t}$ with $\mathrm{Trunc}_0(\mathbb{t}) = (\varphi)$, for φ running on the set $\mathbb{P}(\mathbb{F}[y])$ of all monic irreducible polynomials in $\mathbb{F}[y]$.

The branches of a type $\mathbb{t}$ of order r are parametrised by triplets (ϕ, λ, ψ), where ϕ is a representative of $\mathbb{t}$, λ is a positive rational number and $\psi \in \mathbb{F}_{r+1}[y]$ is a monic irreducible polynomial such that $\psi \neq y$. Such a triplet determines an edge $\mathbb{t} \to \mathbb{t}^*$, where $\mathbb{t}^* = (\mathbb{t}; (\phi, \lambda, \psi))$ is the type obtained by enlarging $\mathbb{t}$ with data (ϕ, λ, ψ) at the $(r+1)$-th level.

Suppose $\mathbb{t} = (\psi_0; (\phi_1, \lambda_1, \psi_1); \ldots; (\phi_r, \lambda_r, \psi_r))$. In practice, when we represent a path like (2.8) we omit the labels of the vertices which are not root nodes and we label the edges with the level data as follows:

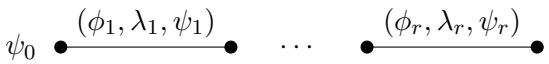


Figure 2.5: Visual path representation of a type $\mathbb{t}$ of order r.

Also, since the direction of the edges is self-evident, we draw them as lines instead of vectors. We recover the real path (2.8) from its practical representation (Figure 2.5) by attaching to each vertex of the path the type obtained by gathering all level data from the previous edges.

All truncates of a strongly optimal type $\mathbb{t}$ are also strongly optimal, hence the subset $\mathcal{T}^{\mathrm{str}} \subseteq \mathcal{T}$ is a full subtree of $\mathcal{T}$. Also, if $\mathbb{t} \equiv \mathbb{t}^*$ are strongly optimal, then $\mathrm{Trunc}_i(\mathbb{t}) \equiv \mathrm{Trunc}_i(\mathbb{t}^*)$ for all $0 \leqslant i \leqslant r$. Therefore, the

tree structure on $\mathcal{T}^{\mathrm{str}}$ induces a natural tree structure on the quotient set $\mathbb{T} = \mathcal{T}^{\mathrm{str}}/\equiv$.

Since the equivalence relation $\equiv$ on $\mathcal{T}^{\mathrm{str}}$ only identifies vertices of the same order, a path of length r in $\mathcal{T}^{\mathrm{str}}$ determines a path of length r in $\mathbb{T}$.

For types of order zero, $\mathbb{t} \equiv \mathbb{t}^*$ holds only for $\mathbb{t} = \mathbb{t}^*$; thus, the root nodes of $\mathbb{T}$ are in 1-1 correspondence with the set $\mathbb{P}(\mathbb{F}[y])$.

Index of coincidence of types

Owing to the tree structure on the set of types, we are able to introduce a measure of similarity on types.

Definition 2.14. *Let $\mathbb{t}, \mathbb{t}' \in T$ be two vertices of a tree of types $T \subseteq \mathcal{T}$ of order r and r' respectively. The* index of coincidence*,*

$$i(\mathbb{t}, \mathbb{t}') = \min\left\{0 \leqslant \ell \leqslant \min\left\{r, r'\right\} : \mathrm{Trunc}_\ell(\mathbb{t}) \neq \mathrm{Trunc}_\ell(\mathbb{t}')\right\},$$

is the lowest index ℓ such that truncation of both types at ℓ is not *equal.*

Two vertices with index of coincidence greater than 0, share the same root node (ψ_0). However, if two types have an index of coincidence of 0, that indicates that they belong to distinct connected trees.

2.4 OM factorisation of polynomials

2.4.1 OM representations of prime polynomials

Let $F \in \mathbb{P}$ be a prime polynomial and let $\mathbb{t}_F$ be a strongly optimal type of order r such that $\mathrm{om}([F]) = [\mathbb{t}_F]$. Let us denote by $\mathbb{T}(F) \subset \mathbb{T}$ the unibranch tree determined by the path joining $[\mathbb{t}_F]$ with its root node in $\mathbb{T}$.

For any polynomial $\phi \in [F] \cap \mathcal{O}[x]$ in the Okutsu class of F and having coefficients in $\mathcal{O}$, the pair $[\mathbb{t}_F, \phi]$ is called an *OM representation* of F. If $\phi = F$ we say that the OM representation is *exact*.

Definition 2.15. *The* quality *of ϕ as an approximation to F is defined as the rational number $v(\phi(\theta))$.*

The polynomial ϕ is a "sufficiently good" approximation to F for many purposes. The discrete data contained in the type $\mathbb{t}_F$ is a kind of DNA sequence common to all individuals in the Okutsu class $[F]$, and many properties of F and the extension K_F/K_v are described by this genetic data, as we have seen in Section 2.3.2. In a more classical approach the computation of these invariants has to be derived from extra routines that may be computationally expensive. Further, the genetic information of F is helpful in the construction of approximations with a prescribed quality and, more generally, it leads to a new design of fast routines carrying out basic arithmetic tasks in number fields and function fields.

2.4.2 OM representation of a square-free polynomial

Let $f = F_1 \cdots F_t$ be the prime factorisation in $\mathcal{O}_v[x]$ of a square-free monic polynomial $f \in \mathcal{O}[x]$. For each $1 \leqslant j \leqslant t$, let r_j be the Okutsu depth of F_j and $\theta_j \in \overline{K}_v$ a root of F_j.

Definition 2.16. *The* genomic tree *of f is the finite tree $\mathbb{T}(f) := \mathbb{T}(F_1) \cup \cdots \cup \mathbb{T}(F_t) \subset \mathbb{T}$.*

Let us extend the notion of Okutsu equivalence in section 2.1 to non-irreducible polynomials

Definition 2.17. *Let $g, h \in \mathcal{O}[x]$ be monic polynomials with prime factorisations $g = G_1 \cdots G_s$, $h = H_1 \cdots H_{s'}$ in $\mathcal{O}_v[x]$. We say that g and h are* Okutsu equivalent*, and we write $g \approx h$, if $s = s'$ and $G_j \approx H_j$ for all $1 \leqslant j \leqslant s$, up to ordering.*

An expression of the form, $g \approx P_1 \cdots P_s$, with $P_1, \ldots, P_s \in \mathbb{P} \cap \mathcal{O}[x]$ is called an Okutsu factorisation *of g.*

Clearly, every $g \in \mathcal{O}[x]$ admits a unique (up to $\approx$) Okutsu factorisation. However, we need a stronger concept for our purposes. For instance, if all factors of g are Okutsu equivalent to P, then $g \approx P^s$ is an Okutsu factorisation of g which is unable to distinguish the true prime factors of g.

Definition 2.18. *We say that $P_j \in [F_j]$ is a* Montes approximation to F_j as a factor of f *if*

$$v\left(P_j(\theta_j)\right) > v\left(P_j(\theta_k)\right), \qquad \forall\ 1 \leqslant k \neq j \leqslant t.$$

An OM factorisation *of f is an Okutsu factorisation $f \approx P_1 \cdots P_t$ such that each approximate factor P_j is a Montes approximation to F_j as a factor of f.*

If $f \approx P_1 \cdots P_t$ is an OM factorisation of f, the types $\mathtt{t}_{F_j}$ may be extended to types

$$\mathtt{t}_j := \begin{cases} \left(\mathtt{t}_{F_j}; (P_j, \lambda_{r_j+1,j}, \psi_{r_j+1,j})\right), & P_j \neq F_j, \\ \left(\mathtt{t}_{F_j}; (P_j, \infty, -)\right), & P_j = F_j, \end{cases} \tag{2.9}$$

satisfying

$$\operatorname{ord}_{\mathtt{t}_j}(F_j) = 1, \quad \mathtt{t}_j \nmid F_k, \qquad \text{for all } 1 \leqslant k \neq j \leqslant t.$$

By Theorem 2.6, the quality of the approximations $P_j \approx F_j$ is given by the formula:

$$v\left(P_j(\theta_j)\right) = \delta_0(F_j) + \frac{\lambda_{r_j+1,j}}{e(F_j)}.$$

If $P_j \nmid f$, the slope $\lambda_{r_j+1,j}$ is an integer which may be computed as the largest slope (in absolute value) of $\mathrm{N}^-_{r_j+1}(f) = \mathrm{N}^-_{v_{r_j},P_j}(f)$. This slope corresponds to a side whose end points have abscissas 0 and 1 (see Figure 2.8). Hence, $\mathrm{R}_{r_j+1}(f) := \mathrm{R}_{v_{r_j},P_j,\lambda_{r_j+1,j}}(f)$ has degree one and $\psi_{r_j+1,j}$ is equal to $\mathrm{R}_{r_j+1}(f)$ divided by its leading coefficient.

The types $\mathtt{t}_j$ are optimal, but not strongly optimal because $e_{r_j+1,j} = f_{r_j+1,j} = 1$, so that $m_{r_j+2,j} = m_{r_j+1,j} = \deg F_j$.

Definition 2.19. *Let $T(f) \subset \mathcal{T}^{\mathrm{str}}$ be a faithful pre-image of the genomic tree of f; that is, $T(f)$ maps to $\mathbb{T}(f)$ under the quotient map $\mathcal{T}^{\mathrm{str}} \to \mathbb{T}$, and the vertices of $T(f)$ are pairwise inequivalent.*

An OM representation *of f is the tree obtained by enlarging $T(f)$ with*

the t new vertices $\mathbb{t}_j$ and edges $\mathbb{t}_{F_j} \to \mathbb{t}_j$ determined by the choice of an OM factorisation of f.

Thus, an OM representation of f gathers the information provided by a family of OM representations of the prime factors. The information added by the choice of an OM factorisation of f allows us to distinguish the different prime factors.

The leaves of an OM representation of f are in 1-1 correspondence with the prime factors of f, whereas the root nodes are in 1-1 correspondence with the monic irreducible factors of $\overline{f}$ in $\mathbb{F}[y]$ (see Figure 2.6).

For instance, in the example presented in Figure 2.6, $f = F_1F_2F_3F_4$ has four prime factors and $\overline{F_1}, \overline{F_2}, \overline{F_3}, \overline{F_4}$ are a power of the same prime polynomial in $\mathbb{F}[y]$. The vertex $\mathbb{t}_{F_1}$ has order 0, $\mathbb{t}_{F_2} = \mathbb{t}_{F_3}$ have order 3 and $\mathbb{t}_{F_4}$ has order 5.

We represent the edges $\mathbb{t}_{F_j} \to \mathbb{t}_j$ with dotted lines to emphasise that the leaves $\mathbb{t}_j$ are not strongly optimal types.

In general, the vertices $\mathbb{t}_{F_i}$ are not necessarily leaves of the tree $T(f)$. It may happen that $\mathbb{t}_{F_i}$ coincides with a vertex in the path joining $\mathbb{t}_{F_j}$ with its root node for some $j \neq i$. Thus, the leaves of an OM representation of f may sprout from arbitrary vertices in $T(f)$.

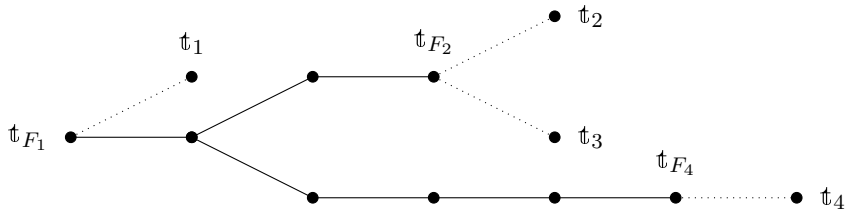


Figure 2.6: OM representation of $f = F_1 \cdots F_4$, with $F_2 \approx F_3$.

The factors F_2, F_3 have been distinguished thanks to a certain (unspecified) OM factorisation of f.

Definition 2.20. *We say that a leaf of an OM representation of f is* isolated *if the previous node has only one branch. For instance, in Figure 2.6 the leaf corresponding to F_4 is isolated, while the other three leaves are not. The $(r_j + 1)$-th Newton polygons for isolated and non-isolated leaves are shown in Figure 2.8.*

2.5 The Montes algorithm

In this section, we describe the algorithm for OM factorisation developed by Montes in 1999 [Mon99]. The algorithm was inspired by the ideas of Ore [Ore23][Ore25] and MacLane [Mac36a][Mac36b].

The aim of the Montes algorithm is the computation of an OM representation of a given square-free polynomial $f \in \mathcal{O}[x]$. Let

$$\mathcal{F} = \{F_1, \dots, F_t\},$$

be the set of prime factors of f in $\mathcal{O}_v[x]$. For any type $\mathfrak{t}$, we denote

$$\mathcal{F}_{\mathfrak{t}} = \{F \in \mathcal{F} : \mathfrak{t} \mid F\} \subseteq \mathcal{F}.$$

Since $\operatorname{ord}_{\mathfrak{t}}(f) = \sum_{1 \leqslant j \leqslant t} \operatorname{ord}_{\mathfrak{t}}(F_j)$, the set $\mathcal{F}_{\mathfrak{t}}$ is empty if and only if $\mathfrak{t} \nmid f$. Also, if $\operatorname{ord}_{\mathfrak{t}}(f) = 1$, then there is an index j such that $\operatorname{ord}_{\mathfrak{t}}(F_j) = 1$ and $\operatorname{ord}_{\mathfrak{t}}(F_k) = 0$ for all $k \neq j$; thus, $\mathcal{F}_{\mathfrak{t}} = \{F_j\}$ is a one-element subset in this case.

The algorithm generates a tree of types $\mathfrak{T}$ which constitutes an OM representation of f. The leaf types $\mathfrak{t}_1, \dots, \mathfrak{t}_t$ of this tree have the property that $\mathcal{F}_{\mathfrak{t}_i} = \{F_i\}$ for all $1 \leqslant i \leqslant t$.

2.5.1 Non-optimised Montes algorithm

Definition 2.21. *A subtree $T \subset \mathcal{T}$ is called* coherent *if for every node $\mathfrak{t} \in T$, all edges with left end point $\mathfrak{t}$ have the same ϕ-polynomial. An example can be seen in Figure 2.7.*

There is a non-optimised version of the Montes algorithm, which outputs a coherent tree of types.

Let us briefly describe this version of the algorithm. Initially, $\overline{f}$ is factorised in $\mathbb{F}[y]$. For each monic irreducible factor φ of $\overline{f}$, a triplet $(\mathfrak{t}, \phi, \omega)$ is considered, where $\mathfrak{t} = (\varphi)$ is the type of order 0 determined by φ, ϕ is a representative of $\mathfrak{t}$ (that is, a monic lift of φ to $\mathcal{O}[x]$), and $\omega = \operatorname{ord}_{\varphi}(\overline{f})$. All these triplets $(\mathfrak{t}, \phi, \omega)$ are stored in a stack.

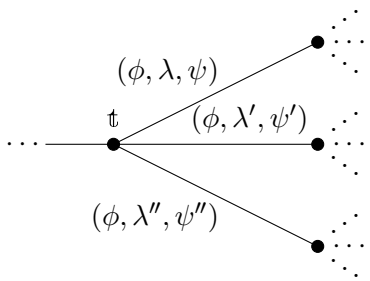


Figure 2.7: A segment of a coherent tree T.

Along the execution of the algorithm the stack always contains triplets $(\mathfrak{t}, \phi, \omega)$, where $\mathfrak{t} \mid f$, ϕ is a representative of $\mathfrak{t}$ and $\omega = \mathrm{ord}_{\mathfrak{t}}(f)$. The main loop of the algorithm takes such a triplet and attaches to the type $\mathfrak{t}$ one or more branches $\mathfrak{t}_{\lambda,\psi} := (\mathfrak{t}; (\phi, \lambda, \psi))$ of $\mathfrak{t}$ such that $\mathfrak{t}_{\lambda,\psi} \mid f$ and the set $\mathcal{F}_{\mathfrak{t}}$ splits as the disjoint union:

$$\mathcal{F}_{\mathfrak{t}} = \bigcup_{(\lambda,\psi)} \mathcal{F}_{\mathfrak{t}_{\lambda,\psi}}.$$

Note that all these branches have the same ϕ-polynomial. The pairs (λ, ψ) are considered as follows,

- $-\lambda$ runs on the slopes of $\mathrm{N}^{\omega}_{v_{\mathfrak{t}},\phi}(f)$, the piece of $\mathrm{N}_{v_{\mathfrak{t}},\phi}(f)$ contained in $[0, \omega] \times \mathbb{R}$.
- ψ runs on the prime factors of $\mathrm{R}_{v_{\mathfrak{t}},\phi,\lambda}(f)$ in $\mathbb{F}_{\mathfrak{t}}[y]$.

Let $\phi_{\lambda,\psi} \in \mathcal{O}[x]$ be a representative of $\mathfrak{t}_{\lambda,\psi}$ and take

$$\omega_{\lambda,\psi} = \mathrm{ord}_{\psi}(\mathrm{R}_{v_{\mathfrak{t}},\phi,\lambda}(f)) = \mathrm{ord}_{\mathfrak{t}_{\lambda,\psi}}(f).$$

If this positive integer is equal to one, then $\mathfrak{t}_{\lambda,\psi}$ singles out one of the prime factors of f in $\mathcal{O}_v[x]$. In this case, we add a final level to $\mathfrak{t}$:

$$\tilde{\mathfrak{t}} = (\mathfrak{t}; (\phi_{\lambda,\psi}, -, -)),$$

as in (2.9) and we store the type $\tilde{\mathfrak{t}}$ in a list of "output types". On the other hand, if $\omega_{\lambda,\psi} > 1$, then the triplet $(\mathfrak{t}_{\lambda,\psi}, \phi_{\lambda,\psi}, \omega_{\lambda,\psi})$ is pushed back onto the stack to bare further branching in future iterations of the main loop.

After a finite number of iterations of this process, the algorithm outputs a list $\mathbb{t}_1, \ldots, \mathbb{t}_t$ of types parametrising the prime factors of f in $\mathcal{O}_v[x]$.

This describes a kind of "non-optimised" Montes algorithm, in which the output tree having the types $\mathbb{t}_1, \ldots, \mathbb{t}_t$ as leaves is coherent. It is non-optimised because the types $\mathbb{t}_{\lambda,\psi}$ may not be optimal. In fact, if $\lambda \in \mathbb{Z}$ and $\deg \psi = 1$, we have

$$\deg \phi_{\lambda,\psi} = e_\lambda \cdot \deg \psi \cdot \deg \phi = \deg \phi,$$

where e_λ is the positive denominator of λ. Hence, the type $\mathbb{t}_{\lambda,\psi}$ is not strongly optimal and its branches may even cease to be optimal. We must avoid this situation, because the numerical data attached to the types will not be intrinsic data of the prime factor of f.

For this reason, we are interested in an optimised version of the Montes algorithm, which will ensure that the types it works with will be strongly optimal and the numerical data attached to them will be the intrinsic genetic data of the prime factors of f.

2.5.2 Optimised Montes algorithm

The optimised version of the Montes algorithm includes a "refinement procedure" which ensures that it only stores strongly optimal types (except for the leaves of the output tree) and yields an OM representation of f. However, a price must be paid; the output tree of OM representatives is no longer coherent.

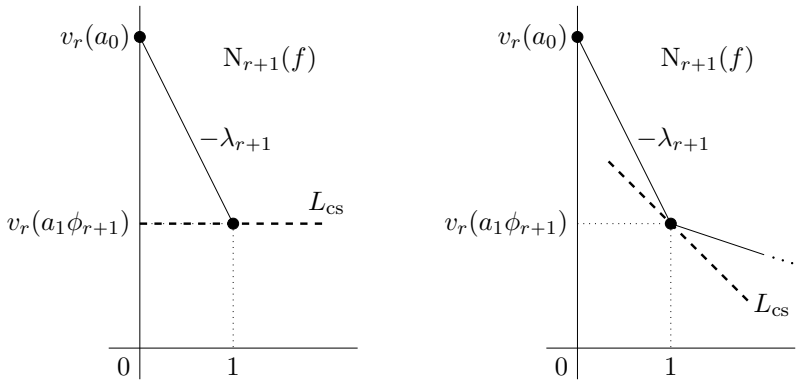
Let us describe the optimised Montes algorithm. The stack stores triplets $(\mathbb{t}, \phi, \omega)$ where $\mathbb{t}$ is a strongly optimal type dividing f, ϕ is a representative of $\mathbb{t}$ and $\omega = \operatorname{ord}_{\mathbb{t}}(f)$ is a positive integer. Initially, the stack stores the triplets determined by the irreducible factors of $\overline{f}$, as in the non-optimised algorithm.

If the main loop finds a "bad" branch (ϕ, λ, ψ) with $\lambda \in \mathbb{Z}$ and $\deg \psi = 1$, then it simply drops this level and pushes the triplet $(\mathbb{t}, \phi_{\lambda,\psi}, \omega_{\lambda,\psi})$ onto the stack instead of $(\mathbb{t}_{\lambda,\psi}, \phi_{\lambda,\psi}, \omega_{\lambda,\psi})$ as we would do in the non-optimised algorithm. This is called a *refinement step*.

The point is that the branching obtained by applying the main loop to both triplets determines the same partition of the set $\mathcal{F}_{\mathbb{t}_{\lambda,\psi}}$. Thus, the two algorithms, optimised and non-optimised, yield the same successive decomposition of the set $\mathcal{F}$ until all its elements are singled out.

The output types store a non-negative integer h_{cs}, called the *cutting slope*. If $\omega_{\lambda,\psi} = 1$ occurs during a refinement step, we take $h_{\mathrm{cs}} = \lambda$, and we take $h_{\mathrm{cs}} = 0$ otherwise. The output type is an isolated leaf of the output tree $\mathfrak{T}$ of OM representations if and only if $h_{\mathrm{cs}} = 0$.

As explained in Section 2.4.2, if the corresponding prime factor of f has Okutsu depth r, then the corresponding output type has order $r+1$ and the side of the maximal slope (in absolute size) of $\mathrm{N}_{r+1}(f)$ has slope $-\lambda_{r+1}$ and end points of abscissas 0 and 1. The line L_{cs} of slope $-h_{\mathrm{cs}}$, first touching $\mathrm{N}_{r+1}(f)$ from below, separates this side from the rest of the sides of $\mathrm{N}_{r+1}(f)$ (see Figure 2.8).



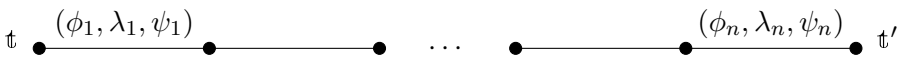
Isolated leaf ($h_{\mathrm{cs}} = 0$). Non-isolated leaf ($h_{\mathrm{cs}} > 0$).

Figure 2.8: Newton polygon $\mathrm{N}^-_{r+1}(f)$ determined by a leaf of order $r+1$ of an OM representation $\mathfrak{T}$ of f. The line L_{cs} has slope $-h_{\mathrm{cs}}$ and $f = \sum_{0 \leqslant s} a_s \phi_{r+1}^s$.

The advantage of the optimised algorithm is twofold: first, it outputs the genomic tree of f and all the canonical data it contains; second, it works with types of smaller order, which saves a lot of execution time due to the highly recursive nature of the routines for the computation of Newton polygons and residual polynomials.

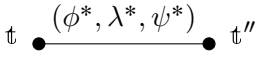
Nevertheless, it is worth keeping in mind the existence of the "non-optimised" coherent tree of types, produced by the non-optimised Montes

algorithm. The optimised tree of OM representations (the real output of the Montes algorithm) may be derived from the non-optimised tree by an iterative application of the following transformation. Any path,



(2.10)

in which all edges except for the final one are bad edges satisfying $\lambda_i \in \mathbb{Z}$, $\deg \psi_i = 1$ for $1 \leqslant i < n$, collapses into



(2.11)

where

$$\phi^* = \phi_n, \qquad \lambda^* = \lambda_1 + \cdots + \lambda_n, \qquad \psi^* = \psi_n.$$

The types $\mathfrak{t}'$ and $\mathfrak{t}''$ are equivalent.

For instance, consider the segment of a non-optimised tree shown in Figure 2.9, where the nodes whose previous edge is "bad" are represented by $\circ$ and the nodes following a "good" edge are represented by $\bullet$. Additionally, the "bad" edges are marked with a dotted line.

The same segment of the optimised tree derived from this coherent tree is presented in Figure 2.10.

where $\mathfrak{t}_i \equiv \mathfrak{t}'_i$ for all $1 \leqslant i \leqslant 5$, and

$$\begin{array}{lll}
\phi_1^* = \phi_1, & \lambda_1^* = \lambda + \lambda_1, & \psi_1^* = \psi_1, \\
\phi_2^* = \phi_2, & \lambda_2^* = \lambda + \lambda'_1 + \lambda_2, & \psi_2^* = \psi_2, \\
\phi_3^* = \phi_2, & \lambda_3^* = \lambda + \lambda'_1 + \lambda'_2, & \psi_3^* = \psi'_2, \\
\phi_4^* = \phi_4, & \lambda_4^* = \lambda + \lambda''_1 + \lambda_4, & \psi_4^* = \psi_4, \\
\phi_5^* = \phi, & \lambda_5^* = \lambda', & \psi_5^* = \psi'.
\end{array}$$

The optimised tree is no longer coherent, because amongst the five branches of $\mathfrak{t}$ we find four different ϕ-polynomials.

Remark 2.22. For a vertex $\mathfrak{m}$ in a tree, we denote by $\mathfrak{t}_\mathfrak{m}$ the type obtained

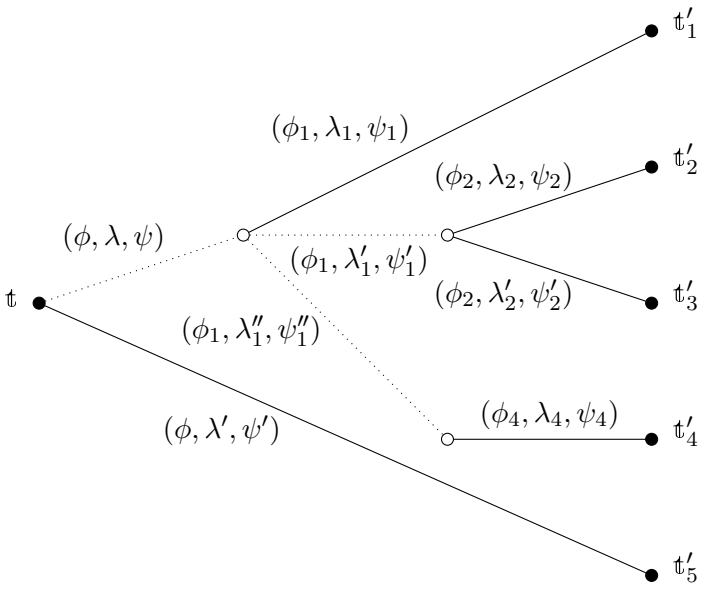


Figure 2.9: Segment of a non-optimised tree starting from $\mathbb{t}$.

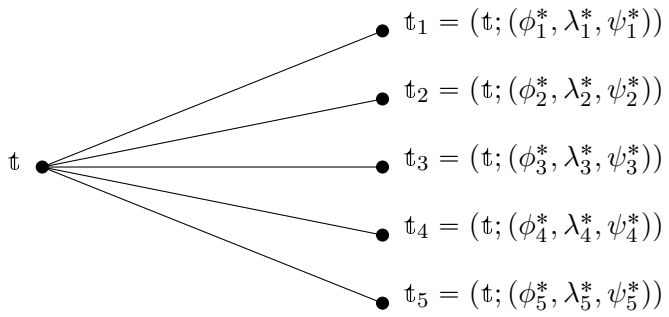


Figure 2.10: Segment of an optimised tree starting from $\mathbb{t}$.

by gathering all level data from the edges of the path joining $\mathbb{n}$ with its root node.

As a merely combinatorial structure, we may identify the set of vertices of the optimised tree $\mathfrak{T}^{\mathrm{op}}$ as a subset of the set of vertices of the non-optimised tree $\mathfrak{T}^{\mathrm{nop}}$. However, for the same vertex $\mathbb{n}$, the types $\mathbb{t}_{\mathbb{n}}$ for the optimised and non-optimised trees are different. But they are equivalent.

The existence of the non-optimised tree is useful in many situations. Let us see an example.

Notation. *For each type* $\mathtt{t} \in \mathfrak{T}$*, denote*

$$\mathbb{P}(\mathtt{t}) := \{F \in \mathbb{P} : \mathtt{t} \mid F\}.$$

Lemma 2.23. *Let* $\mathtt{t}, \mathtt{t}' \in \mathfrak{T}$ *be two different types of order* $r > r'$ *respectively such that* $\mathtt{t}'$ *is the truncation of* $\mathtt{t}$ *at level* r'*. That is to say,* $\mathtt{t}'$ *belongs to the path joining* $\mathtt{t}$ *with its root node. Then,* $\mathbb{P}(\mathtt{t}) \subsetneq \mathbb{P}(\mathtt{t}')$*.*

Proof. We have $\mathtt{t}' = \mathrm{Trunc}_{r'}(\mathtt{t})$. Take any $F \in \mathbb{P}(\mathtt{t})$; by Theorem 2.6, $\mathrm{R}_{r'}(F)$ is a power of $\psi_{r'}$, so that $\mathtt{t}' \mid F$ and $F \in \mathbb{P}(\mathtt{t}')$. This shows that $\mathbb{P}(\mathtt{t}) \subseteq \mathbb{P}(\mathtt{t}')$.

Finally, this inclusion is not an equality because $\mathtt{t}' \mid \phi_{r'+1}$, while $\mathtt{t} \nmid \phi_{r'+1}$. In fact, $\mathrm{R}_{r'+1}(\phi_{r'+1}) = 1$ implies that $\mathrm{Trunc}_{r'+1}(\mathtt{t}) \nmid \phi_{r'+1}$. □

Lemma 2.24. *Let* $T \subseteq \mathfrak{T}$ *be a coherent tree. Let* $\mathtt{t}, \mathtt{t}' \in T$ *such that neither of them is a truncation of the other. Then* $\mathbb{P}(\mathtt{t}) \cap \mathbb{P}(\mathtt{t}') = \varnothing$*.*

Proof. The statement is obvious if $\mathtt{t}$ and $\mathtt{t}'$ have different root nodes, because for all $F \in \mathbb{P}(\mathtt{t})$, the reduction $\overline{F}$ modulo $\mathfrak{m}$ is a power of the monic irreducible polynomial ψ_0 corresponding to the root node of $\mathtt{t}$.

Suppose that $\mathtt{t}, \mathtt{t}'$ have the same root node and let $\mathtt{t}_0$ be the greatest common node in the paths joining $\mathtt{t}, \mathtt{t}'$ with their root node. By Lemma 2.23 we may assume that $\mathtt{t}$ and $\mathtt{t}'$ are branches of $\mathtt{t}_0$, in other words, that $\mathtt{t}_0$ is the previous node of both $\mathtt{t}$ an $\mathtt{t}'$. By the coherence of T we have

$$\mathtt{t} = (\mathtt{t}_0; (\phi, \lambda, \psi)), \qquad \mathtt{t}' = (\mathtt{t}_0; (\phi, \lambda', \psi')),$$

where either $\lambda \neq \lambda'$ or $\lambda = \lambda', \psi \neq \psi'$.

Let r be the order of $\mathtt{t}_0$ and v_r its attached valuation. Now, for any $F \in \mathbb{P}(\mathtt{t})$, $F' \in \mathbb{P}(\mathtt{t}')$, Theorem 2.6 shows that $\mathrm{N}_{v_r,\phi}(F)$ and $\mathrm{N}_{v_r,\phi}(F')$ are one-sided of slopes $-\lambda$ and $-\lambda'$ respectively. Hence, $\lambda \neq \lambda'$ implies $F \neq F'$. On the other hand, if $\lambda = \lambda'$ then $\mathrm{R}_{\mathtt{t}_0,\phi,\lambda}(F) = \psi^a$ and $\mathrm{R}_{\mathtt{t}_0,\phi,\lambda}(F') = (\psi')^{a'}$ and this implies $F \neq F'$, because $\psi \neq \psi'$. □

This result may be false for arbitrary incoherent trees. However, Lemma 2.24 is valid for the OM representations of square-free polynomials.

Theorem 2.25. *Let $T \subseteq \mathfrak{T}$ be an OM representation of a monic squarefree polynomial $f \in \mathcal{O}[x]$. Let $\mathfrak{t}, \mathfrak{t}' \in T$ be two nodes such that neither of them is a truncation of the other. Then $\mathbb{P}(\mathfrak{t}) \cap \mathbb{P}(\mathfrak{t}) = \varnothing$. In particular, $\mathcal{F}_{\mathfrak{t}} \cap \mathcal{F}_{\mathfrak{t}'} = \varnothing$.*

Proof. Clearly, the nodes $\mathfrak{t}, \mathfrak{t}'$ are equivalent to two nodes of the non-optimised tree, neither of them belonging to the path joining the other type with its root node. Since the non-optimised tree is coherent, the statement is an immediate consequence of Lemma 2.24.

The final statement is a consequence of $\mathcal{F}_{\mathfrak{t}} = \mathbb{P}(\mathfrak{t}) \cap \mathcal{F}$. □

While the Montes algorithm only produces optimised trees of types, certain information is stored about the refinements that take place during the execution.

Consider the chain of refinements that take place between (2.10) and (2.11). During each refinement that provokes branching of a type, the intermediate ϕ and λ values are stored.

Let $\mathfrak{t}, \mathfrak{t}^* \in \mathfrak{T}$ be two strongly optimal types with index of coincidence $i(\mathfrak{t}, \mathfrak{t}^*) = \ell$. Then suppose that at level ℓ, each type has a list of refinements,

$$\begin{aligned} \mathrm{Ref}_\ell(\mathfrak{t}) &= \left[(\phi^{\mathfrak{t}}_{(1)}, \lambda^{\mathfrak{t}}_{(1)}, \psi^{\mathfrak{t}}_{(1)}), \dots, (\phi^{\mathfrak{t}}_{(k)}, \lambda^{\mathfrak{t}}_{(k)}, \psi^{\mathfrak{t}}_{(k)})\right], \\ \mathrm{Ref}_\ell(\mathfrak{t}^*) &= \left[(\phi^{\mathfrak{t}^*}_{(1)}, \lambda^{\mathfrak{t}^*}_{(1)}, \psi^{\mathfrak{t}^*}_{(1)}), \dots, (\phi^{\mathfrak{t}^*}_{(k')}, \lambda^{\mathfrak{t}^*}_{(k')}, \psi^{\mathfrak{t}^*}_{(k')})\right]. \end{aligned} \tag{2.12}$$

The final refinement in each list is the ℓ-th level of the types $\mathfrak{t}$ and $\mathfrak{t}^*$. This allows us to extend our index of coincidence from Definition 2.14 to a more precise indicator.

Definition 2.26. *The* minor index of coincidence $\hat{\imath}(\mathfrak{t}, \mathfrak{t}^*)$ *for two types $\mathfrak{t}, \mathfrak{t}^* \in \mathfrak{T}$, is the least index ℓ', such that for the refinement lists given in* (2.12)*,*

$$(\phi^{\mathfrak{t}}_{(\ell')}, \lambda^{\mathfrak{t}}_{(\ell')}, \psi^{\mathfrak{t}}_{(\ell')}) \neq (\phi^{\mathfrak{t}^*}_{(\ell')}, \lambda^{\mathfrak{t}^*}_{(\ell')}, \psi^{\mathfrak{t}^*}_{(\ell')}).$$

We also define the extended index of coincidence *of two types as,*

$$I(\mathfrak{t}, \mathfrak{t}^*) := [i(\mathfrak{t}, \mathfrak{t}^*), \hat{\imath}(\mathfrak{t}, \mathfrak{t}^*)],$$

where extended indices of coincidence are ordered lexicographically, i.e. we have the ordering $[14, 9] < [14, 10]$ *and* $[14, a] < [15, b]$*, regardless of the values of* a *and* b*.*

Definition 2.27. *Let* $\mathfrak{t}, \mathfrak{t}^* \in \mathfrak{T}$ *be strongly optimal types with index of coincidence* $i(\mathfrak{t}, \mathfrak{t}^*) = \ell$ *and let the list of refinements of each type at level* ℓ *be as in* (2.12)*.*

1. *The* greatest common ϕ-polynomial *of the pair* $(\mathfrak{t}, \mathfrak{t}^*)$ *is* $\phi(\mathfrak{t}, \mathfrak{t}^*) = \phi^{\mathfrak{t}}_{(j)} = \phi^{\mathfrak{t}^*}_{(j)}$*, with* j *maximal.*
2. *The* hidden slopes *of the pair* $(\mathfrak{t}, \mathfrak{t}^*)$ *are* $\lambda^{\mathfrak{t}^*}_{\mathfrak{t}} = \lambda^{\mathfrak{t}}_{(j)}$ *and* $\lambda^{\mathfrak{t}}_{\mathfrak{t}^*} = \lambda^{\mathfrak{t}^*}_{(j)}$*, for this maximal value of* j*.*

2.5.3 Complexity

The only algorithmic assumptions on the fields K and $\mathbb{F}$ for the algorithm to work properly, are the existence of efficient routines for the division with remainder of polynomials in $\mathcal{O}[x]$ and the factorisation of polynomials over finite extensions of the residue class field $\mathbb{F}$.

The performance will depend as well on the efficiency of these two tasks. There is still no complexity analysis of the algorithm in the general case, but for $\mathbb{F}$ a finite field, the following complexity estimation was obtained in [BNS13, Thm. 5.14], under the assumption that the field extension L/K, defined by $f \in \mathcal{O}[x]$, is separable.

Theorem 2.28. *If* $\mathbb{F}$ *is a finite field, the complexity of the Montes algorithm, measured in number of operations in* $\mathbb{F}$ *is*

$$\mathcal{C}_{\text{Montes}} = O\left(n^{2+\epsilon} + n^{1+\epsilon}(1+\delta)\log(q) + n^{1+\epsilon}\delta^{2+\epsilon}\right),$$

where $q = \#\mathbb{F}$*,* $n = \deg f$ *and* $\delta := v\left(\operatorname{disc}(f)\right)$*.*

2.6 Single-factor lifting and v-adic factorisation

Let $f \in \mathcal{O}[x]$ be a monic square-free polynomial and let $f = F_1 \cdots F_t$ be its factorisation into a product of prime polynomials in $\mathcal{O}_v[x]$.

A *v-adic factorisation* of f is an approximate factorisation with a prescribed precision; that is, a family of monic polynomials $P_1, \dots, P_t \in \mathcal{O}[x]$ such that $P_j \equiv F_j \pmod{\mathfrak{m}^\nu}$ for all $0 \leqslant j \leqslant t$, for a prescribed positive integer ν.

For many purposes, it may be necessary to find an approximation to a single prime factor F of f with a prescribed quality. This is the aim of the *single-factor lifting algorithm* [GNP12], abbreviated as SFL in what follows.

The starting point of SFL is a leaf $\mathtt{t}$ of an OM representation of f

$$\mathtt{t} = (\psi_0; (\phi_1, \lambda_1, \psi_1); \dots; (\phi_r, \lambda_r, \psi_r); (\phi_{r+1}, \lambda_{r+1}, \psi_{r+1})) \tag{2.13}$$

computed by the Montes algorithm. Let F be the prime factor of f singled out by $\mathtt{t}$, and let $\theta \in \overline{K}_v$ be a root of F. We denote

$$V := V_{r+1}, \quad \phi := \phi_{r+1}, \quad h_\phi := \lambda_{r+1} = h_{r+1}, \quad e := e(F) = e_1 \cdots e_r.$$

The polynomial ϕ is a Montes approximation to F as a factor of f. By Theorem 2.6, the quality of the approximation is:

$$v\left(\phi(\theta)\right) = \frac{V + h_\phi}{e} = \delta_0(F) + \frac{h_\phi}{e}.$$

The main loop of SFL computes a new Montes approximation Φ such that

$$h_\Phi \geqslant 2h_\phi - h_{\mathrm{cs}}.$$

The Newton polygon $\mathrm{N}^-_{v_r,\Phi}(f)$ coincides with $\mathrm{N}^-_{v_r,\phi}(f)$ except for the side of largest slope (in absolute value) $-h_\Phi$, whose end points have abscissas 0 and 1 (see Figure 2.8). In particular, the cutting slope h_{cs} of $\mathtt{t}$ once again separates this initial side from the remainder of the sides. Therefore, we may apply the SFL loop to Φ and iterate the procedure until we get a Montes approximation Φ with h_Φ large enough. By Lemma [GN, Lem. 4.1], if $h_\Phi \geqslant e(\nu + \mathrm{cap}(F) - \delta_0(F))$, then $\Phi \equiv F \pmod{\mathfrak{m}^\nu}$.

After k iterations of the SFL loop we get a Montes approximation Φ_k

with

$$h_{\Phi_k} \geqslant h_\phi + (2^k - 1)(h_\phi - h_{\mathrm{cs}}).$$

Hence, for a given positive integer H, the number of iterations of the SFL loop that are needed to achieve $h_{\Phi_k} \geqslant H$ is $\lceil \log_2((H - h_{\mathrm{cs}})/(h_\phi - h_{\mathrm{cs}})) \rceil$.

2.6.1 Complexity

The complexity of the SFL routine was analysed in [GNP12, Lem. 6.5] and [BNS13, Thm. 5.16]. As in Section 2.5.3, the complexity analysis that follows assumes that the residue class field $\mathbb{F}$ is a finite field and that L/K is separable. In the next result we denote $n = \deg f$, $n_F = \deg F$ and $\delta_F = v(\mathrm{disc}(F))$.

Theorem 2.29. *The SFL routine requires $O\left(nn_F\nu^{1+\epsilon} + n\delta_F^{1+\epsilon}\right)$ operations in $\mathbb{F}$ to compute a Montes approximation Φ to F as a factor of f, with precision ν.*

By applying the SFL routine to each leaf of an OM representation of f, we get an OM factorisation $f \approx P_1 \cdots P_t$ such that $P_j \equiv F_j \pmod{\mathfrak{m}^\nu}$ for all j.

Theorem 2.30. *If $\mathbb{F}$ is a finite field, a combined application of the Montes and SFL algorithms, computes an OM factorisation of f with precision ν, at the cost of*

$$O\left(n^{2+\epsilon} + n^{1+\epsilon}(1+\delta)\log q + n^{1+\epsilon}\delta^{2+\epsilon} + n^2\nu^{1+\epsilon}\right),$$

operations in $\mathbb{F}$.

2.7 OM representations of prime ideals

Suppose we are given an OM representation of f, as computed by the Montes algorithm.

As was mentioned at the beginning of this chapter, the prime ideals $\mathfrak{p} \in \mathcal{P}$ of $\mathcal{O}_L$ are in 1-to-1 correspondence with the prime factors of f in

$\mathcal{O}_v[x]$, which are, in turn, in 1-to-1 correspondence with the leaves of the optimised tree of the OM representation of f. The OM representations of each of these prime factors also hold a great deal of useful information about the associated prime ideal.

Let $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ be prime ideals of $\mathcal{O}_L$ and let $\mathbb{t}_\mathfrak{p}$, $\mathbb{t}_\mathfrak{q}$ be their respective OM representations. That is,

$$\mathbb{t}_\mathfrak{p} = \left(\psi_{0,\mathfrak{p}}; (\phi_{1,\mathfrak{p}}, \lambda_{1,\mathfrak{p}}, \psi_{1,\mathfrak{p}}); \ldots; (\phi_{r_\mathfrak{p},\mathfrak{p}}, \lambda_{r_\mathfrak{p},\mathfrak{p}}, \psi_{r_\mathfrak{p},\mathfrak{p}}); (\phi_\mathfrak{p}, \lambda_{r_\mathfrak{p}+1,\mathfrak{p}}, \psi_{r_\mathfrak{p}+1,\mathfrak{p}})\right),$$

and a similar notation for $\mathbb{t}_\mathfrak{q}$. Note that we denote by $\phi_\mathfrak{p} := \phi_{r_\mathfrak{p}+1,\mathfrak{p}}$ the Montes approximation to $F_\mathfrak{p}$ as a factor of f. Thus, the OM factorisation of f attached to this OM representation is

$$f \approx \prod_{\mathfrak{p}\in\mathcal{P}} \phi_\mathfrak{p}.$$

We allow an abuse of notation regarding the definitions of the indices of coincidence, greatest common ϕ-polynomial, and hidden slopes, which we will write as $i(\mathfrak{p}, \mathfrak{q})$, $\phi(\mathfrak{p}, \mathfrak{q})$, and $\lambda_\mathfrak{p}^\mathfrak{q}$ respectively.

One advantage of working with OM representations of prime ideals, is that we have explicit formulas for the $\mathfrak{p}$-valuation of the ϕ-polynomials at each level of the type $\mathbb{t}_\mathfrak{q}$.

The following proposition will be heavily used throughout the remainder of this memoir. The valuations $w_\mathfrak{p}$ were presented in Definition 1.2.

Proposition 2.31 ([GMN13, Prop. 4.7])**.** *Let $\mathfrak{p} \in \mathcal{P}$, be a prime ideal of $\mathcal{O}_L$ of Okutsu depth $r_\mathfrak{p}$. Then for any $1 \leqslant i \leqslant r_\mathfrak{p} + 1$,*

$$w_\mathfrak{p}\left(\phi_{i,\mathfrak{p}}(\theta)\right) = \frac{V_{i,\mathfrak{p}} + \lambda_{i,\mathfrak{p}}}{e_{1,\mathfrak{p}} \cdots e_{i-1,\mathfrak{p}}}.$$

Let $\mathfrak{q} \in \mathcal{P}$, be another prime ideal of Okutsu depth $r_\mathfrak{q}$ such that $\mathfrak{p} \neq \mathfrak{q}$

and with index of coincidence $\ell = i(\mathfrak{p}, \mathfrak{q})$. *For any* $1 \leqslant i \leqslant r_{\mathfrak{q}} + 1$,

$$w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right) = \begin{cases} 0, & \text{if } \ell = 0, \\ \dfrac{V_i + \lambda_i}{e_1 \cdots e_{i-1}}, & \text{if } i < \ell, \\ \dfrac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}}, & \text{if } i = \ell > 0 \text{ and } \phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{q}), \\ \dfrac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}}, & \text{if } i = \ell > 0 \text{ and } \phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{p}, \mathfrak{q}), \\ \dfrac{m_{i,\mathfrak{q}}}{m_\ell} \cdot \dfrac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}}, & \text{if } i > \ell > 0. \end{cases}$$

In these formulas, we omit the subscript $\mathfrak{p}$, $\mathfrak{q}$ *when the invariants of the two types coincide.*

Corollary 2.32. *Let* $f \approx \prod_{\mathfrak{p} \in \mathcal{P}} \phi_{\mathfrak{p}}$ *be an OM factorisation attached to an OM representation of* f. *Then, for any pair* $\mathfrak{p} \neq \mathfrak{q}$ *of prime ideals, we have* $w_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right) = w_{\mathfrak{p}}\left(F_{\mathfrak{q}}(\theta)\right)$.

Proof. Let $\ell = i(\mathfrak{p}, \mathfrak{q})$. If $\ell < r_{\mathfrak{q}} + 1$, or if $\ell = r_{\mathfrak{q}} + 1$ and $\phi_{\mathfrak{q}} \neq \phi(\mathfrak{p}, \mathfrak{q})$, Proposition 2.31 shows that

$$w_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right) = \frac{n_{\mathfrak{q}}}{m_\ell} \cdot \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}}. \tag{2.14}$$

In the case $\ell = r_{\mathfrak{q}} + 1$ and $\phi_{\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{q})$, we have $m_\ell = n_{\mathfrak{q}}$ and

$$w_{\mathfrak{q}}\left(\phi_{\mathfrak{q}}(\theta)\right) = \frac{V_\ell + \lambda_{\mathfrak{q}}^{\mathfrak{p}}}{e_1 \cdots e_{\ell-1}}, \qquad w_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right) = \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}}.$$

Since $\phi_{\mathfrak{q}}$ is a Montes approximation to $F_{\mathfrak{q}}$ as a factor of f, we have $w_{\mathfrak{q}}\left(\phi_{\mathfrak{q}}(\theta)\right)$ $> w_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right)$, so that $\lambda_{\mathfrak{q}}^{\mathfrak{p}} > \lambda_{\mathfrak{p}}^{\mathfrak{q}}$, and (2.14) is valid in this case too.

Now, let $\mathtt{t}$ be the first type of the non-optimised tree where the branches of $\mathfrak{p}$ and $\mathfrak{q}$ diverge. If we denote $\phi = \phi(\mathfrak{p}, \mathfrak{q})$, let $\mathtt{t}' = (\mathtt{t}; (\phi, \lambda_{\mathfrak{q}}^{\mathfrak{p}}, \psi_{\mathfrak{q}}^{\mathfrak{p}}))$, $\mathtt{t}'' = (\mathtt{t}; (\phi, \lambda_{\mathfrak{p}}^{\mathfrak{q}}, \psi_{\mathfrak{p}}^{\mathfrak{q}}))$ be the types attached to the two branch nodes such that $\mathtt{t}'' \mid F_{\mathfrak{p}}$ and $\mathtt{t}' \mid F_{\mathfrak{q}}$.

By Theorem 2.6, $\mathrm{N}_{v_{\mathtt{t}},\phi}(F_{\mathfrak{q}})$ is one-sided of slope $\lambda_{\mathfrak{q}}^{\mathfrak{p}}$. By Lemma 2.24, $\mathtt{t}'' \nmid F_{\mathfrak{q}}$; hence (2.5) shows that $w_{\mathfrak{p}}\left(F_{\mathfrak{q}}(\theta)\right)$ is given by (2.14) too. □

We will now clarify how the slopes $\lambda_{\mathfrak{p}}^{\mathfrak{q}}$, $\lambda_{\mathfrak{q}}^{\mathfrak{p}}$ are calculated. Consider the segment of the non-optimised tree of types shown in Figure 2.11. We see that the $\mathfrak{p}$ and $\mathfrak{q}$ types have a (non-optimised) index of coincidence $i(\mathbb{t}_{\mathfrak{p}}^{\mathrm{nop}}, \mathbb{t}_{\mathfrak{q}}^{\mathrm{nop}}) = \ell'$. Note that $\mathbb{n}_m$ corresponds to a type in the optimised tree.

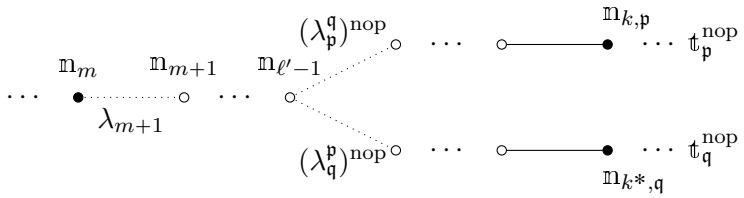


Figure 2.11: The branching between $\mathbb{t}_{\mathfrak{p}}^{\mathrm{nop}}$ and $\mathbb{t}_{\mathfrak{q}}^{\mathrm{nop}}$ in a non-optimised tree.

The slopes $(\lambda_{\mathfrak{p}}^{\mathfrak{q}})^{\mathrm{nop}}$ and $(\lambda_{\mathfrak{q}}^{\mathfrak{p}})^{\mathrm{nop}}$ of Figure 2.11 are called the "hidden slopes", as they are not present in the optimised tree. Actually, they should be called the "non-optimised hidden slopes", because they do not coincide with the "optimised hidden slopes" of Definition 2.27. The relationship between these hidden slopes is:

$$\lambda_{\mathfrak{p}}^{\mathfrak{q}} = \lambda_{m+1} + \cdots + \lambda_{\ell'-1} + (\lambda_{\mathfrak{p}}^{\mathfrak{q}})^{\mathrm{nop}},$$
$$\lambda_{\mathfrak{q}}^{\mathfrak{p}} = \lambda_{m+1} + \cdots + \lambda_{\ell'-1} + (\lambda_{\mathfrak{q}}^{\mathfrak{p}})^{\mathrm{nop}},$$

where λ_i is the slope that corresponds to the path between $\mathbb{n}_{i-1}$ and $\mathbb{n}_i$ for all $m+1 \leqslant i \leqslant \ell'-1$.

3

Optimal polynomials

"It is what you don't expect... that most needs looking for.'

– Neal Stephenson, *Anathem*

We keep dealing with our discrete valued field (K, v) and we keep the notation of the previous chapter.

The aim of this and the next chapter is to construct a triangular, reduced $\mathcal{O}$-basis of $\mathcal{O}_L$ from a given OM representation of f. By Theorem 1.26, it suffices to construct a family $g_0, \ldots, g_{n-1} \in \mathcal{O}[x]$ of monic polynomials, such that for all $0 \leqslant i < n$:

1. $\deg g_i = i$.
2. $w\left(g_i(\theta)\right)$ is maximal amongst all monic polynomials in $\mathcal{O}[x]$ of degree i.

In Section 3.1 we recall the construction of *Okutsu bases*. For a prime ideal $\mathfrak{p} \mid \mathfrak{m}$, corresponding to a prime factor $F_\mathfrak{p}$ of f in $\mathcal{O}_v[x]$, Okutsu constructed an $\mathcal{O}_v$-basis of the completion $\mathcal{O}_\mathfrak{p}$ of $\mathcal{O}_L$ with respect to the $\mathfrak{p}$-adic topology, by considering a similar family of polynomials $g_{0,\mathfrak{p}}, \ldots, g_{n_\mathfrak{p}-1,\mathfrak{p}} \in \mathcal{O}[x]$ having a maximal $w_\mathfrak{p}$-value amongst all monic polynomials of the same degree [Oku82a][Oku82b]. Also, in [GMN10b] it was shown that these polynomials $g_{i,\mathfrak{p}}$ may be derived in a trivial way from any OM representation of $F_\mathfrak{p}$.

Section 3.1 finalises the part of the memoir devoted to preliminary results. From this point onward, the rest of the results are original.

The rest of this chapter is dedicated to showing that the search for these *optimal* polynomials, satisfying (1) and (2) may be restricted to polynomials of a very special form:

$$g_i = \prod_{\mathfrak{p} \mid \mathfrak{m}} g_{i_\mathfrak{p},\mathfrak{p}}, \qquad \deg i = i.$$

That is, g_i may be taken to be a product of exactly one polynomial in each local Okutsu basis.

In Chapter 4, we develop and algorithm that performs an efficient search to find the right choices for the factors $g_{i_\mathfrak{p},\mathfrak{p}}$.

3.1 Okutsu bases

Let $L_\mathfrak{p}$ be the completion of L with respect to the $\mathfrak{p}$-adic topology. We may consider a topological embedding $L \subset L_\mathfrak{p} \subset \overline{K}_v$, so that $L_\mathfrak{p}$ may be identified to a finite extension of K_v of degree $n_\mathfrak{p} = ef$, where $e := e(\mathfrak{p}/\mathfrak{m})$, $f := f(\mathfrak{p}/\mathfrak{m})$. We denote by $\mathcal{O}_\mathfrak{p} := \mathcal{O}_{L_\mathfrak{p}}$ the integral closure of $\mathcal{O}_v$ in $L_\mathfrak{p}$.

Let r be the Okutsu depth of $F_\mathfrak{p}$ and suppose that

$$\mathbb{t} = (\psi_0; (\phi_1, \lambda_1, \psi_1); \ldots; (\phi_{r+1}, \lambda_{r+1}, \psi_{r+1}))\,,$$

is the leaf corresponding to $F_\mathfrak{p}$ in the tree of an OM representation of f. Recall that $m_1 \mid \cdots \mid m_r \mid m_{r+1}$ and $m_1 < \cdots < m_r < m_{r+1}$.

We shall say that the family of ϕ-polynomials

$$[\phi_1, \ldots, \phi_r],$$

is an *Okutsu frame* of $F_{\mathfrak{p}}$. These polynomials determine a family of optimal polynomials $g_{0,\mathfrak{p}}, \ldots, g_{n_{\mathfrak{p}}-1,\mathfrak{p}} \in \mathcal{O}[x]$ as follows. Each $0 \leqslant i < n_{\mathfrak{p}}$ may be expressed in a unique way as:

$$i = a_0 + a_1 m_1 + \cdots + a_r m_r, \qquad 0 \leqslant a_j < m_{j+1}/m_j = e_j f_j.$$

Thus, the polynomials:

$$g_{i,\mathfrak{p}} := x^{a_0} \prod_{j=1} \phi_j^{a_j}, \qquad 0 \leqslant i < n_{\mathfrak{p}},$$

are monic polynomials in $\mathcal{O}[x]$ of degree $\deg g_{i,\mathfrak{p}} = i$.

Theorem 3.1 ([GMN10b, Thm. 2.15, Thm. 3.9])**.** *For all $0 \leqslant i < n_{\mathfrak{p}}$, $w_{\mathfrak{p}}(g_{i,\mathfrak{p}})$ is maximal amongst all monic polynomials in $\mathcal{O}[x]$ of degree i.*

By Theorem 1.26, we get a triangular and reduced $\mathcal{O}_v$-basis of $\mathcal{O}_{\mathfrak{p}}$ by taking:

$$\alpha_i := \pi^{-\lfloor w_{\mathfrak{p}}(g_{i,\mathfrak{p}}(\theta)) \rfloor} g_{i,\mathfrak{p}}(\theta), \qquad 0 \leqslant i < n_{\mathfrak{p}}.$$

We call $\mathcal{B}_{\mathfrak{p}} = (\alpha_i)_{0 \leqslant i < n_{\mathfrak{p}}}$ the *Okutsu basis* of $\mathcal{O}_{\mathfrak{p}}$, or simply the Okutsu $\mathfrak{p}$-basis.

The group of fractional ideals $\mathcal{I}_{\mathcal{O}_{\mathfrak{p}}}$ is a cyclic infinite group generated by $\mathfrak{p}$. Thus, the same family $g_{0,\mathfrak{p}}, \ldots, g_{n_{\mathfrak{p}}-1,\mathfrak{p}}$ that was used for the construction of an $\mathcal{O}_v$-basis of $\mathcal{O}_{\mathfrak{p}}$ yields a triangular and reduced $\mathcal{O}_v$-basis of any fraction ideal. In fact, consider a fractional ideal $I = \mathfrak{p}^a$, for some $a \in \mathbb{Z}$. The function $w_{\mathfrak{m},I}$ introduced in Definition 1.4, differs from $w_{\mathfrak{p}}$ only by a shift:

$$w_{\mathfrak{p},I}(\alpha) = w_{\mathfrak{p}}(\alpha) - \frac{a}{e(\mathfrak{p}/\mathfrak{m})}.$$

Hence, Theorem 3.1 shows that these polynomials have a maximal $w_{\mathfrak{m},I}$-value amongst all monic polynomials in $\mathcal{O}[x]$ of the same degree. Therefore,

Theorem 1.26 shows that

$$\beta_i := \pi^{-\lfloor w_{\mathfrak{m},I}(g_{i,\mathfrak{p}}(\theta))\rfloor} g_{i,\mathfrak{p}}(\theta), \qquad 0 \leqslant i < n_{\mathfrak{p}},$$

is a triangular and reduced $\mathcal{O}_v$-basis of I.

For further purposes, the family of numerators of an Okutsu basis of $\mathcal{O}_{\mathfrak{p}}$ is extended by adding an Okutsu approximation to $F_{\mathfrak{p}}$,

$$\mathcal{N}_{\mathfrak{p}} := \left\{1 =: g_{0,\mathfrak{p}}, \ldots, g_{n_{\mathfrak{p}}-1,\mathfrak{p}}, g_{n_{\mathfrak{p}},\mathfrak{p}} := \phi_{\mathfrak{p}}\right\}. \tag{3.1}$$

3.2 Optimal polynomials as products of ϕ-polynomials

Let $f \in \mathcal{O}[x]$ be a monic, irreducible polynomial of degree n and fix a root $\theta \in \overline{K}$ of f. Let $L = K(\theta)$ be the finite extension of K defined by f and $\mathcal{O}_L$ the integral closure of $\mathcal{O}$ in L which is a Dedekind domain.

We assume that $\mathcal{O}_L$ is finitely generated as an $\mathcal{O}$-module, and so $\mathcal{O}_L$ is a free $\mathcal{O}$-module of rank $n = \deg f$. Since $\mathcal{O}$ is a local ring, $\mathcal{O}_L$ is a semilocal Dedekind domain. Let $\mathcal{P} = \mathrm{Max}(\mathcal{O}_L)$ be the finite set of non-zero prime ideals of $\mathcal{O}_L$.

The Montes algorithm with input (f, v), produces a tree $\mathfrak{T} = \{\mathbb{t}_{\mathfrak{p}} : \mathfrak{p} \in \mathcal{P}\}$ of types. Each type is an OM representation of a prime factor $F_{\mathfrak{p}}$ of f in $\mathcal{O}_v[x]$, corresponding to a prime ideal $\mathfrak{p} \in \mathcal{P}$, which has been extended to a type of order $r_{\mathfrak{p}} + 1$, where $r_{\mathfrak{p}}$ is the Okutsu depth of $F_{\mathfrak{p}}$.

Definition 3.2. *For a set S of prime ideals, the S-valuation of an element in L is the minimum of the $\mathfrak{p}$-valuations for all primes in S,*

$$w_S(\alpha) := \min\left\{w_{\mathfrak{p}}(\alpha)\right\}_{\mathfrak{p}\in S} = \min\left\{\frac{v_{\mathfrak{p}}(\alpha)}{e(\mathfrak{p}/\mathfrak{m})}\right\}_{\mathfrak{p}\in S}, \qquad \forall\ \alpha \in L.$$

By convention, we take $w := w_{\mathcal{P}}$ to indicate the minimum of the $\mathfrak{p}$-valuations for all the prime ideals $\mathfrak{p}$ of $\mathcal{O}_L$ (i.e. $\mathfrak{p} \in \mathcal{P}$).

Definition 3.3. *Let $g \in \mathcal{O}_v[x]$. The* degree adjusted $w_{\mathfrak{p}}$*-valuation of the*

element $g(\theta) \in \mathcal{O}_L$ is defined as

$$\hat{w}_{\mathfrak{p}}\left(g(\theta)\right) := \frac{w_{\mathfrak{p}}\left(g(\theta)\right)}{\deg g}.$$

The same concept holds for w_S-valuations for any set S of prime ideals,

$$\hat{w}_S\left(g(\theta)\right) := \frac{w_S\left(g(\theta)\right)}{\deg g}.$$

This brings us to the purpose of this section, $\mathfrak{p}$-optimal polynomials.

Definition 3.4. *A monic polynomial $g \in \mathcal{O}[x]$ of degree d is called $\mathfrak{p}$-optimal if it has valuation $w_{\mathfrak{p}}\left(g(\theta)\right)$ maximal amongst all monic polynomials in $\mathcal{O}[x]$ also of degree d.*

A polynomial $g \in \mathcal{O}[x]$ of degree d is v-optimal if $w\left(g(\theta)\right)$ is maximal amongst all monic polynomials in $\mathcal{O}[x]$ of the same degree.

A type $\mathtt{t}_{\mathfrak{p}}$ corresponding to a prime ideal $\mathfrak{p}$ of depth $r_{\mathfrak{p}}$ contains ϕ-polynomials at each level,

$$\phi_{1,\mathfrak{p}}, \ldots, \phi_{r_{\mathfrak{p}},\mathfrak{p}}, \phi_{r_{\mathfrak{p}}+1,\mathfrak{p}} =: \phi_{\mathfrak{p}},$$

of degree $m_1 < m_2 < \cdots < m_{r_{\mathfrak{p}}} < m_{r_{\mathfrak{p}}+1} = n_{\mathfrak{p}}$.

Lemma 3.5. *For any prime ideal $\mathfrak{p} \in \mathcal{P}$,*

$$\hat{w}_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}}(\theta)\right) < \hat{w}_{\mathfrak{p}}\left(\phi_{i+1,\mathfrak{p}}(\theta)\right), \qquad 1 \leqslant i \leqslant r_{\mathfrak{p}}.$$

Proof. By Proposition 2.31,

$$w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}}(\theta)\right) = \frac{V_i + \lambda_i}{e_1 \cdots e_{i-1}} = \frac{e_i V_i + h_i}{e_1 \cdots e_i}.$$

Recall that $V_{i+1} = e_i f_i (e_i V_i + h_i)$ for all $1 \leqslant i \leqslant r_{\mathfrak{p}}$, so that

$$\frac{m_{i+1}}{m_i} \cdot w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}}(\theta)\right) = e_i f_i \cdot w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}}(\theta)\right) = \frac{V_{i+1}}{e_1 \cdots e_i} < w_{\mathfrak{p}}\left(\phi_{i+1,\mathfrak{p}}(\theta)\right).$$

□

The final polynomial $\phi_{\mathfrak{p}}$ is a Montes approximation to $F_{\mathfrak{p}}$ as a factor of

f and,

$$f \approx \prod_{\mathfrak{p}\in\mathcal{P}} \phi_{\mathfrak{p}}.$$

The ϕ-polynomials for all the prime ideals generate a semigroup.

Definition 3.6. *Let $S \subseteq \mathcal{P}$ be a set of prime ideals. Then, $\Phi(S) \subset \mathcal{O}[x]$ is the multiplicative semigroup generated by*

$$1, \{\phi_{i,\mathfrak{p}}\}_{\mathfrak{p}\in S,\ 0\leqslant i\leqslant r_{\mathfrak{p}}}, \bigcup_{\mathfrak{p}\in S} \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}}),$$

where $\mathrm{Rep}(\mathbb{t}_{\mathfrak{p}}) = [F_{\mathfrak{p}}] \cap \mathcal{O}[x]$ is the set of all representatives of $\mathbb{t}_{\mathfrak{p}}$.

We use $\Phi(\mathfrak{p})$ to denote the semigroup generated by the ϕ-polynomials belonging to a single prime ideal $\mathfrak{p} \in \mathcal{P}$ and fix $\Phi := \Phi(\mathcal{P})$.

Recall that by Corollary 2.32, for $\mathfrak{p} \neq \mathfrak{q}$, all $\phi \in \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}})$ Montes approximations to $F_{\mathfrak{p}}$ as a factor of f have the same $\mathfrak{q}$-valuation $w_{\mathfrak{q}}(\phi(\theta))$

We are interested in showing that we can restrict our search for v-optimal polynomials of a given degree d to those in Φ.

Definition 3.7. *For any node $\mathbb{n} \in \mathfrak{T}$ or $\mathbb{n} \in \mathfrak{T}^{\mathrm{nop}}$, Let $S_{\mathbb{n}} \subset \mathcal{P}$ be the subset of prime ideals $\mathfrak{p}$ such that $\mathbb{t}_{\mathbb{n}} \mid F_{\mathfrak{p}}$. Equivalently, $\mathbb{n}$ belongs to the path joining the leaf of $\mathfrak{T}$ (or $\mathfrak{T}^{\mathrm{nop}}$) corresponding to $\mathfrak{p}$, with the root node.*

We recall that $\mathbb{t}_{\mathbb{n}}$ is the type obtained by gathering the data corresponding to all edges in the path joining $\mathbb{n}$ with its root node.

Lemma 3.8. *Let $\mathbb{n}$ be a node in the non-optimised (connected) tree $\mathfrak{T}^{\mathrm{nop}}_{\psi_0}$ and let $g, h \in \mathbb{P}(\mathcal{O}_v[x])$ be two prime polynomials divisible by $\mathbb{t}_{\mathbb{n}}$. Then, for any prime ideal $\mathfrak{p} \in \mathcal{P}\backslash S_{\mathbb{n}}$ we have $\hat{w}_{\mathfrak{p}}(g(\theta)) = \hat{w}_{\mathfrak{p}}(h(\theta))$.*

Proof. Let $\mathbb{m}$ be the greatest common node in the paths of the non-optimised tree $\mathfrak{T}^{\mathrm{nop}}_{\psi_0}$ joining $\mathbb{t}_{\mathfrak{p}}$ and $\mathbb{n}$ with the root node. Since $\mathfrak{p} \notin S_{\mathbb{n}}$, the node $\mathbb{m}$ cannot be equal to $\mathbb{n}$. Since $\mathbb{t}_{\mathfrak{p}}$ is a leaf of the tree, $\mathbb{m}$ cannot be equal to $\mathbb{t}_{\mathfrak{p}}$ either. The structure of the non-optimised tree is shown in Figure 3.1.

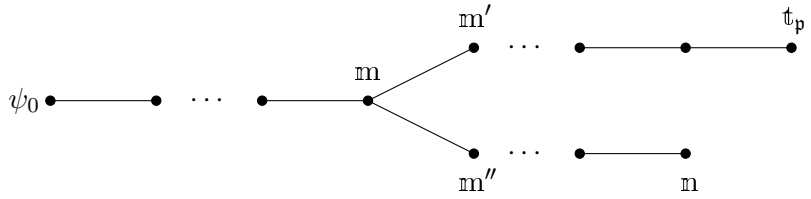


Figure 3.1: The node $\mathbb{m}$ is the greatest common node of $\mathbb{n}$ and $\mathfrak{t}_\mathfrak{p}$.

Let $\mathbb{m}', \mathbb{m}''$ be the nodes following $\mathbb{m}$ in each of the two paths. Since the non-optimised tree is coherent, we have

$$\mathfrak{t}_{\mathbb{m}'} = (\mathfrak{t}_\mathbb{m}; (\phi_\mathbb{m}, \lambda', \psi')), \qquad \mathfrak{t}_{\mathbb{m}''} = (\mathfrak{t}_\mathbb{m}; (\phi_\mathbb{m}, \lambda'', \psi'')),$$

with a common choice for the representative $\phi_\mathbb{m}$ of $\mathfrak{t}_\mathbb{m}$. Let us denote simply by $v_\mathbb{m}$ the valuation $v_{\mathfrak{t}_\mathbb{m}}$ attached to (the last level of) the type $\mathfrak{t}_\mathbb{m}$.

By Theorem 2.6, $\mathrm{N}_{v_\mathbb{m},\phi_\mathbb{m}}(g)$ is one-sided of slope $-\lambda''$ and $\mathrm{R}_{v_\mathbb{m},\phi_\mathbb{m},\lambda''}(g)$ is a power of ψ''. Since $(\lambda', \psi') \neq (\lambda'', \psi'')$, Theorem 2.6 shows that $\mathfrak{t}_{\mathbb{m}'} \nmid g$. By (2.5), we have

$$w_\mathfrak{p}(g(\theta)) = \frac{\deg g}{\deg \phi_\mathbb{m}} \cdot \frac{V_r + \min\{\lambda', \lambda''\}}{e_1 \cdots e_{r-1}}, \tag{3.2}$$

where $r = \mathrm{ord}(\mathfrak{t}_\mathbb{m}) + 1$. The same arguments show that $\mathfrak{t}_{\mathbb{m}'} \nmid h$ and a formula analogous to (3.2) holds for $w_\mathfrak{p}(h(\theta))$. Hence,

$$\hat{w}_\mathfrak{p}(g(\theta)) = \frac{w_\mathfrak{p}(g(\theta))}{\deg g} = \frac{w_\mathfrak{p}(h(\theta))}{\deg h} = \hat{w}_\mathfrak{p}(h(\theta)).$$

□

The next result is the main aim of this section. For the search for v-optimal polynomials, we may consider only polynomials in the semigroup $\Phi(\mathcal{P})$.

Theorem 3.9. *Let $S \subseteq \mathcal{P}$ be a set of prime ideals. For any $h \in \mathcal{O}[x]$ monic of degree $0 \leqslant d < n$, there exists $\phi \in \Phi(S)$ also of degree d such that,*

$$w_\mathfrak{p}(\phi(\theta)) \geqslant w_\mathfrak{p}(h(\theta)), \qquad \forall\ \mathfrak{p} \in S. \tag{3.3}$$

Proof. The proof will proceed by induction on the degree d of the polynomial. We will work in steps, in each one reducing the space in which we need to consider h.

If $d = 0$, then $\phi = h = 1 \in \Phi(S)$.

Claim. *It is sufficient to check* (3.3) *for* $h \in \mathbb{P}(\mathcal{O}_v[x])$.

Let $h = h_1 h_2$, with $h_1, h_2 \in \mathcal{O}_v[x]$ monic of degree $d_1, d_2 > 0$ respectively.

Now, consider $h \equiv H_1 H_2 \pmod{\mathfrak{m}^N}$, with $N > \max\{w_\mathfrak{p}(h(\theta)) : \mathfrak{p} \in S\}$ and $H_1, H_2 \in \mathcal{O}[x]$ monic, also of degree $d_1, d_2 > 0$ respectively.

By an inductive argument on the number of prime factors of h, there exist $\phi_i \in \Phi(S)$ of degree d_i such that,

$$w_\mathfrak{p}(\phi_i(\theta)) \geqslant w_\mathfrak{p}(H_i(\theta)), \qquad \forall\ \mathfrak{p} \in S,$$

for $i = 1, 2$. Take $\phi = \phi_1\phi_2 \in \Phi(S)$. For any $\mathfrak{p} \in S$ we have

$$\begin{aligned} w_\mathfrak{p}(\phi(\theta)) &= w_\mathfrak{p}(\phi_1(\theta)\phi_2(\theta)) \\ &= w_\mathfrak{p}(\phi_1(\theta)) + w_\mathfrak{p}(\phi_2(\theta)) \\ &\geqslant w_\mathfrak{p}(H_1(\theta)) + w_\mathfrak{p}(H_2(\theta)) \\ &= w_\mathfrak{p}(H_1(\theta)H_2(\theta)) \\ &= w_\mathfrak{p}(h(\theta)). \end{aligned}$$

The final equality is due to the triangle inequality of the $\mathfrak{p}$-adic valuation. We can express $h = H_1H_2 + \pi^N G$ for some $G \in \mathcal{O}[x]$, and since $w_\mathfrak{p}\left(\pi^N G(\theta)\right) > w_\mathfrak{p}(h(\theta))$ for all $\mathfrak{p} \in S$, we deduce the equality $w_\mathfrak{p}(h(\theta)) = w_\mathfrak{p}(H_1(\theta)H_2(\theta))$. This proves the claim.

Now, we only need to consider the case $h \in \mathbb{P}(\mathcal{O}_v[x])$, which we will divide into two cases, depending on whether the reductions mod $\mathfrak{m}$ of f and h share a common factor.

Case 1. $h \in \mathbb{P}(\mathcal{O}_v[x])$, and $\gcd(\overline{f}, \overline{h}) = 1$.

In this case $w_\mathfrak{p}(h(\theta)) = 0$ for all $\mathfrak{p} \in S$. Thus, (3.3) is obviously satisfied.

Case 2. $h \in \mathbb{P}(\mathcal{O}_v[x])$, $\overline{h} = \psi_0^b$, $b \in \mathbb{N}$, for $\psi_0 \in \mathbb{F}[y]$ a prime factor of $\overline{f}$.

Let $S_{\psi_0} \subset S$ be the subset of prime ideals $\mathfrak{p} \in S$ such that $\psi_{0,\mathfrak{p}} = \psi_0$. Let

$$\Phi(S_{\psi_0}) = \left\{\phi \in \Phi(S) : \overline{\phi} \text{ is a power of } \psi_0\right\}.$$

It is sufficient to find $\phi \in \Phi(S_{\psi_0})$ of degree d such that (3.3) holds for all $\mathfrak{p} \in S_{\psi_0}$, since

$$w_{\mathfrak{p}}\left(\phi(\theta)\right) = 0 = w_{\mathfrak{p}}\left(h(\theta)\right), \qquad \forall\ \mathfrak{p} \in S \backslash S_{\psi_0}.$$

By hypothesis, the root node of $\mathfrak{T}_{\psi_0}^{\mathrm{nop}}$ divides h. Let $\mathtt{n}$ be the highest order node in the non-optimised tree $\mathfrak{T}_{\psi_0}^{\mathrm{nop}}$ such that $\mathtt{t}_{\mathtt{n}} \mid h$, and let $i-1$ be the order of $\mathtt{t}_{\mathtt{n}}$. We distinguish two cases according to $\mathtt{n}$ being a leaf or not.

Case 2A. $\mathtt{n}$ is a leaf.

In this case, $S_{\mathtt{n}} = \{\mathfrak{p}_0\}$ contains only one prime ideal. The polynomial ϕ_{i-1} is an Okutsu approximation to $F_{\mathfrak{p}_0}$. Since $\mathtt{t}_{\mathtt{n}} \mid h$, Theorem 2.6 shows that $\deg h = a \cdot \deg \phi_{i-1} = a \cdot \deg F_{\mathfrak{p}_0}$ for some positive integer a. If $a = 1$, then h is a polynomial with minimal degree such that $\mathtt{t}_{\mathtt{n}} \mid h$; that is, h is a representative of $\mathtt{t}_{\mathtt{n}}$, so that $h \in \Phi(S)$ and the statement of the theorem is obvious.

Suppose $a > 1$. Since $\mathtt{n}$ is a leaf of $\mathfrak{T}_{\psi_0}^{\mathrm{nop}}$, any representative of $\mathtt{t}_{\mathtt{n}}$ is an Okutsu approximation to $F_{\mathfrak{p}_0}$ and we may take $\phi_0 \in \mathrm{Rep}(\mathtt{t}_{\mathtt{n}})$ with $w_{\mathfrak{p}_0}\left(\phi_0(\theta)\right)$ arbitrarily large. Let us consider ϕ_0 with

$$w_{\mathfrak{p}_0}\left(\phi_0(\theta)\right) \geqslant \frac{w_{\mathfrak{p}_0}\left(h(\theta)\right)}{a},$$

and take $\phi = \phi_0^a \in \Phi(S_{\psi_0})$. By construction, $w_{\mathfrak{p}_0}\left(\phi(\theta)\right) = a \cdot w_{\mathfrak{p}_0}\left(\phi_0(\theta)\right) \geqslant w_{\mathfrak{p}_0}\left(h(\theta)\right)$. On the other hand, for any $\mathfrak{p} \in S_{\psi_0}$, $\mathfrak{p} \neq \mathfrak{p}_0$, Lemma 3.8 shows

that $\hat{w}_{\mathfrak{p}}(\phi_0(\theta)) = \hat{w}_{\mathfrak{p}}(h(\theta))$, so that

$$\begin{aligned} w_{\mathfrak{p}}(\phi(\theta)) &= a \cdot w_{\mathfrak{p}}(\phi_0(\theta)) \\ &= a \cdot \deg \phi_0 \cdot \hat{w}_{\mathfrak{p}}(\phi_0(\theta)) \\ &= \deg h \cdot \hat{w}_{\mathfrak{p}}(h(\theta)) \\ &= w_{\mathfrak{p}}(h(\theta)). \end{aligned} \tag{3.4}$$

This proves $w_{\mathfrak{p}}(\phi(\theta)) \geqslant w_{\mathfrak{p}}(h(\theta))$ for all $\mathfrak{p} \in S$.

Case 2B. $\mathbb{n}$ is not a leaf.

For a certain choice $\phi_{\mathbb{n}}$ of a representative of $\mathbb{t}_{\mathbb{n}}$, the node $\mathbb{n}$ has several branches $\mathbb{m}$ in the non-optimised tree, with

$$\mathbb{t}_{\mathbb{m}} = (\mathbb{t}_{\mathbb{n}}; (\phi_{\mathbb{n}}, \lambda, \psi)).$$

By the maximality of $\mathbb{n}$, we have $\mathbb{t}_{\mathbb{m}} \not| \, h$ for all these branch nodes $\mathbb{m}$. Let $\lambda_{\max}$ be the greatest slope (in absolute size) of these branches and let $\mathbb{m}_{\max}$ be any branch node of $\mathbb{n}$ with slope $\lambda_{\max}$.

If $\mathbb{m}_{\max}$ is a node of the optimised tree, take $\mathbb{n}_{\max} = \mathbb{m}_{\max}$. Otherwise, let $\mathbb{n}_{\max}$ be any node of the optimised tree which has been derived from $\mathbb{m}_{\max}$ by a series of refinement steps as presented in Figure 3.2.

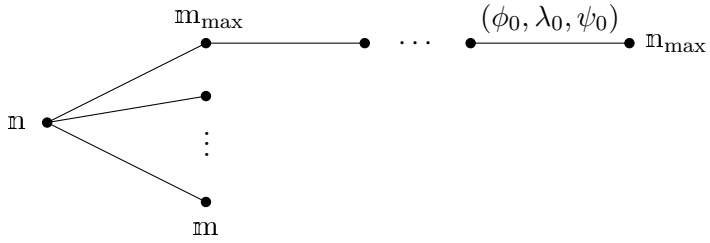


Figure 3.2: The node $\mathbb{n}_{\max}$ belongs to the optimised tree.

Let $(\phi_0, \lambda_0, \psi_0)$ be the last level of $\mathbb{t}_{\mathbb{n}_{\max}}$ in the non-optimised tree. As explained in Section 2.5.2 the last level of $\mathbb{t}_{\mathbb{n}_{\max}}$ as a type from the optimised tree will be $(\phi_0, \lambda_0^*, \psi_0)$, where λ_0^* is the sum of all the slopes of all bad levels between $\mathbb{n}_{\max}$ and its previous node in the optimised tree. This will be $\mathbb{n}$ if $\mathbb{n}$ belongs to the optimised tree or some node prior to $\mathbb{n}$ if it does not.

Thus, $\phi_0 = \phi_{j,\mathfrak{p}_0}$ for all $\mathfrak{p}_0 \in S_{\mathbb{n}_{\max}}$, where j is the order of $\mathbb{n}_{\max}$ as a node of the optimised tree. By construction, $\deg \phi_0 = \deg \phi_{\mathbb{n}}$ and since $\mathbb{t}_{\mathbb{n}} \mid h$, Theorem 2.6 shows that $\deg h = a \cdot \deg \phi_{\mathbb{n}} = a \cdot \deg \phi_0$, for a certain positive integer a. Let us take $\phi = \phi_0^a \in \Phi(S_{\psi_0})$ and let us show that $w_{\mathfrak{p}}(\phi(\theta)) \geqslant w_{\mathfrak{p}}(h(\theta))$ for all $\mathfrak{p} \in S_{\psi_0}$.

For $\mathfrak{p} \notin S_{\mathbb{n}}$ we have $\hat{w}_{\mathfrak{p}}(h(\theta)) = \hat{w}_{\mathfrak{p}}(\phi_0(\theta))$ by Lemma 3.8, and (3.4) shows that $w_{\mathfrak{p}}(h(\theta)) = w_{\mathfrak{p}}(\phi(\theta))$.

For any $\mathfrak{p} \in S_{\mathbb{n}}$, Theorem 2.7 shows that

$$w_{\mathfrak{p}}(h(\theta)) = a \cdot \frac{V_i + \min\{\lambda_{\mathfrak{p}}, \lambda_h\}}{e_1 \cdots e_{i-1}},$$

where $-\lambda_h$ is the slope of $\mathrm{N}_{v_{\mathbb{n}},\phi_{\mathbb{n}}}(h)$ (which is one-sided) and $-\lambda_{\mathfrak{p}}$ is the slope of the unique branch $\mathbb{m}$ of $\mathbb{n}$ for which $\mathbb{t}_{\mathbb{m}} \mid F_{\mathfrak{p}}$.

If $\mathbb{m} \neq \mathbb{m}_{\max}$, Proposition 2.31 shows that

$$w_{\mathfrak{p}}(\phi(\theta)) = a \cdot w_{\mathfrak{p}}(\phi_0(\theta)) = a \cdot \frac{V_i + \min\{\lambda_{\mathfrak{p}}, \lambda_{\max}\}}{e_1 \cdots e_{i-1}},$$

so that $w_{\mathfrak{p}}(\phi(\theta)) \geqslant w_{\mathfrak{p}}(h(\theta))$, because

$$\min\{\lambda_{\mathfrak{p}}, \lambda_h\} \leqslant \lambda_{\mathfrak{p}} = \min\{\lambda_{\mathfrak{p}}, \lambda_{\max}\}.$$

If $\mathbb{m} = \mathbb{m}_{\max}$, that is $\mathbb{t}_{\mathbb{m}_{\max}} \mid F_{\mathfrak{p}}$, then $i(\mathfrak{p}, \mathfrak{p}_0) \geqslant i$ and by Proposition 2.31

$$w_{\mathfrak{p}}(\phi(\theta)) = \begin{cases} a \cdot \dfrac{V_i + \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}}{e_1 \cdots e_{i-1}}, & i(\mathfrak{p}, \mathfrak{p}_0) = i, \phi_{i,\mathfrak{p}} = \phi(\mathfrak{p}, \mathfrak{p}_0), \\ a \cdot \dfrac{V_i + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}_0}, \lambda_{\mathfrak{p}_0}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{i-1}}, & i(\mathfrak{p}, \mathfrak{p}_0) = i, \phi_{i,\mathfrak{p}} \neq \phi(\mathfrak{p}, \mathfrak{p}_0), \\ a \cdot \dfrac{V_i + \lambda_{i,\mathfrak{p}_0}}{e_1 \cdots e_{i-1}}, & i(\mathfrak{p}, \mathfrak{p}_0) > i. \end{cases}$$

If $i(\mathfrak{p}, \mathfrak{p}_0) = i$ then we have $\lambda_{\mathfrak{p}}^{\mathfrak{p}_0} \geqslant \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}_0}, \lambda_{\mathfrak{p}_0}^{\mathfrak{p}}\right\} > \lambda_{\max}$ because in the refinement procedure the slope grows strictly. On the other hand, if $i(\mathfrak{p}, \mathfrak{p}_0) > i$ then $\lambda_{i,\mathfrak{p}_0} \geqslant \lambda_{\max}$ and $w_{\mathfrak{p}}(\phi(\theta)) \geqslant w_{\mathfrak{p}}(h(\theta))$ for all $\mathfrak{p} \in S_{\psi_0}$. □

3.3 Optimal polynomials as products of numerators of Okutsu bases

3.3.1 Partial Okutsu bases

Let $S \subseteq \mathcal{P}$ be a subset of $\mathcal{P}$ and let $n_S = \sum_{\mathfrak{p}\in S} n_{\mathfrak{p}}$ be the degree of S. Consider a sequence of monic polynomials:

$$1, g_1, \ldots, g_{n_S-1} \in \mathcal{O}[x], \qquad \deg g_i = i,\ 0 \leqslant i < n_S, \tag{3.5}$$

such that $g_i(\theta)$ has maximal S-valuation amongst all monic polynomials of the same degree:

$$w_S\left(g_i(\theta)\right) = \max\left\{w_S\left(g(\theta)\right) : g \in \mathcal{O}[x], g \text{ monic}, \deg g = i\right\}, \quad 0 \leqslant i < n_S.$$

These conditions imply that the sequence of all

$$\frac{g_i(\theta)}{\pi^{\lfloor w_S(g_i(\theta))\rfloor}}, \qquad 0 \leqslant i < n_S,$$

is a reduced triangular S-basis of L. That is, the images of these elements under the topological embeddings

$$(\iota_{\mathfrak{p}})_{\mathfrak{p}\in S} : L \hookrightarrow \bigoplus_{\mathfrak{p}\in S} L_{\mathfrak{p}},$$

form an $\mathcal{O}_v$-basis of $\bigoplus_{\mathfrak{p}\in S} \mathcal{O}_{\mathfrak{p}}$.

Since $\bigoplus_{\mathfrak{p}\in\mathcal{P}} \mathcal{O}_{\mathfrak{p}}$ is isomorphic to $\mathcal{O}_L \otimes_{\mathcal{O}} \mathcal{O}_v$, a $\mathcal{P}$-basis is a v-integral basis. Also, for a one-element subset $S = \{\mathfrak{p}\}$, a reduced triangular S-basis is just a reduced triangular $\mathcal{O}_v$-basis of $\mathcal{O}_{\mathfrak{p}}$ with numerators having coefficients in $\mathcal{O}$.

As in Section 3.1, we shall consider extended families of Okutsu S-numerators of S-bases by adding the (formal) polynomial $g_{n_S} = \phi_S$ as the numerator of degree n_S,

$$\mathcal{N}_S := \left\{ g_{0,S}, \ldots, g_{n_S-1,S}, g_{n_S,S} = \phi_S := \prod_{\mathfrak{p}\in S} \phi_{\mathfrak{p}} \right\}.$$

Let $\mathfrak{T}$ be the tree of types of an OM representation of f. For any $\mathfrak{p} \in \mathcal{P}$ we denote by $\mathfrak{t}_\mathfrak{p}$ the leaf of $\mathfrak{T}$ corresponding to $\mathfrak{p}$.

Definition 3.10. *For $S \subseteq \mathcal{P}$, let $\mathfrak{T}_S \subseteq \mathfrak{T}$ be the subtree formed by all paths joining the leaves $\mathfrak{t}_\mathfrak{p}$, for $\mathfrak{p} \in S$, with each of their respective root nodes.*

For $\mathfrak{t} \in \mathfrak{T}_S$ we denote by $S_\mathfrak{t} \subseteq S$ the subset of all $\mathfrak{p} \in S$ such that $\mathfrak{t}$ belongs to the path joining $\mathfrak{t}_\mathfrak{p}$ to its root node.

Definition 3.11. *For a set S of prime ideals, we define,*

$$\mathrm{Ok}(S) = \left\{ \prod_{\mathfrak{p} \in S} g_{i_\mathfrak{p},\mathfrak{p}} : 0 \leqslant i_\mathfrak{p} \leqslant n_\mathfrak{p} \right\},$$

as the set of all polynomials that are a product of exactly one Okutsu $\mathfrak{p}$-numerator for each $\mathfrak{p} \in S$.

For a set $\{\mathfrak{p}\}$ containing a single prime ideal $\mathfrak{p} \in \mathcal{P}$, we simply use $\mathrm{Ok}(\mathfrak{p})$.

It should be noted that $\mathrm{Ok}(\mathfrak{p})$ coincides with the extended family $\mathcal{N}_\mathfrak{p}$ of numerators of the Okutsu $\mathfrak{p}$-basis $\mathcal{B}_\mathfrak{p}$ given in (3.1).

Here we are interested in how closely we can replicate the results of Lemma 3.5 for cross valuations, that is to say where the ϕ-polynomial belongs to a different prime to that of the valuation.

Lemma 3.12. *Let $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ be two different prime ideals with index of coincidence $\ell = i(\mathfrak{p}, \mathfrak{q})$. Then:*

1. $\hat{w}_\mathfrak{p}\left(\phi_{i,\mathfrak{q}}(\theta)\right) < \hat{w}_\mathfrak{p}\left(\phi_{i+1,\mathfrak{q}}(\theta)\right), \quad 1 \leqslant i < \ell,$

2. $\hat{w}_\mathfrak{p}\left(\phi_{i,\mathfrak{q}}(\theta)\right) = \hat{w}_\mathfrak{p}\left(\phi_{i+1,\mathfrak{q}}(\theta)\right), \quad \ell < i \leqslant r_\mathfrak{p}.$

Proof. For $i, j > \ell$, Proposition 2.31 shows that

$$\hat{w}_\mathfrak{p}\left(\phi_{i,\mathfrak{q}}(\theta)\right) = \hat{w}_\mathfrak{p}\left(\phi_{j,\mathfrak{q}}(\theta)\right).$$

This proves item (2).

For $i < \ell - 1$, we have $\phi_{i,\mathfrak{q}} = \phi_{i,\mathfrak{p}}$, $\phi_{i+1,\mathfrak{q}} = \phi_{i+1,\mathfrak{p}}$ and the inequality of (1) is a direct consequence of Lemma 3.5.

Assume $i = \ell - 1$. By Proposition 2.31, we have

$$\begin{aligned}\hat{w}_{\mathfrak{p}}\left(\phi_{\ell-1,\mathfrak{q}}(\theta)\right) &= \frac{1}{m_{\ell-1}} \cdot \frac{V_{\ell-1} + \lambda_{\ell-1}}{e_1 \cdots e_{\ell-2}} \\ &= \frac{1}{e_{\ell-1} f_{\ell-1} m_{\ell-1}} \cdot \frac{e_{\ell-1} f_{\ell-1} (e_{\ell-1} V_{\ell-1} + h_{\ell-1})}{e_1 \cdots e_{\ell-1}} \\ &= \frac{1}{m_\ell} \cdot \frac{V_\ell}{e_1 \cdots e_{\ell-1}},\end{aligned}$$

$$\hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right) = \frac{1}{m_\ell} \cdot \begin{cases} \dfrac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}}, & \text{or} \\ \dfrac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}}. \end{cases}$$

Hence, $\hat{w}_{\mathfrak{p}}\left(\phi_{\ell-1,\mathfrak{q}}(\theta)\right) < \hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right)$ and this ends the proof of (1). □

It is easy to find examples where

$$\hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right) > \hat{w}_{\mathfrak{p}}\left(\phi_{\ell+1,\mathfrak{q}}(\theta)\right). \tag{3.6}$$

Because of this fact, the proof of the validity of the MaxMin algorithm in Chapter 4 is rather involved.

This pathology occurs with $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{q})$ and $\lambda_{\mathfrak{p}}^{\mathfrak{q}}$ is much larger than $\lambda_{\mathfrak{q}}^{\mathfrak{p}}$ (see Proposition 2.31). Hence, it is also easy to find specific conditions that avoid (3.6).

Lemma 3.13. *Let $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ be two different prime ideals with $\ell = i(\mathfrak{p}, \mathfrak{q}) > 0$ and such that $\lambda_{\mathfrak{q}}^{\mathfrak{p}} \geqslant \lambda_{\mathfrak{p}}^{\mathfrak{q}}$. Then,*

$$\hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right) = \hat{w}_{\mathfrak{p}}\left(\phi_{\ell+1,\mathfrak{q}}(\theta)\right).$$

Proof. By the hypothesis, $\min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\} = \lambda_{\mathfrak{p}}^{\mathfrak{q}}$ and both cases for $i = \ell$ in Proposition 2.31 are equal, giving

$$\begin{aligned}\hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right) &= \frac{1}{m_\ell} \cdot \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}} \\ &= \frac{1}{m_{\ell+1}} \cdot \frac{m_{\ell+1}}{m_\ell} \cdot \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}} = \hat{w}_{\mathfrak{p}}\left(\phi_{\ell+1,\mathfrak{q}}(\theta)\right).\end{aligned}$$

□

Definition 3.14. *An* Okutsu S*-basis of* L *is a triangular* S*-basis with numerators* $\{g_i : 0 \leqslant i < n_S\}$*, such that* $\deg g_i = i$ *and* $g_i \in \mathrm{Ok}(S)$*, for all* $0 \leqslant i < n_S$*.*

Theorem 3.15, below, shows that Okutsu S-bases exist

3.3.2 Existence of partial Okutsu bases

A simple and very efficient algorithm, presented in Section 4.2, can be employed to choose an optimal combination of basis numerators for each degree d. In this section, we show that such an optimal combination will be optimal amongst all elements of Φ and therefore, by Theorem 3.9, amongst all polynomials with coefficients in the discrete valuation ring $\mathcal{O}$.

Theorem 3.15. *Let* $S \subseteq \mathcal{P}$ *be a set of prime ideals. For any* $\phi \in \Phi(S)$*, monic of degree* $0 \leqslant d \leqslant n_S$*, there exists* $g \in \mathrm{Ok}(S)$ *also monic and of degree* d *such that,*

$$w_{\mathfrak{p}}\left(g(\theta)\right) \geqslant w_{\mathfrak{p}}\left(\phi(\theta)\right), \qquad \forall\ \mathfrak{p} \in S.$$

In order to prove this theorem, we define a pair of transforms which, when used in conjunction, are able to convert any polynomial $\phi \in \Phi(S)$ to another polynomial $g \in \mathrm{Ok}(S)$, of equal or greater value, in a finite number of steps.

Firstly, we require certain measures on polynomials in $\Phi(S)$, which define how close they are to also being included in $\mathrm{Ok}(S)$.

Definition 3.16. *The irreducible factors of a polynomial* $\phi \in \Phi(S)$ *can be grouped by the prime ideal* $\mathfrak{p} \in S \subseteq \mathcal{P}$ *to which they belong,*

$$\begin{aligned} \phi &= \prod_{\mathfrak{p} \in S} g_{\mathfrak{p}}, \\ g_{\mathfrak{p}} &= \prod_{i=1}^{r_{\mathfrak{p}}} \phi_{i,\mathfrak{p}}^{a_i} \prod_{\varphi \in [F_{\mathfrak{p}}]} \varphi^{a_\varphi}, \qquad \mathfrak{p} \in S. \end{aligned}$$

We call $g_{\mathfrak{p}}$ *the* $\mathfrak{p}$-part *of* ϕ*. Using this grouping into* $\mathfrak{p}$*-parts, a measure*

of disorder can placed be on ϕ,

$$\begin{aligned} \mathrm{D} : \Phi(S) &\longrightarrow \mathbb{N} \\ \phi &\longmapsto \sum_{\mathfrak{p}\in S} max\left\{\deg(g_{\mathfrak{p}}) - n_{\mathfrak{p}}, 0\right\}. \end{aligned}$$

We recall that $\phi_{0,\mathfrak{p}} = x$ for all $\mathfrak{p} \in \mathcal{P}$ by convention.

Definition 3.17. *Let $\phi \in \Phi(\mathfrak{p})$ and $\ell \in \mathbb{N}$. We say that ϕ is* ℓ-canonical *if for all $0 \leqslant i \leqslant \min\{\ell, r_{\mathfrak{p}}\}$ we have $0 \leqslant \mathrm{ord}_{\phi_{i,\mathfrak{p}}}(\phi) < e_{i,\mathfrak{p}} f_{i,\mathfrak{p}}$.*

A polynomial $\varphi \in \Phi(S)$ is ℓ-canonical if each $\mathfrak{p}$-part of φ is ℓ-canonical.

Note that for any $m \in \mathbb{N}$, there is a unique $r_{\mathfrak{p}}$-canonical polynomial $\phi \in \Phi(\mathfrak{p})$ of degree m. We have necessarily $\phi = g_{i,\mathfrak{p}}\phi_{\mathfrak{p}}^a$, with $m = an_{\mathfrak{p}} + i$ and $0 \leqslant i < n_{\mathfrak{p}}$.

Recall that $e_{0,\mathfrak{p}} = 1$, $f_{0,\mathfrak{p}} = \deg(\psi_{0,\mathfrak{p}})$ for all $\mathfrak{p} \in \mathcal{P}$. We understand that "being (-1)-canonical" is an empty condition, so that all polynomials are (-1)-canonical.

By the construction of $\mathrm{Ok}(S)$, a polynomial $\phi \in \Phi(S)$ belongs to $\mathrm{Ok}(S)$ if, and only if, it is r-canonical for $r = \max\{r_{\mathfrak{p}}\}_{\mathfrak{p}\in S}$, and has disorder $\mathrm{D}(g) = 0$.

Making a polynomial $\varphi \in \Phi(S)$ r-canonical for $r = \max\{r_{\mathfrak{p}}\}_{\mathfrak{p}\in S}$ is not necessarily a straightforward task. Specifically, it is not sufficient to simply replace the $\mathfrak{p}$-part of φ with its r-canonical equivalent for each $\mathfrak{p} \in S$.

As an example, consider $S = \{\mathfrak{p}, \mathfrak{q}\} \subseteq \mathcal{P}$, such that $i(\mathfrak{p}, \mathfrak{q}) = 1$ and $\phi_{1,\mathfrak{p}} = \phi(\mathfrak{p}, \mathfrak{q}) = \phi_{1,\mathfrak{q}}$. Let $\mathfrak{p}$ and $\mathfrak{q}$ have the following OM invariants:

$$\begin{aligned} &\mathfrak{p} : e_1 = 1,\ f_1 = 4,\ h_1 = 1; \qquad e_2 = 1,\ f_2 = 3,\ h_2 = 9;\ \ldots \\ &\mathfrak{q} : e_1 = 1,\ f_1 = 3,\ h_1 = 2; \end{aligned}$$

for the first two levels for $\mathfrak{p}$ and the first level for $\mathfrak{q}$, and additionally $f_0 = 1$. Now consider $\phi = \phi_{2,\mathfrak{p}}\phi_{1,\mathfrak{p}}^4 \in \Phi(S)$, a polynomial of degree $\deg\phi = 8$. By Proposition 2.31, the $\mathfrak{p}$- and $\mathfrak{q}$-valuations for ϕ are respectively

$$\begin{aligned} w_{\mathfrak{p}}\left(\phi(\theta)\right) &= 13 \cdot 1 + 1 \cdot 4 = 17, \\ w_{\mathfrak{q}}\left(\phi(\theta)\right) &= 4 \cdot 1 + 2 \cdot 4 = 12. \end{aligned}$$

The polynomial ϕ is a product of numerators of the Okutsu $\mathfrak{p}$-basis only. The 2-canonical polynomial of the same degree is $g = \phi_{2,\mathfrak{p}}^2$. Again, referring to Proposition 2.31, we may calculate the valuations of g as

$$w_{\mathfrak{p}}\left(g(\theta)\right) = 13 \cdot 2 = 26,$$
$$w_{\mathfrak{q}}\left(g(\theta)\right) = 4 \cdot 2 = 8.$$

Here we see that not only is the $\mathfrak{q}$-valuation of g less than ϕ, but this is also the case for the S-valuation, $w_S\left(g(\theta)\right) < w_S\left(\phi(\theta)\right)$. As such, we require a more advanced transformation, one that does not just operate separately within each individual $\mathfrak{p}$-part of a polynomial.

Consider the transformation Canonify, which is presented in Algorithm 3.1. For the case $\ell = 0$, in step 2, we agree that $\{\mathfrak{t} \in \mathfrak{T}(S) : \mathfrak{t} \text{ of order } -1\}$ is the set of all root nodes of $\mathfrak{T}(S)$.

Lemma 3.18. *Let $U = \{\varphi_{\mathfrak{p}} \in \Phi(\mathfrak{p})\}_{\mathfrak{p}\in S}$ be a set of polynomials that are $(\ell-1)$-canonical. Let $U', b \leftarrow \text{Canonify}(U, \ell)$ be the result of the algorithm* Canonify. *Then either*

- *b is* `false` *and all polynomials in U' are ℓ-canonical; or*
- *b is* `true` *and the number of $r_{\mathfrak{p}}$-canonical polynomials in U' is strictly greater than those in U.*

Additionally, let $\phi = \prod_{\varphi_{\mathfrak{p}}\in U} \varphi_{\mathfrak{p}}$ and $g = \prod_{g_{\mathfrak{p}}\in U'} g_{\mathfrak{p}}$, then

$$w_{\mathfrak{q}}\left(g(\theta)\right) \geqslant w_{\mathfrak{q}}\left(\phi(\theta)\right), \qquad \forall\ \mathfrak{q} \in S,$$
$$D\left(g\right) \leqslant D\left(\phi\right).$$

Proof. There are four distinct cases in the **while** loop of Algorithm 3.1. In each case, one or more $g_{\mathfrak{p}}$ are changed. We will show that for each of these cases, the constraints of the lemma are maintained.

At each iteration of the **for** loop, a prime $\mathfrak{p} \in S\backslash S_0$ has two possibilities: either $\mathfrak{p} \notin S_{\mathfrak{t}}$, and then $i(\mathfrak{p}, \mathfrak{q}) < \ell$ for all $\mathfrak{q} \in S_{\mathfrak{t}}$, or $\mathfrak{p} \in S_{\mathfrak{t}}$ and then $g_{\mathfrak{p}}$ is an $r_{\mathfrak{p}}$-canonical polynomial of degree $d_{\mathfrak{p}} \geqslant n_{\mathfrak{p}}$.

Algorithm 3.1 Canonify$(\{\varphi_{\mathfrak{p}}\}_{\mathfrak{p}\in S}, \ell)$ transformation

Input: An integer $\ell \geqslant 0$ and a set of polynomials $\{\varphi_{\mathfrak{p}} \in \Phi(\mathfrak{p})\}_{\mathfrak{p}\in S}$ which are all $(\ell-1)$-canonical.

Output: A set of polynomials $\{g_{\mathfrak{p}} \in \Phi(\mathfrak{p})\}_{\mathfrak{p}\in S}$ and a boolean value b. If b is `false` then all $g_{\mathfrak{p}}$ are ℓ-canonical.

$g_{\mathfrak{p}} \leftarrow \varphi_{\mathfrak{p}}$, for all $\mathfrak{p} \in S$
for $\mathbb{t}$ **in** $\{\mathbb{t} \in \mathfrak{T}(S) : \mathbb{t}$ of order $\ell-1\}$ **do**
  $S_0 \leftarrow \{\mathfrak{p} \in S_{\mathbb{t}} : g_{\mathfrak{p}}$ not $r_{\mathfrak{p}}$-canonical or $\deg(g_{\mathfrak{p}}) < n_{\mathfrak{p}}\}$
  Order $S_0 = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{s_0}\}$ so that $\lambda_{\mathfrak{p}_i}^{\mathfrak{p}_j} \leqslant \lambda_{\mathfrak{p}_j}^{\mathfrak{p}_i}$ for all $1 \leqslant i < j \leqslant s_0$
  for $\mathfrak{p}_0$ **in** S_0 **do**
    while $\mathrm{ord}_{\phi_{\ell,\mathfrak{p}_0}}(g_{\mathfrak{p}_0}) \geqslant e_{\ell,\mathfrak{p}_0} f_{\ell,\mathfrak{p}_0}$ **do**
      if $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{p}_0}^{e_{\ell,\mathfrak{p}_0} f_{\ell,\mathfrak{p}_0}}(\theta)\right) \leqslant w_{\mathfrak{p}}\left(\phi_{\ell+1,\mathfrak{p}_0}(\theta)\right)$, $\forall\ \mathfrak{p} \in S_0$ **then**
        $g_{\mathfrak{p}_0} \leftarrow g_{\mathfrak{p}_0} \cdot \left(\phi_{\ell,\mathfrak{p}_0}^{e_{\ell,\mathfrak{p}_0} f_{\ell,\mathfrak{p}_0}}\right)^{-1} \cdot \phi_{\ell+1,\mathfrak{p}_0}$
      else
        $S_{0,\phi} = \{\mathfrak{p} \in S_0 : \phi_{\ell,\mathfrak{p}_0} \in \mathrm{Ref}_{\ell}(\mathbb{t}_{\mathfrak{p}})\}$
        $\mathfrak{q} \leftarrow$ prime ideal in $S_{0,\phi}$ such that $\lambda_{\mathfrak{q}}^{\mathfrak{p}} \geqslant \lambda_{\mathfrak{p}}^{\mathfrak{q}}$, $\forall\ \mathfrak{p} \in S_{0,\phi}$
        if $\deg(g_{\mathfrak{q}}) \geqslant n_{\mathfrak{q}}$ **then**
          $g_{\mathfrak{q}} \leftarrow g_{m,\mathfrak{q}}$, $r_{\mathfrak{q}}$-canonical polynomial of degree $m = \deg g_{\mathfrak{q}}$
          $S_0 \leftarrow S_0 \backslash \{\mathfrak{q}\}$
        else if $n_{\mathfrak{q}} - \deg(g_{\mathfrak{q}}) > m_{\ell}$ **then**
          $g_{\mathfrak{p}_0} \leftarrow g_{\mathfrak{p}_0} \cdot (\phi_{\ell,\mathfrak{p}_0})^{-1}$
          $g_{\mathfrak{q}} \leftarrow g_{\mathfrak{q}} \cdot \phi_{\ell,\mathfrak{q}}$
        else
          $m \leftarrow \deg(g_{\mathfrak{q}}) + m_{\ell} - n_{\mathfrak{q}}$
          $g_{\mathfrak{p}_0} \leftarrow g_{\mathfrak{p}_0} \cdot (\phi_{\ell,\mathfrak{p}_0})^{-1} \cdot g_{m,\mathfrak{p}_0}$
          $g_{\mathfrak{q}} \leftarrow \phi_{\mathfrak{q}}$
          $b \leftarrow$ `true`
          Exit algorithm.
        end if
      end if
    end while
  end for
end for
$b \leftarrow$ `false`

In the first case, the $\mathfrak{p}$-valuation of the product $g_{\mathfrak{q}}g_{\mathfrak{q}'}$ for any two $\mathfrak{q}, \mathfrak{q}' \in S_0$ will be unchanged, so long as $g_{\mathfrak{q}}$ and $g_{\mathfrak{q}'}$ both remain $(\ell-1)$-canonical and the sum of their degrees is constant.

In the second case, there is some $\phi_{\mathfrak{p}} \in [F_{\mathfrak{p}}] \cap \mathcal{O}[x]$ that divides $g_{\mathfrak{p}}$. We can apply the Single Factor Lifting algorithm (Section 2.6) to $\phi_{\mathfrak{p}}$ to raise its $\mathfrak{p}$-valuation as high as necessary to get $w_{\mathfrak{p}}(g(\theta)) \geqslant w_{\mathfrak{p}}(\phi(\theta))$ no matter what changes in $g_{\mathfrak{q}}$ for $\mathfrak{q} \neq \mathfrak{p}$ have been made.

We will, therefore, only consider the valuations of those prime ideals in S_0.

Case 1. $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{p}_0}^{e_{\ell,\mathfrak{p}_0}f_{\ell,\mathfrak{p}_0}}(\theta)\right) \leqslant w_{\mathfrak{p}}(\phi_{\ell+1,\mathfrak{p}_0}(\theta))$, $\forall\ \mathfrak{p} \in S_0$ (line 7). By the condition of this case, $w_{\mathfrak{p}}\left(g_{\mathfrak{p}_0}^{(\text{new})}(\theta)\right) \geqslant w_{\mathfrak{p}}\left(g_{\mathfrak{p}_0}^{(\text{old})}(\theta)\right)$, for all $\mathfrak{p} \in S_0$.

Case 2. $\exists\ \mathfrak{p} \in S_0$ such that $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{p}_0}^{e_{\ell,\mathfrak{p}_0}f_{\ell,\mathfrak{p}_0}}(\theta)\right) > w_{\mathfrak{p}}(\phi_{\ell+1,\mathfrak{p}_0}(\theta))$ (line 9). Here, we select $\mathfrak{q} \in S_{0,\phi}$ such that $\lambda_{\mathfrak{q}}^{\mathfrak{p}} \geqslant \lambda_{\mathfrak{p}}^{\mathfrak{q}}$, for all $\mathfrak{p} \in S_{0,\phi}$. The prime ideal $\mathfrak{q}$ has the property that,

$$\hat{w}_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}(\theta)) \leqslant \hat{w}(\phi_{i+1,\mathfrak{q}}(\theta)), \qquad \forall\ \mathfrak{p} \in S_0,\ \ell \leqslant i \leqslant r_{\mathfrak{q}}. \tag{3.7}$$

By Lemma 3.13, this is true for all $\mathfrak{p} \in S_{0,\phi}$ and for $\mathfrak{p} \in S_0 \backslash S_{0,\phi}$, we have $\phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{p},\mathfrak{q})$ and by Proposition 2.31 $\hat{w}_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}(\theta)) = \hat{w}(\phi_{\ell+1,\mathfrak{q}}(\theta))$. This will be important in the following sub-cases.

Case 2A. $\deg(g_{\mathfrak{q}}) \geqslant n_{\mathfrak{q}}$ (line 12). At this step, we replace $g_{\mathfrak{q}}$ with the $r_{\mathfrak{q}}$-canonical polynomial of the same degree. By (3.7), we can safely perform this operation. Since $\mathfrak{q}$ no longer meets the inclusion criteria, we remove it from S_0. We will now return to the beginning of the **while** loop, since it is possible that with the exclusion of $\mathfrak{q}$ from S_0, the condition for Case 1 will now be met.

Case 2B. $n_{\mathfrak{q}} - \deg(g_{\mathfrak{q}}) > m_{\ell}$ (line 15). In this case, for all $\mathfrak{p} \in S_{0,\phi}$ we have $\lambda_{\mathfrak{p}_0}^{\mathfrak{p}} = \min\left\{\lambda_{\mathfrak{p}_0}^{\mathfrak{p}}, \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}\right\} \leqslant \lambda_{\mathfrak{p}}^{\mathfrak{q}} = \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}$ which implies that $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{p}_0}(\theta)) \leqslant w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}(\theta))$.

For $\mathfrak{p} \in S_0 \backslash S_{0,\phi}$, due to the refinement process, $\lambda_{\mathfrak{p}_0}^{\mathfrak{p}} = \lambda_{\mathfrak{q}}^{\mathfrak{p}}$ and $\lambda_{\mathfrak{p}}^{\mathfrak{p}_0} = \lambda_{\mathfrak{p}}^{\mathfrak{q}}$ and so $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{p}_0}(\theta)) = w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}(\theta))$.

Case 2C. $n_{\mathfrak{q}} - \deg(g_{\mathfrak{q}}) \leqslant m_\ell$ (line 18). In this case, we must be careful not to increase the disorder of ϕ. Here we remove a single $\phi_{\ell,\mathfrak{p}}$ from $g_{\mathfrak{p}}$ and insert $g_{m,\mathfrak{p}_0}$, where $0 \leqslant m < m_\ell$. At the same time we replace $g_{\mathfrak{q}}$ with $\phi_{\mathfrak{q}}$. Actually, we can consider this step as

$$g_{\mathfrak{q}} \longrightarrow g_{\mathfrak{q}}(g_{m,\mathfrak{q}})^{-1}\phi_{\ell,\mathfrak{q}} \longrightarrow \phi_{\mathfrak{q}},$$

where $m < m_\ell$ implies that $g_{m,\mathfrak{q}} = g_{m,\mathfrak{p}}$. So for all $\mathfrak{p} \in S_0$, we have

$$\begin{aligned} w_{\mathfrak{p}}\left(g_{\mathfrak{p}_0}(\theta)g_{\mathfrak{q}}(\theta)\right) &\leqslant w_{\mathfrak{p}}\left(g_{\mathfrak{p}_0}(\theta)\phi_{\ell,\mathfrak{p}_0}(\theta)^{-1}g_{m,\mathfrak{p}_0}(\theta)\cdot g_{\mathfrak{q}}(\theta)g_{m,\mathfrak{q}}(\theta)^{-1}\phi_{\ell,\mathfrak{q}}(\theta)\right) \\ &\leqslant w_{\mathfrak{p}}\left(g_{\mathfrak{p}_0}(\theta)\phi_{\ell,\mathfrak{p}_0}(\theta)^{-1}g_{m,\mathfrak{p}_0}(\theta)\cdot \phi_{\mathfrak{q}}(\theta)\right). \end{aligned}$$

The first inequality is true by the argument given in Case 2B and the second inequality is given by (3.7) using the same argument as Case 2A.

Since we may have moved a polynomial of degree less than m_ℓ from the $\mathfrak{q}$-part to the $\mathfrak{p}_0$-part of ϕ, we cannot guarantee that the $\mathfrak{p}_0$-part will remain $(\ell-1)$-canonical. However, this change will not affect the $\mathfrak{p}$-valuation for any $\mathfrak{p} \in S$, since $g_{m,\mathfrak{q}} = g_{m,\mathfrak{p}}$.

At this step, $g_{\mathfrak{q}}$, which was previously of degree less than $n_{\mathfrak{q}}$ will become $r_{\mathfrak{q}}$-canonical of degree $n_{\mathfrak{q}}$, fulfilling the requirement attached to b being returned `true`.

All polynomials $\mathfrak{p} \in S_{\mathbb{t}}$ which are $r_{\mathfrak{p}}$-canonical of degree $d_{\mathfrak{p}} \geqslant n_{\mathfrak{p}}$ are not included in S_0 and so the the $\mathfrak{p}$-part is not changed during Canonify. This means that the number of polynomials fulfilling the condition attached to b being `true` will never decrease.

Since the algorithm works through each $\mathfrak{p}_0$ in S_0, if b is not set to `true` then all $g_{\mathfrak{p}_0}$ will be made ℓ-canonical by the condition of the while loop. □

Remark. It should be noted that when we run the Canonify transformation for $\ell = 0$, we may have the case that we transfer a $\phi_{0,\mathfrak{p}} = x$ from $\varphi_{\mathfrak{p}}$ for a $\mathfrak{p}$ with $f_{0,\mathfrak{p}} > 1$ to a $g_{\mathfrak{q}}$ where $f_{0,\mathfrak{q}} = 1$. In this situation, $\phi_{0,\mathfrak{q}} = \phi_{1,\mathfrak{q}} = x$ and although $\mathfrak{p}$ and $\mathfrak{q}$ belong to different disconnected trees, $w_{\mathfrak{q}}\left(\phi_{0,\mathfrak{p}}(\theta)\right) = w_{\mathfrak{q}}\left(\phi_{1,\mathfrak{q}}(\theta)\right) > 0$.

However, since this is only a conceptual change and doesn't materially

affect g as it differs from ϕ, there is no difference in the valuation.

Remark. We will never need to implement Algorithm 3.1! It is only a theoretical construction whose aim is to justify Theorem 3.15.

Definition 3.19. *The transformation* $\underset{\mathfrak{p}\to\mathfrak{q}}{\text{Transfer}}$ *for two prime ideals* $\mathfrak{p},\mathfrak{q}\in\mathcal{P}$*, converts a single* ϕ*-polynomial from the Okutsu* $\mathfrak{p}$*-frame into an element from the Okutsu* $\mathfrak{q}$*-basis, multiplied by a power of the Okutsu approximation to* $F_{\mathfrak{q}}$*.*

$$\begin{aligned}\underset{\mathfrak{p}\to\mathfrak{q}}{\text{Transfer}}:\ \Phi(\mathfrak{p})&\longrightarrow\Phi(\mathfrak{q})\\ \phi_{i,\mathfrak{p}}&\longmapsto g_{m,\mathfrak{q}},\end{aligned}$$

where $g_{m,\mathfrak{q}}\in\Phi(\mathfrak{q})$ *is the* $r_{\mathfrak{q}}$*-canonical polynomial of degree* $m=m_{i,\mathfrak{p}}$*.*

The transformation Transfer has non-trivial valuation properties, so $\mathfrak{p}$ and $\mathfrak{q}$ must be chosen carefully so that the $\mathfrak{l}$-valuation for specific $\mathfrak{l}\in S$ of the resulting polynomial is not less than the original $\phi_{i,\mathfrak{p}}$. To choose these polynomials, we require further information about the relative valuations of polynomials from the Okutsu bases of our prime ideals.

Using the extended index of coincidence presented in Definition 2.26, the following Lemma gives us a link between the relative alikeness of prime ideals and their respective cross-valuations.

Lemma 3.20. *Let* $S\in\mathcal{P}$ *be a set of prime ideals and fix a prime ideal* $\mathfrak{p}_0\in S$*. Now select* $\mathfrak{q}\in S\backslash\{\mathfrak{p}_0\}$ *such that* $I(\mathfrak{p}_0,\mathfrak{q})\geqslant I(\mathfrak{p}_0,\mathfrak{p})$ *and in the case of equality* $\lambda_{\mathfrak{q}}^{\mathfrak{p}_0}\geqslant\lambda_{\mathfrak{p}}^{p_0}$ *for all* $\mathfrak{p}\in S\backslash\{\mathfrak{p}_0\}$*. Then, for any* $1\leqslant i\leqslant r_{\mathfrak{p}_0}$*, if we take* $g_{\mathfrak{q}}:=\underset{\mathfrak{p}_0\to\mathfrak{q}}{\text{Transfer}}\left(\phi_{i,\mathfrak{p}_0}\right)$*, we have,*

$$w_{\mathfrak{p}}\left(g_{\mathfrak{q}}(\theta)\right)\geqslant w_{\mathfrak{p}}\left(\phi_i,\mathfrak{p}_0(\theta)\right),\qquad\forall\ \mathfrak{p}\in S\backslash\{\mathfrak{p}_0\}\,.\tag{3.8}$$

Proof. Let $\ell=i(\mathfrak{p}_0,\mathfrak{q})$, then if $i<\ell$ then $g_{\mathfrak{q}}=\phi_{i,\mathfrak{q}}=\phi_{i,\mathfrak{p}_0}$, so we only need to consider $i\geqslant\ell$, in which case $m_\ell\mid m_{i,\mathfrak{p}_0}$ and so $g_{\mathfrak{q}}$ is the product of ϕ-polynomials of index ℓ and greater. Meanwhile, by the maximality of the numerators of an Okutsu basis, $w_{\mathfrak{q}}\left(g_{\mathfrak{q}}\right)$ is maximal amongst all monic

polynomials of degree m. Hence the property required of (3.8) is clear for $\mathfrak{p} = \mathfrak{q}$.

Consider the following subsets of S,

$$S_1 := \{\mathfrak{p} \in S \setminus \{\mathfrak{p}_0, \mathfrak{q}\} : i(\mathfrak{p}_0, \mathfrak{p}) \geqslant i(\mathfrak{p}_0, \mathfrak{q})\},$$
$$S_0 := \{\mathfrak{p} \in S_1 : I(\mathfrak{p}_0, \mathfrak{p}) \geqslant I(\mathfrak{p}_0, \mathfrak{q})\}.$$

By the choice of $\mathfrak{q}$, there are no prime ideals in S that satisfy the strict inequality in the set inclusion conditions and they may be replaced by $i(\mathfrak{p}_0, \mathfrak{p}) = i(\mathfrak{p}_0, \mathfrak{q})$ and $I(\mathfrak{p}_0, \mathfrak{p}) = I(\mathfrak{p}_0, \mathfrak{q})$ respectively.

We will examine the prime ideals delimited by these sets separately.

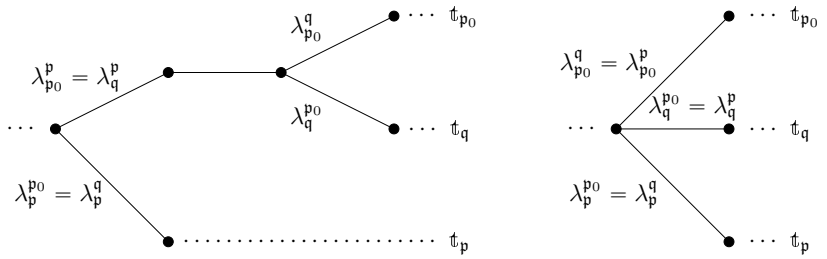
Case 1. $\mathfrak{p} \in S \setminus S_1$. We have $i(\mathfrak{p}_0, \mathfrak{p}) < i(\mathfrak{p}_0, \mathfrak{q})$, which implies $i(\mathfrak{p}_0, \mathfrak{p}) = i(\mathfrak{q}, \mathfrak{p})$ and so by Proposition 2.31,

$$w_{\mathfrak{p}}(g_{\mathfrak{q}}(\theta)) = w_{\mathfrak{p}}(\phi_{i,\mathfrak{p}_0}(\theta)), \qquad 1 \leqslant i \leqslant r_{\mathfrak{p}_0}, \ \forall\ \mathfrak{p} \in S \setminus S_1,$$

In fact, since $i(\mathfrak{p}_0, \mathfrak{p}) < i(\mathfrak{p}_0, \mathfrak{q})$, the result of Transfer is $g_{\mathfrak{q}} = \phi_{i,\mathfrak{q}} = \phi_{i,\mathfrak{p}_0}$ for all $i \leqslant i(\mathfrak{p}_0, \mathfrak{p})$. On the other hand, if $i > i(\mathfrak{p}_0, \mathfrak{p}) = i(\mathfrak{q}, \mathfrak{p})$, the $\mathfrak{p}$-valuation depends only on the degree.

The following two cases are illustrated in Figure 3.3. Note that although Figure 3.3 represents non-optimised trees, the hidden slopes we have written are the optimised ones.

Case 2. $\mathfrak{p} \in S_1 \setminus S_0$. Here, $\mathfrak{p}_0$, $\mathfrak{q}$, and $\mathfrak{p}$ share a common index of coincidence $\ell = i(\mathfrak{p}_0, \mathfrak{q}) = i(\mathfrak{p}_0, \mathfrak{p}) = i(\mathfrak{q}, \mathfrak{p})$, but $I(\mathfrak{p}_0, \mathfrak{q}) > I(\mathfrak{p}_0, \mathfrak{p})$. From Figure 3.3 we



$I(\mathfrak{p}_0, \mathfrak{q}) > I(\mathfrak{p}_0, \mathfrak{p})$ $(\mathfrak{q} \notin S_0)$. $I(\mathfrak{p}_0, \mathfrak{q}) = I(\mathfrak{p}_0, \mathfrak{p})$ $(\mathfrak{q} \in S_0)$.

Figure 3.3: Relative positions of $\mathbb{t}_{\mathfrak{p}_0}$, $\mathbb{t}_{\mathfrak{q}}$, and $\mathbb{t}_{\mathfrak{p}}$ in the non-optimised tree.

can see that in this case $\phi_{\ell,\mathfrak{p}_0} \neq \phi(\mathfrak{p}_0,\mathfrak{p})$, $\phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{q},\mathfrak{p})$ and also $\lambda_{\mathfrak{p}_0}^{\mathfrak{p}} = \lambda_{\mathfrak{q}}^{\mathfrak{p}}$ and $\lambda_{\mathfrak{p}}^{\mathfrak{p}_0} = \lambda_{\mathfrak{p}}^{\mathfrak{q}}$.

Therefore, by Proposition 2.31,

$$\begin{aligned} w_{\mathfrak{p}}\left(g_{\mathfrak{q}}(\theta)\right) &= \frac{\deg g_{\mathfrak{q}}}{m_\ell} \cdot \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}} \\ &= \frac{m_{i,\mathfrak{p}_0}}{m_\ell} \cdot \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}_0}, \lambda_{\mathfrak{p}_0}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}} \\ &= w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}_0}(\theta)\right). \end{aligned}$$

Case 3. $\mathfrak{p} \in S_0$. Finally, we have the case where $I(\mathfrak{p}_0,\mathfrak{q}) = I(\mathfrak{p}_0,\mathfrak{p})$, as shown in Figure 3.3, which implies $\phi(\mathfrak{p}_0,\mathfrak{q}) = \phi(\mathfrak{q},\mathfrak{p})$ and by the hypothesis on $\mathfrak{q}$ we have $\min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\} = \lambda_{\mathfrak{p}}^{\mathfrak{q}}$ and so by Proposition 2.31,

$$\begin{aligned} w_{\mathfrak{p}}\left(g_{\mathfrak{q}}(\theta)\right) &= \frac{\deg g_{\mathfrak{q}}}{m_\ell} \cdot \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}} \\ &= \frac{m_{i,\mathfrak{p}_0}}{m_\ell} \cdot \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}}{e_1 \cdots e_{\ell-1}} \\ &\geqslant \begin{cases} \dfrac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}}{e_1 \cdots e_{\ell-1}} = w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}_0}(\theta)\right), & \text{if } i = \ell, \\ \dfrac{m_{i,\mathfrak{p}_0}}{m_\ell} \cdot \dfrac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}_0}, \lambda_{\mathfrak{p}_0}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}} = w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}_0}(\theta)\right), & \text{if } i > \ell. \end{cases} \end{aligned}$$

□

We may now prove Theorem 3.15.

Proof of Theorem 3.15. We will use the previously defined transformations Canonify and Transfer iteratively, to convert a polynomial $\phi \in \Phi(S)$ into a polynomial $g \in \mathrm{Ok}(S)$. At each step the $\mathfrak{p}$-valuation will be preserved or raised for all $\mathfrak{p} \in S$ and the disorder will be reduced.

Consider the polynomial ϕ', which we initially set to ϕ.

Step 1. First, we consider the $\mathfrak{p}$-part $g_{\mathfrak{p}}$ of ϕ' for all $\mathfrak{p}$ in S. For $0 \leqslant \ell \leqslant r_{\max} = \max\left\{r_{\mathfrak{p}}\right\}_{\mathfrak{p}\in S}$, apply Canonify$(\left\{g_{\mathfrak{p}}\right\}_{\mathfrak{p}\in S}, \ell)$. If b is returned `true`, then restart ℓ at 1.

By the condition in Lemma 3.18 that when b returns `true` the number of $r_{\mathfrak{p}}$-canonical polynomials with degree $d_{\mathfrak{p}} \geqslant n_{\mathfrak{p}}$ increases (and even when

b returns `false` it may never decrease), b can only return `true` up to $\#S$ times. Therefore, this process will complete in a finite number of iterations. Once Canonify has been run successfully up to $\ell = r_{\max}$, all $g_{\mathfrak{p}}$ will be $r_{\mathfrak{p}}$-canonical.

Set $\phi' = \prod_{\mathfrak{p}\in S} g_{\mathfrak{p}}$. By Lemma 3.18, this ϕ' will have greater or equal $\mathfrak{p}$-valuation for all $\mathfrak{p} \in S$ and will have lesser or equal disorder. Additionally, it will be $r_{\max}$-canonical.

Step 2. Consider the set,

$$\text{Overloaded}(\phi') = \{\mathfrak{p} \in S : \deg(g_{\mathfrak{p}}) > n_{\mathfrak{p}}\},$$

of all prime ideals $\mathfrak{p}$ for which the degree of $g_{\mathfrak{p}}$, the $\mathfrak{p}$-component of ϕ', is greater than $n_{\mathfrak{p}}$ the degree of the prime ideal itself.

If $\text{Overloaded}(\phi')$ is empty, then the disorder $\mathrm{D}(\phi')$ of ϕ' is 0 and we have $g := \phi' \in \mathrm{Ok}(S)$, so we have finished.

Step 3. In the case that $\text{Overloaded}(\phi')$ is not empty, we select an arbitrary $\mathfrak{p}_0 \in \text{Overloaded}(\phi')$ and consider the set $S' = S \setminus \text{Overloaded}(\phi')$. Now, select a prime ideal $\mathfrak{q} \in S'$ such that $I(\mathfrak{p}_0, \mathfrak{q}) \geqslant I(\mathfrak{p}_0, \mathfrak{p})$ and in the case of equality $\lambda_{\mathfrak{q}}^{\mathfrak{p}_0} \geqslant \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}$ for all $\mathfrak{p} \in S'$ as in Lemma 3.20.

We then apply $\underset{\mathfrak{p}_0\to\mathfrak{q}}{\text{Transfer}}$ on ϕ', converting the least degree $\phi_{i,\mathfrak{p}_0}$ dividing $g_{\mathfrak{p}_0}$ into the $r_{\mathfrak{q}}$-canonical polynomial $g_{m,\mathfrak{q}} \in \Phi(\mathfrak{q})$ of degree $m = m_{i,\mathfrak{p}_0}$.

By the selection of $\mathfrak{q}$, we have $I(\mathfrak{p}_0, \mathfrak{q}) \geqslant I(\mathfrak{p}_0, \mathfrak{p})$ for all prime ideals $\mathfrak{p} \in S \setminus \text{Overloaded}(\phi')$, and in the case of equality $\lambda_{\mathfrak{q}}^{\mathfrak{p}_0} \geqslant \lambda_{\mathfrak{p}}^{\mathfrak{p}_0}$. By these conditions, Lemma 3.20 shows that

$$w_{\mathfrak{p}}\left(g_{m,\mathfrak{q}}(\theta)\right) \geqslant w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{p}_0}(\theta)\right), \qquad \forall\ \mathfrak{p} \in S'.$$

For the remaining $\mathfrak{p} \in \text{Overloaded}(\phi')$, the $\mathfrak{p}$-valuation of ϕ' can be raised by applying the Single Factor Lifting algorithm to $\phi_{\mathfrak{p}}$, and so this covers all $\mathfrak{p} \in S$.

Since we are removing a degree $m_{i,\mathfrak{p}_0}$ polynomial from $g_{\mathfrak{p}_0}$ and including a polynomial of the same degree in $g_{\mathfrak{q}}$, the disorder will be reduced by $\min\{n_{\mathfrak{q}} - \deg g_{\mathfrak{q}}, m_{i,\mathfrak{p}_0}\}$. Since $\mathfrak{q}$ was chosen so that $\deg(g_{\mathfrak{q}}) < n_{\mathfrak{q}}$, this

reduction in disorder must be at least 1.

We now return to Step 1.

Clearly, this process will terminate after at most $\mathrm{D}(\phi)$ iterations. □

4

MaxMin

"Never accept the proposition that just because a solution satisfies a problem, that it must be the only solution."

– Raymond E. Feist, *Magician*

In this chapter we describe the MaxMin algorithm, its input requirements and output properties.

We retain the setting from the previous chapter.

4.1 Formal extension of the Okutsu $\mathfrak{p}$-bases

By Theorem 1.16, in order to construct a reduced triangular $\mathcal{O}$-basis of $\mathcal{O}_L$ we need only to find a family of monic w-optimal polynomials of degree $0, 1, \ldots, n-1$. By Theorem 3.9 and Theorem 3.15, we may restrict our

search to polynomials constructed as the product of exactly one numerator of each Okutsu $\mathfrak{p}$-basis, for $\mathfrak{p} \in \mathcal{P}$.

The aim of the MaxMin algorithm is to perform an efficient search for these optimal polynomials in $\mathrm{Ok}(\mathcal{P})$.

To decide which numerators are chosen for each degree, we need only to know the values $w_{\mathfrak{q}}\left(g_{i_{\mathfrak{p}},\mathfrak{p}}(\theta)\right)$ for all $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ and $0 \leqslant i_{\mathfrak{p}} \leqslant n_{\mathfrak{p}}$. As presented in Chapter 2, these values are given by OM invariants present in an OM representation of f, which is provided by the Montes algorithm. The exception is $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right)$, which can be arbitrarily large, depending on the choice of $\phi_{\mathfrak{p}}$ the Montes approximation to $F_{\mathfrak{p}}$ as a factor of f.

For this reason, we do not choose a concrete polynomial $\phi_{\mathfrak{p}}$ beforehand, but rather run the algorithm as if $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right)$ (formally) takes the value ∞. We define the following valuation function on polynomials in Φ:

Definition 4.1. *For all* $\mathfrak{p} \in \mathcal{P}$*,*

$$
\begin{aligned}
w_{\mathfrak{p}} : \Phi &\longrightarrow \mathbb{Q} \cup \{\infty\} \\
\phi &\longmapsto \begin{cases} w_{\mathfrak{p}}\left(\phi(\theta)\right), & \text{if } \phi^* \nmid \phi, \forall\ \phi^* \in \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}}), \\ \infty, & \text{if } \phi^* \mid \phi, \text{ for some } \phi^* \in \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}}). \end{cases}
\end{aligned}
$$

Therefore, $w_{\mathfrak{p}}\left(g\right) = w_{\mathfrak{p}}\left(g(\theta)\right)$ if $\phi_{\mathfrak{p}} \nmid g$, for all $\phi_{\mathfrak{p}} \in \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}})$, however $w_{\mathfrak{p}}\left(g\right) = \infty$ if $\phi_{\mathfrak{p}} \mid g$ for some $\phi_{\mathfrak{p}} \in \mathrm{Rep}(\mathbb{t}_{\mathfrak{p}})$. This is practical as by Corollary 2.32 the value of $w_{\mathfrak{q}}\left(\phi_{\mathfrak{p}}(\theta)\right)$ for $\mathfrak{q} \neq \mathfrak{p}$ only depends on $\mathfrak{q}$ and not the choice of $\phi_{\mathfrak{p}}$.

This valuation also extends to the $\mathcal{P}$-valuation for the set $\mathcal{P}$ of prime ideals (see Section 3.2). Therefore, $w\left(g\right) = w\left(g(\theta)\right)$ if for every $\phi_{\mathfrak{p}}$ dividing g, we take a concrete choice for this polynomial with $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right)$ sufficiently large.

In other words, $\phi_{\mathfrak{p}}$ is considered to be a symbolic polynomial. Its degree is known to be $n_{\mathfrak{p}}$ and its $\mathfrak{q}$-valuation for all $\mathfrak{q} \neq \mathfrak{p}$, which is given by Proposition 2.31, does not depend on the concrete choice of $\phi_{\mathfrak{p}}$ by Corollary 2.32. The algorithm will provide a recipe to construct numerators $g_i \in \mathcal{O}[x]$ of degree i as a product of Oktusu $\mathfrak{p}$-numerators for each $\mathfrak{p}$ in $\mathcal{P}$. The

corresponding member of the triangular basis will be

$$\alpha_i = g_i(\theta)\pi^{-\lfloor w(g_i)\rfloor}, \qquad 0 \leqslant i < n.$$

In order to compute α_i, we must apply the Single Factor Lifting algorithm to find concrete Okutsu approximations $\phi_{\mathfrak{p}}$, with a valuation $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right)$ large enough to guarantee that $w\left(g_i\right) = w\left(g_i(\theta)\right)$ for all $0 \leqslant i < n$.

4.2 The MaxMin algorithm

We fix an ordering $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_N\}$ on the set $\mathcal{P}$, with the property that for all types $\mathfrak{t}$ in the tree of types $\mathfrak{T}$, the subset $\mathcal{P}_{\mathfrak{t}} \subseteq \mathcal{P}$ of prime ideals whose genomic tree contains the type $\mathfrak{t}$ is an interval of $\mathcal{P}$. That is, there exist indices $1 \leqslant a_{\mathfrak{t}} \leqslant b_{\mathfrak{t}} \leqslant N$ such that,

$$\mathcal{P}_{\mathfrak{t}} = [a_{\mathfrak{t}}, b_{\mathfrak{t}}] := \{\mathfrak{p}_j : a_{\mathfrak{t}} \leqslant j \leqslant b_{\mathfrak{t}}\}. \tag{4.1}$$

As the branches of $\mathfrak{T}$ do not cross one-another, the reader will easily be convinced that it is always possible to consider such an ordering.

Consider a subset $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\} \subseteq \mathcal{P}$ with the induced ordering, and extended Okutsu $\mathfrak{p}$-numerators $\left\{g_{i_{\mathfrak{p}},\mathfrak{p}} : 0 \leqslant i_{\mathfrak{p}} \leqslant n_{\mathfrak{p}}\right\}$ for each $\mathfrak{p} \in S$, as indicated in Section 4.1.

We consider multi-indices $\mathbb{i} = (i_{\mathfrak{q}})_{\mathfrak{q}\in S}$ of degree

$$\deg \mathbb{i} := \sum_{\mathfrak{q}\in S} i_{\mathfrak{q}},$$

leading to monic polynomials in $\mathcal{O}[x]$:

$$g_{\mathbb{i}} := \prod_{\mathfrak{q}\in S} g_{i_{\mathfrak{q}},\mathfrak{q}},$$

with $\deg g_{\mathbb{i}} = \deg \mathbb{i}$.

Definition 4.2. *A multi-index $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ is said to be* maximal *if*

$$w_S\left(g_{\mathbb{i}}\right) \geqslant w_S\left(g_{\mathbb{j}}\right),$$

for all multi-indices $\mathbb{j}$ *with* $\deg \mathbb{j} = \deg \mathbb{i}$.

In this case, we also say that $g_{\mathbb{i}}$ *is a* maximal numerator.

Notation. *For* $1 \leqslant j \leqslant s$ *we denote by* $\mathbb{u}_j$ *the multi-index with coordinates* $i_{\mathfrak{q}} = 0$ *for all* $\mathfrak{q} \neq \mathfrak{q}_j$ *and* $i_{\mathfrak{q}_j} = 1$.

Algorithm 4.1 MaxMin[S] algorithm

Input: A subset $S = \{\mathfrak{q}_1, \ldots, \mathfrak{q}_s\} \subseteq \mathcal{P}$ and Okutsu numerators $\{g_{i,\mathfrak{q}} : 0 \leqslant i \leqslant n_{\mathfrak{q}}\}$ of $\mathfrak{q}$-bases for each $\mathfrak{q} \in S$.

Output: A family $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n_S} \in \mathbb{N}^s$ of multi-indices of degree $0, 1, \ldots, n_S$ respectively.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0)$
2: **for** $k = 0 \rightarrow n_S - 1$ **do**
3: $\quad j \leftarrow \min\{1 \leqslant i \leqslant s : w_{\mathfrak{q}_i}(g_{\mathbb{i}_k}) = w_S(g_{\mathbb{i}_k})\}$
4: $\quad \mathbb{i}_{k+1} \leftarrow \mathbb{i}_k + \mathbb{u}_j$
5: **end for**

The main aim of this chapter is to prove the following result.

Theorem 4.3. *If* $\mathfrak{T}_S$ *is a connected tree, then all output multi-indices of* MaxMin[S] *are maximal.*

This gives the name MaxMin for the algorithm, because it finds the maximal value amongst the minima of certain numerical data. This provides a computation of an Okutsu S-basis of L as follows.

Theorem 4.4. *Let* $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n_S}$ *be an output of* MaxMin[S]. *Choose Okutsu approximations* $\phi_{\mathfrak{p}}$ *of all* $\mathfrak{p} \in S$, *such that*

$$w_S(g_{\mathbb{i}_k}) = w_S(g_{\mathbb{i}_k}(\theta)), \qquad 0 \leqslant k < n_S.$$

Then, $g_{\mathbb{i}_0}, g_{\mathbb{i}_1}, \ldots, g_{\mathbb{i}_{n_S-1}}$ *are numerators of an Okutsu* S*-basis of* L.

In fact, Theorems 3.9 and 3.15 allow us to conclude from Theorem 4.3 that $g_{\mathbb{i}_0}, \ldots, g_{\mathbb{i}_{n_S-1}}$ are numerators of a reduced triangular S-basis of L. Since all $g_{\mathbb{i}_k}$ belong to $\mathrm{Ok}(S)$, this triangular basis is an Okutsu S-basis.

The discussion about finding the required valuation of all $w_{\mathfrak{p}}(\phi_{\mathfrak{p}}(\theta))$ is postponed until Section 4.7.

We will now present some remarks about the behaviour of the algorithm.

4.2.1 Guaranteed termination

It is trivial to see that the algorithm MaxMin[S] always terminates after exactly n_S iterations.

Thanks to the convention $w_{\mathfrak{p}}(\phi_{\mathfrak{p}}) = \infty$, the index j in step 3 indicates a prime $\mathfrak{q}_j$ such that for the multi-index $\mathbb{i}_k = (i_{\mathfrak{q}})_{\mathfrak{q}\in S}$, we will always have $i_{\mathfrak{q}_j} < r_{\mathfrak{q}_j}$. Therefore, the next multi-index $\mathbb{i}_{k+1} = (i'_{\mathfrak{q}})_{\mathfrak{q}\in S}$ constructed in step 4 has indices $i'_{\mathfrak{q}} \leqslant r_{\mathfrak{q}}$ for all $\mathfrak{q} \in S$.

Furthermore, the first and last output multi-indices are $\mathbb{i}_0 = (0, \ldots, 0)$ and $\mathbb{i}_{n_S} = (n_{\mathfrak{q}_1}, \ldots, n_{\mathfrak{q}_s})$. As such, $g_{\mathbb{i}_0} = 1$ and $g_{\mathbb{i}_{n_S}} = \prod_{\mathfrak{q}\in S} \phi_{\mathfrak{q}}$ and $g_{\mathbb{i}_0}, \ldots, g_{\mathbb{i}_{n_S}}$ is an extended family of numerators of an Okutsu S-basis of L, according to the convention introduced in Section 3.3.1.

4.2.2 Polynomial products are not computed

The algorithm does not compute the products $g_{\mathbb{i}_k}$. It only computes the values $w_{\mathfrak{q}}(g_{\mathbb{i}_k})$ for $\mathfrak{q} \in S$, which are determined by the 3-dimensional array of data $w_{\mathfrak{q}_k}(g_{j,\mathfrak{q}_i})$ indexed by i, j, and k in the ranges $1 \leqslant i \leqslant s$, $0 \leqslant j_i \leqslant n_{\mathfrak{q}_i}$, and $1 \leqslant k \leqslant s$, respectively.

If these numbers are replaced by arbitrary, non-negative rational numbers $\nu_{k,j_i,i} \in \mathbb{Q}_{>0}$ and we take

$$\nu_{k,\mathbb{i}} := \sum_{i=1}^{s} \nu_{k,j_i,i},$$

with $\mathbb{i} = (j_i)_{1\leqslant i\leqslant s}$ a multi-index as above, the MaxMin routine may fail to compute

$$\max\{\min\{\nu_{k,\mathbb{i}} : 1 \leqslant k \leqslant s\} : \deg \mathbb{i} = d\},$$

a maximal multi-index of degree d.

4.2.3 Initial conditions

Suppose $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ is a multi-index with degree $\deg \mathbb{i} = d$, such that $w_S(g_{\mathbb{i}})$ is maximal amongst all multi-indices of degree d. Then, it may not be true that by increasing an adequate index by one, we get a multi-index $\mathbb{j}$, of

degree $d+1$, which renders a maximal value of $w_S(g_{\mathbb{j}})$ amongst all multi-indices of degree $k+1$.

For instance, let us consider the example presented in Section 4.2.5. The output index of degree 3 is $\mathbb{i}_3 = (1,2,0)$, resulting in the polynomial $g_3 = \phi_{1,\mathfrak{p}}\phi_{1,\mathfrak{q}}^2$ with valuations $w(g_3) = (10,8,8)$ for $w_{\mathfrak{p}}$, $w_{\mathfrak{q}}$ and $w_{\mathfrak{l}}$ respectively.

We could choose an alternative index $\mathbb{j}_3 = (1,1,1)$ which would give a polynomial $g_3' = \phi_{1,\mathfrak{p}}\phi_{1,\mathfrak{q}}\phi_{1,\mathfrak{l}}$ with the exact same valuations $\vec{w}(g_3') = (10,8,8)$. However, in the second case, the next index would be $\mathbb{j}_4 = (1,2,1)$ giving the polynomial $g_4' = \phi_{1,\mathfrak{p}}\phi_{1,\mathfrak{q}}^2\phi_{1,\mathfrak{l}}$ with valuations $\vec{w}(g_4') = (12,11,11)$. This is clearly not maximal as the polynomial constructed by the example $g_4 = \phi_{1,\mathfrak{p}}\phi_{2,\mathfrak{q}}$ has valuations $w(g_4) = (12,17,16)$.

However, the maximal multi-indices that are met along the flow of the algorithm will lead to subsequent maximal indices. It is curious that the (extremely) simple strategy that MaxMin employs to choose successive maximal multi-indices is able to avoid these pathological cases.

4.2.4 Ordering of input prime ideals

Theorem 4.3 shows that MaxMin produces a sequence of maximal multi-indices regardless of the choice of ordering on $\mathcal{P}$, as long as it satisfies (4.1). However, the numerators $g_{\mathbb{i}_k}$ produced from these multi-indices do depend on the choice of ordering.

Condition (4.1) on the ordering of $\mathcal{P}$ has only been imposed for the purpose of simplifying the proof of Theorem 4.3, which appears to be true for an arbitrary ordering on $\mathcal{P}$. However, we have not been able to give a rigorous proof of this fact.

4.2.5 MaxMin Example

We will now present a small example for $S = \{\mathfrak{p},\mathfrak{q},\mathfrak{l}\}$ where $\mathfrak{T}_S$ is connected.

Consider the tree $\mathfrak{T}_S^{\text{nop}}$ shown in Figure 4.1. We indicate only the data (ϕ,λ) for each edge.

Since all slopes have integer values, all denominators e_i are equal to one. Hence, for our choices of "good" and "bad" edges to be coherent, we must

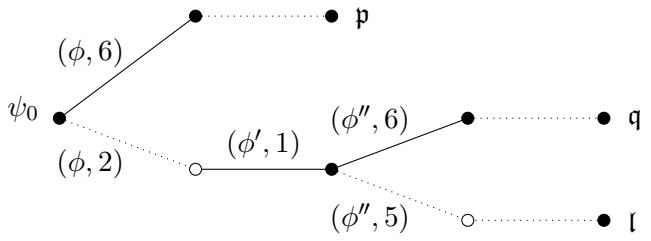


Figure 4.1: Example non-optimised connected tree $\mathfrak{T}_S^{\text{nop}}$ of types.

have $f_{1,\mathfrak{p}}, f_{1,\mathfrak{q}} = f_{1,\mathfrak{l}}$, and $f_{2,\mathfrak{q}}$ greater than one. For instance:

$$\begin{aligned}
\mathfrak{p}: &\quad e_1 = 1, f_1 = 4, h_1 = 6; \\
\mathfrak{q}: &\quad e_1 = 1, f_1 = 3, h_1 = 3; \quad e_2 = 1, f_2 = 2, h_2 = 6; \\
\mathfrak{l}: &\quad e_1 = 1, f_1 = 3, h_1 = 3.
\end{aligned}$$

The data corresponding to the edges leading to a leaf are not specified as we do not need them to run MaxMin.

The optimised tree $\mathfrak{T}_S^{\text{op}}$ is shown in Figure 4.2.

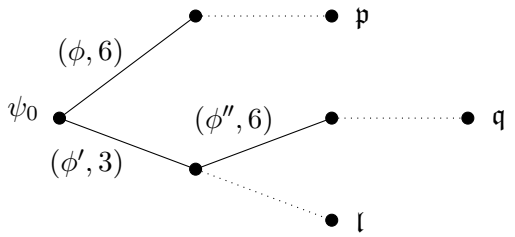


Figure 4.2: Example optimised connected tree $\mathfrak{T}_S^{\text{op}}$ of types.

Here we can see the relationship between the polynomials in the non-optimised and optimised trees,

$$\begin{aligned}
\phi(\mathfrak{p}, \mathfrak{q}) = \phi(\mathfrak{p}, \mathfrak{l}) = \phi_{1,\mathfrak{p}} &= \phi, \\
\phi_{1,\mathfrak{q}} = \phi_{1,\mathfrak{l}} &= \phi', \\
\phi(\mathfrak{q}, \mathfrak{l}) = \phi_{2,\mathfrak{q}} &= \phi'',
\end{aligned}$$

and the optimised hidden slopes are:

$$\lambda_{\mathfrak{p}}^{\mathfrak{q}} = \lambda_{\mathfrak{p}}^{\mathfrak{l}} = 6, \qquad \lambda_{\mathfrak{q}}^{\mathfrak{p}} = \lambda_{\mathfrak{l}}^{\mathfrak{p}} = 2,$$
$$\lambda_{\mathfrak{q}}^{\mathfrak{l}} = 6, \qquad \lambda_{\mathfrak{l}}^{\mathfrak{q}} = 5.$$

The numerators of the extended Okutsu bases of each of the three prime ideals will be,

$$\begin{aligned}
\mathcal{N}_{\mathfrak{p}} &: \ 1,\ \phi_{1,\mathfrak{p}},\ \phi_{1,\mathfrak{p}}^2,\ \phi_{1,\mathfrak{p}}^3,\ \phi_{\mathfrak{p}};\\
\mathcal{N}_{\mathfrak{q}} &: \ 1,\ \phi_{1,\mathfrak{q}},\ \phi_{1,\mathfrak{q}}^2,\ \phi_{2,\mathfrak{q}},\ \phi_{2,\mathfrak{q}}\phi_{1,\mathfrak{q}},\ \phi_{2,\mathfrak{q}}\phi_{1,\mathfrak{q}}^2,\ \phi_{\mathfrak{q}};\\
\mathcal{N}_{\mathfrak{l}} &: \ 1,\ \phi_{1,\mathfrak{l}},\ \phi_{1,\mathfrak{l}}^2,\ \phi_{\mathfrak{l}}.
\end{aligned}$$

Using the explicit formulas of Proposition 2.31, we may compute the valuations of each of the ϕ-polynomials. We write them as a tuple $\vec{w} = (w_{\mathfrak{p}}, w_{\mathfrak{q}}, w_{\mathfrak{l}})$.

$$\begin{aligned}
&\vec{w}\,(\phi_{1,\mathfrak{p}}) = (6,2,2), \qquad &&\vec{w}\,(\phi_{\mathfrak{p}}) = (\infty,8,8),\\
&\vec{w}\,(\phi_{1,\mathfrak{q}}) = (2,3,3), \qquad &&\vec{w}\,(\phi_{2,\mathfrak{q}}) = (6,15,14) \qquad \vec{w}\,(\phi_{\mathfrak{q}}) = (12,\infty,28),\\
&\vec{w}\,(\phi_{1,\mathfrak{l}}) = (2,3,3), \qquad &&\vec{w}\,(\phi_{\mathfrak{l}}) = (6,14,\infty).
\end{aligned}$$

We can now step through the results of running MaxMin$[S]$. The "minimal" valuation is underlined at each step. This indicates the index which will be incremented in the following step.

i	g_i	$\vec{w}\,(g_i)$	$w\,(g_i)$
0	$1 \cdot 1 \cdot 1$	$(\underline{0}, 0, 0)$	0
1	$\phi_{1,\mathfrak{p}} \cdot 1 \cdot 1$	$(6, \underline{2}, 2)$	2
2	$\phi_{1,\mathfrak{p}} \cdot \phi_{1,\mathfrak{q}} \cdot 1$	$(8, \underline{5}, 5)$	5
3	$\phi_{1,\mathfrak{p}} \cdot \phi_{1,\mathfrak{q}}^2 \cdot 1$	$(10, \underline{8}, 8)$	8
4	$\phi_{1,\mathfrak{p}} \cdot \phi_{2,\mathfrak{q}} \cdot 1$	$(\underline{12}, 17, 16)$	12
5	$\phi_{1,\mathfrak{p}}^2 \cdot \phi_{2,\mathfrak{q}} \cdot 1$	$(\underline{18}, 19, 18)$	18
6	$\phi_{1,\mathfrak{p}}^3 \cdot \phi_{2,\mathfrak{q}} \cdot 1$	$(24, 21, \underline{20})$	20
7	$\phi_{1,\mathfrak{p}}^3 \cdot \phi_{2,\mathfrak{q}} \cdot \phi_{1,\mathfrak{l}}$	$(26, 24, \underline{23})$	23
8	$\phi_{1,\mathfrak{p}}^3 \cdot \phi_{2,\mathfrak{q}} \cdot \phi_{1,\mathfrak{l}}^2$	$(28, 27, \underline{26})$	26
9	$\phi_{1,\mathfrak{p}}^3 \cdot \phi_{2,\mathfrak{q}} \cdot \phi_{\mathfrak{l}}$	$(\underline{30}, 35, \infty)$	30

10	$\phi_{\mathfrak{p}} \cdot \phi_{2,\mathfrak{q}} \cdot \phi_{\mathfrak{l}}$	$(\infty, \underline{37}, \infty)$	37
11	$\phi_{\mathfrak{p}} \cdot \phi_{2,\mathfrak{q}}\phi_{1,\mathfrak{q}} \cdot \phi_{\mathfrak{l}}$	$(\infty, \underline{40}, \infty)$	40
12	$\phi_{\mathfrak{p}} \cdot \phi_{2,\mathfrak{q}}\phi_{1,\mathfrak{q}}^2 \cdot \phi_{\mathfrak{l}}$	$(\infty, \underline{43}, \infty)$	43
13	$\phi_{\mathfrak{p}} \cdot \phi_{\mathfrak{q}} \cdot \phi_{\mathfrak{l}}$	(∞, ∞, ∞)	∞

The final element g_{13} is the "extended" element, and is not included in the v-integral basis of S.

4.3 Precomputation

Consider a partition of a subset $S = \{\mathfrak{q}_1, \ldots, \mathfrak{q}_s\} \subseteq \mathcal{P}$:

$$S = S_1 \cup \cdots \cup S_t.$$

That is, a decomposition of S into the disjoint union of several subsets. We require that the ordering of each S_j adhere to (4.1), and that S maintains the ordering of the subsets, so that for $1 \leqslant i < j \leqslant t$ all elements of S_i come before all elements of S_j. Denote

$$n_j := n_{S_j}, \qquad 0 \leqslant j \leqslant t.$$

Take extended families of numerators $g_{0,S_j}, \ldots, g_{n_j,S_j}$ of Okutsu S_j-bases, for all $0 \leqslant j \leqslant t$.

Consider multi-indices $\mathbb{i} = (i_1, \ldots, i_t)$ of degree $\deg \mathbb{i} = i_1 + \cdots + i_t$ and monic polynomials $g_{\mathbb{i}} := g_{i_1,S_1} \cdots g_{i_t,S_t} \in \mathcal{O}[x]$.

We may consider the version of MaxMin presented in Algorithm 4.2.

Algorithm 4.2 MaxMin$[S = S_1 \cup \cdots \cup S_t]$ algorithm

Input: A partition $S = S_1 \cup \cdots \cup S_t$ of $S \subseteq \mathcal{P}$, and extended families $\left\{g_{i,S_j} : 0 \leqslant i \leqslant n_{S_j}\right\}$ of numerators of Okutsu S_j-bases for all $1 \leqslant j \leqslant t$.

Output: A family $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n_S} \in \mathbb{N}^t$ of multi-indices of degree $0, 1, \ldots, n_S$, respectively.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0) \in \mathbb{N}^t$
2: **for** $k = 0 \to n_S - 1$ **do**
3: $\quad j \leftarrow \min\{1 \leqslant i \leqslant t : w_{S_i}(g_{\mathbb{i}_k}) = w_S(g_{\mathbb{i}_k})\}$
4: $\quad \mathbb{i}_{k+1} \leftarrow \mathbb{i}_k + \mathbb{u}_j$
5: **end for**

There is a double motivation for the consideration of this algorithm. On one hand, the Montes algorithm is able to provide Okutsu S-bases for certain subsets $S \in \mathcal{P}$ in a very natural way. Thus, in practice we are going to use this MaxMin$[S = S_1 \cup \cdots \cup S_t]$ algorithm instead of the "global" one. On the other hand, such a decomposition of MaxMin will be useful for the proof of Theorem 4.3 (see Section 4.5).

Definition 4.5. *For indices $1 \leqslant a \leqslant b \leqslant s$, consider the following interval of S:*

$$I = [a, b] := \{\mathfrak{q}_j : a \leqslant j \leqslant b\} \subseteq S.$$

We say that I admits precomputation *if, after natural identifications, the algorithm* MaxMin$[S]$ *has the same output as*

$$\text{MaxMin}\left[S = \{\mathfrak{q}_1\} \cup \cdots \cup \{\mathfrak{q}_{a-1}\} \cup I \cup \{\mathfrak{q}_{b+1}\} \cup \cdots \cup \{\mathfrak{q}_s\}\right], \tag{4.2}$$

where we consider the output of the algorithm MaxMin$[I]$ *as an extended Okutsu I-basis.*

By "natural identifications" we mean that if the k^{th} output of MaxMin$[S]$ *is $\mathbb{i}_k = (i_{\mathfrak{q}_1}, \ldots, i_{\mathfrak{q}_s})$, then the k^{th} output of the algoritm* (4.2) *is:*

$$\mathbb{j}_k = (i_{\mathfrak{q}_1}, \ldots, i_{\mathfrak{q}_a-1}, i_I, i_{\mathfrak{q}_b+1}, \ldots, i_{\mathfrak{q}_s}),$$

while the i_I^{th} output of MaxMin$[I]$ *is $(i_{\mathfrak{q}_a}, \ldots, i_{\mathfrak{q}_b})$.*

Let $(\mathbb{i}_k)_{0 \leqslant k \leqslant n_S}$ be the output of MaxMin$[S]$, leading to numerators

$$\mathbb{i}_k = (i_{\mathfrak{q}_1}, \ldots, i_{\mathfrak{q}_s}) \implies g_{\mathbb{i}_k} = \prod_{\mathfrak{q} \in S} g_{i_\mathfrak{q}, \mathfrak{q}}.$$

Let $g'_0, g'_1, \ldots, g'_{n_I}$ be the numerators deduced from the application of the algorithm MaxMin$[I]$. Then, let $(\mathbb{j}_k)_{0 \leqslant k \leqslant n_S}$ be the output of the MaxMin algorithm (4.2). If I admits precomputation, these multi-indices lead to

numerators

$$\mathbb{j}_k = (i_{\mathfrak{q}_1}, \dots, i_{\mathfrak{q}_a - 1}, i_I, i_{\mathfrak{q}_b + 1}, \dots, i_{\mathfrak{q}_s}) \implies$$
$$g'_{\mathbb{j}_k} = g'_{i_I} \prod_{\mathfrak{q} \in S \setminus I} g_{i_\mathfrak{q}, \mathfrak{q}} = \prod_{\mathfrak{q} \in I} g_{i_\mathfrak{q}, \mathfrak{q}} \prod_{\mathfrak{q} \in S \setminus I} g_{i_\mathfrak{q}, \mathfrak{q}} = g_{\mathbb{i}_k}, \quad (4.3)$$

so that MaxMin$[S]$ and (4.2) lead to the same family of numerators of an Okutsu S-basis.

The next result is an immediate consequence of the definition.

Corollary 4.6. *Let $S = I_1 \cup \cdots \cup I_t$ be a decomposition of S into the disjoint union of intervals $I_j = [a_j, b_j]$ with increasing end points $b_1 < \cdots < b_t$.*

If all intervals I_j admit precomputation, then MaxMin$[S = I_1 \cup \cdots \cup I_t]$ *has the same output as* MaxMin$[S]$*, after natural identifications.* □

Suppose that $\mathbb{i}_0, \dots, \mathbb{i}_{n_S}$ and $\mathbb{j}_0, \dots, \mathbb{j}_{n_S}$ are the outputs of MaxMin$[S]$ and MaxMin$[S = I_1 \cup \cdots \cup I_t]$, respectively. The natural identifications in this case are

$$\mathbb{i}_k = (i_{\mathfrak{q}_1}, \dots, i_{\mathfrak{q}_s}) \implies \mathbb{j}_k = (i_1, \dots, i_t)\,,$$

and for all $1 \leqslant j \leqslant t$, the i_j^{th} output of MaxMin$[S_j]$ is the multi-index $(i_{\mathfrak{q}_m})_{a_j \leqslant m \leqslant b_j}$.

Let us give a criterion for an interval to admit precomputation.

Lemma 4.7. *Let $\mathbb{i}_0, \mathbb{i}_1, \dots, \mathbb{i}_{n_S}$ be the output of* MaxMin$[S]$ *and let $I \subseteq S$ be an interval of S. For each $0 \leqslant k \leqslant n_S$, let $\mathbb{i}_k = (i_\mathfrak{q})_{\mathfrak{q} \in S}$ and denote*

$$g_{\mathbb{i}_k} = \prod_{\mathfrak{q} \in S} g_{i_\mathfrak{q}, \mathfrak{q}}, \qquad G_{\mathbb{i}_k} = \prod_{\mathfrak{q} \in S \setminus I} g_{i_\mathfrak{q}, \mathfrak{q}}.$$

Suppose that for each $0 \leqslant k \leqslant n_S$ the following condition holds

$$w_I\left(g_{\mathbb{i}_k}\right) = w_S\left(g_{\mathbb{i}_k}\right) \implies w_\mathfrak{p}\left(G_{\mathbb{i}_k}\right) = w_\mathfrak{q}\left(G_{\mathbb{i}_k}\right), \qquad \forall\ \mathfrak{p}, \mathfrak{q} \in I.$$

Then, I admits precomputation.

Proof. Let $(\mathbb{i}_k)_{0 \leqslant k \leqslant n_S}$ be the output of MaxMin$[S]$ and $(\mathbb{j}_k)_{0 \leqslant k \leqslant n_S}$ be the output of the precomputed MaxMin algorithm (4.2).

Clearly, $\mathbb{i}_0$ and $\mathbb{j}_0$ may be identified. For $k \geqslant 0$, suppose that $\mathbb{i}_k$ may be identified with $\mathbb{j}_k$. This means

$$\begin{aligned}\mathbb{i}_k &= \left(i_{\mathfrak{q}_1}, \ldots, i_{\mathfrak{q}_s}\right),\\ \mathbb{j}_k &= \left(i_{\mathfrak{q}_1}, \ldots, i_{\mathfrak{q}_a-1}, i_I, i_{\mathfrak{q}_b+1}, \ldots, i_{\mathfrak{q}_s}\right),\end{aligned}$$

while the i_I^{th} output of MaxMin$[I]$ is the multi-index $\left(i_{\mathfrak{q}_j}\right)_{a \leqslant j \leqslant b}$.

With the notation of (4.3),

$$g_{\mathbb{i}_k} = g'_{\mathbb{j}_k}, \qquad g'_{i_I} = \prod_{m=a}^{b} g_{i_{\mathfrak{q}_m}, \mathfrak{q}_m}.$$

The algorithm MaxMin$[S]$ outputs $\mathbb{i}_{k+1} = \mathbb{i}_k + \mathbb{u}_j$, where

$$j = \min\left\{1 \leqslant m \leqslant s : w_{\mathfrak{q}_m}\left(g_{\mathbb{i}_k}\right) = w_S\left(g_{\mathbb{i}_k}\right)\right\}.$$

If $\mathfrak{q}_j \notin I$, then the $\mathfrak{q}_j$-index in $\mathbb{j}_k$ will also be the least index satisfying $w_{\mathfrak{q}_j}\left(g'_{\mathbb{j}_k}\right) = w_S\left(g'_{\mathbb{j}_k}\right)$, since $g'_{\mathbb{j}_k} = g_{\mathbb{i}_k}$. Thus, the algorithm in (4.2) will also increase the $\mathfrak{q}_j$-coordinate.

If $\mathfrak{q}_j \in I$, then $w_I\left(g_{\mathbb{i}_k}\right) = w_S\left(g_{\mathbb{i}_k}\right)$ and $w_{\mathfrak{q}_m}\left(g_{\mathbb{i}_k}\right) > w_S\left(g_{\mathbb{i}_k}\right)$ for all $m < a$; thus, (4.2) will increase i_I by one. In this case, we must show that the $(i_I + 1)$-th output of MaxMin$[I]$ is the multi-index obtained from $\left(i_{\mathfrak{q}_j}\right)_{a \leqslant j \leqslant b}$ by increasing the $\mathfrak{q}_j$-coordinate by one.

The index increased by MaxMin$[I]$ will be:

$$J = \min\left\{a \leqslant m \leqslant b : w_{\mathfrak{q}_m}\left(g'_{i_I}\right) = w_I\left(g'_{i_I}\right)\right\}.$$

By hypothesis, $\nu := w_{\mathfrak{q}}\left(G_{\mathbb{i}_k}(\theta)\right)$ is independent of the choice of $\mathfrak{q} \in I$. Since $g_{\mathbb{i}_k} = G_{\mathbb{i}_k} g'_{i_I}$, we have:

$$w_{\mathfrak{q}}\left(g_{\mathbb{i}_k}\right) = w_{\mathfrak{q}}\left(g'_{i_I}\right) + \nu, \qquad \forall\ \mathfrak{q} \in I.$$

In particular, $w_S\left(g_{\mathbb{i}_k}\right) = w_I\left(g_{\mathbb{i}_k}\right) = w_I\left(g'_{i_I}\right) + \nu$, so that $J = j$. □

One specific case of precomputation which we will make use of, is the precomputation of certain intervals $S_{\mathbb{t}} \subseteq S$ defined by a type $\mathbb{t}$.

Lemma 4.8. *For any $\mathbb{t} \in \mathfrak{T}_S$, the interval $S_{\mathbb{t}} \subseteq S$ admits precomputation.*

Proof. For every $\mathfrak{p} \in S_{\mathbb{t}}$ and every $\mathfrak{q} \notin S_{\mathbb{t}}$, the explicit formulas from Proposition 2.31 show that $w_{\mathfrak{q}}(\phi_{i,\mathfrak{q}})$ is independent of $\mathfrak{p}$, for all i. Hence, the same is true for all polynomials $G_{\mathbb{i}_k}$ that are a product of these ϕ-polynomials.

Thus, $S_{\mathbb{t}}$ meets the criterion of Lemma 4.7. □

4.3.1 Precomputation counter-example

To illustrate the necessity of the precomputation criterion, we present a small example where an incompatible interval is precomputed and this leads to output multi-indices which are not optimal. Let $S = \{\mathfrak{p}, \mathfrak{q}, \mathfrak{l}\}$ and consider the first levels of a non-optimised tree shown in Figure 4.3.

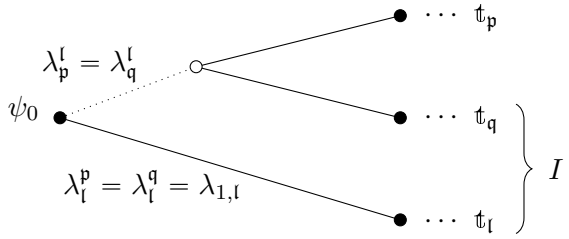


Figure 4.3: Non-optimised tree with interval that does not meet the precomputation criterion.

In this example, the branch that divides to become part of $\mathbb{t}_{\mathfrak{p}}$ and $\mathbb{t}_{\mathfrak{q}}$ includes a refinement step after its division from the branch that is part of $\mathbb{t}_{\mathfrak{l}}$. Let us consider the following values for the slopes shown in Figure 4.3:

$$\lambda_{1,\mathfrak{p}} = 6, \qquad \lambda_{1,\mathfrak{q}} = 4, \qquad \lambda_{1,\mathfrak{l}} = 4, \qquad \lambda_{\mathfrak{p}}^{\mathfrak{l}} = 2.$$

The first monic lifting of ψ_0 to $\mathcal{O}[x]$ chosen by the Montes algorithm is $\phi(\mathfrak{p}, \mathfrak{l}) = \phi(\mathfrak{q}, \mathfrak{l}) = \phi_{1,\mathfrak{l}}$. The branch of slope 2 suffers refinement and this polynomial is replaced with $\phi(\mathfrak{p}, \mathfrak{q}) = \phi_{1,\mathfrak{p}} = \phi_{1,\mathfrak{q}}$. This polynomial leads to two branches which constitute the first level of two nodes of order 1 belonging to the paths joining $\mathbb{t}_{\mathfrak{p}}$ and $\mathbb{t}_{\mathfrak{q}}$ with the root node, respectively.

The optimised hidden slopes $\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}$ coincide with $\lambda_{1,\mathfrak{p}}$ and $\lambda_{1,\mathfrak{q}}$ respectively.

For the purpose of this example, this is sufficient information about the types $\mathbb{t}_{\mathfrak{p}}$, $\mathbb{t}_{\mathfrak{q}}$, and $\mathbb{t}_{\mathfrak{l}}$, except to say that the order of all three types is greater than 1.

According to the explicit formulas in Proposition 2.31, we can calculate the following valuations for the first level ϕ-polynomials of each type. The valuations are given as a tuple $\vec{w} = (w_{\mathfrak{p}}, w_{\mathfrak{q}}, w_{\mathfrak{l}})$. Note also that $\phi_{1,\mathfrak{p}} = \phi(\mathfrak{p}, \mathfrak{q}) = \phi_{1,\mathfrak{q}}$.

$$\vec{w}\,(\phi_{1,\mathfrak{p}}) = \vec{w}\,(\phi_{1,\mathfrak{q}}) = (6, 4, 2),$$
$$\vec{w}\,(\phi_{1,\mathfrak{l}}) = (2, 2, 4).$$

The interval $I = \{\mathfrak{q}, \mathfrak{l}\}$ does not meet the precomputation criterion, as we cannot guarantee that when either the $\mathfrak{q}$- or the $\mathfrak{l}$-valuation of an output numerator is minimal, the $\mathfrak{q}$- and $\mathfrak{l}$-valuations of the $\mathfrak{p}$-part of that numerator will be equal.

Let us now consider the first three basis numerators computed by the MaxMin$[S]$. The minimal valuation (and as such, the index to increment for the following numerator) is marked with an underline.

i	g_i	$\vec{w}\,(g_i)$	$w\,(g_i)$
0	$1 \cdot 1 \cdot 1$	$(\underline{0}, 0, 0)$	0
1	$\phi_{1,\mathfrak{p}} \cdot 1 \cdot 1$	$(6, 4, \underline{2})$	2
2	$\phi_{1,\mathfrak{p}} \cdot 1 \cdot \phi_{1,\mathfrak{l}}$	$(8, \underline{6}, 6)$	6

Let $I = \{\mathfrak{q}, \mathfrak{l}\} \subset S$ be the interval shown in Figure 4.3 and consider the first three output numerators of MaxMin$[I]$.

i	h_i	$\vec{w}\,(h_i)$	$w\,(h_i)$
0	$1 \cdot 1$	$(\underline{0}, 0)$	0
1	$\phi_{1,\mathfrak{q}} \cdot 1$	$(4, \underline{2})$	2
2	$\phi_{1,\mathfrak{q}} \cdot \phi_{1,\mathfrak{l}}$	$(\underline{6}, 6)$	6

Finally, we may consider the output of MaxMin$[S = \{\mathfrak{p}\} \cup I]$, the algorithm using the results from the precomputed interval I.

i	g'_i	$\vec{w}(g'_i)$	$w(g'_i)$
0	$1 \cdot 1$	$(\underline{0}, 0, 0)$	0
1	$\phi_{1,\mathfrak{p}} \cdot 1$	$(6, \underline{4}, 2)$	2
2	$\phi_{1,\mathfrak{p}} \cdot h_1 = \phi_{1,\mathfrak{p}} \cdot \phi_{1,\mathfrak{q}}$	$(12, \underline{8}, 4)$	4

Since $w(g_2) = 6 > w(g'_2) = 4$, the output of MaxMin$[S = \{\mathfrak{p}\} \cup I]$ is not optimal. From this example, we see that precomputing an interval which does not meet the precomputation criterion may lead to non-optimal numerators.

4.4 The block-wise MaxMin algorithm

Consider an ordered subset $S = \{\mathfrak{q}_1, \ldots, \mathfrak{q}_s\} \subseteq \mathcal{P}$, such that the corresponding tree $\mathfrak{T}_S$ is connected, and take

$$\ell = i(S) := \min\{i(\mathfrak{p}, \mathfrak{q}) : \mathfrak{p}, \mathfrak{q} \in S\},$$

so that $\ell - 1$ is the order of the greatest common node of all paths joining the leaves of $\mathfrak{T}_S$ with the root node.

The Okutsu frames of all primes $\mathfrak{p} \in S$ have the same first $\ell - 1$ key polynomials $\phi_1, \ldots, \phi_{\ell-1}$. Thus, the first m_ℓ numerators of the Okutsu $\mathfrak{p}$-bases coincide for all $\mathfrak{p} \in S$. Let

$$\mathcal{N} = \{1 = h_0, h_1, \ldots, h_{m_\ell - 1}\},$$

be the family of these common numerators. Note that

$$w_{\mathfrak{p}}(h) = w_{\mathfrak{q}}(h), \qquad \forall\ \mathfrak{p}, \mathfrak{q} \in S,\ \forall\ h \in \mathcal{N}. \tag{4.4}$$

Lemma 4.9. *For all $\mathfrak{p}, \mathfrak{q} \in S$ and all $0 \leqslant r, s < m_\ell$:*

$$w_{\mathfrak{q}}(h_r h_s) \leqslant \begin{cases} w_{\mathfrak{q}}(h_{r+s}), & \text{if } r + s < m_\ell, \\ w_{\mathfrak{q}}(\phi_{\ell,\mathfrak{p}}) + w_{\mathfrak{q}}(h_k), & \text{if } r + s = m_\ell + k. \end{cases}$$

Proof. If $r + s < m_\ell$, the inequality $w_{\mathfrak{q}}(h_r h_s) \leqslant w_{\mathfrak{q}}(h_{r+s})$ is a consequence

of the maximality of $w_{\mathfrak{q}}(h_{r+s})$ amongst all monic polynomials of degree $r+s$.

Suppose now $r+s = m_\ell + k$. The recurrence $V_i = e_{i-1}f_{i-1}(e_{i-1}V_{i-1} + h_{i-1})$ shows that

$$\frac{V_i}{e_1\cdots e_{i-1}} = w_{\mathfrak{q}}\left(\phi_{i-1}^{e_{i-1}f_{i-1}}\right) < w_{\mathfrak{q}}(\phi_i) = \frac{V_i+\lambda_i}{e_1\cdots e_{i-1}}, \qquad \forall\ i < m_\ell.$$

Hence, in any product of powers of $\phi_1,\dots,\phi_{\ell-1}$ we may replace $\phi_{i-1}^{e_{i-1}f_{i-1}}$ with ϕ_i to increase the $\mathfrak{q}$-valuation. Therefore,

$$w_{\mathfrak{q}}(h_r h_s) \leqslant w_{\mathfrak{q}}\left(\phi_{\ell-1}^{e_{\ell-1}f_{\ell-1}}\right) + w_{\mathfrak{q}}(h_k) = \frac{V_\ell}{e_1\cdots e_{\ell-1}} + w_{\mathfrak{q}}(h_k), \qquad (4.5)$$

because the term on the right hand side is the maximal $\mathfrak{q}$-valuation of a product of powers of $\phi_1,\dots,\phi_{\ell-1}$ of degree $m_\ell + k$. Now, the formulas in Proposition 2.31 show the existence a slope λ (either hidden or not) for which

$$w_{\mathfrak{q}}(\phi_{\ell,\mathfrak{p}}) = \frac{V_\ell+\lambda}{e_1\cdots e_{\ell-1}} > \frac{V_\ell}{e_1\cdots e_{\ell-1}}.$$

This ends the proof of the second inequality. □

Lemma 4.10. *Let $\mathbb{i}$ be a maximal multi-index of degree divisible by m_ℓ.*

1. *There exists a maximal multi-index $\mathbb{i}' = (i'_{\mathfrak{p}})_{\mathfrak{p}\in S}$ of the same degree, having all its coordinates $i'_{\mathfrak{p}}$ divisible by m_ℓ.*
2. *The elements in the family $g_{\mathbb{i}}\mathcal{N}$ are maximal numerators of degree $\deg(\mathbb{i}), \deg(\mathbb{i})+1, \dots, \deg(\mathbb{i})+m_\ell-1$.*

Proof. For $0 \leqslant j < m_\ell$, let $\mathbb{j} = (j_{\mathfrak{p}})_{\mathfrak{p}\in S}$ be a multi-index of degree $im_\ell + j$. Each index $j_{\mathfrak{p}}$ may be written

$$j_{\mathfrak{p}} = q_{\mathfrak{p}}m_\ell + k_{\mathfrak{p}}, \qquad 0 \leqslant k_{\mathfrak{p}} < m_\ell,$$

and the numerators $g_{j_{\mathfrak{p}},\mathfrak{p}}$ of the Okutsu $\mathfrak{p}$-basis may be written

$$g_{j_{\mathfrak{p}},\mathfrak{p}} = G_{\mathfrak{p}}h_{k_{\mathfrak{p}}}, \qquad \deg G_{\mathfrak{p}} = q_{\mathfrak{p}}m_\ell.$$

By (4.4), we have

$$w_S\left(g_{\mathbb{j}}\right) = w_S\left(\prod_{\mathfrak{p}\in S} G_{\mathfrak{p}} h_{k_{\mathfrak{p}}}\right) = w_S\left(\prod_{\mathfrak{p}\in S} G_{\mathfrak{p}}\right) + w_{\mathfrak{p}_0}\left(\prod_{\mathfrak{p}\in S} h_{k_{\mathfrak{p}}}\right),$$

where $\mathfrak{p}_0$ is an arbitrary choice of a prime ideal in S.

Since all polynomials $G_{\mathfrak{p}}$ have a degree which is a multiple of m_ℓ, we have $\sum_{\mathfrak{p}\in S} k_{\mathfrak{p}} = qm_\ell + j$, for some non-negative integer q. Consider the polynomial $h := \phi_{\ell-1}^{e_{\ell-1}f_{\ell-1}}$ of degree m_ℓ. By an iterative application of the inequalities in (4.5), we get

$$w_{\mathfrak{q}}\left(\prod_{\mathfrak{p}\in S} h_{k_{\mathfrak{p}}}\right) \leqslant w_{\mathfrak{q}}\left(h^q\right) + w_{\mathfrak{q}}\left(h_j\right), \qquad \forall\ \mathfrak{q} \in S.$$

Hence,

$$\begin{aligned} w_S\left(g_{\mathbb{j}}\right)) &\leqslant w_S\left(h^q \prod_{\mathfrak{p}\in S} G_{\mathfrak{p}}\right) + w_{\mathfrak{p}_0}\left(h_j\right) \\ &= w_S\left(h^q h_j \prod_{\mathfrak{p}\in S} G_{\mathfrak{p}}\right) < w_S\left(\phi_{\ell,\mathfrak{p}_0}^q h_j \prod_{\mathfrak{p}\in S} G_{\mathfrak{p}}\right). \end{aligned}$$

The final inequality is a consequence of $\hat{w}_{\mathfrak{p}}\left(\phi_{k,\mathfrak{p}_0}\right) < \hat{w}_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{p}_0}\right)$, for all $k < \ell$ and $\mathfrak{p} \in S$, which is shown in Lemma 3.12.

These arguments, applied to $\mathbb{j} = \mathbb{i}$ (and $j = 0$) prove item (1). Also, applied to an arbitrary $\mathbb{j}$ of degree $\deg(\mathbb{i}) + j$ show that

$$\begin{aligned} w_S\left(g_{\mathbb{j}}\right) &\leqslant w_S\left(\phi_{\ell,\mathfrak{p}_0}^q \prod_{\mathfrak{p}\in S} G_{\mathfrak{p}}\right) + w_{\mathfrak{p}_0}\left(h_j\right) \\ &\leqslant w_S\left(g_{\mathbb{i}}\right) + w_{\mathfrak{p}_0}\left(h_j\right) \\ &= w_S\left(g_{\mathbb{i}} h_j\right), \end{aligned}$$

by the maximality of $g_{\mathbb{i}}$. This proves item (2). □

Lemma 4.11. *Let $\mathbb{i} = (i_{\mathfrak{q}})_{\mathfrak{q}\in S}$ be an output multi-index of* MaxMin$[S]$ *of degree divisible by m_ℓ.*

1. *All coordinates $i_{\mathfrak{q}}$ are divisible by m_ℓ.*

2. *Let* $j = \min\{1 \leqslant m \leqslant s : w_{\mathfrak{q}_m}(g_{\mathbb{i}}) = w_S(g_{\mathbb{i}})\}$. *Then, the next* m_ℓ *iterations of* MaxMin$[S]$ *increase the coordinate* $\mathfrak{q}_j$.

Proof. All coordinates of $\mathbb{i}_0$ are zero; hence divisible by m_ℓ. Thus, it suffices to prove that any output multi-index $\mathbb{i} = (i_\mathfrak{q})_{\mathfrak{q}\in S}$ whose coordinates are all divisible by m_ℓ satisfies (2).

Let $j = \min\{1 \leqslant m \leqslant s : w_{\mathfrak{q}_m}(g_{\mathbb{i}}) = w_S(g_{\mathbb{i}})\}$. If $\mathbb{i} = \mathbb{i}_k$ is the k-th output multi-index of MaxMin$[S]$, the algorithm selects $\mathbb{i}_{k+1} = \mathbb{i}_k + \mathbb{u}_j$. Since $i_{\mathfrak{q}_j}$ is a multiple of m_ℓ, we have $g_{\mathbb{i}_{k+1}} = g_{\mathbb{i}_k} h_1$; hence,

$$\begin{aligned} w_\mathfrak{q}\left(g_{\mathbb{i}_{k+1}}\right) &= w_\mathfrak{q}\left(g_{\mathbb{i}_k}\right) + w_\mathfrak{q}\left(h_1\right) \\ &\geqslant w_S\left(g_{\mathbb{i}_k}\right) + w_\mathfrak{q}\left(h_1\right) \\ &= w_S\left(g_{\mathbb{i}_{k+1}}\right), \end{aligned}$$

for all $\mathfrak{q} \in S$. Thus, $w_\mathfrak{q}\left(g_{\mathbb{i}_{k+1}}\right) = w_S\left(g_{\mathbb{i}_{k+1}}\right)$ if and only if $w_\mathfrak{q}\left(g_{\mathbb{i}_k}\right) = w_S\left(g_{\mathbb{i}_k}\right)$, so that the next iteration increases the $\mathfrak{q}_j$-coordinate again. By iterating this argument, we get $g_{\mathbb{i}_{k+m_\ell-1}} = g_{\mathbb{i}_k} h_{m_\ell-1}$. At this point, the $\mathfrak{q}_j$-coordinate will be increased once more to yield $\mathbb{i}_{k+m_\ell} = \mathbb{i}_k + m_\ell \mathbb{u}_j$. □

This result shows that MaxMin$[S]$ works by blocks of length m_ℓ. Thus, we may consider Algorithm 4.3.

Algorithm 4.3 MaxMin$[S; m_\ell]$ algorithm

Input: An ordered subset $S = \{\mathfrak{q}_1, \ldots, \mathfrak{q}_s\} \subseteq \mathcal{P}$ such that $\mathfrak{T}_S$ is connected, and extended families $\{g_{i,\mathfrak{q}} : 0 \leqslant i \leqslant n_\mathfrak{q}\}$ of numerators of Okutsu $\mathfrak{q}$-bases of each $\mathfrak{q} \in S$.

Output: A family $\mathbb{i}_0, \mathbb{i}_{m_\ell}, \mathbb{i}_{2m_\ell}, \ldots, \mathbb{i}_{n_S/m_\ell}$ of multi-indices with $\deg \mathbb{i}_k = k$, having all coordinates divisible by m_ℓ.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0)$
2: **for** $k = 0 \to (n_S/m_\ell) - 1$ **do**
3: $\quad j \leftarrow \min\left\{1 \leqslant i \leqslant s : w_{\mathfrak{q}_i}\left(g_{\mathbb{i}_{km_\ell}}\right) = w_S\left(g_{\mathbb{i}_{km_\ell}}\right)\right\}$
4: $\quad \mathbb{i}_{(k+1)m_\ell} \leftarrow \mathbb{i}_{km_\ell} + m_\ell \mathbb{u}_j$
5: **end for**

Theorem 4.3 will be a consequence of the following result.

Theorem 4.12. *The output multi-indices of* MaxMin$[S; m_\ell]$ *are maximal amongst all multi-indices of the same degree with coordinates divisible by* m_ℓ.

In fact, by Lemma 4.10, all output multi-indices of MaxMin$[S; m_\ell]$ will be maximal and by Lemma 4.11 these multi-indices coincide with the output multi-indices of degree divisible by m_ℓ of MaxMin$[S]$.

Finally, Lemma 4.11 shows how to derive all other output multi-indices of MaxMin$[S]$ and Lemma 4.10 shows that these multi-indices are maximal too.

4.5 Proof of Theorem 4.12

The proof of Theorem 4.12 makes a heavy use of the structure of the non-optimised tree with base type $\mathbb{t}_{\ell-1}$ which is the greatest common node in all paths joining the leaves of $\mathfrak{T}_S$ with the root node.

Let ϕ_ℓ be the first representative of $\mathbb{t}_{\ell-1}$ which leads to branching. Thus, before constructing ϕ_ℓ, the Montes algorithm may have constructed other representatives of $\mathbb{t}_{\ell-1}$ admitting unibranch refinements.

Let $\lambda_{\min}$ be the least slope (in absolute size) occurring in the branching based on ϕ_ℓ. Let $S_{\min} \subseteq S$ be the subset of all prime ideals derived from a branches of slope $\lambda_{\min}$ of ϕ_ℓ.

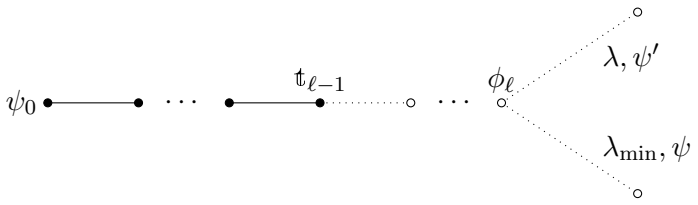


Figure 4.4: Tree $\mathfrak{T}$ with common node $\mathbb{t}_{\ell-1}$.

The basic idea is to split $S = U \cup D$ (U for "up" and D for "down") into the disjoin union of two intervals which admit precomputation and then analyse the behaviour of MaxMin$[S = U \cup D]$ for which the multi-indices have only two coordinates.

Lemma 4.10 and Lemma 4.11 show that the output multi-indices of MaxMin$[S; m_\ell]$ coincide with the output of an ordinary application of the 2-dimensional MaxMin applied to the precomputations MaxMin$[U; m_\ell]$ and MaxMin$[D; m_\ell]$. We shall denote this algorithm by MaxMin$[S = U \cup D; m_\ell]$.

We distinguish three cases according to the structure of the non-optimised tree:

Case (A). *There exists a branch with slope $\lambda_{\min}$ which suffered refinement. In this case, we take D to be the set of all prime ideals derived from this branch. Note that $\phi_{\ell,\mathfrak{p}} \neq \phi_\ell \ \forall \ \mathfrak{p} \in D$, and that there may be other $\lambda_{\min}$-branches.*

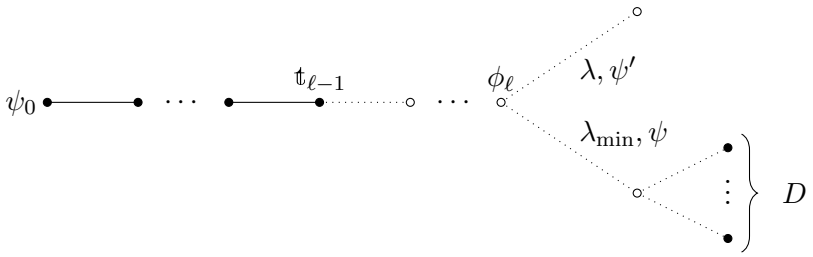


Figure 4.5: Case (A): Tree $\mathfrak{T}$ with common node $\mathfrak{t}_{\ell-1}$ and at least one refined $\lambda_{\min}$-branch.

Case (B). *None of the $\lambda_{\min}$-branches suffered refinement, and there are no other slopes. In other words, $\lambda_{\ell,\mathfrak{p}} = \lambda_{\min}$ and $\phi_{\ell,\mathfrak{p}} = \phi_\ell$ for all $\mathfrak{p} \in S = S_{\min}$.*

In this case, we take $D = S_{\mathfrak{t}}$, where $\mathfrak{t} := \mathrm{Trunc}_\ell(\mathfrak{t}_{\mathfrak{p}_0})$, for an arbitrary choice of $\mathfrak{p}_0 \in S$.

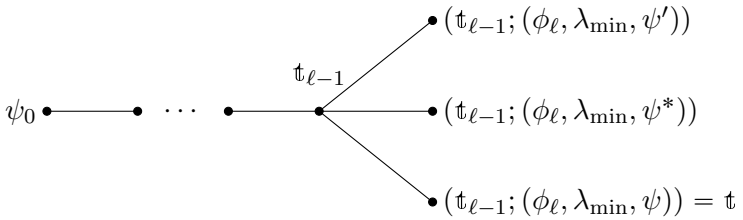


Figure 4.6: Case (B): Tree $\mathfrak{T}$ with common node $\mathfrak{t}_{\ell-1}$ with only unrefined $\lambda_{\min}$-branches.

Case (C). *None of the $\lambda_{\min}$-branches suffered refinement, but there are other slopes. In other words, $\lambda_{\ell,\mathfrak{p}} = \lambda_{\min}$ and $\phi_{\ell,\mathfrak{p}} = \phi_\ell$, for all $\mathfrak{p} \in S_{\min}$, and $S_{\min} \subsetneq S$.*

In this case, we take $D = S_{\min}$.

In all cases, we may change the ordering of $\mathcal{P}$ so that D and $U = S \backslash D$ are intervals.

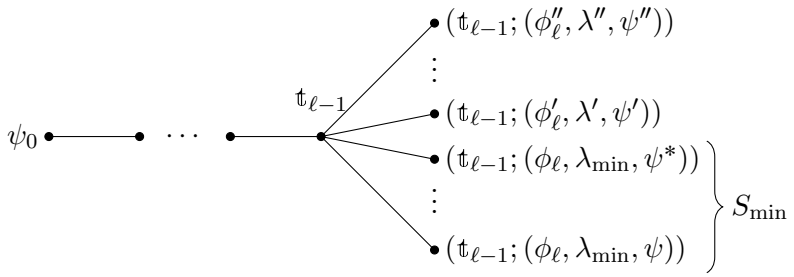


Figure 4.7: Case (C): Tree $\mathfrak{T}$ with common node $\mathbb{t}_{\ell-1}$ with unrefined $\lambda_{\min}$-branches and other slopes.

4.5.1 Proof of the Theorem in cases (A) and (B)

Let $U = S\backslash D$ and denote

$$c := \frac{V_\ell + \lambda_{\min}}{e_1 \cdots e_{\ell-1}}.$$

The explicit formulas from Proposition 2.31 show that

$$w_{\mathfrak{p}}\left(\phi_{m,\mathfrak{q}}\right) = (m/m_\ell)c = w_{\mathfrak{q}}\left(\phi_{m,\mathfrak{p}}\right), \qquad \forall\ \mathfrak{p} \in U,\ \mathfrak{q} \in D,\ \forall\ m \geqslant \ell. \tag{4.6}$$

On the other hand, all ideas and criteria about precomputation apply to the MaxMin algorithms restricted to all multi-indices whose coordinates are divisible by m_ℓ. Hence, (4.6) shows that D and $U = S\backslash D$ meet the condition of Lemma 4.7 and both intervals admit precomputation.

Denote the respective output families of numerators of MaxMin$[U; m_\ell]$ and MaxMin$[D; m_\ell]$ by:

$$1, g_1, \ldots, g_{n_U/m_\ell},$$
$$1, g'_1, \ldots, g'_{n_D/m_\ell},$$

respectively. Note that $\deg g_k = \deg g'_k = km_\ell$, for all k.

By Corollary 4.6, MaxMin$[S; m_\ell]$ has the same output as MaxMin$[S = U \cup D; m_\ell]$, after natural identifications of the respective multi-indices. In other words, if (i, j) is the k-th output of MaxMin$[S = U \cup D; m_\ell]$ (so that $k = i + j$), then the k-th numerator provided by MaxMin$[S; m_\ell]$ is $g_i g'_j$.

Definition 4.13. *We say that a monic polynomial $G \in \mathcal{O}[x]$* has support *in a subset $S' \subset S$ if it is a product of polynomials $\phi_{m,\mathfrak{p}}$ for $\mathfrak{p} \in S'$ and $m \geqslant \ell$.*

Note that the degree of G is necessarily a multiple of m_ℓ.

In order to prove Theorem 4.12, we must show that the output numerators of MaxMin$[S; m_\ell]$ are maximal amongst all polynomials of the same degree with support in S.

We proceed by induction on $\#S$. The case $\#S = 1$ being trivial, we may assume by the induction hypothesis that both sequences of numerators are maximal amongst all polynomials of the same degree with support in U and D, respectively.

For all $0 \leqslant i \leqslant n_U/m_\ell$ and all $0 \leqslant j \leqslant n_D/m_\ell$, denote

$$\begin{aligned} \nu_i &:= w_U(g_i) - ic, \\ \nu'_j &:= w_D\left(g'_j\right) - jc. \end{aligned} \tag{4.7}$$

We agree that $\nu_{-1} = \nu'_{-1} = -1$.

Lemma 4.14. *For all $i, j \geqslant 0$,*

$$\nu_i \leqslant \nu_{i+1}, \qquad \nu'_j \leqslant \nu'_{j+1}.$$

Proof. By Proposition 2.31, $w_\mathfrak{q}(\phi_{\ell,\mathfrak{q}}(\theta)) = (V_\ell + \lambda_{\ell,\mathfrak{q}})/(e_1 \cdots e_{\ell-1})$ for all $\mathfrak{q} \in U$. Since $\lambda_{\ell,\mathfrak{q}} \geqslant \lambda_{\min}$, the maximality of g_{i+1} implies

$$w_U(g_{i+1}) \geqslant w_U(g_i\phi_{\ell,\mathfrak{q}}) \geqslant w_U(g_i) + c, \qquad \forall\ \mathfrak{q} \in U.$$

Similarly, the maximality of g'_{j+1} implies

$$w_D\left(g'_{j+1}\right) \geqslant w_D\left(g'_j\phi_\ell\right) = w_D\left(g'_j\right) + c.$$

By the definition (4.7) of ν_i, ν'_j, this ends the proof of the lemma. □

For any bi-index $\mathbb{i} = (i, j)$, and any $\mathfrak{p} \in U$, $\mathfrak{q} \in D$, we have

$$\begin{aligned} w_\mathfrak{p}(g_\mathbb{i}) &= w_\mathfrak{p}\left(g_i g'_j\right) = w_\mathfrak{p}(g_i(\theta)) + jc, \\ w_\mathfrak{q}(g_\mathbb{i}) &= w_\mathfrak{q}\left(g_i g'_j\right) = w_\mathfrak{q}\left(g'_j(\theta)\right) + ic. \end{aligned}$$

Hence,

$$\begin{aligned} w_U\left(g_{\mathbb{i}}\right) &= \nu_i + (\deg \mathbb{i})c, \\ w_D\left(g_{\mathbb{i}}\right) &= \nu'_j + (\deg \mathbb{i})c, \\ w_S\left(g_{\mathbb{i}}\right) &= w_S\left(g_i g'_j\right) = \min\left\{\nu_i, \nu'_j\right\} + (\deg \mathbb{i})c. \end{aligned}$$

Therefore, these numbers ν_i, ν'_j determine the flow of $\mathrm{MaxMin}[S = U \cup D; m_\ell]$. If (i, j) is an output pair, the next output pair is decided as follows:

$$\begin{aligned} w_U\left(g_i g'_j\right) = w_S\left(g_i g'_j\right) &\Longleftrightarrow \nu_i \leqslant \nu'_j, && \text{“}U\text{-minimal”}, \\ w_D\left(g_i g'_j\right) = w_S\left(g_i g'_j\right) &\Longleftrightarrow \nu'_j < \nu_i, && \text{“}D\text{-minimal”}. \end{aligned}$$

The next output pair is $(i+1, j)$ in the U-minimal case, and $(i, j+1)$ in the D-minimal case.

Proposition 4.15. *The output bi-indices (i, j) of $\mathrm{MaxMin}[S = U \cup D; m_\ell]$ satisfy the following properties:*

1. *Either $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$, or $\nu_{i-1} \leqslant \nu'_j < \nu_i$.*

2. *The output multi-indices $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p} \in S}$ of $\mathrm{MaxMin}[S; m_\ell]$ which are obtained by joining the i-th output of $\mathrm{MaxMin}[U; m_\ell]$ and the j-th output of $\mathrm{MaxMin}[D; m_\ell]$ are maximal.*

Proof. Clearly, the initial output pair $(0, 0)$ satisfies (1). Let us check that if an output pair (i, j) satisfies (1), then the next output pair satisfies (1) as well.

Suppose that $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$, so that the next output pair is $(i+1, j)$.

$$\begin{aligned} \nu_{i+1} \leqslant \nu'_j &\implies \nu'_{j-1} \leqslant \nu_i \leqslant \nu_{i+1} \leqslant \nu'_j, \\ \nu_{i+1} > \nu'_j &\implies \nu_i \leqslant \nu'_j < \nu_{i+1}. \end{aligned}$$

Suppose that $\nu_{i-1} \leqslant \nu'_j < \nu_i$, so that the next output pair is $(i, j+1)$.

$$\begin{aligned} \nu_i \leqslant \nu'_{j+1} &\implies \nu'_j < \nu_i \leqslant \nu'_{j+1}, \\ \nu_i > \nu'_{j+1} &\implies \nu_{i-1} \leqslant \nu'_j \leqslant \nu'_{j+1} < \nu_i. \end{aligned}$$

This proves item (1). As a consequence, for any $k \in \mathbb{Z}$ such that $0 \leqslant i - k \leqslant n_U/m_\ell$ and $0 \leqslant j + k \leqslant n_D/m_\ell$, we have:

$$\min\left\{\nu_{i-k}, \nu'_{j+k}\right\} \leqslant \min\left\{\nu_i, \nu'_j\right\}. \tag{4.8}$$

In fact, if $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$, then $\min\left\{\nu_{i-k}, \nu'_{j+k}\right\} \leqslant \nu_i$, whereas in the case $\nu_{i-1} \leqslant \nu'_j < \nu_i$, we have $\min\left\{\nu_{i-k}, \nu'_{j+k}\right\} \leqslant \nu'_j$.

In order to prove (2), suppose that (i, j) is an output pair of $\mathrm{MaxMin}[S = U \cup D; m_\ell]$ and let g be a polynomial of degree $(i+j)m_\ell$ with support in S. We may write $g = GG'$, with G, G' polynomials with support in U and D, respectively.

Suppose $\deg G = (i-k)m_\ell$, $\deg G' = (j+k)m_\ell$, for certain $k \in \mathbb{Z}$. By (4.6) and the maximality of the numerators g_{i-k}, g'_{j+k}, we have:

$$\begin{aligned} w_U(g) &= w_U\left(GG'\right) \\ &= w_U(G) + (j+k)c \\ &\leqslant \nu_{i-k} + (i-k)c + (j+k)c \\ &= \nu_{i-k} + (i+j)c, \end{aligned}$$

$$\begin{aligned} w_D(g) &= w_D\left(GG'\right) \\ &= w_D\left(G'\right) + (i-k)c \\ &\leqslant \nu'_{j+k} + (j+k)c + (i-k)c \\ &= \nu'_{j+k} + (i+j)c. \end{aligned}$$

Hence, by using (4.8), we get:

$$\begin{aligned} w_S(g) &= \min\{w_U(g), w_D(g)\} \\ &= \min\left\{\nu_{i-k}, \nu'_{j+k}\right\} + (i+j)c \\ &\leqslant \min\left\{\nu_i, \nu'_j\right\} + (i+j)c \\ &= w_S\left(g_i g'_j\right). \end{aligned}$$

□

This ends the proof of Theorem 4.12 in cases (A) and (B).

4.5.2 Precomputation in Case (C)

Recall that $D = S_{\min}$ and $U = S \backslash D$. In this case, we have:

$$\begin{aligned} \phi_{\ell,\mathfrak{q}} &= \phi_\ell, && \forall\ \mathfrak{q} \in D, \\ \phi(\mathfrak{p}, \mathfrak{q}) &= \phi_\ell, && \forall\ \mathfrak{p} \in U, \mathfrak{q} \in D. \end{aligned}$$

For each $\mathfrak{p} \in S$ we denote by $\lambda_\mathfrak{p}$ the slope of the branch of ϕ_ℓ in the non-optimised tree to which the leaf of $\mathfrak{p}$ belongs. Also, we denote

$$c := \frac{V_\ell + \lambda_{\min}}{e_1 \cdots e_{\ell-1}}, \qquad \delta_\mathfrak{p} := \frac{\lambda_\mathfrak{p} - \lambda_{\min}}{e_1 \cdots e_{\ell-1}}.$$

The explicit formulas presented in Proposition 2.31 show that for all $\mathfrak{p} \in U, \mathfrak{q} \in D$:

$$\begin{aligned} w_\mathfrak{q}(\phi_{i,\mathfrak{p}}) &= (m_i/m_\ell)c, && \forall\ i \geqslant \ell, \\ w_\mathfrak{p}(\phi_{i,\mathfrak{q}}) &= \begin{cases} (m_i/m_\ell)c, & \text{if } i > \ell, \\ \delta_\mathfrak{p} + c, & \text{if } i = \ell. \end{cases} \end{aligned} \tag{4.9}$$

Let G be a polynomial of degree im_ℓ with support in U, and let G' be a polynomial of degree jm_ℓ with support in D. If $m := \operatorname{ord}_{\phi_\ell}(G')$, the

formulas (4.9) show that:

$$\begin{aligned} w_U\left(GG'\right) &= \min\left\{w_{\mathfrak{p}}\left(G\right) + m\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + jc, \\ w_D\left(GG'\right) &= w_D\left(G'\right) + ic. \end{aligned} \tag{4.10}$$

The first formula of (4.9) shows that D meets the criterion of Lemma 4.7 and admits precomputation. In order to show that U admits precomputation too, we need another lemma.

Notation. *For each $\mathfrak{p} \in D$, we denote $m_{\mathfrak{p}} := m_{\ell+1,\mathfrak{p}} = e_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}} m_\ell$.*

Note that $e_{\ell,\mathfrak{p}}$ is independent of $\mathfrak{p}$, because it is the least positive denominator of $\lambda_{\min}$.

Lemma 4.16. *Let $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ be an output of* MaxMin$[S; m_\ell]$ *and $g = g_{\mathbb{i}}$ the corresponding numerator. Let $\mathfrak{p} \in S$ be the least prime with $w_{\mathfrak{p}}(g) = w_S(g)$.*

1. *If $\mathfrak{p} \in D$ and $m_{\mathfrak{p}} \mid i_{\mathfrak{p}}$, then the next $e_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}}$ output numerators are*

$$g\phi_\ell,\ g\phi_\ell^2,\ \ldots,\ g\phi_\ell^{e_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}} - 1},$$

and finally

$$\left(\prod_{\mathfrak{q}\neq\mathfrak{p}} g_{i_{\mathfrak{q}},\mathfrak{q}}\right) \cdot g_{i_{\mathfrak{p}}+m_{\mathfrak{p}},\mathfrak{p}}.$$

2. *If $\mathfrak{p} \in U$, then $m_{\mathfrak{q}} \mid i_{\mathfrak{q}}$ for all $\mathfrak{q} \in D$.*

Proof. Suppose $\mathfrak{p} \in D$ and $m_{\mathfrak{p}} \mid i_{\mathfrak{p}}$. Since the element $g_{i_{\mathfrak{p}},\mathfrak{p}}$, a numerator of the Okutsu $\mathfrak{p}$-basis has degree divisible by $m_{\mathfrak{p}}$, it is not divisible by ϕ_ℓ, and $g_{i_{\mathfrak{p}}+1,\mathfrak{p}} = g_{i_{\mathfrak{p}},\mathfrak{p}}\phi_\ell$. Hence, the next output numerator is $g\phi_\ell$.

By (4.9), $w_{\mathfrak{p}}(g\phi_\ell) = w_{\mathfrak{p}}(g) + c$, while $w_{\mathfrak{q}}(g\phi_\ell) \geqslant w_{\mathfrak{q}}(g) + c$ for all $\mathfrak{q} \in S$. Thus, the least prime with $w_{\mathfrak{q}}(g\phi_\ell) = w_S(g\phi_\ell)$ is, once again, the prime $\mathfrak{p}$.

This argument may be iterated as long as $\operatorname{ord}_{\phi_\ell}(g_{i_{\mathfrak{p}}+km_\ell,\mathfrak{p}}) = k < e_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}}$. For $k = f_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}} - 1$, the prime $\mathfrak{p}$ is still the least one satisfying $w_{\mathfrak{p}}\left(g\phi_\ell^k\right) = w_S\left(g\phi_\ell^k\right)$, so that the component of the multi-index corresponding to $\mathfrak{p}$ is increased and the output multi-index is $\mathbb{i} + m_{\mathfrak{p}}\mathbb{u}_{\mathfrak{p}}$.

Since $i_{\mathfrak{p}} + m_{\mathfrak{p}} \equiv 0 \pmod{m_{\mathfrak{p}}}$, the polynomial $g_{i_{\mathfrak{p}}+m_{\mathfrak{p}},\mathfrak{p}}$ is not divisible by ϕ_ℓ; in particular, it is not equal to ϕ_ℓ times the previous polynomial and the prime $\mathfrak{p}$ may cease to satisfy $w_{\mathfrak{p}}\left(g_{\mathbb{i}+m_{\mathfrak{p}}\mathbb{u}_{\mathfrak{p}}}\right) = w_S\left(g_{\mathbb{i}+m_{\mathfrak{p}}\mathbb{u}_{\mathfrak{p}}}\right)$.

The second item follows immediately from the first. □

Corollary 4.17. *U admits precomputation.*

Proof. Let us show that U meets the criterion of Lemma 4.7.

Let $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ be an output of MaxMin$[S; m_\ell]$ and let $g = g_{\mathbb{i}}$ be the corresponding numerator. Suppose that $w_U(g) = w_S(g)$. With respect to the ordering of S, all elements in U are less than all elements in D; hence, the least prime $\mathfrak{p}$ with $w_{\mathfrak{p}}(g) = w_S(g)$ belongs to U. By (2) of Lemma 4.16, $m_{\mathfrak{q}} \mid i_{\mathfrak{q}}$ for all $\mathfrak{q} \in D$, and this implies that none of the numerators $g_{i_{\mathfrak{q}},\mathfrak{q}}$, for $\mathfrak{q} \in D$, is divisible by ϕ_ℓ.

Therefore, (4.9) shows that $w_{\mathfrak{p}}\left(g_{i_{\mathfrak{q}},\mathfrak{q}}\right) = (i_{\mathfrak{q}}/m_\ell)c$ for all $\mathfrak{p} \in U$, and the value $w_{\mathfrak{p}}(G_{\mathbb{i}}) = w_{\mathfrak{p}}\left(\prod_{\mathfrak{q}\in D} g_{i_{\mathfrak{q}},\mathfrak{q}}\right)$ is independent of $\mathfrak{p} \in U$. □

4.5.3 Proof of the Theorem in Case (C)

Denote the respective output families of numerators of MaxMin$[U; m_\ell]$ and MaxMin$[D; m_\ell]$ by:

$$1, g_1, \ldots, g_{n_U/m_\ell},$$
$$1, g'_1, \ldots, g'_{n_D/m_\ell},$$

respectively. Note that $\deg g_k = \deg g'_k = km_\ell$, for all k.

Let $\mathbb{i} = (i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ be an output of MaxMin$[S; m_\ell]$. Since U and D admit precomputation, Corollary 4.6, Lemma 4.10, and Lemma 4.11 show that $g_{\mathbb{i}} = g_i g'_j$, for the k-th output (i, j) of MaxMin$[S = U \cup D; m_\ell]$.

Notation. *We denote $[j] := \mathrm{ord}_{\phi_\ell}(g'_j)$, for $0 \leqslant j \leqslant n_D/m_\ell$.*

By Lemma 4.16, all indices $i_{\mathfrak{q}}$, for $\mathfrak{q} \in D$ are divisible by $m_{\mathfrak{q}}$ except eventually for one, say $i_{\mathfrak{q}_0}$. Hence, $[j]$ is the residue of the euclidian division of $i_{\mathfrak{q}_0}$ by $m_{\mathfrak{q}_0}$. Note that $[j] = 0$ if and only if $m_{\mathfrak{q}} \mid i_{\mathfrak{q}}$ for all $\mathfrak{q} \in D$.

Let us consider rational numbers ν_i, ν'_j as in (4.7). The formulas (4.10) translate into

$$\begin{aligned} w_U\left(g_i g'_j\right) &= \min\left\{w_{\mathfrak{p}}\left(g_i\right) + [j]\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + jc, \\ w_D\left(g_i g'_j\right) &= w_D\left(g'_j\right) + ic = \nu'_j + (i+j)c. \end{aligned} \tag{4.11}$$

Lemma 4.18. *These data ν_i, ν'_j satisfy the following properties for all $i, j > 0$:*

1. *$\nu_{i-1} < \nu_i$.*
2. *$\nu'_{j-1} \leqslant \nu'_j$ and if $[j] \neq 0$ then equality holds.*

Proof. Take any $\mathfrak{p} \in U$. By Proposition 2.31, we have the value $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{p}}\right) = (V_\ell + \lambda_{\ell,\mathfrak{p}})/(e_1 \cdots e_{\ell-1})$. Since $\lambda_{\ell,\mathfrak{q}} > \lambda_{\min}$, the maximality of g_i implies

$$w_U\left(g_i\right) \geqslant w_U\left(g_{i-1}\phi_{\ell,\mathfrak{p}}\right) > w_U\left(g_{i-1}\right) + c,$$

because $w_{\mathfrak{q}}\left(\phi_{\ell,\mathfrak{p}}\right) = (V_\ell + \lambda)/(e_1 \cdots e_{\ell-1})$ for some $\lambda > \lambda_{\min}$, for all $\mathfrak{q} \in U$.

This proves (1). Similarly, the maximality of g'_j implies $\nu'_{j-1} \leqslant \nu'_j$.

By Lemma 4.16, $[j] \neq 0$ implies that $g'_j = g'_{j-1}\phi_\ell$. Since $w_{\mathfrak{q}}\left(\phi_\ell\right) = c$ for all $\mathfrak{q} \in D$, this implies $w_D\left(g'_j\right) = w_D\left(g'_{j-1}\right) + c$. This proves (2). □

Lemma 4.19. *Let $\mathbb{i} = (i, j)$ be an output pair of* MaxMin$[S = U \cup D; m_\ell]$. *Then,*

1. *Either $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$, or $\nu_{i-1} \leqslant \nu'_j < \nu_i$.*
2. *The next output pair is $(i+1, j)$ in the first case, and $(i, j+1)$ in the second case.*

Proof. Clearly, the initial pair $(0, 0)$ satisfies (1), the next output pair is $(1, 0)$, and it satisfies (1) too. Let us show by induction that if an output pair satisfies (1) then the next output pair is given as indicated in (2) and it satisfies (1) as well.

Suppose that $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$. If the previous output pair was $(i-1, j)$, the induction hypothesis implies that we had $\nu'_{j-1} \leqslant \nu_{i-1} \leqslant \nu'_j$. Since $\nu'_{j-1} \leqslant \nu_{i-1} < \nu_i \leqslant \nu'_j$, we have $[j] = 0$ by item (2) of Lemma 4.18. If

the previous output pair was $(i, j-1)$, then $\nu'_{j-1} < \nu_i$ by the induction hypothesis. This leads again to $\nu'_{j-1} < \nu'_j$ and to $[j] = 0$.

Thus, (4.11) shows that

$$\begin{aligned} w_U\left(g_i g'_j\right) &= w_U\left(g_i\right) + jc \\ &= \nu_i + (i+j)c \\ &\leqslant \nu'_j + (i+j)c \\ &= w_D\left(g_i g'_j\right). \end{aligned}$$

Thus, $w_U\left(g_i g'_j\right) = w_S\left(g_i g'_j\right)$ and the next output pair is $(i+1, j)$. The arguments of the proof of Proposition 4.15 show that $(i+1, j)$ satisfies (1).

Suppose that $\nu_{i-1} \leqslant \nu'_j < \nu_i$. By (4.11), we have

$$\begin{aligned} w_D\left(g_i g'_j\right) &= \nu'_j + (i+j)c \\ &< \nu_i + (i+j)c \\ &= w_U\left(g_i\right) + jc \\ &\leqslant \min\left\{w_{\mathfrak{p}}\left(g_i\right) + [j]\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + jc \\ &= w_U\left(g_i g'_j\right). \end{aligned}$$

Thus, $w_D\left(g_i g'_j\right) = w_S\left(g_i g'_j\right)$ and the next output pair is $(i, j+1)$. The arguments of the proof of Proposition 4.15 show that $(i, j+1)$ satisfies (1). □

Lemma 4.20. *Consider indices $0 \leqslant k \leqslant i$ and let G be a polynomial of degree $(i-k)m_\ell$ with support in U. Then,*

$$\min\left\{w_{\mathfrak{p}}\left(G\right) + k\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} \leqslant \nu_i + (i-k)c.$$

Proof. Let $\mathfrak{q} \in U$ be a prime ideal with a maximal value of $\lambda_{\mathfrak{q}}$. The statement

follows from the following chain of inequalities:

$$\begin{aligned} \min\left\{w_{\mathfrak{p}}(G)+k\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U}+kc &\leqslant w_U\left(G\phi^k_{\ell,\mathfrak{q}}\right)\\ &\leqslant w_U(g_i)\\ &=\nu_i+ic. \end{aligned} \tag{4.12}$$

The second inequality of (4.12) follows from the maximality of g_i. The first inequality is deduced from the formulas from Proposition 2.31. In fact, for any $\mathfrak{p}\in U$, these formulas yield $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}})=(V_\ell+\lambda)/(e_1\cdots e_{\ell-1})$, for a certain slope λ, depending on $\mathfrak{p}$, such that $\lambda\geqslant\lambda_{\mathfrak{p}}$; hence,

$$\begin{aligned} w_{\mathfrak{p}}\left(G\phi^k_{\ell,\mathfrak{q}}\right) &= w_{\mathfrak{p}}(G)+k\frac{V_\ell+\lambda}{e_1\cdots e_{\ell-1}}\\ &\geqslant w_{\mathfrak{p}}(G)+k(c+\delta_{\mathfrak{p}}), \end{aligned}$$

for all $\mathfrak{p}\in U$, which implies the first inequality in (4.12).

More precisely, if $i(\mathfrak{p},\mathfrak{q})=\ell$, then $\lambda=\lambda^{\mathfrak{q}}_{\mathfrak{p}}$ or $\lambda=\min\left\{\lambda^{\mathfrak{q}}_{\mathfrak{p}},\lambda^{\mathfrak{p}}_{\mathfrak{q}}\right\}$, according to $\phi(\mathfrak{p},\mathfrak{q})$ being equal to $\phi_{\ell,\mathfrak{q}}$ or not. Now, if $\mathfrak{p}$ and $\mathfrak{q}$ belong to the same ϕ_ℓ-branch of the non-optimised tree, we have (see Section 2.7)

$$\lambda_{\mathfrak{p}}=\lambda_{\mathfrak{q}}<\min\left\{\lambda^{\mathfrak{q}}_{\mathfrak{p}},\lambda^{\mathfrak{p}}_{\mathfrak{q}}\right\}\leqslant\lambda.$$

If $\mathfrak{p}$ and $\mathfrak{q}$ belong to different ϕ_ℓ-branches of the non-optimised tree, then $\lambda^{\mathfrak{q}}_{\mathfrak{p}}=\lambda_{\mathfrak{p}}$, $\lambda^{\mathfrak{p}}_{\mathfrak{q}}=\lambda_{\mathfrak{q}}$, so that, again,

$$\lambda_{\mathfrak{p}}=\min\left\{\lambda_{\mathfrak{p}},\lambda_{\mathfrak{q}}\right\}=\min\left\{\lambda^{\mathfrak{q}}_{\mathfrak{p}},\lambda^{\mathfrak{p}}_{\mathfrak{q}}\right\}\leqslant\lambda.$$

Finally, if $i(\mathfrak{p},\mathfrak{q})>\ell$, then $\lambda=\lambda_{\ell,\mathfrak{q}}=\lambda_{\ell,\mathfrak{p}}\geqslant\lambda_{\mathfrak{p}}$, by Proposition 2.31. □

We are ready to prove Theorem 4.12 in Case (C).

Proposition 4.21. *In Case (C), all output multi-indices of* $\mathrm{MaxMin}[S;m_\ell]$ *are maximal amongst the multi-indices of the same degree whose coordinates are all divisible by* m_ℓ.

Proof. Let $\mathbb{i}=(i_{\mathfrak{p}})_{\mathfrak{p}\in S}$ be an output multi-index of $\mathrm{MaxMin}[S;m_\ell]$, obtained by joining the i-th output of $\mathrm{MaxMin}[U]$ and the j-th output of $\mathrm{MaxMin}[D]$.

Let g be a polynomial of degree $(i+j)m_\ell$ with support in S. We may write $g = GG'$, with G, G' polynomials with support in U and D, respectively.

Suppose $\deg G = (i-k)m_\ell$, $\deg G' = (j+k)m_\ell$, for certain $k \in \mathbb{Z}$. Let us write

$$G' = H\phi_\ell^m, \qquad \phi_\ell \nmid H,\ \deg H = qm_\ell.$$

Note that $q + m = j + k$. By (4.10),

$$\begin{aligned} w_U\left(GG'\right) &= \min\left\{w_{\mathfrak{p}}\left(G\right) + m\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + (j+k)c, \\ w_D\left(GG'\right) &= w_D\left(G'\right) + (i-k)c. \end{aligned}$$

Since $w_{\mathfrak{p}}\left(\phi_\ell\right) = c$ for all $\mathfrak{q} \in D$, the last equality leads to

$$\begin{aligned} w_D\left(GG'\right) &= w_D\left(G'\right) + (i-k)c \\ &= w_D\left(H\right) + (m+i-k)c \\ &\leqslant w_D\left(g'_q\right) + (m+i-k)c \\ &= \nu'_q + (q+m+i-k)c \\ &= \nu'_q + (i+j)c. \end{aligned} \tag{4.13}$$

By Lemma 4.19, we may distinguish two cases according to the comparison of ν_i with ν'_j.

Case 1. $\nu'_{j-1} \leqslant \nu_i \leqslant \nu'_j$. In this case, we saw during the proof of Lemma 4.19 that $[j] = 0$. Hence, $w_S\left(g_i g'_j\right) = w_U\left(g_i g'_j\right) = \nu_i + (i+j)c$, by (4.11). We want to show that

$$\begin{aligned} w_S\left(GG'\right) &= \min\left\{w_U\left(GG'\right), w_D\left(GG'\right)\right\} \\ &\leqslant \nu_i + (i+j)c. \end{aligned}$$

If $m \leqslant k$, then Lemma 4.20 shows that

$$\begin{aligned} w_U\left(GG'\right) &= \min\left\{w_{\mathfrak{p}}\left(G\right) + m\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + (j+k)c \\ &\leqslant \min\left\{w_{\mathfrak{p}}\left(G\right) + k\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + (j+k)c \\ &\leqslant \nu_i + (i-k)c + (j+k)c \\ &= \nu_i + (i+j)c. \end{aligned}$$

If $m > k$, then $q < j$, or equivalently $q \leqslant j-1$. Thus, (4.13) shows that

$$\begin{aligned} w_D\left(GG'\right) &\leqslant \nu'_q + (i+j)c \\ &\leqslant \nu'_{j-1} + (i+j)c \\ &\leqslant \nu_i + (i+j)c. \end{aligned}$$

Case 2. $\nu_{i-1} \leqslant \nu'_j < \nu_i$. In this case, Lemma 4.19 and (4.11) show that $w_S\left(g_i g'_j\right) = w_D\left(g_i g'_j\right) = \nu'_j + (i+j)c$. We want to show that

$$\begin{aligned} w_S\left(GG'\right) &= \min\left\{w_U\left(GG'\right), w_D\left(GG'\right)\right\} \\ &\leqslant \nu'_j + (i+j)c. \end{aligned}$$

If $m < k$, then $m \leqslant k-1$. Having in mind that $\deg G/m_\ell = i-k = (i-1)-(k-1)$, Lemma 4.20 shows that

$$\begin{aligned} w_U\left(GG'\right) &= \min\left\{w_{\mathfrak{p}}\left(G(\theta)\right) + m\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + (j+k)c \\ &\leqslant \min\left\{w_{\mathfrak{p}}\left(G(\theta)\right) + (k-1)\delta_{\mathfrak{p}}\right\}_{\mathfrak{p}\in U} + (j+k)c \\ &\leqslant \nu_{i-1} + (i-k)c + (j+k)c \\ &= \nu_{i-1} + (i+j)c \\ &\leqslant \nu'_j + (i+j)c. \end{aligned}$$

If $m \geqslant k$, then $q \leqslant j$ and (4.13) shows that

$$w_D\left(GG'\right) \leqslant \nu'_q + (i+j)c \leqslant \nu'_j(i+j)c.$$

□

4.6 MaxMin **for unconnected trees**

In this section we will discuss the application of the MaxMin algorithm to a set of unconnected trees.

4.6.1 The separated MaxMin algorithm

Consider a partition

$$S = S_1 \cup \cdots \cup S_t \subseteq \mathcal{P}, \tag{4.14}$$

of a subset of $\mathcal{P}$such that all elements in S_i are less than all elements of S_j for $i < j$. We say that the partition of S is "separated" if for any $1 \leqslant i \neq j \leqslant t$ and any choice of $\mathfrak{p} \in S_i$, $\mathfrak{q} \in S_j$, the following equivalent conditions hold:

1. $i(\mathfrak{p}, \mathfrak{q}) = 0$.
2. $\mathbb{t}_\mathfrak{p}$ and $\mathbb{t}_\mathfrak{q}$ do not belong to the same connected subtree of $\mathfrak{T}$.
3. $\gcd(\overline{F}_\mathfrak{p}, \overline{F}_\mathfrak{q}) = 1$.

In this case, the formulas in Proposition 2.31 show that:

$$w_\mathfrak{p}(\phi_{i,\mathfrak{q}}) = 0, \qquad 1 \leqslant i \leqslant r_\mathfrak{q} + 1. \tag{4.15}$$

For any $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$, the condition "$\mathbb{t}_\mathfrak{p}$ and $\mathbb{t}_\mathfrak{q}$ belong to the same connected subtree of $\mathfrak{T}$" defines an equivalence relation $\sim$ on $\mathcal{P}$. Hence, every subset $S \subset \mathcal{P}$ admits a unique separated partition (4.14) satisfying moreover:

$$\mathfrak{p}, \mathfrak{q} \in S_i \Longrightarrow \mathfrak{p} \sim \mathfrak{q},$$

for all $1 \leqslant i \leqslant t$.

For separated partitions, we may consider the "separated MaxMin", presented in Algorithm 4.4.

The aim of this section is to prove the following result.

Theorem 4.22. *The output multi-indices of* SepMaxMin$[S = S_1 \cup \cdots \cup S_t]$ *are maximal.*

Algorithm 4.4 SepMaxMin$[S = S_1 \cup \cdots \cup S_t]$ algorithm

Input: A separated partition $S = S_1 \cup \cdots \cup S_t$ of S, and extended families $\{G_{k,i} : 0 \leqslant k \leqslant n_{S_i}\}$ of numerators of Okutsu S_i-bases for all $1 \leqslant i \leqslant t$.

Output: A family $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n_S} \in \mathbb{N}^t$ of multi-indices $\mathbb{i}_k = (k_i)_{1 \leqslant i \leqslant t}$ of degree $0, 1, \ldots, n_S$, respectively.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0) \in \mathbb{N}^t$
2: **for** $k = 0 \to n_S - 1$ **do**
3: $\quad j \leftarrow \min\{1 \leqslant i \leqslant t : w_{S_i}(G_{k_i,i}) = w_S(G_{\mathbb{i}_k})\}$
4: $\quad \mathbb{i}_{k+1} \leftarrow \mathbb{i}_k + \mathbb{u}_j$
5: **end for**

Thanks to the separateness of the partition, condition (4.15) shows that

$$w_{S_i}(G_{k_i,i}) = w_{S_i}(G_{\mathbb{i}_k}),$$

for all $1 \leqslant i \leqslant t$ and all $\mathbb{i}_k$. Hence SepMaxMin$[S = S_1 \cup \cdots \cup S_t]$ has the same output as MaxMin$[S = S_1 \cup \cdots \cup S_t]$.

The only difference between the two algorithms is that in the separated algorithm for the computation of $w_{S_i}(G_{\mathbb{i}_k})$ we only need to concern ourselves with the S_i-valuation of the polynomial $G_{k_i,i}$.

On the other hand, (4.15) implies that the criterion of Lemma 4.7 is fulfilled, and so SepMaxMin$[S = S_1 \cup \cdots \cup S_t]$ has the same output as MaxMin$[S]$ after natural identification of the output multi-indices.

Therefore, we may drop the condition of the connectedness of $\mathfrak{T}_S$ in Theorem 4.3. In fact, Theorem 4.3 and Theorem 4.22 imply the main result of this memoir.

Theorem 4.23. *All output multi-indices of* MaxMin$[S]$ *are maximal.*

In the remainder of this section, we will agree that $w_{S_i}(G_{-1,i}) = -1$ for all $1 \leqslant i \leqslant t$.

Lemma 4.24. *Let* $\mathbb{i}_k = (k_i)_{1 \leqslant i \leqslant t}$ *be the* k*-th output multi-index resulting from* SepMaxMin$[S = S_1 \cup \cdots \cup S_t]$*, and let* j *be the minimal index as in line 3 of the algorithm. Then, for all* $1 \leqslant i \leqslant t$*,*

$$w_{S_i}(G_{k_i,i}) \geqslant w_{S_j}\left(G_{k_j,j}\right) \geqslant w_{S_i}(G_{k_i-1,i}). \tag{4.16}$$

Proof. The first inequality is a direct consequence of

$$w_{S_j}\left(G_{k_j,j}\right) = w_S\left(g_{\mathbb{i}_k}\right) \leqslant w_{S_i}\left(G_{k_i}\right), 1 \leqslant i \leqslant t.$$

Let us prove the second inequality by induction on the degree k of the output multi-index. Clearly, the initial multi-index $\mathbb{i}_0 = (0, \ldots, 0) \in \mathbb{N}^t$ satisfies (4.16). Let ℓ be the minimal index in the $(k-1)$-th iteration, so that

$$\mathbb{i}_{k-1} = \mathbb{i}_k - \mathbb{u}_\ell, \qquad \mathbb{i}_{k+1} = \mathbb{i}_k + \mathbb{u}_j.$$

Let us assume that $\mathbb{i}_{k-1}$ satisfied (4.16). Thus

$$w_{S_\ell}\left(G_{k_\ell - 1,\ell}\right) \geqslant w_{S_i}\left(G_{k_i-1,i}\right), \qquad 1 \leqslant i \leqslant t.$$

If $\ell = j$, we immediately deduce:

$$w_{S_j}\left(G_{k_j,j}\right) \geqslant w_{S_j}\left(G_{k_j-1,j}\right) \geqslant w_{S_i}\left(G_{k_i-1,i}\right), \qquad 1 \leqslant i \leqslant t.$$

On the other hand, if $\ell \neq j$, by construction ℓ satisfies:

$$w_{S_\ell}\left(G_{k_\ell-1,\ell}\right) = w_S\left(g_{\mathbb{i}_{k-1}}\right) \leqslant w_{S_j}\left(G_{k_j,j}\right),$$

and we also deduce the inequality we are looking for:

$$w_{S_j}\left(G_{k_j,j}\right) \geqslant w_{S_\ell}\left(G_{k_\ell-1,\ell}\right) \geqslant w_{S_i}\left(G_{k_i-1,i}\right), \qquad 1 \leqslant i \leqslant t.$$

□

We can now proceed to prove this section's main theorem.

Proof of Theorem 4.22. We will proceed by induction on the iteration m. For $m = 0$, $\mathbb{i}_0 = (0, \ldots, 0)$ which is maximal by virtue of being unique.

Suppose all output multi-indices up to and including $\mathbb{i}_k = (k_1, \ldots, k_t)$ are maximal. Let S_j be minimal in iteration m so that $\mathbb{i}_{k+1} = \mathbb{i}_k + \mathbb{u}_j$. It

will be shown that

$$w_S\left(G_{\mathbb{i}_{k+1}}\right) \geqslant w_S\left(G_{\mathbb{j}}\right) \qquad \forall\ \mathbb{j},\ \deg \mathbb{j} = k+1.$$

In order that $\mathbb{j} = (j_1, \ldots, j_t) \neq \mathbb{i}_{k+1}$ have degree $k+1$, it is necessary that either

1. $\mathbb{j} = \mathbb{i}_k + \mathbb{u}_\ell$ for $\ell \neq j$; or

2. there exists at least one coordinate ℓ such that $j_\ell < k_\ell$.

For case (1), $w_S\left(G_{\mathbb{j}}\right) = w_{S_j}\left(G_{k_j,j}\right) = w_S\left(G_{\mathbb{i}_k}\right)$. In the case of (2), by (4.16) we can see that,

$$\begin{aligned} w_S\left(G_{\mathbb{i}_k}\right) &= w_{S_j}\left(G_{k_j,j}\right) \\ &\geqslant w_{S_\ell}\left(G_{k_\ell-1,\ell}\right) \\ &\geqslant w_{S_\ell}\left(G_{j_\ell,\ell}\right) \\ &\geqslant w_S\left(G_{\mathbb{j}}\right). \end{aligned}$$

Meanwhile, by the construction of $\mathbb{i}_{k+1}$ from $\mathbb{i}_k$ we have,

$$w_S\left(G_{\mathbb{i}_{k+1}}\right) \geqslant w_S\left(G_{\mathbb{i}_k}\right) \geqslant w_S\left(G_{\mathbb{j}}(\theta)\right),$$

and so $\mathbb{i}_{k+1}$ is maximal. □

4.7 Improvement of Okutsu approximations

During the execution of the MaxMin algorithm, as presented in Section 4.2, we have taken the $\mathfrak{p}$-valuation of $\phi_\mathfrak{p}$ the Okutsu approximation to $F_\mathfrak{p}$ to be formally infinite. This is practical, as $w_\mathfrak{p}\left(\phi_\mathfrak{p}(\theta)\right)$ depends on the choice of $\phi_\mathfrak{p}$ as an Okutsu approximation to $F_\mathfrak{p}$ and can be arbitrarily large.

However, in order to construct a basis, a concrete approximation must be chosen. The Montes algorithm provides us with an approximation and by using the Single Factor Lifting (SLF) algorithm, we can efficiently improve the approximation and raise its $\mathfrak{p}$-valuation.

In this section, we will compute a lower bound on the required valuation $w_{\mathfrak{p}}(\phi_{\mathfrak{p}}(\theta))$ so that we may use the minimum number of iterations of the SFL algorithm.

Let $G_0, G_1, \ldots, G_{n-1}$ be the family of numerators of a v-integral basis created by the MaxMin algorithm and let $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n-1}$ be the multi-indices that define them so that

$$G_k = \prod_{\mathfrak{p} \in \mathcal{P}} g_{\mathbb{i}_k[\mathfrak{p}],\mathfrak{p}}, \qquad 0 \leqslant k < n,$$

where $\mathbb{i}[\mathfrak{p}]$ is the $\mathfrak{p}$-coordinate of the multi-index $\mathbb{i}$ and $g_{j,\mathfrak{p}}$ is the j-th numerator of the Okutsu $\mathfrak{p}$-basis.

Let $\nu_k \in \mathbb{Q}$ be the maximal w-valuation for a monic polynomial in $\mathcal{O}[x]$ of degree k evaluated in θ. If we take the formally infinite $\mathfrak{p}$-valuation for $\phi_{\mathfrak{p}}$ for all $\mathfrak{p} \in \mathcal{P}$, then $w(G_k(\theta)) = \nu_k$.

For all $\mathfrak{p} \in \mathcal{P}$, $w_{\mathfrak{p}}(\phi_{\mathfrak{p}}(\theta))$ must be large enough that for all $1 \leqslant k < n$ such that $\phi_{\mathfrak{p}} \mid G_k$, we have

$$w_{\mathfrak{p}}(G_k(\theta)) \geqslant \nu_k.$$

Definition 4.25. *We define the required $\mathfrak{p}$-valuation for the Okutsu approximation to $F_{\mathfrak{p}}$ to be,*

$$W_{\mathfrak{p}} := \max\left\{ \nu_k - \sum_{\mathfrak{q} \neq \mathfrak{p}} w_{\mathfrak{p}}\left(g_{\mathbb{i}_k[\mathfrak{q}],\mathfrak{q}}(\theta)\right) : \mathbb{i}_k[\mathfrak{p}] = r_{\mathfrak{p}} + 1 \right\}.$$

Computationally, the value of $W_{\mathfrak{p}}$ is simple to calculate. At each iteration k where $\mathbb{i}_k[\mathfrak{p}]$ has reached the value $r_{\mathfrak{p}} + 1$ we sum the precomputed $\mathfrak{p}$-valuations for all the numerators being included in G_k, except the $\mathfrak{p}$-numerator and then subtract this from minimum valuation (which we must find to increment the multi-index for the next iteration anyway). We compare this to the previously stored $W_{\mathfrak{p}}$ and keep the greater value.

4.8 Further optimisation

The MaxMin algorithm can benefit from a number of optimisations, depending on the structure of the genetic tree $\mathfrak{T}$ of types which represent the prime ideals $\mathfrak{p} \in \mathcal{P}$ of $\mathcal{O}_L$.

One optimisation is the precomputation of a subset $S_{\mathbb{t}} \subseteq \mathcal{P}$ of prime ideals which share a common type in their genetic tree. Another, related, optimisation is the case where a set S of prime ideals has an index of coincidence $i(S) = \ell > 1$ greater than one. In this case, we can use $\mathrm{MaxMin}[S; m_\ell]$ to only calculate the indices divisible by m_ℓ, and fill in the remaining indices as per Lemma 4.10.

Both of these optimisations will be further explored during the complexity analysis in Chapter 6.

4.8.1 Terminal sides of a type

Let $\mathbb{t}$ be a type of order r with representative ϕ encountered during the execution of the Montes algorithm. We consider the order r Newton polygon $\mathrm{N}^-_{v_{\mathbb{t}},\phi,\omega}(f)$ created using the data inherent in the type, as shown in Figure 4.8.

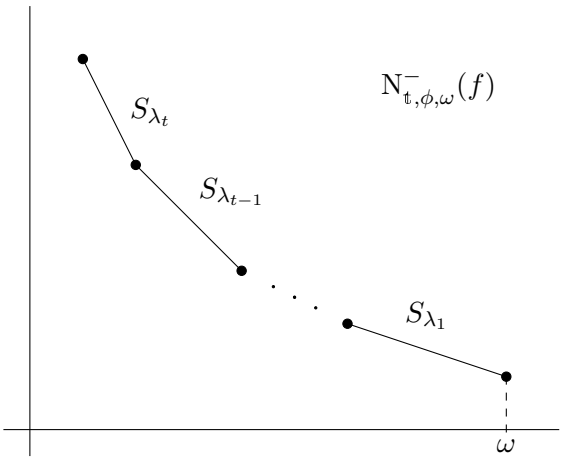


Figure 4.8: Higher order Newton polygon of f with multiple slopes.

For each side S_λ of the Newton polygon, we calculate a residual polynomial and consider its irreducible factors in the extension of the residual

class field $\mathbb{F}_r$.

$$\mathrm{R}_{v_{\mathbb{t}},\phi,\lambda}(f) = \zeta \prod \psi^{\omega_\psi}, \qquad \zeta \in \mathbb{F}_r, \psi \in \mathbb{F}_r[y].$$

A new type is created from the data $\mathbb{t}' = (\mathbb{t}; (\phi, \lambda, \psi))$. In the case that $\omega_\psi = 1$, this type is a leaf node of the genetic tree describing f and is an OM representation of a prime ideal $\mathfrak{p}_{\mathbb{t}'} \in \mathcal{P}$ of $\mathcal{O}_L$. As such, we call this new type $\mathbb{t}'$ *terminal*. We also call the prime ideal $\mathfrak{p}_{\mathbb{t}'}$ a *terminal prime ideal* of $\mathbb{t}$.

Definition 4.26. *Let $\mathbb{t}$ be a type of order r and ϕ a representative of $\mathbb{t}$. Consider the side S_λ of slope $-\lambda$ of the Newton polygon $\mathrm{N}^-_{\mathbb{t},\phi,\omega}(f)$ and let $S_{\mathbb{t},\phi,\lambda}$ be the set of all prime ideals of this side S_λ. We call S_λ a* terminal side *if it has one or more irreducible factors ψ with exponent $\omega_\psi = 1$.*

For a terminal side S_λ, the set of terminal prime ideals of that side is $I_\lambda \subset S_{\mathbb{t},\phi,\lambda}$. Additionally,

$$I = \bigcup_\lambda I_\lambda,$$

is the set of all terminal prime ideals of $\mathbb{t}$.

Let S_λ be a terminal side of $\mathbb{t}, \phi$, then the *terminal length* of S_λ is given by

$$\ell_{\mathrm{term}}(S_\lambda) = e_\lambda \sum_{\{\psi:\omega_\psi=1\}} f_\psi,$$

where e_λ is the denominator of $\lambda = h_\lambda/e_\lambda$ and f_ψ is the degree of ψ.

Let $\phi_1, \ldots, \phi_r$ be the representatives at each level of the type $\mathbb{t}$. For all prime ideals $\mathfrak{p} \in S_\mathbb{t}$ whose genetic tree contains $\mathbb{t}$, the first m_r elements of the Okutsu $\mathfrak{p}$-basis will coincide. Let,

$$\mathcal{N}_\mathbb{t} = \{1 = h_0, h_1, \ldots, h_{r-1}\},$$

be the family of these common numerators. This is a similar construction to that given in Section 4.4.

Proposition 4.27. *Let $\mathfrak{t}$ be a type of order $\ell-1$ and ϕ a representative of $\mathfrak{t}$ such that $\mathfrak{t}, \phi$ have terminal prime ideals $\mathfrak{p} \in I = I_{\lambda_1} \cup \cdots \cup I_{\lambda_t}$. Let $\mathcal{N}_{\mathfrak{t}}$ be the common family of numerators of $\mathfrak{t}$. Then for all λ,*

$$\mathcal{N}_\lambda = \bigcup_{a=0}^{\ell_{\text{term}}(S_\lambda)} \phi^a \mathcal{N}_{\mathfrak{t}} \cup \{\phi_\lambda\}, \qquad \phi_\lambda = \prod_{\mathfrak{p} \in I_\lambda} \phi_{\mathfrak{p}},$$

is an extended family of numerators of an Okutsu I_λ-basis.

Proof. We will first show that I_λ admits precomputation. We will then show how the precomputed numerators can be substituted for powers of ϕ and remain valid.

For $\mathfrak{p} \in I_\lambda$ and $\mathfrak{q} \in S_{\mathfrak{t},\phi,\lambda'} \subseteq S_{\mathfrak{t}} \backslash I_\lambda$,

$$w_{\mathfrak{p}}\left(\phi_{k,\mathfrak{q}}(\theta)\right) = \begin{cases} \dfrac{V_\ell + \lambda}{e_1 \cdots e_{\ell-1}}, & k = \ell \text{ and } \phi_{\ell,\mathfrak{q}} = \phi, \\ \dfrac{m_{k,\mathfrak{q}}}{m_\ell} \cdot \dfrac{V_\ell + \min\{\lambda, \lambda'\}}{e_1 \cdots e_{\ell-1}}, & k > \ell \text{ or } \phi_{\ell,\mathfrak{q}} \neq \phi. \end{cases}$$

Since these values depend only upon λ' which will be fixed for each $\mathfrak{q} \in S_{\mathfrak{t},\phi,\lambda'}$ and λ which is equal for all $\mathfrak{p} \in I_\lambda$, then the precomputation criterion given in Lemma 4.7 is met, and I_λ admits precomputation in $S_{\mathfrak{t}}$.

The index of coincidence of any two prime ideals $\mathfrak{p}, \mathfrak{q} \in S_{\mathfrak{t}}$ in the subset $I_\lambda \subseteq S_{\mathfrak{t}}$ is $i(\mathfrak{p}, \mathfrak{q}) = \ell$, so we can apply MaxMin$[I_\lambda; m_\ell]$ as given in Algorithm 4.3. However, as we will see, the output of the MaxMin algorithm for this set is fixed.

For two distinct prime ideals $\mathfrak{p}, \mathfrak{q} \in I_\lambda$, we have the degree adjusted $\mathfrak{p}$-valuations,

$$\hat{w}_{\mathfrak{p}}\left(\phi(\theta)\right) = \frac{1}{m_\ell} \cdot \frac{V_\ell + \lambda}{e_1 \cdots e_{\ell-1}} = \hat{w}_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right) < \hat{w}_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right), \quad \mathfrak{p} \neq \mathfrak{q}. \tag{4.17}$$

As such, the output multi-indices of $\mathrm{MaxMin}[I_\lambda; m_\ell]$ will be,

$$\begin{aligned}
\mathbb{i}_0 &= (0, 0, \ldots, 0, 0),\\
\mathbb{i}_1 &= (1, 0, \ldots, 0, 0),\\
&\vdots\\
\mathbb{i}_{e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}} &= (e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}, 0, \ldots, 0, 0),\\
\mathbb{i}_{e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}+1} &= (e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}, 1, \ldots, 0, 0),\\
&\vdots\\
\mathbb{i}_{\ell_{\mathrm{term}}(S_\lambda)-1} &= (e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}, e_{\ell,\mathfrak{p}_2} f_{\ell,\mathfrak{p}_2}, \ldots, e_{\ell,\mathfrak{p}_{s-1}} f_{\ell,\mathfrak{p}_{s-1}}, e_{\ell,\mathfrak{p}_s} f_{\ell,\mathfrak{p}_s} - 1),\\
\mathbb{i}_{\ell_{\mathrm{term}}(S_\lambda)} &= (e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}, e_{\ell,\mathfrak{p}_2} f_{\ell,\mathfrak{p}_2}, \ldots, e_{\ell,\mathfrak{p}_{s-1}} f_{\ell,\mathfrak{p}_{s-1}}, e_{\ell,\mathfrak{p}_s} f_{\ell,\mathfrak{p}_s}),
\end{aligned}$$

which will generate an extended family of numerators of degree divisible by m_ℓ,

$$\mathcal{N}_{I_\lambda;m_\ell} = \left\{1, \phi, \ldots, \phi^{e_{\ell,\mathfrak{p}_1} f_{\ell,\mathfrak{p}_1}-1}, \phi_{\mathfrak{p}_1}, \ldots, \phi^{e_{\ell,\mathfrak{p}_s} f_{\ell,\mathfrak{p}_s}-1}\phi_{\mathfrak{p}_1}\cdots\phi_{\mathfrak{p}_{s-1}}; \phi_{\mathfrak{p}_1}\cdots\phi_{\mathfrak{p}_s}\right\}.$$

Using (4.17), we can see that

$$w_{I_\lambda}\left(\phi_{\mathfrak{p}}(\theta)\right) = w_{I_\lambda}\left(\phi^{e_{\ell,\mathfrak{p}} f_{\ell,\mathfrak{p}}}(\theta)\right), \qquad \forall\ \mathfrak{p} \in I_\lambda,$$

and so we can replace all but the last of the numerators in $\mathcal{N}_{I_\lambda;m_\ell}$ with powers of ϕ,

$$\mathcal{N}'_{I_\lambda;m_\ell} = \left\{1, \phi, \phi^2, \ldots, \phi^{\ell_{\mathrm{term}}(S_\lambda)-1}; \phi_{\mathfrak{p}_1}\cdots\phi_{\mathfrak{p}_s}\right\}.$$

By applying part (2) of Lemma 4.10 to $\mathcal{N}_{\mathbb{t}}$ and $\mathcal{N}'_{I_\lambda;m_\ell}$, we can construct an extended family of numerators of an Okutsu I_λ-basis. It can be seen that this family of numerators will coincide with $\mathcal{N}_\lambda$. □

4.9 Basis element reduction modulo an $\mathfrak{m}$-power

During the construction of a triangular v-integral basis, the coefficients of the numerators can grow larger than is necessary. It is beneficial to reduce

the coefficients of each element of the basis modulo a power of $\mathfrak{m}$. This leaves us with smaller basis elements and less computation should we wish to then convert the basis to Hermite Normal Form or, in fact, perform any computation with the basis. This is especially important in the case of function fields, as the coefficients are themselves polynomials.

The following result is obvious.

Lemma 4.28. *Let $g, G \in \mathcal{O}[x]$ be monic polynomials of degree m and let $\nu = w(g(\theta))$. Then,*

$$G \equiv g \pmod{\mathfrak{m}^{\lfloor \nu \rfloor}} \implies w(G(\theta)) \geqslant \lfloor \nu \rfloor,$$
$$G \equiv g \pmod{\mathfrak{m}^{\lceil \nu \rceil}} \implies w(G(\theta)) \geqslant \nu.$$

Corollary 4.29. *Let $\pi^{-\lfloor \nu_0 \rfloor} g_0(\theta), \ldots, \pi^{-\lfloor \nu_{n-1} \rfloor} g_{n-1}(\theta)$ be a triangular v-integral basis, where $\nu_i = w(g_i(\theta))$. Let $G_i \in \mathcal{O}[x]$ be monic polynomials such that $G_i \equiv g_i \pmod{\mathfrak{m}^{\lfloor \nu_i \rfloor}}$ for all $0 \leqslant i < n$. Then the elements $\pi^{-\lfloor \nu_0 \rfloor} G_0(\theta), \ldots, \pi^{-\lfloor \nu_{n-1} \rfloor} G_{n-1}(\theta)$ also form a triangular v-integral basis.*

Proof. By Theorem 1.23 it suffices to show that $\lfloor w(G_i(\theta)) \rfloor$ is maximal amongst all monic polynomials in $\mathcal{O}[x]$ of the same degree. By Lemma 4.28, $\lfloor w(G_i(\theta)) \rfloor = \lfloor w(g_i(\theta)) \rfloor$ is maximal. □

If $g_0, \ldots, g_{n-1}$ are the numerators of a reduced triangular v-integral basis as in Definition 1.25, then they must be reduced modulo a slightly higher power of $\mathfrak{m}$ to remain reduced.

Corollary 4.30. *Let $\pi^{-\lfloor \nu_0 \rfloor} g_0(\theta), \ldots, \pi^{-\lfloor \nu_{n-1} \rfloor} g_{n-1}(\theta)$ be an $\mathfrak{m}$-reduced triangular v-integral basis, where $\nu_i = w(g_i(\theta))$. Let $G_i \in \mathcal{O}[x]$ be monic polynomials such that $G_i = g_i \pmod{\mathfrak{m}^{\lceil \nu_i \rceil}}$ for all $0 \leqslant i < n$. Then $\pi^{-\lfloor \nu_0 \rfloor} G_0(\theta)$, $\ldots, \pi^{-\lfloor \nu_{n-1} \rfloor} G_{n-1}(\theta)$ is also an $\mathfrak{m}$-reduced triangular v-integral basis.*

Proof. By Theorem 1.26, for a triangular v-integral basis to be reduced, the numerators must have maximal valuation amongst all polynomials of the same degree. By the second part of Lemma 4.28, $w(G_i) = w(g_i)$ is maximal. □

5

Triangular bases of fractional ideals

"Everything is hard before it is easy."

– Johann Wolfgang von Goethe

This chapter will provide details of how the MaxMin algorithm is adapted to compute local bases of fractional ideals.

Let (K, v) be a discrete valued field with valuation ring $\mathcal{O}$. Let $\mathfrak{m}$ be the maximal ideal, $\pi \in \mathfrak{m}$ a generator of $\mathfrak{m}$ and $\mathbb{F} = \mathcal{O}/\mathfrak{m}$ the residue class field.

Let K_v be the completion of K, and retain $v : \overline{K}_v^* \to \mathbb{Q}$ the canonical extension of v to a fixed algebraic closure of K_v. Let $\mathcal{O}_v$ be the valuation ring of K_v, and $\mathfrak{m}_v$ its maximal ideal.

Let $f \in \mathcal{O}[x]$ be a monic, irreducible polynomial of degree n and fix a root $\theta \in \overline{K}$ in the algebraic closure of K. Let $L = K(\theta)$ be the finite extension of K defined by f and $\mathcal{O}_L$ the integral closure of $\mathcal{O}$ in L, which is a Dedekind domain. We suppose that $\mathcal{O}_L$ is finitely generated as an $\mathcal{O}$-module and denote by $\mathcal{P}$ the set of prime ideals of $\mathcal{O}_L$.

Let $\mathcal{I}_L$ be the set of non-zero fractional ideals of $\mathcal{O}_L$ and let $I \in \mathcal{I}_L$ be one such ideal,

$$I = \prod_{\mathfrak{p} \mid \mathfrak{m}} \mathfrak{p}^{a_\mathfrak{p}}.$$

We consider the map analogous to $w_{\mathfrak{m},I}$, which is detailed in Definition 1.4, for a prime ideal $\mathfrak{p} \in \mathcal{P}$,

$$\begin{aligned} w_{\mathfrak{p},I} : L &\longrightarrow \mathbb{Q} \cup \{\infty\}, \\ \alpha &\longmapsto w_{\mathfrak{p},I}(\alpha) = w_\mathfrak{p}(\alpha) - a_\mathfrak{p}/e(\mathfrak{p}/\mathfrak{m}). \end{aligned} \tag{5.1}$$

Let $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\} \subseteq \mathcal{P}$ be a set of prime ideals of $\mathcal{O}_L$, then we may define $w_{S,I}(\alpha) := \min\{w_{\mathfrak{p},I}(\alpha)\}_{\mathfrak{p} \in S}$ as expected.

Let $\mathcal{O}_\mathfrak{p}$ be the completion of $\mathcal{O}_L$ with respect to the $\mathfrak{p}$-adic topology. As we saw in Section 3.1, the numerators of an Okutsu basis of $\mathcal{O}_\mathfrak{p}$ are also numerators for a $\mathfrak{p}$-integral basis of the fractional ideal $I = I \otimes_{\mathcal{O}_v} \mathcal{O}_\mathfrak{p}$ of $\mathcal{O}_\mathfrak{p}$. Let us now see that we may take combinations of these numerators to form polynomials of degrees $0, 1, \ldots, n-1$ which are maximal with respect to the valuation w_I, leading in this way to a reduced integral $\mathcal{O}$-basis of I.

5.1 Okutsu bases

Let $S \subseteq \mathcal{P}$ be a subset of the prime ideals of $\mathcal{O}_L$ and let $n_S = \sum_{\mathfrak{p} \in S} n_\mathfrak{p}$ be the degree of S.

A triangular S-basis of a prime ideal I is determined by a sequence of monic polynomials:

$$1, g_1, \ldots, g_{n_S - 1} \in \mathcal{O}[x], \qquad \deg g_i = i,$$

such that $w_{S,I}(g_i)$ is maximal amongst all monic polynomials of degree i.

In the next section, we will see how the MaxMin algorithm can also be used to find these maximal polynomials amongst all polynomials in $\mathrm{Ok}(S)$. As such, we are again interested in showing that $\mathrm{Ok}(S)$ contains polynomials which are maximal amongst all polynomials in $\mathcal{O}[x]$.

Fix a prime ideal $\mathfrak{p} \in \mathcal{P}$ and a fractional ideal $I \in \mathcal{I}_L$. It is clear, from the definition of $w_{\mathfrak{p},I}$ that,

$$w_{\mathfrak{p}}(g) > w_{\mathfrak{p}}(h) \iff w_{\mathfrak{p},I}(g) > w_{\mathfrak{p},I}(h), \tag{5.2}$$

for all $g, h \in \mathcal{O}[x]$.

Theorem 5.1. *Let $S \subseteq \mathcal{P}$ be a set of prime ideals. For any $h \in \mathcal{O}[x]$ monic of degree $0 \leqslant d \leqslant n$, there exists $g \in \mathrm{Ok}(S)$ also of degree d such that,*

$$w_{\mathfrak{p},I}(g) \geqslant w_{\mathfrak{p},I}(h), \qquad \forall\ \mathfrak{p} \in S.$$

Proof. From (5.2) it is clear that this follows from Theorem 3.9 and Theorem 3.15, which state the same condition for the valuations $w_{\mathfrak{p}}$. □

5.2 MaxMin for fractional ideals

The MaxMin algorithm can be adapted to produce an S-integral basis of the fractional ideal I by substituting the valuation $w_{\mathfrak{p}}$ for $w_{\mathfrak{p},I}$ for all $\mathfrak{p} \in S$, as presented in Algorithm 5.1.

Algorithm 5.1 MaxMin$[S, I]$ algorithm

Input: A subset $S = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\} \subseteq \mathcal{P}$ and Okutsu $\mathfrak{p}$-numerators $\{g_{i,\mathfrak{p}} : 0 \leqslant i \leqslant n_{\mathfrak{p}}\}$ for each $\mathfrak{p} \in S$. Additionally, a fractional ideal I.

Output: A family $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_{n_S} \in \mathbb{N}^s$ of multi-indices of degree $0, 1, \ldots, n_S$ respectively.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0)$
2: **for** $k = 0 \to (n_S - 1)$ **do**
3: $\quad j \leftarrow \min\{1 \leqslant i \leqslant s : w_{\mathfrak{p}_i,I}(g_{\mathbb{i}_k}) = w_{S,I}(g_{\mathbb{i}_k})\}$
4: $\quad \mathbb{i}_{k+1} \leftarrow \mathbb{i}_k + \mathbb{u}_j$
5: **end for**

Proposition 5.2. *All output multi-indices of* $\mathrm{MaxMin}[S, I]$ *are* I*-maximal.*

Let us consider the series of $(\mathfrak{p}, I)$-valuations of the Okutsu $\mathfrak{p}$-numerators,

$$w_{\mathfrak{p},I}\left(g_{0,\mathfrak{p}}\right), w_{\mathfrak{p},I}\left(g_{1,\mathfrak{p}}\right), \ldots, w_{\mathfrak{p},I}\left(g_{n_{\mathfrak{p}}-1,\mathfrak{p}}\right), w_{\mathfrak{p},I}\left(g_{n_{\mathfrak{p}},\mathfrak{p}}\right) = \infty.$$

This series of valuations is a "shifted" version of the normal $w_{\mathfrak{p}}$ valuations of the same numerators. Clearly, fixing $\mathfrak{p}$ and I,

$$w_{\mathfrak{p}}\left(g\right) - w_{\mathfrak{p}}\left(h\right) = w_{\mathfrak{p},I}\left(g\right) - w_{\mathfrak{p},I}\left(h\right),$$

for all $g, h \in \mathcal{O}[x]$. As such, the difference between any valuation and the next is the same in both the $w_{\mathfrak{p}}$ and the $w_{\mathfrak{p},I}$ cases.

This readily leads to the conclusion that by using the $w_{\mathfrak{p},I}$ valuations the MaxMin algorithm, which otherwise produces numerators of a v-integral basis of the maximal order, will do the same for a fractional ideal.

However, some clarification is required to follow the proofs in Chapter 4 in the fractional ideal case.

Valuation of products

Let $S \subseteq \mathcal{P}$ be a set of prime ideals and let I be a fractional ideal. If $h \in \mathcal{O}[x]$ has the same $\mathfrak{q}$-valuation for all $\mathfrak{q} \in S$, then we may write,

$$w_{S,I}\left(gh\right) = w_{S,I}\left(g\right) + w_{\mathfrak{q}}\left(h\right), \tag{5.3}$$

for all $g \in \mathcal{O}[x]$. This form is used extensively for w_S during the proof of Theorem 4.12. As an example, let g_i, g'_j be output numerators of $\mathrm{MaxMin}[U; m_\ell]$ and $\mathrm{MaxMin}[D, m_\ell]$ respectively. In the fractional ideal case we would then have, $w_{D,I}\left(g_i g'_j\right) = \nu'_j + (i+j)c$, where c retains the same value as in the maximal order case.

Precomputation

Let $S \subseteq \mathcal{P}$ be a set of prime ideals and let $J \subseteq S$ be an interval of S. Lemma 4.7 gives a criterion which ensures that J admits precomputation as given in Definition 4.5.

Recall that for a multi-index $\mathbb{i}_k = (i_\mathfrak{q})_{\mathfrak{q}\in S}$, we may divide the k-th numerator $g_{\mathbb{i}_k} = g'_{\mathbb{i}_k} G_{\mathbb{i}_k}$ into the J-part and the $(S\backslash J)$-part respectively,

$$g'_{\mathbb{i}_k} = \prod_{\mathfrak{q}\in J} g_{i_\mathfrak{q},\mathfrak{q}}, \qquad G_{\mathbb{i}_k} = \prod_{\mathfrak{q}\in S\backslash J} g_{i_\mathfrak{q},\mathfrak{q}}.$$

Let I be a fractional ideal, then the criterion so that J admits precomputation should be interpreted as

$$w_{J,I}\left(g_{\mathbb{i}_k}\right) = w_{S,I}\left(g_{\mathbb{i}_k}\right) \implies w_\mathfrak{p}\left(G_{\mathbb{i}_k}\right) = w_\mathfrak{q}\left(G_{\mathbb{i}_k}\right), \qquad \forall\ \mathfrak{p},\mathfrak{q}\in J. \tag{5.4}$$

This states that when the minimal prime ideal for a given muliti-index lies in the interval J, then the $\mathfrak{p}$-valuation of the polynomial defined by the part of that index not belonging to J must be the same for all $\mathfrak{p}\in J$. It is important to note that the valuation which must be the same is $w_\mathfrak{p}$ and not $w_{\mathfrak{p},I}$.

The reason that we require the $w_\mathfrak{p}$ valuation to be equal is so that when the minimal prime ideal is in J we have

$$w_{\mathfrak{p},I}\left(g_{\mathbb{i}_k}\right) - w_{\mathfrak{q},I}\left(g_{\mathbb{i}_k}\right) = w_{\mathfrak{p},I}\left(g'_{\mathbb{i}_k}\right) - w_{\mathfrak{q},I}\left(g'_{\mathbb{i}_k}\right), \qquad \forall\ \mathfrak{p},\mathfrak{q}\in J.$$

This is sufficient to ensure that the decision which the MaxMin algorithm takes when precomputing J is the same that would have been taken in the analogous iteration computing MaxMin for S. By utilising the expansion in (5.3), we may show that

$$\begin{aligned} w_{\mathfrak{p},I}\left(g_{\mathbb{i}_k}\right) - w_{\mathfrak{q},I}\left(g_{\mathbb{i}_k}\right) &= w_{J,\mathfrak{p}}\left(g'_{\mathbb{i}_k} G_{\mathbb{i}_k}\right) - w_{J,\mathfrak{q}}\left(g'_{\mathbb{i}_k} G_{\mathbb{i}_k}\right) \\ &= \left(w_{\mathfrak{p},I}\left(g'_{\mathbb{i}_k}\right) + w_\mathfrak{p}\left(G_{\mathbb{i}_k}\right)\right) - \left(w_{\mathfrak{q},I}\left(g'_{\mathbb{i}_k}\right) + w_\mathfrak{q}\left(G_{\mathbb{i}_k}\right)\right) \\ &= w_{\mathfrak{p},I}\left(g'_{\mathbb{i}_k}\right) - w_{\mathfrak{q},I}\left(g'_{\mathbb{i}_k}\right), \end{aligned}$$

are equal in the case where the fractional ideal precomputation criterion (5.4) holds.

Series of ν-values

In proving Theorem 4.12 we consider the division of a set of prime ideals into two intervals U and D such that $S = U \cup D$. Given the numerators of a basis for each of them, we represent the principal part of the w_U-valuation of the numerators of the U basis by a series of rational numbers ν_i with $0 \leqslant i \leqslant n_U$. There is an equivalent series for the subset D, which are ν'_j with $0 \leqslant j \leqslant n_D$.

For the purpose of the proofs of Proposition 4.15 and Lemma 4.19 we agree that $\nu_{-1}, \nu'_{-1} = -1$.

In fractional ideal case, we must instead specify that

$$\nu_{-1}, \nu'_{-1} = \min\left\{\nu_0, \nu'_0\right\} - 1.$$

This is compatible with the maximal order case, where $\nu_0 = \nu'_0 = 0$.

5.3 Basis element reduction modulo an $\mathfrak{m}$-power

Just as in the case of bases of the maximal order, it is possible to reduce the elements of a v-integral basis of I modulo a power of $\mathfrak{m}$. However, due to the nature of the map $w_{\mathfrak{m},I}$, the exponent may be different.

Consider the numerators $g_0, \ldots, g_{n-1}$ of a v-integral basis of a fractional ideal I, generated by the MaxMin algorithm, and let $\nu_m = w_I(g_m)$. The v-integral basis of I is then

$$\mathcal{B} = \left(\pi^{-\lfloor\nu_0\rfloor} g_0(\theta), \ldots, \pi^{-\lfloor\nu_{n-1}\rfloor} g_{n-1}(\theta)\right)$$

Theorem 5.3. *Let $\pi^{-\lfloor\nu_0\rfloor} g_0(\theta), \ldots, \pi^{-\lfloor\nu_{n-1}\rfloor} g_{n-1}(\theta)$ be a triangular v-integral basis $\mathcal{B}$ of a fractional ideal I, where $\nu_i = w_I(g_i(\theta))$ for all i. Let $G_0, \ldots, G_{n-1} \in \mathcal{O}[x]$ be monic polynomials of degree $0, \ldots, n-1$, respectively, such that*

$$G_i \equiv g_i \pmod{\mathfrak{m}^{\lceil\zeta_i\rceil}}, \qquad 0 \leqslant i < n,$$

where $\zeta_i = \lfloor\nu_i\rfloor + \max\left\{0, \max\left\{a_\mathfrak{p}/e(\mathfrak{p}/\mathfrak{m})\right\}_{\mathfrak{p}\,|\,\mathfrak{m}}\right\}$.

Then, $\mathcal{B}' = \left(\pi^{-\lfloor\nu_0\rfloor} G_0(\theta), \ldots, \pi^{-\lfloor\nu_{n-1}\rfloor} G_{n-1}(\theta)\right)$ is also a triangular v-

integral basis of I.

If the basis $\mathcal{B}$ is v-reduced and $G_i \equiv g_i \pmod{\mathfrak{m}^{\lceil \zeta_i' \rceil}}$, where $\zeta_i' = \nu_i + \max\left\{0, \max\left\{a_\mathfrak{p}/e(\mathfrak{p}/\mathfrak{m})\right\}_{\mathfrak{p}\,|\,\mathfrak{m}}\right\}$ for all $0 \leqslant i < n$, then the resulting basis $\mathcal{B}'$ will also be v-reduced.

Proof. We follow the proof of Corollary 4.29, which proves the same claim for bases of the maximal order. By Theorem 1.23 it suffices to show that $\lfloor w_I\left(G_i(\theta)\right)\rfloor$ is maximal for all $0 \leqslant i < n$.

To ensure this, it is sufficient to show that $w_{\mathfrak{p},I}\left(G\right) \geqslant \lfloor \nu_i \rfloor$ for all $\mathfrak{p} \in \mathcal{P}$, and that is an easy consequence of

$$w_{\mathfrak{p},I}\left(g_i\right) \leqslant \nu_i \geqslant \lfloor \nu_i \rfloor, \qquad \lceil \zeta_i \rceil \geqslant \lfloor \nu_i \rfloor + \frac{a_\mathfrak{p}}{e(\mathfrak{p}/\mathfrak{m})}.$$

By Theorem 1.26, in the case that $\mathcal{B}$ is v-reduced we must show that $w_I\left(G_i(\theta)\right)$ is maximal for all i. It suffices to show that $w_{\mathfrak{p},I}\left(G_i(\theta)\right) \geqslant \nu_i$ and this is an immediate consequence of

$$w_{\mathfrak{p},I}\left(g_i\right) \geqslant \nu_i, \qquad \lceil \zeta_i \rceil \geqslant \nu_i + \frac{a_\mathfrak{p}}{e(\mathfrak{p}/\mathfrak{m})}.$$

□

5.4 Advantages of the application of MaxMin in function fields

Let $A = k[t]$, where k is a perfect field and t is an indeterminate. Let $K = k(t)$ be the field of fractions of A.

Let $f \in A[x]$ be a monic separable polynomial, so that $L = K[x]/(f)$ is a function field over k.

A *place* of K is a discrete valuation

$$v : K^* \twoheadrightarrow \mathbb{Z},$$

which is trivial on the elements of k (the "constants"). The set of all places of K may be identified with $\mathbb{P}(A) \cup \{\infty\}$. Every monic irreducible polynomial

$p(t) \in A$ determines a discrete valuation

$$v = \mathrm{ord}_p : A \longrightarrow \mathbb{Z},$$

which extends to a place of K. Also, there is a place at infinity, defined as $\mathrm{ord}_\infty(a/b) = \deg b - \deg a$, for any $a, b \in A$. Let $A_\infty := k[t^{-1}]_{(t^{-1})} \subseteq K$ be the valuation ring of ord_∞.

A *place* of L is a discrete valuation

$$v_P : L^* \longrightarrow \mathbb{Q},$$

which extends a place of K. Let $\mathbb{P}_L$ be the set of all places of L. We may split $\mathbb{P}_L = \mathbb{P}_0(L) \cup \mathbb{P}_\infty(L)$ into the disjoint union of two subsets containing the *finite* places and the *infinite* places, according to $v_{P|K} = \mathrm{ord}_p$ for some $p \in \mathbb{P}(A)$ or $v_{P|K} = \mathrm{ord}_\infty$, respectively.

A *divisor* D of L/k is a formal finite $\mathbb{Z}$-linear combination of places of L. It may be written in a unique way as $D = D_0 + D_\infty$, where D_0, D_∞ are divisors with support in finite places and infinite places respectively. Let $\mathrm{Div}(L/k)$ be the group of all divisors of L/k.

Denote the integral closures of A, A_∞ in L respectively as

$$\mathcal{O}_L := \mathrm{Cl}(A, L), \qquad \mathcal{O}_{L,\infty} := \mathrm{Cl}(A_\infty, L).$$

These two rings are Dedekind domains. There are natural bijections:

$$\mathbb{P}_0(L) \longrightarrow \mathrm{Max}(\mathcal{O}_L), \qquad \mathbb{P}_\infty(L) \longrightarrow \mathrm{Max}(\mathcal{O}_{L,\infty}),$$

which induce a group isomorphism:

$$\mathrm{Div}(L/k) \longrightarrow \mathcal{I}_{\mathcal{O}_L} \times \mathcal{I}_{\mathcal{O}_{L,\infty}},$$

between the group of divisors and the product of the groups of fractional ideals of $\mathcal{O}_L$ and $\mathcal{O}_{L,\infty}$.

A divisor $D_0 = \sum_{P \in \mathbb{P}_0(L)} n_P P$ with finite support determines the fractional ideal $\prod_{P \in \mathbb{P}_0(L)} \mathfrak{p}_P^{-n_P}$, where $\mathfrak{p}_P \in \mathrm{Max}(\mathcal{O}_L)$ is the non-zero prime ideal

attached to P. There is a similar identification of the divisors with support at infinity and $\mathcal{I}_{\mathcal{O}_{L,\infty}}$.

Given a divisor D, the Riemann-Roch space,

$$\mathcal{L}(D) := \{a \in L^* : \operatorname{div}(a) \geqslant -D\} \cup \{0\},$$

is a fundamental invariant of D. It is a finite-dimensional k-vector space whose computation is crucial for many arithmetic questions on the function field.

If D corresponds to the pair of fractional ideals (I, I_∞), then $\mathcal{L}(D) = I \cap I_\infty$. For certain arithmetic questions concerning the divisor D and the space $\mathcal{L}(D)$, a *reduction* procedure is introduced, in the following terms:

- Compute an A-basis of I considered as an A-lattice of L.
- Consider a certain length function w_{I_∞} on L determined by I_∞.
- Compute a *reduced* basis of I with respect to this length function.

An efficient implementation of the reduction algorithm requires the computation of an ord_∞-reduced basis of I_∞. In the language of [Bau14], this basis leads to an "orthonormal basis" of L with respect to the considered length function. Then we need to compute a transition matrix between the input A-basis of L and the orthonormal basis. The reduction algorithm consists of the application of a series of *reduction steps* to this matrix.

The implementation of this method uses the method of the quotients to compute this ord_∞-reduced basis of $\mathcal{O}_{L,\infty}$. This non-triangular basis could not be triangularised because the standard triangularisation routine destroys the ord_∞-reducedness property. The input A-basis of L was computed by applying the global method described in Section 1.6. Local triangular bases are computed and gathered into a global triangular matrix. Of course, the transition matrix between the two bases was not triangular, and this slowed the reduction algorithm.

The application of the MaxMin algorithm leads to a substantial practical improvement of this strategy.

1. MaxMin is applied to compute a triangular ord_∞-reduced basis of $\mathcal{O}_{L,\infty}$.

2. MaxMin is applied to compute triangular local bases which are gathered into a global triangular basis of I.

In this way, the transition matrix between the two bases is a triangular matrix and this accelerates the reduction routine.

6

Complexity analysis

"Sometimes six and six make a dozen, and sometimes they make a mess."

– Robert Jordan, *The Path of Daggers*

Let (K, v) be a discrete valued field with valuation ring $\mathcal{O}$, $\mathfrak{m}$ the maximal ideal, $\pi \in \mathfrak{m}$ a generator of $\mathfrak{m}$ and $\mathbb{F} = \mathcal{O}/\mathfrak{m}$ the residue class field.

Let $f \in \mathcal{O}[x]$ be a monic, irreducible and separable polynomial of degree n and fix a root $\theta \in \overline{K}$ in the algebraic closure of K. Let $L = K(\theta)$ be the finite separable extension of K defined by f, and let $\mathcal{O}_L$ be the integral closure of $\mathcal{O}$ in L. Let $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ be the set of non-zero prime ideals of $\mathcal{O}_L$.

In this chapter we will analyse the computational complexity of an algorithm which computes a v-integral basis of L. This process requires a num-

ber of steps, based on the algorithms presented in Chapter 2 and Chapter 4:

1. Use the Montes algorithm to produce an OM representation $\mathfrak{T}$ of f.
2. Run the MaxMin algorithm to generate a family of maximal indices $\mathbb{i}_0, \ldots, \mathbb{i}_{n-1}$.
3. Apply the Single Factor Lifting algorithm to get an adequate improvement of the Montes approximation of each prime factor of f.
4. Compute the Okutsu numerators $g_0, \ldots, g_{n-1}$ specified by the maximal indices.
5. Divide the Okutsu numerators by the appropriate power of π to create an integral basis.

In Chapter 2 we presented complexity estimates for steps (1) and (3), although in the case of (3) we need to define the precision which we require the SFL algorithm to reach, which is discussed in Section 6.3. Computationally, the division in step (5) is negligible, as the numerator and denominator will be stored separately and by construction, π does not divide the basis numerators.

In Section 6.1 and Section 6.2 we will provide an analysis of the complexity of the remaining two steps. In Section 6.4 the additional space requirements of the MaxMin algorithm will be detailed.

Notation. *All logarithms are base-2 unless otherwise stated.*

From now on, we denote $\delta := v(\mathrm{disc}(f))$.

If $\delta = 0$ then $\mathcal{O}_L = \mathcal{O}[\theta]$ and $1, \theta, \ldots, \theta^{n-1}$ is a v-integral basis of L. So, we can assume $\delta > 0$ for the purpose of our analysis.

The results of this chapter yield the following total estimation.

Theorem 6.1. *Suppose that* $\mathbb{F}$ *is a finite field with* q *elements. The total cost of the computation of a* v*-integral basis of* L *by the application of the Montes and the MaxMin algorithms is*

$$O\left(n^{2+\epsilon}\delta^{1+\epsilon} + n^{1+\epsilon}\delta\log(q) + n^{1+\epsilon}\delta^{2+\epsilon}\right),$$

operations in $\mathbb{F}$. *If we assume* q *small, this will give us an estimation of* $O\left(n^{2+\epsilon}\delta^{1+\epsilon} + n^{1+\epsilon}\delta^{2+\epsilon}\right)$ *bit operations.*

This theorem will be proved in Section 6.3.

6.1 Complexity analysis of the MaxMin algorithm

Compared to the other processes that are required, such as the Montes algorithm, or polynomial multiplication to create the elements of a basis, the computational cost of the MaxMin algorithm is small.

Proposition 6.2. *The computational cost of the* MaxMin *algorithm is*

$$O\left(n^2 \log(n\delta)^{1+\epsilon}\right),$$

bit operations

The computational complexity of the MaxMin algorithm itself can be divided into two parts, preprocessing and the MaxMin loop. We shall study them in Section 6.1.2 and Section 6.1.3, respectively. We shall see that both tasks require at most $O\left(n^2 \log(n\delta)^{1+\epsilon}\right)$ bit operations. This will confirm Proposition 6.2.

6.1.1 Upper bound on valuations

The *height* of a rational number $a/b \in \mathbb{Q}$, expressed as the quotient of two coprime integers a, b is defined as:

$$h(a/b) = \max\{|a|, |b|\}.$$

The MaxMin algorithm primarily works with valuations of basis elements. The aim of this section is to find upper bounds for the height of the rational numbers $w_{\mathfrak{p}}(g)$, for $\mathfrak{p} \in \mathcal{P}$ and $g \in \mathrm{Ok}(\mathcal{P})$.

These bounds will be easily deduced from bounds for the values $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}})$, for $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ and $0 \leqslant i \leqslant r_{\mathfrak{q}} + 1$.

Ideally, we would like to bound these numbers in terms of the parameters $n = \deg f$ and $\delta = v(\mathrm{disc}(f))$. Unfortunately, this is not always possible,

as the following example shows. For a prime number p, take $\mathcal{O} = \mathbb{Z}_{(p)}$ the valuation ring of the p-adic valuation $v = \mathrm{ord}_p$, and consider the following irreducible polynomial

$$f = x^3 + px + p^{N+1},$$

where N is a large positive integer.

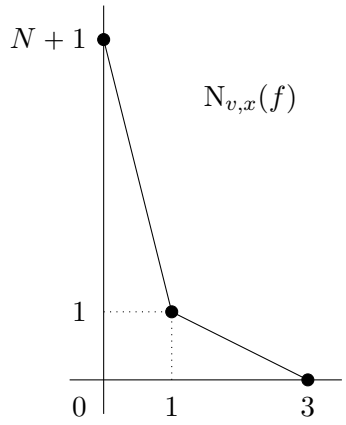


Figure 6.1: Newton polygon with "unbounded' valuation.

The Newton polygon $\mathrm{N}_{v,x}(f)$, shown in Figure 6.1, reveals that $\mathfrak{m} = \mathfrak{pq}$ splits as the product of two prime ideals $\mathfrak{p}$, $\mathfrak{q}$ admitting the following OM representations:

$$\mathbb{t}_\mathfrak{p} = (y; (\phi_\mathfrak{p}, \lambda_\mathfrak{p}, \psi_\mathfrak{p})), \qquad \mathbb{t}_\mathfrak{q} = (y; (x, 1/2, y+1); (\phi_\mathfrak{q}, \lambda_\mathfrak{q}, \psi_\mathfrak{q})).$$

We have $r_\mathfrak{p} = 0$, $n_\mathfrak{p} = 1$, and $r_\mathfrak{q} = 1$, $n_\mathfrak{q} = 2$. The valuation $w_\mathfrak{p}(\phi_{1,\mathfrak{p}}) = w_\mathfrak{p}(x) = N$ may be arbitrarily large, while $n = \delta = 3$.

Nevertheless, this value $w_\mathfrak{p}(\phi_{1,\mathfrak{q}}) = N$ is irrelevant for the execution of the MaxMin algorithm. The only information we need about this value is $w_\mathfrak{p}(\phi_{1,\mathfrak{q}}) > 1/2$. In fact, the numerators of the respective extended Okutsu bases are:

$$\mathcal{N}_\mathfrak{p} = \{1, \phi_\mathfrak{p}\}, \qquad \mathcal{N}_\mathfrak{q} = \{1, x, \phi_\mathfrak{q}\}.$$

Thus, if we consider the ordered set $S = \{\mathfrak{p}, \mathfrak{q}\}$, MaxMin$[S]$ runs as:

i	g_i	$\vec{w}(g_i)$	$w(g_i)$
0	$1 \cdot 1$	$(\underline{0}, 0)$	0
1	$\phi_{\mathfrak{p}} \cdot 1$	$(\infty, \underline{1/2})$	1/2
2	$\phi_{\mathfrak{p}} \cdot x$	$(\infty, \underline{1})$	1

While for the reverse ordered set $S' = \{\mathfrak{q}, \mathfrak{p}\}$, MaxMin$[S']$ runs as:

i	g_i	$\vec{w}(g_i)$	$w(g_i)$
0	$1 \cdot 1$	$(\underline{0}, 0)$	0
1	$x \cdot 1$	$(\underline{1/2}, N)$	1/2
2	$\phi_{\mathfrak{q}} \cdot 1$	$(\infty, \underline{1})$	1

Note that in both cases, if we had worked with the symbolic value $w_{\mathfrak{p}}(x) = \infty$, the output of MaxMin would have been the same.

This example illustrates the strategy we are going to follow in order to avoid the computation of the "unbounded" values $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}})$:

1. Detect under what exact conditions on $\mathfrak{p}$, $\mathfrak{q}$ and i the value $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}})$ cannot be bounded in terms of n and δ.

2. Show that if these "bad" values are taken symbolically to be $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) = \infty$, this does not affect the execution of MaxMin.

3. Find explicit bounds for the "boundable" values.

Notation. *For any two prime ideals $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ denote:*

$$\delta_{\mathfrak{p}} := v(\operatorname{disc}(F_{\mathfrak{p}})), \qquad \delta_{\mathfrak{p},\mathfrak{q}} := v(\operatorname{Res}(F_{\mathfrak{p}}, F_{\mathfrak{q}})).$$

The well-known formula:

$$\delta = \sum_{\mathfrak{p} \in \mathcal{P}} \delta_{\mathfrak{p}} + \sum_{\mathfrak{p},\mathfrak{q} \in \mathcal{P}} \delta_{\mathfrak{p},\mathfrak{q}}$$

shows that $\delta_{\mathfrak{p}}, \delta_{\mathfrak{p},\mathfrak{q}} \leqslant \delta$ for all $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$.

Definition 6.3. *Let $\mathfrak{p} \neq \mathfrak{q}$ be two different prime ideals and consider an index $0 \leqslant \ell < r_{\mathfrak{q}} + 1$. We say that $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}})$ is* virtually unbounded *if the following four conditions are satisfied:*

1. $\ell = i(\mathfrak{p}, \mathfrak{q}) = r_{\mathfrak{p}} + 1$.

2. *If* $\mathfrak{t}$ *is the last node of the non-optimised tree satisfying* $\mathfrak{t} \mid F_{\mathfrak{p}}$, $\mathfrak{t} \mid F_{\mathfrak{q}}$, *and* $\mathfrak{t}'$ *is the branch node dividing* $F_{\mathfrak{p}}$, *we have* $S_{\mathfrak{t}'} = \{\mathfrak{p}\}$.

3. $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{q})$.

4. $\lambda_{\mathfrak{p}}^{\mathfrak{q}} > \lambda_{\mathfrak{l}}^{\mathfrak{p}}$ *for all* $\mathfrak{l} \in S_{\mathfrak{t}}$, $\mathfrak{l} \neq \mathfrak{p}$.

Recall that for two prime ideals, $I(\mathfrak{p}, \mathfrak{q})$ is the extended index of coincidence as given in Definition 2.26.

Lemma 6.4. *Let* $\mathfrak{q} \in \mathcal{P}$ *be a prime ideal and let* $0 \leqslant \ell < r_{\mathfrak{q}} + 1$. *Let* $\mathfrak{p}, \mathfrak{p}'$ *be two prime ideals such that*

1. $\ell = i(\mathfrak{p}, \mathfrak{q}) = r_{\mathfrak{p}} + 1 = i(\mathfrak{p}', \mathfrak{q}) = r_{\mathfrak{p}'} + 1$.

2. $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{q}) = \phi(\mathfrak{p}', \mathfrak{q})$.

Then, $\min\left\{w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right), w_{\mathfrak{p}'}\left(\phi_{\ell,\mathfrak{q}}\right)\right\} \leqslant \delta_{\mathfrak{p},\mathfrak{p}'}/n_{\mathfrak{p}}$.

Proof. Conditions (1) and (2) imply that $I(\mathfrak{p}, \mathfrak{q}) = I(\mathfrak{p}', \mathfrak{q})$. Assume first that $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{p}')$. Then, $I(\mathfrak{p}, \mathfrak{q}) = I(\mathfrak{p}, \mathfrak{p}')$ and the relative position of the three primes in the non-optimised tree is as shown in Figure 6.2.

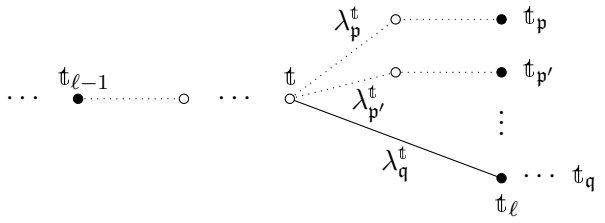


Figure 6.2: Non-optimised tree with potentially unbounded prime ideals, $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p}, \mathfrak{p}')$.

As shown, let $\mathfrak{t}$ be the last type in the non-optimised tree such that $\mathfrak{t} \mid F_{\mathfrak{p}}$, $\mathfrak{t} \mid F_{\mathfrak{p}'}$, and $\mathfrak{t} \mid F_{\mathfrak{q}}$, then $\lambda_{\mathfrak{p}}^{\mathfrak{t}}$, $\lambda_{\mathfrak{p}'}^{\mathfrak{t}}$, and $\lambda_{\mathfrak{q}}^{\mathfrak{t}}$ are the slopes corresponding to the branches of $\mathfrak{t}$ that lead to $\mathfrak{p}$, $\mathfrak{p}'$, and $\mathfrak{q}$ respectively.

If λ is the sum of all slopes corresponding to the edges between $\mathbb{t}_{\ell-1}$ and $\mathbb{t}$, then we have the following cutting slopes (from the optimised tree),

$$\begin{aligned}
\lambda_{\mathfrak{p}} &:= \lambda_{\mathfrak{p}}^{\mathfrak{p}'} = \lambda_{\mathfrak{p}}^{\mathfrak{q}} = \lambda_{\mathfrak{p}}^{\mathbb{t}} + \lambda, \\
\lambda_{\mathfrak{p}'} &:= \lambda_{\mathfrak{p}'}^{\mathfrak{p}} = \lambda_{\mathfrak{p}'}^{\mathfrak{q}} = \lambda_{\mathfrak{p}'}^{\mathbb{t}} + \lambda, \\
\lambda_{\mathfrak{q}} &:= \lambda_{\mathfrak{q}}^{\mathfrak{p}} = \lambda_{\mathfrak{q}}^{\mathfrak{p}'} = \lambda_{\mathfrak{q}}^{\mathbb{t}} + \lambda = \lambda_{\ell,\mathfrak{q}}.
\end{aligned}$$

By Proposition 2.31, we have,

$$\begin{aligned}
w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right) = w_{\mathfrak{p}'}\left(\phi_{\mathfrak{p}}\right) &= \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}, \lambda_{\mathfrak{p}'}\right\}}{e_1 \cdots e_{\ell-1}}, \\
w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right) &= \frac{V_\ell + \lambda_{\mathfrak{p}}}{e_1 \cdots e_{\ell-1}}, \\
w_{\mathfrak{p}'}\left(\phi_{\ell,\mathfrak{q}}\right) &= \frac{V_\ell + \lambda_{\mathfrak{p}'}}{e_1 \cdots e_{\ell-1}},
\end{aligned}$$

so either $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right) = w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right)$ or $w_{\mathfrak{p}'}\left(\phi_{\ell,\mathfrak{q}}\right) = w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right)$. As we shall see in Lemma 6.7, $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right)$ is equal to $\delta_{\mathfrak{p},\mathfrak{p}'}/n_{\mathfrak{p}}$.

Assume now $\phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{p}, \mathfrak{p}')$ as presented in Figure 6.3.

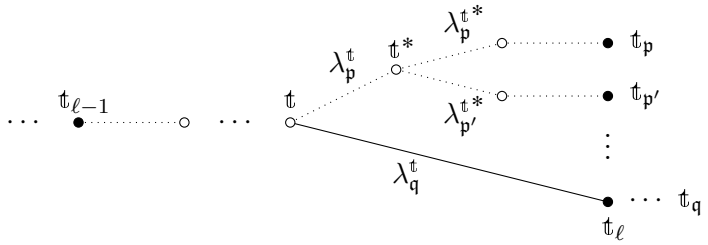


Figure 6.3: Non-optimised tree with potentially unbounded prime ideals, $\phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{p}, \mathfrak{p}')$.

In this case, Proposition 2.31 shows that

$$w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right) = \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}}, \qquad w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right) = \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}'}, \lambda_{\mathfrak{p}'}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}}.$$

Since $\lambda_{\mathfrak{p}}^{\mathfrak{q}} < \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{p}'}, \lambda_{\mathfrak{p}'}^{\mathfrak{p}}\right\} = \lambda_{\mathfrak{p}}^{\mathfrak{q}} + \min\left\{\lambda_{\mathfrak{p}}^{\mathbb{t}*}, \lambda_{\mathfrak{p}'}^{\mathbb{t}*}\right\}$, we have $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right) < w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right)$, and in Lemma 6.7 we will see that $w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}'}\right) \leqslant \delta_{\mathfrak{p},\mathfrak{p}'}/n_{\mathfrak{p}}$. □

The following lemma is a generalisation of Lemma 4.8, for types in an

optimised tree.

Lemma 6.5. *Let $S \subseteq \mathcal{P}$ be a set of prime ideals. For any type $\mathbb{t} \in \mathfrak{T}_S^{\mathrm{nop}}$ in the non-optimised tree, the interval $S_{\mathbb{t}} \subseteq S$ admits precomputation.*

Proof. Let $\mathbb{t}_0$ be the last type in the optimised tree which precedes $\mathbb{t}$. By Lemma 4.8, $S_{\mathbb{t}_0} \subseteq S$ admits precomputation, so we only need to show that $S_{\mathbb{t}} \subseteq S_{\mathbb{t}_0}$ admits precomputation with respect to the prime ideals in $S_{\mathbb{t}_0} \backslash S_{\mathbb{t}}$.

Take any $\mathfrak{p} \in S_{\mathbb{t}}$ and $\mathfrak{q} \in S_{\mathbb{t}_0} \backslash S_{\mathbb{t}}$. Let $\mathbb{t}^*$ be the last type in the non-optimised tree such that $\mathbb{t}^* \mid F_{\mathfrak{p}}$ and $\mathbb{t}^* \mid F_{\mathfrak{q}}$. We take λ to be the sum of all slopes corresponding to the edges linking $\mathbb{t}_0$ with $\mathbb{t}^*$ and denote by $\lambda_{\mathbb{t}}$ and $\lambda_{\mathfrak{q}}$ the slopes corresponding to the branches of $\mathbb{t}^*$ leading to $\mathbb{t}$ and $\mathbb{t}_{\mathfrak{q}}$ respectively. The relative positions of $\mathfrak{p}$ and $\mathfrak{q}$ are shown in Figure 6.4.

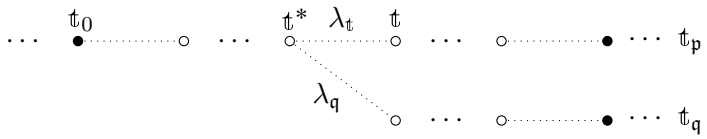


Figure 6.4: $S_{\mathbb{t}}$ is precomputable for $\mathbb{t}$ in the non-optimised tree.

Then the hidden slopes between $\mathfrak{p}$ and $\mathfrak{q}$ are:

$$\lambda_{\mathfrak{q}}^{\mathfrak{p}} = \lambda_{\mathfrak{q}} + \lambda, \qquad \lambda_{\mathfrak{p}}^{\mathfrak{q}} = \lambda_{\mathbb{t}} + \lambda.$$

Since both hidden slopes do not depend on the chosen $\mathfrak{p} \in S_{\mathbb{t}}$, Proposition 2.31 shows that the valuations $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}})$ for $1 \leqslant i \leqslant r_{\mathfrak{q}} + 1$ only depend on $\mathfrak{q}$ and i. Hence, the precomputation criterion of Lemma 4.7 is obviously satisfied. □

Lemma 6.6. *Suppose that all virtually unbounded values are given the value $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}) = \infty$. Accordingly, we take $w_{\mathfrak{p}}(g) = \infty$ for all $g \in \mathrm{Ok}(\mathcal{P})$ containing a factor $\phi_{i,\mathfrak{q}}$ with $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) = \infty$. Then, these conventions do not affect the output of the* MaxMin *algorithm.*

Proof. Let $\mathfrak{p} \in \mathcal{P}$ be a prime ideal admitting virtually unbounded values and let $\mathbb{t}$ be the last type in the non-optimised tree such that $\mathbb{t} \mid F_{\mathfrak{p}}$ and $\mathbb{t} \mid F_{\mathfrak{q}}$ for all $\mathfrak{q}$ such that $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}) = \infty$ as depicted in Figure 6.2. Note that there

may be different $\mathfrak{q}$ with $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}}) = \infty$, but the polynomial $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p},\mathfrak{q})$ is the same for all of them.

By Lemma 6.5, the interval $S_{\mathbb{t}}$ admits precomputation; thus in order to show that our convention on $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}})$ does not affect the output of MaxMin, it suffices to check this for MaxMin$[S_{\mathbb{t}}]$. Since MaxMin works block-wise (Section 4.4), we may suppose moreover that $S_{\mathbb{t}} = \mathcal{P}$, and the least index of coincidence in $\mathcal{P}$ is $\ell = 1$ and that $m_\ell = \deg \phi(\mathfrak{p},\mathfrak{q}) = 1$.

By item (2) of Definition 6.3, the branch of $\mathbb{t}$ which leads to $\mathfrak{p}$ contains no other prime ideals. As such, we may assume that $\mathfrak{p}$ is the first prime ideal in $\mathcal{P}$ and satisfy the ordering criterion given in (4.1). The maximality of the output of MaxMin under this ordering implies the maximality of the output under any other ordering satisfying (4.1).

We have $\mathcal{N}_{\mathfrak{p}} = \{1, \phi_{\mathfrak{p}}\}$ with $\deg \phi_{\mathfrak{p}} = 1$. Since, $\mathfrak{p}$ is the minimal element in $\mathcal{P} = S_{\mathbb{t}}$, MaxMin starts by choosing $g_{\mathbb{i}_0} = 1$, $g_{\mathbb{i}_1} = \phi_{\mathfrak{p}}$. From this point onward, we shall have $w_{\mathfrak{p}}(g_{\mathbb{i}_k}) = \infty$ for all $k \geqslant 1$, so that any assumption on the values $w_{\mathfrak{p}}(\phi_{\ell,\mathfrak{q}})$ is irrelevant. □

Lemma 6.7. *Let $\mathfrak{p} \neq \mathfrak{q}$ be two different prime ideals, and take $0 \leqslant i \leqslant r_{\mathfrak{q}}+1$.*

1. *For any $0 \leqslant m < r_{\mathfrak{p}} + 1$, we have $\hat{w}_{\mathfrak{p}}(\phi_{m,\mathfrak{p}}) \leqslant 2\delta_{\mathfrak{p}}/n_{\mathfrak{p}}^2$.*
2. *If $i = r_{\mathfrak{q}} + 1$ then $\hat{w}_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) = \delta_{\mathfrak{p},\mathfrak{q}}/(n_{\mathfrak{p}} n_{\mathfrak{q}})$.*
3. *If $i < r_{\mathfrak{p}} + 1$ then $\hat{w}_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) \leqslant 2\delta_{\mathfrak{p}}/n_{\mathfrak{p}}^2$.*
4. *If $r_{\mathfrak{p}} + 1 \leqslant i < r_{\mathfrak{q}} + 1$ and $w_{\mathfrak{p}}(\phi_{i,\mathfrak{q}})$ is not virtually unbounded, then either $\hat{w}_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) \leqslant 2\delta_{\mathfrak{l}}/(n_{\mathfrak{p}} n_{\mathfrak{l}})$ or $\hat{w}_{\mathfrak{p}}(\phi_{i,\mathfrak{q}}) \leqslant \delta_{\mathfrak{p},\mathfrak{l}}/(n_{\mathfrak{p}} n_{\mathfrak{l}})$, for some $\mathfrak{l} \in \mathcal{P}$.*

Proof. For item (1), if we denote $r = r_{\mathfrak{p}}$, $e_i = e_{i,\mathfrak{p}}$, $h_r = h_{r_{\mathfrak{p}},\mathfrak{p}}$, then Lemma 3.5 shows that

$$\hat{w}_{\mathfrak{p}}(\phi_{1,\mathfrak{p}}) < \hat{w}_{\mathfrak{p}}(\phi_{2,\mathfrak{p}}) < \cdots < \hat{w}_{\mathfrak{p}}(\phi_{r,\mathfrak{p}}).$$

Now, the common bound $\hat{w}_{\mathfrak{p}}(\phi_{r,\mathfrak{p}})$ can be expressed as:

$$\hat{w}_{\mathfrak{p}}(\phi_{r,\mathfrak{p}}) = \frac{1}{m_{r,\mathfrak{p}}} \cdot \frac{V_r + \lambda_{r,\mathfrak{p}}}{e_1 \cdots e_{r-1}} = \frac{e_r f_r}{e_r f_r m_{r,\mathfrak{p}}} \cdot \frac{e_r V_r + h_r}{e_1 \cdots e_r} = \frac{V_{r+1}}{n_{\mathfrak{p}} e(\mathfrak{p}/\mathfrak{m})} = \frac{\delta_0(F_{\mathfrak{p}})}{n_{\mathfrak{p}}},$$

where $\delta_0(F_\mathfrak{p})$ is the Okutsu bound introduced in Section 2.1. On the other hand, it was shown in [BNS13, Lem. 2.2] that $\delta_0(F_\mathfrak{p}) \leqslant 2\delta_\mathfrak{p}/n_\mathfrak{p}$. This ends the proof of the first item.

The second item follows directly from the equalities:

$$w_\mathfrak{p}(\phi_\mathfrak{q}) = w_\mathfrak{p}(F_\mathfrak{q}) = \delta_{\mathfrak{p},\mathfrak{q}}/n_\mathfrak{p},$$

because $\phi_{i,\mathfrak{q}} = \phi_\mathfrak{q}$ with $\deg(\phi_{i,\mathfrak{q}}) = n_\mathfrak{q}$. The first equality was proved in Corollary 2.32. The second equality follows from the well-known formula

$$\mathrm{Res}(F_\mathfrak{p}, F_\mathfrak{q}) = \prod\nolimits_{\theta_\mathfrak{p}} F_\mathfrak{q}(\theta_\mathfrak{p}),$$

for $\theta_\mathfrak{p}$ running on the roots of $F_\mathfrak{p}$ in $\overline{K}_v$. Since $F_\mathfrak{p}$ is irreducible, its roots are Galois conjugate and have the same v-value. Since $F_\mathfrak{q}$ has coefficients in $\mathcal{O}_v$, all elements $F_\mathfrak{q}(\theta_\mathfrak{p})$ have the same v-valuation too. Hence,

$$\delta_{\mathfrak{p},\mathfrak{q}} = n_\mathfrak{p} v\left(F_\mathfrak{q}(\theta_\mathfrak{p})\right) = n_\mathfrak{p} w_\mathfrak{p}\left(F_\mathfrak{q}(\theta)\right) = n_\mathfrak{p} w_\mathfrak{p}\left(F_\mathfrak{q}\right).$$

Let us prove the third item. Let $\ell = i(\mathfrak{p},\mathfrak{q})$. If $i < \ell$, then $\phi_{i,\mathfrak{q}} = \phi_{i,\mathfrak{p}}$. Hence, Lemma 3.5 and the first item show that

$$\hat{w}_\mathfrak{p}(\phi_{i,\mathfrak{q}}) = \hat{w}_\mathfrak{p}(\phi_{i,\mathfrak{p}}) \leqslant \hat{w}_\mathfrak{p}\left(\phi_{r_\mathfrak{p},\mathfrak{p}}\right) \leqslant 2\delta_\mathfrak{p}/n_\mathfrak{p}^2.$$

If $i \geqslant \ell$, the explicit formulas of Proposition 2.31 show that

$$\hat{w}_\mathfrak{p}(\phi_{i,\mathfrak{q}}) = \frac{1}{m_\ell}\begin{cases} \dfrac{V_\ell + \lambda_\mathfrak{p}^\mathfrak{q}}{e_1\cdots e_{\ell-1}}, & \text{if } i=\ell,\ \phi_{i,\mathfrak{q}} = \phi(\mathfrak{p},\mathfrak{q}) \text{ and } \lambda_\mathfrak{p}^\mathfrak{q} > \lambda_\mathfrak{q}^\mathfrak{p}, \\ \dfrac{V_\ell + \min\left\{\lambda_\mathfrak{p}^\mathfrak{q}, \lambda_\mathfrak{q}^\mathfrak{p}\right\}}{e_1\cdots e_{\ell-1}}, & \text{otherwise.} \end{cases}$$

Since $\lambda_\mathfrak{p}^\mathfrak{q} \leqslant \lambda_{\ell,\mathfrak{p}}$ and $\ell \leqslant i < r_\mathfrak{p} + 1$, in both cases we get the same bound as in the case $i < \ell$:

$$\hat{w}_\mathfrak{p}(\phi_{i,\mathfrak{q}}) \leqslant \hat{w}_\mathfrak{p}(\phi_{\ell,\mathfrak{p}}) \leqslant \hat{w}_\mathfrak{p}\left(\phi_{r_\mathfrak{p},\mathfrak{p}}\right) \leqslant 2\delta_\mathfrak{p}/n_\mathfrak{p}^2.$$

For the fourth item, let $\ell = i(\mathfrak{p},\mathfrak{q})$, so that $i \geqslant r_\mathfrak{p} + 1 \geqslant \ell$. If $i > \ell$, or if

$i = \ell$ and $\phi_{\ell,\mathfrak{q}} \neq \phi(\mathfrak{p},\mathfrak{q})$, Proposition 2.31 and item (1) yield:

$$\hat{w}_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right) = \frac{1}{m_\ell} \cdot \frac{V_\ell + \min\left\{\lambda_{\mathfrak{p}}^{\mathfrak{q}}, \lambda_{\mathfrak{q}}^{\mathfrak{p}}\right\}}{e_1 \cdots e_{\ell-1}} \leqslant \hat{w}_{\mathfrak{q}}\left(\phi_{\ell,\mathfrak{q}}(\theta)\right) \leqslant 2\delta_{\mathfrak{q}}/n_{\mathfrak{q}}^2.$$

Finally, suppose that $i = r_{\mathfrak{p}} + 1 = \ell$ and $\phi_{\ell,\mathfrak{q}} = \phi(\mathfrak{p},\mathfrak{q})$. Let $\mathfrak{t}$ be the last type in the non-optimised tree such that $\mathfrak{t} \mid F_{\mathfrak{p}}$ and $\mathfrak{t} \mid F_{\mathfrak{q}}$, and let $\mathfrak{t}'$ be the branch of $\mathfrak{t}$ dividing $F_{\mathfrak{p}}$. Since $w_{\mathfrak{p}}\left(\phi_{\ell,\mathfrak{q}}\right)$ is not virtually unbounded, either $S_{\mathfrak{t}'}$ contains some other $\mathfrak{l} \neq \mathfrak{p}$, or there exists $\mathfrak{l} \in S_{\mathfrak{t}}$, $\mathfrak{l} \neq \mathfrak{p}$ such that $\lambda_{\mathfrak{p}}^{\mathfrak{q}} \leqslant \lambda_{\mathfrak{l}}^{\mathfrak{q}}$. In both cases we have $\lambda_{\mathfrak{p}}^{\mathfrak{q}} \leqslant \lambda_{\ell,\mathfrak{l}}$.

Proposition 2.31 shows that

$$\hat{w}_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}\right) \leqslant \frac{1}{m_\ell} \cdot \frac{V_\ell + \lambda_{\mathfrak{p}}^{\mathfrak{q}}}{e_1 \cdots e_{\ell-1}} \leqslant \frac{1}{m_\ell} \cdot \frac{V_\ell + \lambda_{\ell,\mathfrak{l}}}{e_1 \cdots e_{\ell-1}} = \hat{w}_{\mathfrak{l}}\left(\phi_{\ell,\mathfrak{l}}\right).$$

If $i(\mathfrak{p},\mathfrak{l}) = \ell < r_{\mathfrak{l}} + 1$, then $n_{\mathfrak{p}} < n_{\mathfrak{l}}$ and

$$\hat{w}_{\mathfrak{l}}\left(\phi_{\ell,\mathfrak{l}}\right) \leqslant \hat{w}_{\mathfrak{l}}\left(\phi_{r_{\mathfrak{l}},\mathfrak{l}}\right) \leqslant 2\delta_{\mathfrak{l}}/n_{\mathfrak{l}}^2 \leqslant 2\delta_{\mathfrak{l}}/(n_{\mathfrak{p}} n_{\mathfrak{l}}).$$

by Lemma 3.5 and item (1).

If $\ell = r_{\mathfrak{l}}+1$, then $m_{i,\mathfrak{q}} = n_{\mathfrak{p}} = n_{\mathfrak{l}}$ and Lemma 6.4 shows that $w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right) \leqslant \delta_{\mathfrak{p},\mathfrak{l}}/n_{\mathfrak{p}}$, so that $\hat{w}_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right) \leqslant \delta_{\mathfrak{p},\mathfrak{l}}/(n_{\mathfrak{p}} n_{\mathfrak{l}})$. □

Corollary 6.8. *Let $g = \prod_{\mathfrak{p}} g_{i_{\mathfrak{p}},\mathfrak{p}}$ be any element in $\mathrm{Ok}(\mathcal{P})$ with $w_{\mathfrak{p}}(g) < \infty$. Then, for all $\mathfrak{p} \in \mathcal{P}$ we have $h(w_{\mathfrak{p}}(g)) \leqslant 2n\delta$.*

Proof. Since, $\delta_{\mathfrak{p}}, \delta_{\mathfrak{p},\mathfrak{q}} \leqslant \delta$ for all $\mathfrak{p},\mathfrak{q} \in \mathcal{P}$, Lemma 6.7 shows that $\hat{w}_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}\right) \leqslant 2\delta/n_{\mathfrak{p}}$, for each factor $\phi_{i,\mathfrak{q}}$ of g. As a consequence,

$$\hat{w}_{\mathfrak{p}}(g) = \left(\sum_{i,\mathfrak{q}} w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}\right)\right) \Big/ \left(\sum_{i,\mathfrak{q}} m_{i,\mathfrak{q}}\right) \leqslant 2\delta/n_{\mathfrak{p}},$$

where the sum runs on all factors $\phi_{i,\mathfrak{q}}$ of g, with due count of multiplicities.

Therefore, $w_{\mathfrak{p}}(g) \leqslant \deg(g) 2\delta/n_{\mathfrak{p}} \leqslant 2n\delta/n_{\mathfrak{p}}$. Since the denominator of $w_{\mathfrak{p}}(g)$ is a divisor of $e(\mathfrak{p}/\mathfrak{m})$, the numerator is bounded by $2ne(\mathfrak{p}/\mathfrak{m})\delta/n_{\mathfrak{p}} \leqslant 2n\delta$. □

6.1.2 Preprocessing for MaxMin$[S]$

It is sufficient to discus the case $S = \mathcal{P}$.

The MaxMin algorithm requires the $w_{\mathfrak{p}}$-valuation of $\phi_{i,\mathfrak{q}}$ for all $1 \leqslant i \leqslant r_{\mathfrak{q}} + 1$ for all $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$. By Proposition 2.31, we can calculate these values via explicit formulas of MacLane invariants, reducing this operation to a small number of calculations in $\mathbb{Q}$ for each valuation of each ϕ-polynomial.

In order to further reduce the complexity of the MaxMin algorithm, we will store the $w_{\mathfrak{p}}$-valuations of each Okutsu numerator $g_{m,\mathfrak{q}}$ for all $0 \leqslant m \leqslant n_{\mathfrak{q}} + 1$ for all $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$. As these numerators are products of ϕ-polynomials, the valuations are sums of the precomputed valuations.

By Corollary 6.8, any sum of two of these valuations has a cost of $O\left(\log(n\delta)^{1+\epsilon}\right)$ word operations.

Lemma 6.9. *Let $\mathbb{V}_0, \ldots, \mathbb{V}_n$ be the vectors of valuations:*

$$\mathbb{V}_k[i] := w_{\mathfrak{p}_i}\left(g_k(\theta)\right),$$

where $g_0, \ldots, g_n$ are the output numerators of MaxMin. *The cost of computing the vectors $\mathbb{V}_0, \ldots, \mathbb{V}_n$ is $\mathcal{C}_{\text{pre}} = O(n^2 \log(n\delta)^{1+\epsilon})$ word operations.*

Proof. Let us first evaluate the cost of the computing valuations of all ϕ-polynomials, excluding the trivial case $w_{\mathfrak{p}}(\phi_{\mathfrak{p}}) = \infty$. For each prime ideal $\mathfrak{q} \in S$, there are $r_{\mathfrak{q}} + 1$ ϕ-polynomials of degree,

$$m_1 \mid m_2 \mid \cdots \mid m_{r_{\mathfrak{q}}} \mid m_{r_{\mathfrak{q}}+1} = n_{\mathfrak{q}},$$

with $m_i < m_{i+1}$ for $1 \leqslant i \leqslant r_{\mathfrak{q}}$. Since $m_{i+1} \geqslant 2m_i$ for all $1 \leqslant i \leqslant r_{\mathfrak{q}}$, we have $r_{\mathfrak{q}} \leqslant \log(n_{\mathfrak{q}})$.

By Proposition 2.31, for any $\mathfrak{p}, \mathfrak{q} \in \mathcal{P}$ we have

$$w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right) = \begin{cases} \dfrac{V+\lambda}{e}, & \text{if } \mathfrak{p} = \mathfrak{q} \text{ or } i < i(\mathfrak{p},\mathfrak{q}), \\ \dfrac{m_i}{m_\ell}\dfrac{V+\lambda}{e}, & \text{if } \mathfrak{p} \neq \mathfrak{q} \text{ and } i \geqslant \ell = i(\mathfrak{p},\mathfrak{q}). \end{cases}$$

All ingredients of these formulas have been computed and stored during the execution of the Montes algorithm. Hence, the computation of $w_{\mathfrak{p}}\left(\phi_{i,\mathfrak{q}}(\theta)\right)$

requires $O\left(\log(n\delta)^{1+\epsilon}\right)$ operations, because it needs a few sums and multiplications of integers of size $O(n\delta)$, as indicated in Corollary 6.8.

Since $r_{\mathfrak{p}} \leqslant \log(n_{\mathfrak{p}})$, for the computation of all $\mathfrak{p}$-valuations of each ϕ-polynomial of each Okutsu $\mathfrak{q}$-frame, we must calculate,

$$O\left((\log(n_{\mathfrak{p}_1}) + \cdots + \log(n_{\mathfrak{p}_s})) \cdot s\right) = O\left(\log(n_{\mathfrak{p}_1} \cdots n_{\mathfrak{p}_s}) \cdot s\right),$$

valuations. It is easy to see that this function maximises when we set $s = n/2$, with $n_{\mathfrak{p}} = 2$ for all $\mathfrak{p} \in \mathcal{P}$. This gives a computational complexity of,

$$\log(2^s) \cdot s = s^2 = O\left(n^2\right),$$

valuations, with a cost of $\mathcal{C}_{\text{pre}_1} = O\left(n^2 \log(n\delta)^{1+\epsilon}\right)$ bit operations.

Let $\nu_{k,i,j} := w_{\mathfrak{p}_i}\left(g_{k,\mathfrak{p}_j}\right)$ for $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ and $1 \leqslant i, j \leqslant s$, $0 \leqslant k \leqslant n_{\mathfrak{p}_j}$ be the valuations of the Okutsu numerators which we wish to store.

There are $s(n+s)$ valuations in total, and each valuation can be constructed as the sum of a prior valuation and the valuation of a ϕ polynomial for the same pair of primes $\mathfrak{p}_i, \mathfrak{p}_j$. As such, we have a computational complexity of,

$$\begin{aligned}\mathcal{C}_{\text{pre}_2} &= O\left(s(n+s)\log(n\delta)^{1+\epsilon}\right) \\ &= O\left(n^2 \log(n\delta)^{1+\epsilon}\right).\end{aligned}$$

This gives a final computational complexity for precomputation of,

$$\mathcal{C}_{\text{pre}} = \mathcal{C}_{\text{pre}_1} + \mathcal{C}_{\text{pre}_2} = O\left(n^2 \log(n\delta)^{1+\epsilon}\right).$$

bit operations. □

6.1.3 MaxMin[S] main loop

Again, we discuss only the case $S = \mathcal{P}$. In the version of MaxMin[$\mathcal{P}$] given in Algorithm 4.1, in each iteration we compute valuations of the output polynomial from the previous iteration. Since this output polynomial is the

product of Okutsu $\mathfrak{p}$-numerators for each $\mathfrak{p} \in \mathcal{P}$, we will make use of the preprocessed valuations discussed in Section 6.1.2.

Lemma 6.10. *The cost of executing the main loop of* MaxMin$[\mathcal{P}]$ *is* $\mathcal{C}_{\text{loop}} = O\left(n^2 \log(n\delta)^{1+\epsilon}\right)$ *word operations.*

Proof. The numbers $\nu_{k,i,j} := w_{\mathfrak{p}_i}\left(g_{k,\mathfrak{p}_j}\right)$ for $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ have already been computed and stored. Now, consider the modifications presented in Algorithm 6.1.

Algorithm 6.1 MaxMin$[\mathcal{P}]$ algorithm using preprocessed valuations

Input: Okutsu numerators $\{g_{i,\mathfrak{q}} : 0 \leqslant i \leqslant n_{\mathfrak{p}}\}$ of $\mathcal{O}$-bases of $\mathcal{O}_{L_{\mathfrak{q}}}$ for each $\mathfrak{q} \in \mathcal{P}$.

Output: A family $\mathbb{i}_0, \mathbb{i}_1, \ldots, \mathbb{i}_n$ of multi-indices of degree $0, 1, \ldots, n$ respectively.

1: $\mathbb{i}_0 \leftarrow (0, \ldots, 0)$
2: $\mathbb{V}_0 \leftarrow (0, \ldots, 0)$
3: **for** $k = 0 \to (n-1)$ **do**
4: $\quad j \leftarrow \min\{1 \leqslant i \leqslant s : \mathbb{V}_k[i] = \min(\mathbb{V}_k)\}$
5: $\quad \mathbb{i}_{k+1} \leftarrow \mathbb{i}_k + \mathbb{u}_j$
6: $\quad \mathbb{V}_{k+1} \leftarrow \mathbb{V}_k + \left(\nu_{\mathbb{i}_{k+1}[j],i,j} - \nu_{\mathbb{i}_k[j],i,j}\right)_{1 \leqslant i \leqslant s}$
7: **end for**

During each of the n iterations, there are two points where a variable number of operations are performed.

On Line 4, the algorithm performs $(s-1)$ comparisons of rational numbers. Then, on Line 6, $\mathbb{V}_{k+1}$ is calculated from $\mathbb{V}_k$, which requires s addition and subtraction operations. By Corollary 6.8 any of these operations has a cost of $O\left(\log(n\delta)^{1+\epsilon}\right)$ bit operations.

This gives a computational cost,

$$\begin{aligned}\mathcal{C}_{\text{loop}} &= O\left(n(s-1+2s)\log(n\delta)^{1+\epsilon}\right) = O\left(ns\log(n\delta)^{1+\epsilon}\right) \\ &= O\left(n^2\log(n\delta)^{1+\epsilon}\right),\end{aligned}$$

word operations. □

6.2 Complexity analysis of basis numerator computation

In this section, we will analyse the computational complexity of the computation of the numerators of an integral basis, $g_0, \ldots, g_n$, for the set $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ of prime ideals of $\mathcal{O}_L$.

We keep the notation $\delta := v(\mathrm{disc}(f))$. By [BNS13, Thm. 3.14], for the computation of a v-integral basis of $\mathcal{O}_L$ we may work modulo $\mathfrak{m}^{\delta+1}$. Hence, we assume that the elements of $\mathcal{O}$ are finite π-adic developments of length $\delta + 1$.

Definition 6.11. *An operation in $\mathcal{O}$ is called $\mathfrak{m}$-small if it involves two elements belonging to a fixed system of representatives of $\mathbb{F} = \mathcal{O}/\mathfrak{m}$.*

Each multiplication in $\mathcal{O}$ costs $O\left(\delta^{1+\epsilon}\right)$ $\mathfrak{m}$-small operations, if we assume fast multiplication, using the techniques of Schönhage-Strassen [SS71]. It is natural to assume that the ring $\mathcal{O}$ is given in a sufficiently good computational representation, so that the cost of an $\mathfrak{m}$-small operation coincides with the cost of one operation in $\mathbb{F}$. For instance, if $\mathbb{F}$ is a finite field and $q := \#\mathbb{F}$, then an $\mathfrak{m}$-small operation in $\mathcal{O}$ requires $O\left(\log(q)^{1+\epsilon}\right)$ word operations.

Lemma 6.12. *The complexity of computing the numerators for a basis from the output of the* MaxMin *algorithm is $\mathcal{C}_{\mathrm{num}} = O\left(n^{2+\epsilon}\delta^{1+\epsilon}\right)$ operations in $\mathbb{F}$.*

Proof. The construction of the basis numerators requires all the numerators of the Okutsu bases $\mathcal{B}_{\mathfrak{p}}$ for all $\mathfrak{p} \in \mathcal{P}$. All the degree 0 numerators are 1, so there are n non-trivial numerators to be computed as a product of ϕ-polynomials. Each numerator of an Okutsu basis can be constructed as the product of a prior numerator and a ϕ-polynomial.

Therefore, we will require $n_{\mathfrak{p}}$ multiplications of polynomials of degree bounded by $n_{\mathfrak{p}}$ to construct all the numerators of the Okutsu basis $\mathcal{B}_{\mathfrak{p}}$, which amounts to $n_{\mathfrak{p}}^{2+\epsilon}$ operations in $\mathcal{O}$ if we assume fast multiplication. The total cost is

$$O\left(n_{\mathfrak{p}_1}^{2+\epsilon} + \cdots + n_{\mathfrak{p}_s}^{2+\epsilon}\right) = O\left(n^{2+\epsilon}\right),$$

operations in $\mathcal{O}$, or equivalently, $O\left(n^{2+\epsilon}\delta^{1+\epsilon}\right)$ operations in $\mathbb{F}$.

Now that the numerators for each Okutsu basis have been computed, we may continue to construct the numerators of a v-integral basis. Let g_i be the i-th element of the basis numerator, the next numerator can be constructed as

$$g_{i+1} = g_i \cdot (g_{i_\mathfrak{p},\mathfrak{p}})^{-1} \cdot g_{i_\mathfrak{p}+1,\mathfrak{p}},$$

for the prime ideal $\mathfrak{p}$ determined by the MaxMin algorithm.

At each step in the construction of the basis numerators a single polynomial division and a single polynomial multiplication are required. The complexity of each of these operations is $O\left(n^{1+\epsilon}\right)$ operations in $\mathcal{O}$ in both cases if we assume fast multiplication. This gives a complexity for computing all n numerators in the basis of

$$\begin{aligned}\mathcal{C}_{\text{num}} &= O\left(n^{2+\epsilon}\delta^{1+\epsilon} + n^{2+\epsilon}\delta^{1+\epsilon}\right)\\ &= O\left(n^{2+\epsilon}\delta^{1+\epsilon}\right),\end{aligned}$$

operations in $\mathbb{F}$. □

6.3 Complexity of computing a v-integral basis

Before providing the proof of Theorem 6.1, we must specify the precision which we require of the Okutsu approximations $\phi_\mathfrak{p}$ for each prime ideal $\mathfrak{p} \in \mathcal{P}$. Recall that the precision is $v_0\left(\phi_\mathfrak{p} - F_\mathfrak{p}\right)$, or equivalently the minimum of the v-values of the coefficients of $\phi_\mathfrak{p} - F_\mathfrak{p}$.

Lemma 6.13. *Let $\phi_\mathfrak{p}$ be an Okutsu approximation to $F_\mathfrak{p}$, a prime factor of f for all $\mathfrak{p} \in \mathcal{P}$. Then, for the construction of the numerators of a v-integral basis chosen by the* MaxMin *algorithm, $\phi_\mathfrak{p}$ requires, at most, a precision of $\delta = v\left(\operatorname{disc}(f)\right)$.*

Proof. As was explained in Section 4.7, we require the valuation $w_\mathfrak{p}\left(\phi_\mathfrak{p}(\theta)\right)$ to be high enough that

$$w\left(g_k\right) = w\left(g_k(\theta)\right), \qquad 0 \leqslant k < n.$$

Since $w\left(g_k\right)$ grows with k, it suffices to achieve

$$w_{\mathfrak{p}}\left(\phi_{\mathfrak{p}}(\theta)\right) \geqslant w\left(g_{n-1}\right).$$

Now, the numerators $g_0(\theta) = 1, g_1(\theta), \ldots, g_{n-1}(\theta)$ are an $\mathcal{O}$-basis of $\mathcal{O}[\theta]$. Hence,

$$\sum_{i=0}^{n-1} \lfloor w\left(g_i\right) \rfloor = \operatorname{ind}(f) := v\left([\mathcal{O}_L : \mathcal{O}[\theta]]\right).$$

On the other hand, in this separable context, there is a well known relationship linking the index with the discriminant:

$$\delta = 2\operatorname{ind}(f) + v\left(\operatorname{disc}(L/K)\right) \geqslant 2\operatorname{ind}(f) + \sum_{\mathfrak{p}\in\mathcal{P}}(e_{\mathfrak{p}} - 1).$$

Let $\mathfrak{q} \in \mathcal{P}$ be a prime for which $w\left(g_{n-1}\right) = w_{\mathfrak{q}}\left(g_{n-1}(\theta)\right)$. Then,

$$\frac{\delta}{2} \geqslant \operatorname{ind}(f) + \frac{e_{\mathfrak{q}} - 1}{e_{\mathfrak{q}}} \geqslant \lfloor w\left(g_{n-1}\right) \rfloor + \frac{e_{\mathfrak{q}} - 1}{e_{\mathfrak{q}}} \geqslant w\left(g_{n-1}\right).$$

Therefore, if we achieve a precision

$$v_0\left(\phi_{\mathfrak{p}} - F_{\mathfrak{p}}\right) \geqslant \frac{\delta}{2} \geqslant w\left(g_{n-1}\right),$$

we get $w_{\mathfrak{p}}\left(\phi_{\mathfrak{q}}(\theta)\right) \geqslant v_0\left(\phi_{\mathfrak{p}} - F_{\mathfrak{p}}\right) \geqslant w\left(g_{n-1}\right)$, as desired. □

We may now prove the main theorem of this chapter.

If $\mathbb{F}$ is a finite field with q elements, then Theorem 2.30 gives the complete complexity for an OM factorisation of f with precision ν as,

$$O\left(n^{2+\epsilon} + n^{1+\epsilon}(1+\delta)\log q + n^{1+\epsilon}\delta^{2+\epsilon} + n^2\nu^{1+\epsilon}\right),$$

operations in $\mathbb{F}$. This includes the complexity for both the Montes algorithm and the Single Factor Lifting algorithm for each prime ideal $\mathfrak{p} \in \mathcal{P}$.

Substituting in our precision bound δ for ν, this a complexity of

$$\mathcal{C}_{\text{Montes}} + \mathcal{C}_{\text{SFL}} = O\left(n^{2+\epsilon} + n^{1+\epsilon}(1+\delta)\log q + n^{1+\epsilon}\delta^{2+\epsilon} + n^2\delta^{1+\epsilon}\right), \quad (6.1)$$

operations in $\mathbb{F}$.

Proof of Theorem 6.1. As stated at the beginning of this chapter, there are four non-negligible steps involved in computing a v-integral basis using the MaxMin algorithm.

By Proposition 6.2 and Lemma 6.12 the cost of $\mathcal{C}_{\text{MaxMin}}$ is dominated by that of $\mathcal{C}_{\text{num}}$. Using the estimation (6.1), we can now specify the total complexity for computing a v-integral basis,

$$\begin{aligned}\mathcal{C}_{\text{basis}} &= \mathcal{C}_{\text{Montes}} + \mathcal{C}_{\text{SFL}} + \mathcal{C}_{\text{MaxMin}} + \mathcal{C}_{\text{num}} \\ &= O\left(n^{2+\epsilon} + n^{1+\epsilon}(1+\delta)\log q + n^{1+\epsilon}\delta^{2+\epsilon} + n^2\delta^{1+\epsilon}\right) + O\left(n^{2+\epsilon}\delta^{1+\epsilon}\right) \\ &= O\left(n^{2+\epsilon}\delta^{1+\epsilon} + n^{1+\epsilon}\delta\log(q) + n^{1+\epsilon}\delta^{2+\epsilon}\right),\end{aligned}$$

operations in $\mathbb{F}$. Clearly, if q is small, this gives $O\left(n^{2+\epsilon}\delta^{1+\epsilon} + n^{1+\epsilon}\delta^{2+\epsilon}\right)$. □

6.4 Space complexity analysis

To complete the complexity analysis of the MaxMin algorithm, we will consider the additional space that is required during its execution.

In order to run, the MaxMin algorithm requires the $\mathfrak{q}$-valuations for all numerators of the Okutsu bases $\mathcal{B}_{\mathfrak{p}}$. It produces a series of multi-indices, which can be used, along with the numerators of each of the Okutsu $\mathfrak{p}$-bases, to compute the numerators of the final basis.

As such, there are three sets of data which must be stored in memory during the process of running the MaxMin algorithm.

1. Valuations
2. Final basis indices
3. Okutsu bases numerators

While the Okutsu bases numerators are not required by the MaxMin algorithm, they are an output of the Montes algorithm, so they must be stored during the running of MaxMin.

Proposition 6.14. *If $\mathbb{F}$ is a finite field with q elements, the additional space required by* MaxMin *to compute an integral basis of $\mathcal{O}_L$ is $\mathcal{S}_{\text{MaxMin}} = O\left(n^2\delta\log(q)\right)$ bits.*

Proof. As stated in Section 6.1.2, the valuations required by MaxMin take the form $\nu_{k,i,j} := w_{\mathfrak{p}_i}\left(g_{k,\mathfrak{p}_j}\right)$ for $\mathcal{P} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$ and $1 \leqslant i, j \leqslant s$, $0 \leqslant k \leqslant n_{\mathfrak{p}_j}$. There are $n + s$ numerators, for each of which we require s valuations, we must store

$$\mathcal{S}_{\text{val}} = O\left((n+s)s\right) = O\left(n^2\right),$$

valuations. We have already seen that these valuations are positive rational numbers of size at most $\delta = v\left(\text{disc}(f)\right)$.

The MaxMin algorithm does not compute the numerators of the basis directly, but rather computes a family of multi-indices $\mathbb{i}_0, \ldots, \mathbb{i}_n$ which describe them. There are $n + 1$, each of which contains s integers. This gives

$$\mathcal{S}_{\text{ind}} = O\left((n+1)s\right) = O\left(n^2\right),$$

integers. As the sum of each component of the multi-index $\mathbb{i}_k$ is k, all these positive integers must be less than n.

The final, and largest, set of elements which must be stored are the numerators of the Okutsu $\mathfrak{p}$-bases. As we saw discussing the valuations, there are $n + s$ numerators. Each numerator is stored as a polynomial of degree less than or equal to n with coefficients in $\mathcal{O}$. As stated in Section 6.2, we may consider elements in $\mathcal{O}$ to be a finite π-adic development of length $\delta + 1$.

So, the space required to store these numerators is

$$\mathcal{S}_{\text{num}} = O\left((n+s)n(\delta+1)\right) = O\left(n^2\delta\right),$$

elements of the residue field $\mathbb{F}$. We have assumed a finite residue field, which is of size $q = \#\mathbb{F}$.

From these three space complexity estimates, we can give an over space complexity for computing a basis with the MaxMin algorithm. The space requirement is

$$\begin{aligned}\mathcal{S}_{\text{MaxMin}} &= \mathcal{S}_{\text{val}} + \mathcal{S}_{\text{ind}} + \mathcal{S}_{\text{num}} \\ &= O\left(n^2 \log(\delta)\right) + O\left(n^2 \log(n)\right) + O\left(n^2 \delta \log(q)\right) \\ &= O\left(n^2 \delta \log(q)\right),\end{aligned}$$

bits. □

7

Example computations

"When you want to know how things really work, study them when they're coming apart."

– William Gibson, *Zero History*

The algorithms for computing triangular bases of integral closures presented in this work have been implemented as part of the "+Ideals" package for the computer algebra system Magma [BCP97].

In this chapter, we will present a number of example computations, comparing running times of the MaxMin based routines with pre-existing routines used by Magma as well as two other OM-based methods

All executions were performed on GNU/Linux running on 8-core 3.0GHz nodes with 32GB main memory. Each execution ran in a single core, using Magma 2.18-5.

The defining polynomials of the algebraic fields used in the examples in this chapter are detailed in Section 7.5.

7.1 Algorithms

We will present results from four different algorithms in this chapter. One of these algorithms is that present in the Magma software package. The remaining three are based on the use of OM representations of prime ideals.

The first, is our own MaxMin algorithm, as described in Chapter 4. As presented in that chapter, the MaxMin algorithm produces triangular v-integral bases, which are also v-reduced.

We will compare our algorithm with an improvement, due to Jens-Dietrich Bauch [Bau14], of the "Multipliers" method of constructing an integral basis presented in [GMN13]. In this algorithm, Okutsu bases are multiplied by a product of Okutsu approximations to form a v-integral basis. This method does not guarantee that the basis is triangular.

The third OM-based algorithm is the *method of the quotients*, described in [GMN]. This method uses the quotients of certain divisions with remainder generated as a byproduct of the Montes algorithm to construct an v-integral basis.

These three OM-methods first apply the Montes algorithm and then some specific ideas to compute v-integral bases. The running times of the figures in this chapter are the total time in seconds of the concatenation of the two procedures.

Finally, we consider the standard implementation found in Magma as a single algorithm. In reality this is either the Round-2 [Coh93] or Round-4 [Poh93] algorithm, which Magma selects depending on the field which it is being applied to.

7.2 Bases of p-maximal orders

Let p be a prime number and let $f \in \mathbb{Z}[x]$ be a monic, irreducible polynomial of degree n. Fix a root $\theta \in \overline{\mathbb{Q}}$ of f and let $K = \mathbb{Q}(\theta)$ be the corresponding

algebraic number field. In this section we will present examples of computing a p-integral basis of the maximal order of K.

7.2.1 Single prime ideal

There are cases where there is only a single prime ideal $\mathfrak{p}$ lying over the prime p. As mentioned in Chapter 3, when $\mathcal{P} = \{\mathfrak{p}\}$ an Okutsu basis is already a p-integral basis of the maximal order. In this case, no application of the MaxMin algorithm is required, the Okutsu basis can be constructed directly from the OM representation of $\mathfrak{p}$. Similar *direct* constructions are performed by the Multipliers and Quotients algorithms, so there is essentially no difference in execution time between the three OM-based algorithms.

For the sake of completeness, we will compare the construction of an Okutsu basis using the OM factorisation algorithm against the current routines existing in Magma. Although it is not guaranteed, Magma almost always produces integral bases of number fields in Hermite normal form. To make a correct comparison, we include the time to convert the Okutsu basis into Hermite normal form as well.

Figure 7.1 shows the running time of the MaxMin and Magma algorithms applied to number fields defined by the polynomials $f = A_{101,n,211,0}(x)$, where $n = \deg f$ ranges from 2 to 200 by increments of three.

It can be seen that Magma and MaxMin are equivalent for $n < 10$, however Magma's computational time quickly increases, while the OM-based method does not increase much beyond a single second all the way up to $n = 200$. For $n > 107$, Magma took over an hour to compute the p-integral basis and the time is not included.

The second example we present also has only a single prime ideal dividing p. Figure 7.2 shows the running times of the OM-based routine and the Magma routine computing integral basis of the maximal order of number field defined by the polynomials $E_{p,j}(x)$ with $1 \leqslant j \leqslant 8$.

In this case, Magma cannot compute a basis for the number field defined by this polynomial for $j > 5$ in less than 24 hours. The OM-based algorithm requires 260 seconds in the final case, however most of this time is computing the Hermite Normal Form, while the OM factorisation requires less than two

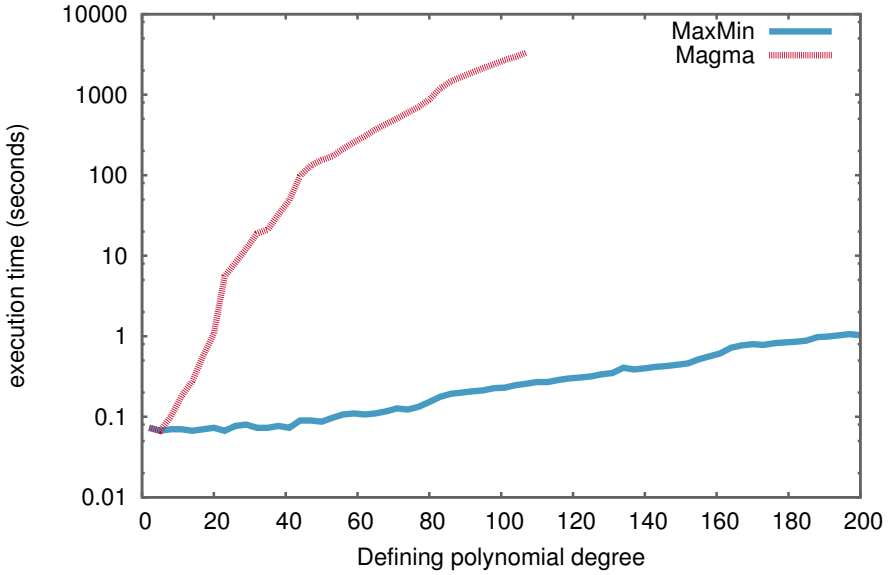


Figure 7.1: Running time for maximal order Hermitian p-basis computation defined by polynomials $A_{101,n,211,0}(x)$ with $2n \in \{2, 5, 8, \ldots, 200\}$.

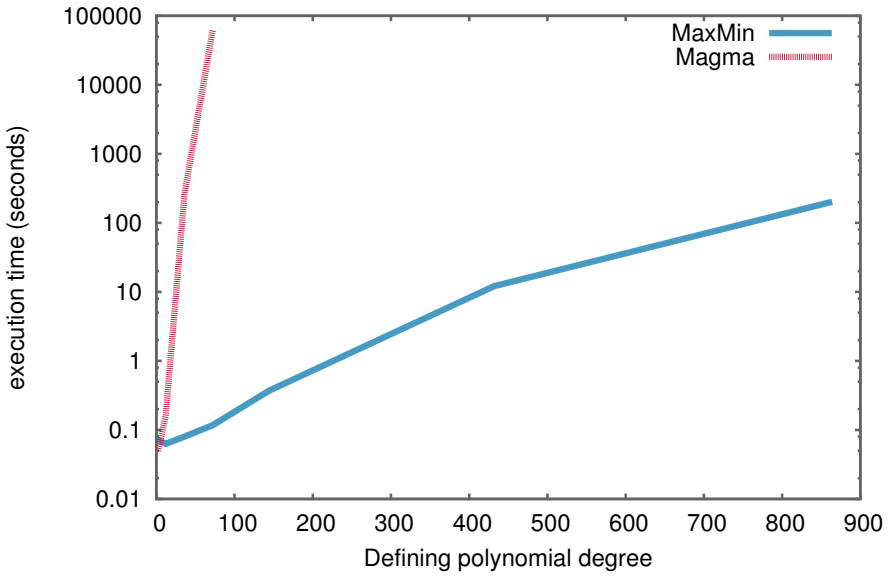


Figure 7.2: Running time for maximal order Hermitian p-basis computation defined by polynomials $E_{13,j}(x)$ with $1 \leqslant j \leqslant 8$ of degree 2, 4, 12, 36, 72, 144, 432, 864.

seconds in all cases.

From these examples it can be seen that the running time of the Magma routine increases drastically with the degree of the defining polynomial, whereas the Montes algorithm increases at a much slower rate.

7.2.2 Multiple prime ideals

In order to evaluate the performance of the MaxMin algorithm as compared to other OM-based methods, it is necessary to choose defining polynomials so that there are multiple prime ideals dividing p.

In this section, we are interested in showing the running time of the different algorithms, varying some of the important characteristics of the number field defined by these polynomials, which affect the computational complexity of generating a p-integral basis of the maximal ideal.

In this section, we compare the time to compute a basis, however the bases are not equal. The MaxMin algorithm will, of course, compute a triangular basis, while the Multipliers and Quotients routines do not have this guarantee.

Running time vs width

The width of a prime polynomial in $\mathbb{Z}_p[x]$ is defined in [GNP12]. It is an upper bound for the number of refinement steps that may occur during an execution of the Montes algorithm.

The first defining polynomial to be shown whose prime factors have a varying width is $f = B_{101,k}(x)$ with $k \leqslant 5000$. The $B_{p,k}(x)$ polynomials have 2 prime ideals dividing p, the width of each of the corresponding prime factors of f in $\mathbb{Z}_p[x]$ is $\lceil k/3 \rceil$. Since the polynomial f has small degree $n = 6$, it is possible to include in Figure 7.3 the times for Magma as well as the three OM-based algorithms. To correctly compare with the Magma implementation, we have included the time to put the OM-based routine output in HNF, however since the basis matrix is small, this requires a negligible amount of time compared to the computation of the basis in each case.

It can be seen that in this case, the three OM-based algorithms, MaxMin,

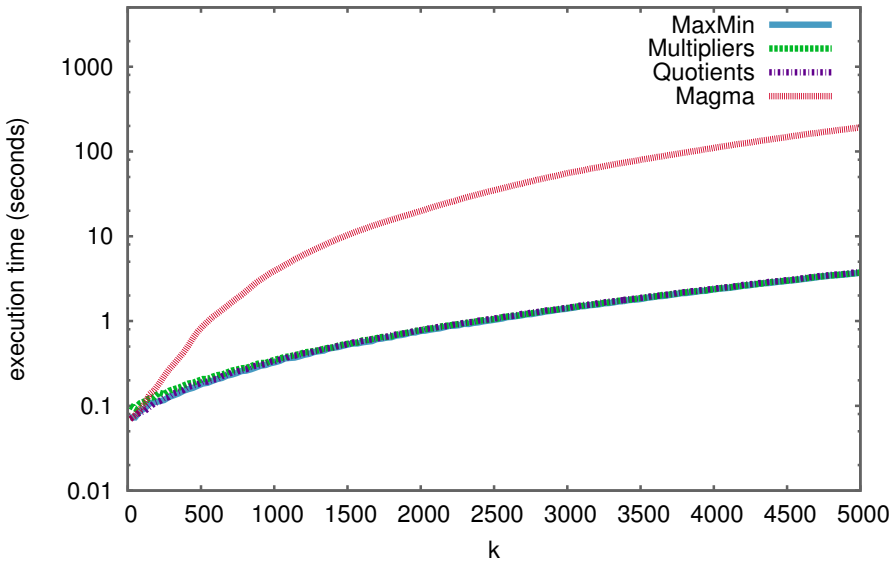


Figure 7.3: Running time for maximal order Hermitian p-basis computation defined by polynomials $B_{101,k}(x)$ with $k \leqslant 5000$.

Multipliers and Quotients perform almost identically. The Magma routine is faster while $k < 200$, but quickly increases to be several orders of magnitude greater than the other routines.

From this point onward, we will not include times for Magma in the examples given. In all cases, Magma was significantly slower than the OM-based algorithms, often unable to complete some of the larger examples in less than 24 hours.

In Figure 7.4 a second example of a defining polynomial with varying width is given. The polynomial $f = C_{101,k}(x)$ defines number fields where 6 prime ideals divide p, each of the corresponding prime factors in $\mathbb{Z}_p[x]$ has width $6k - 90$. Different to the previous example, the prime ideals dividing p have depth 3. Once again, the three OM-based algorithms are quite similar, but this time the Multipliers routine takes longer than the MaxMin and Quotients algorithms.

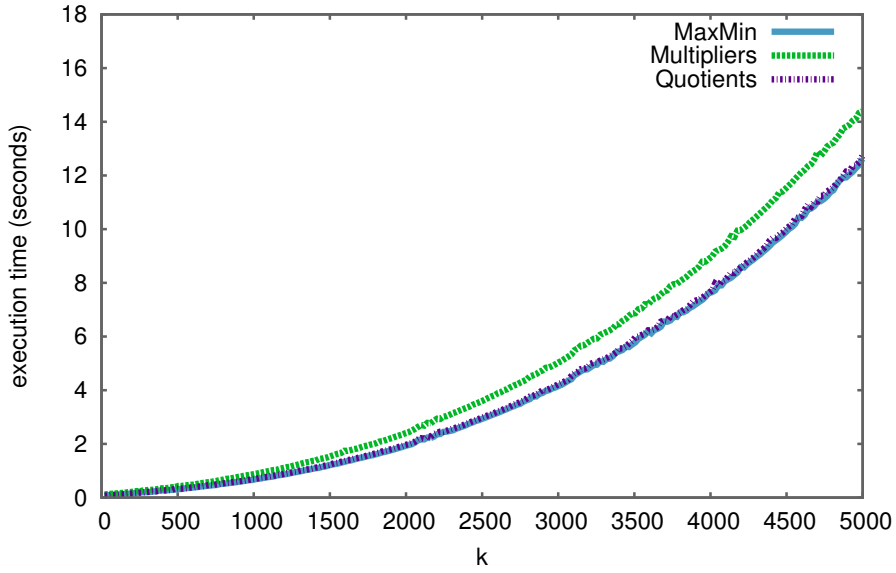


Figure 7.4: Running time for maximal order p-basis computation defined by polynomials $C_{101,k}(x)$ with $k \leqslant 5000$.

Running time vs number of prime factors over $\mathbb{Z}_p[x]$

To compare running times against the number of prime ideal factors of p, we will use two further sets of defining polynomials.

Figure 7.5 shows the time required by the three OM-based routines to compute a p-integral basis of the number fields defined by the polynomials $f = A^m_{1009,n,211}(x)$. The degree of the number field is $\deg f = nm = 1000$ in all cases, with the number of prime ideals dividing p equal to $m \in \{5, 10, 20, 50, 100, 200, 500\}$.

In this example, the Quotients routine begins slower than Multipliers and MaxMin, but as the number of factors increases, it performs better than Multipliers, although still slower than MaxMin which remains fastest in all cases.

Another example with a variable number of factors is shown in Figure 7.6. This example uses the set of defining polynomials $f = D_{101,p,2,21}(x)$, with $p \in \{1069, 1087, 1051, 1117, 1097, 919, 1009\}$.

In this example we have up to 50 prime ideals dividing p. We see that the three OM-based methods perform similarly for a small number of prime

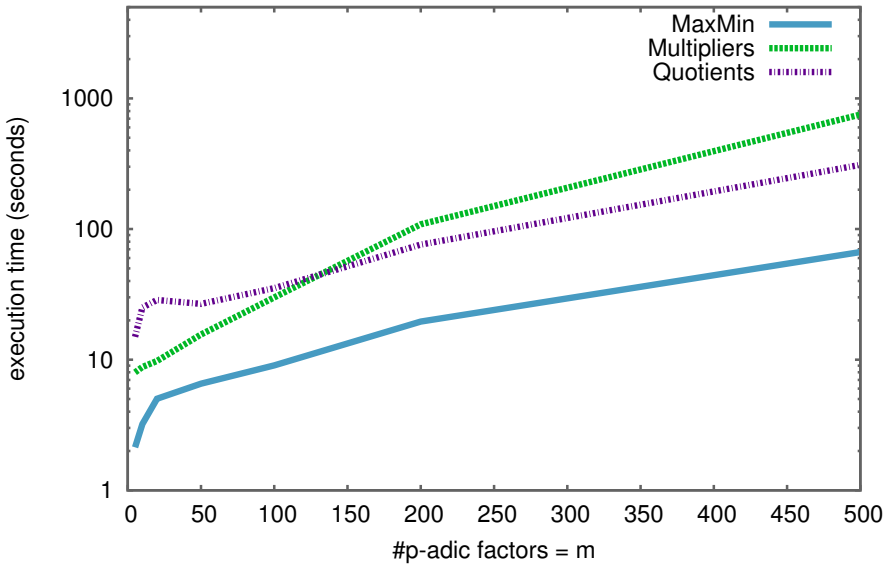


Figure 7.5: Running time for maximal order p-basis computation defined by polynomials $A^m_{1009,n,211}(x)$ with $nm = 1000$ and $m \in \{5, 10, 20, 50, 100, 200, 500\}$.

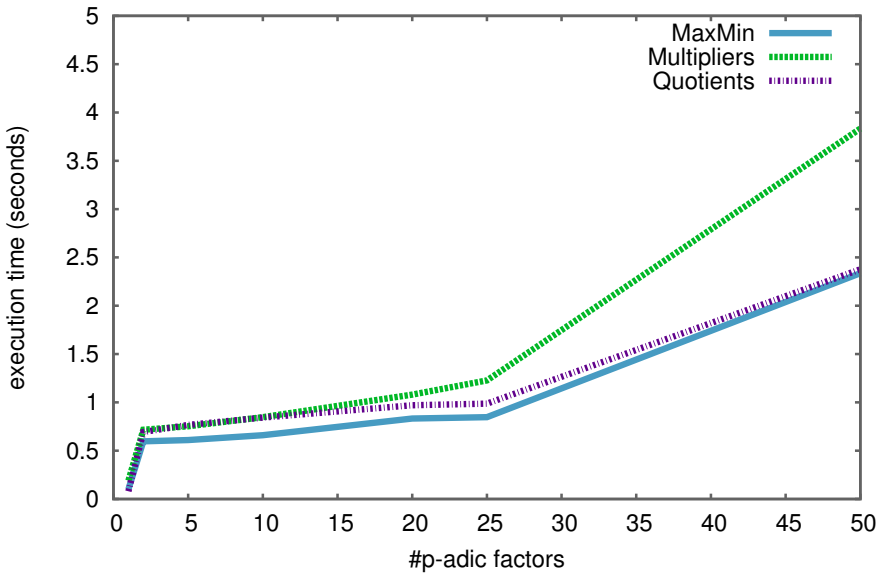


Figure 7.6: Running time for maximal order p-basis computation defined by polynomials $D_{101,p,2,21}(x)$ with $p \in \{1069, 1087, 1051, 1117, 1097, 919, 1009\}$ of degree 1, 2, 5, 10, 20, 25, 50.

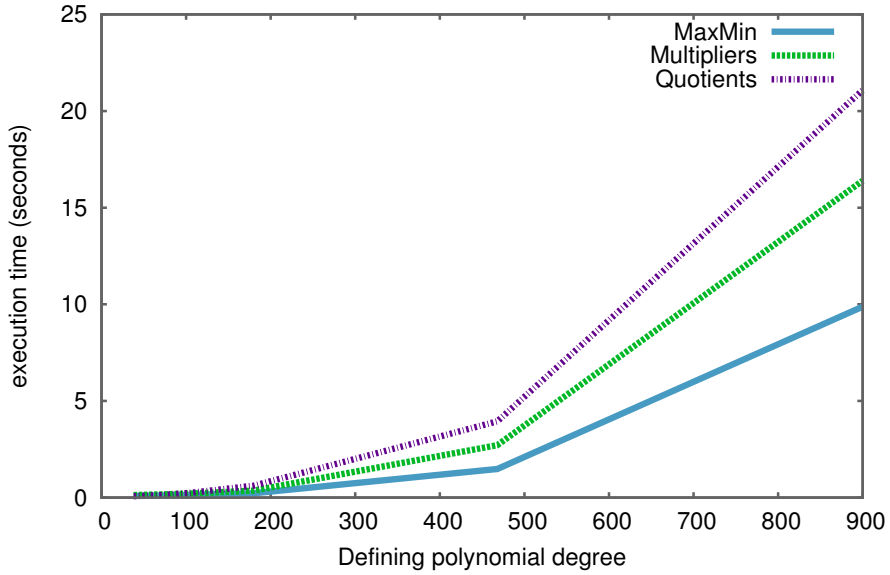


Figure 7.7: Running time for maximal order p-basis computation defined by polynomials $EC_{101,j}(x)$ with $1 \leqslant j \leqslant 8$ of degree 38, 40, 48, 72, 108, 180, 468, 900.

ideals and then the Multipliers routine takes longer for the final case, while Quotients and MaxMin are very similar.

Running time vs depth

In order to show how the running time of the OM-based routines varies with the depth of a prime ideal dividing p, we construct a set of composite defining polynomials $f = EC_{101,j}(x)$ with $1 \leqslant j \leqslant 8$. In this case, there will be a single prime ideal with variable depth equal to j, and 6 further prime ideals of constant depth 3. The time to compute a p-integral basis using each of the algorithms is shown in Figure 7.7.

Due to the difference in depth of the prime ideals dividing p, the MaxMin and Multipliers algorithms must both use the SFL algorithm to improve the quality of some of the Okutsu approximations used in the construction of their respective bases. The Quotients method does not require this step, however it is still slower in practice.

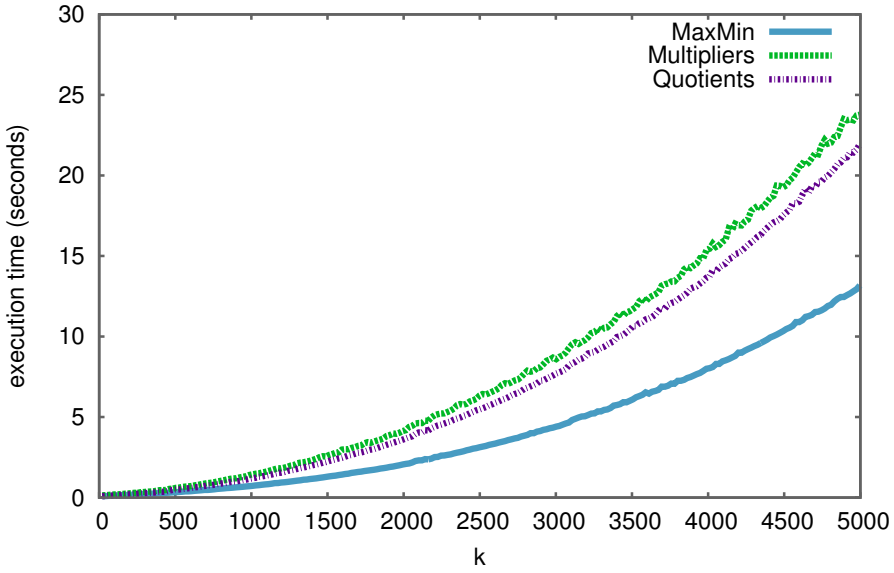


Figure 7.8: Running time for maximal order Hermitian p-basis computation defined by polynomials $C_{101,k}(x)$ with $k \leqslant 5000$.

7.2.3 Hermitian bases

In Section 7.2.2, we have simply compared the time to compute a local basis for each of the OM-based routines. However, often a basis is required in a specific format. The goal of the MaxMin algorithm is to compute triangular bases, however in some cases a Hermitian basis may be required. In this case, MaxMin has an advantage over the Multipliers and Quotients routines, as less work is required to put a triangular matrix in Hermite Normal Form (HNF) than to do the same for an arbitrary matrix.

We will look at two examples to see how the additional time required to compute a Hermitian basis changes the comparison between OM-based routines. The first example is a small (degree 36) polynomial, the second is of variable degree.

Figure 7.8 shows the time required to compute a Hermitian basis of the maximal order of number fields defined by the polynomials $f = C_{101,k}$ with $k \leqslant 5000$. This should be compared to Figure 7.4, which shows the computation time required for a basis (not in Hermite normal form).

We see that, while the Multipliers routine takes only slightly longer than

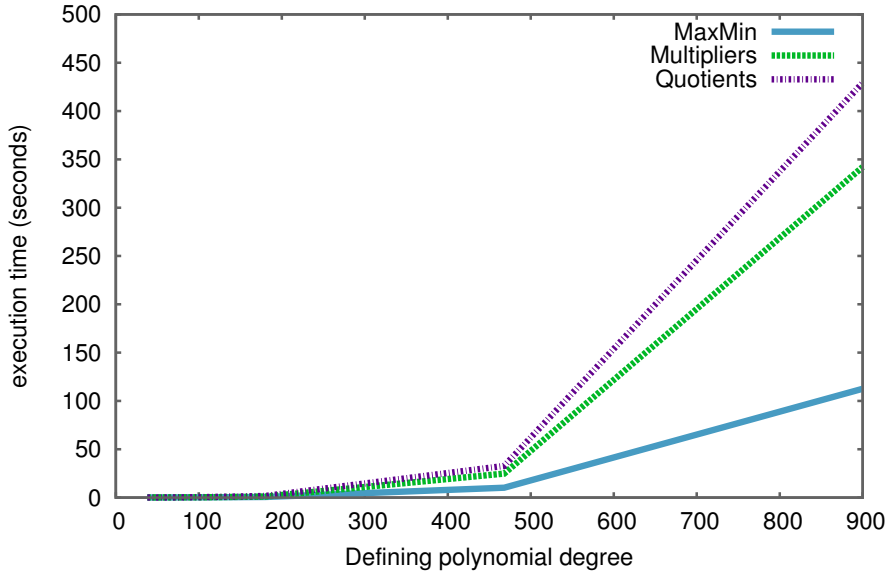


Figure 7.9: Running time for maximal order Hermitian basis computation defined by polynomials $EC_{101,j}(x)$ with $1 \leqslant j \leqslant 8$ of degree 38, 40, 48, 72, 108, 180, 468, 900.

MaxMin to compute a basis, since it is not triangular in this case, it requires much longer to compute the HNF. In the final case, $k = 5000$, the bases produced by the Multipliers and Quotients routines required an additional 9 seconds for the HNF computation, whereas the basis produced by MaxMin required only 0.5 seconds.

In Figure 7.9 we can see the time required to compute a Hermitian basis of the maximal order of number fields defined by the polynomials $f = EC_{101,j}$ with $1 \leqslant j \leqslant 8$. The non-Hermitian basis computation time is shown in Figure 7.7.

In this case, although MaxMin takes considerably less time to compute the basis than both the Multipliers and Quotients routines, the difference in time required to make that basis Hermitian is not as pronounced. For the smaller degree polynomials, the MaxMin is approximately twice as fast. For $EC_{101,8}$ the basis is 900×900 and MaxMin is 3.5 times faster than Multipliers and 4.5 times faster than Quotients.

7.3 Bases of $p(t)$-maximal orders

For a given prime number q, denote by $A := \mathbb{F}_q[t]$ and $K := \mathbb{F}_q(t)$ the polynomial ring and rational function field in the indeterminate t over the finite field with q elements.

Let $p(t) \in A$ be a prime polynomial, that is, monic and irreducible, and let $f \in A[x]$ be a monic irreducible separable polynomial of degree n. Fix a root $\theta \in \overline{K}$ of f and let $L = K(\theta)$ be the function field defined by f.

In the same way that we did for algebraic number fields, in this section, we will show examples of computing a $p(t)$-integral basis of the maximal order of K.

7.3.1 Single prime ideal

As in the number field case, construction of a $p(t)$-integral basis of the maximal order of a function field where only a single prime ideal divides $p(t)$ do not make use of the MaxMin algorithm itself. As such, we will consider only a single OM-based routine compared to the Magma routine.

The times compared here are to compute a Hermitian basis, as the Magma routine produces bases in HNF in all these cases.

Figure 7.10 shows the running time for computing the $p(t)$-integral basis of the maximal order for function fields defined by polynomials $f = A_{p(t),n,3,0}(x)$ for $2 \leqslant n \leqslant 200$. The prime polynomial used is $p(t) = t^2 + 1 \in \mathbb{F}_3[t]$.

Magma was unable to compute a basis for $n > 95$ in less then an hour. The MaxMin algorithm took less than 400ms in all cases.

The running time for function fields defined by polynomials $f = E_{p(t),j}$ for $1 \leqslant j \leqslant 8$ and $p(t) = t^2 + 1 \in \mathbb{F}_7[t]$ are shown in Figure 7.11.

For $j > 5$, Magma was unable to compute a $p(t)$-integral basis of the maximal order in less than 24 hours, while the OM-based routine completed up to $j = 8$. At the final computation, the OM-based routine required 8,580 seconds to complete, however only 173 seconds of those was the actual OM routines. Most of the time was spent computing the HNF, which can be especially costly for function fields.

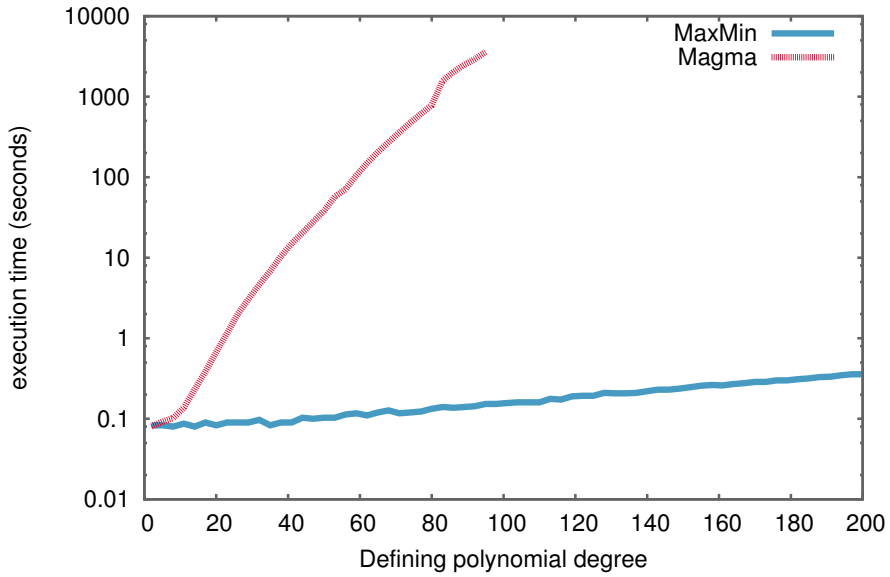


Figure 7.10: Running time for maximal order Hermitian $p(t)$-basis computation defined by polynomials $A_{t^2+1,n,3,0}(x)$ with $n \in \{2, 5, 8, \ldots, 200\}$.

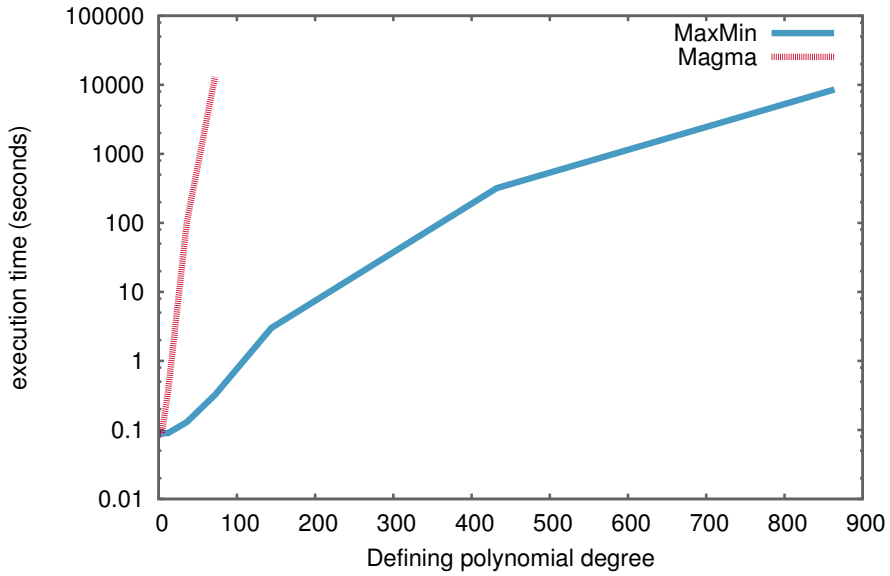


Figure 7.11: Running time for maximal order Hermitian $p(t)$-basis computation defined by polynomials $E_{t^2+1,j}(x)$ for $1 \leqslant j \leqslant 8$ of degree 2, 4, 12, 36, 72, 144, 432, 864.

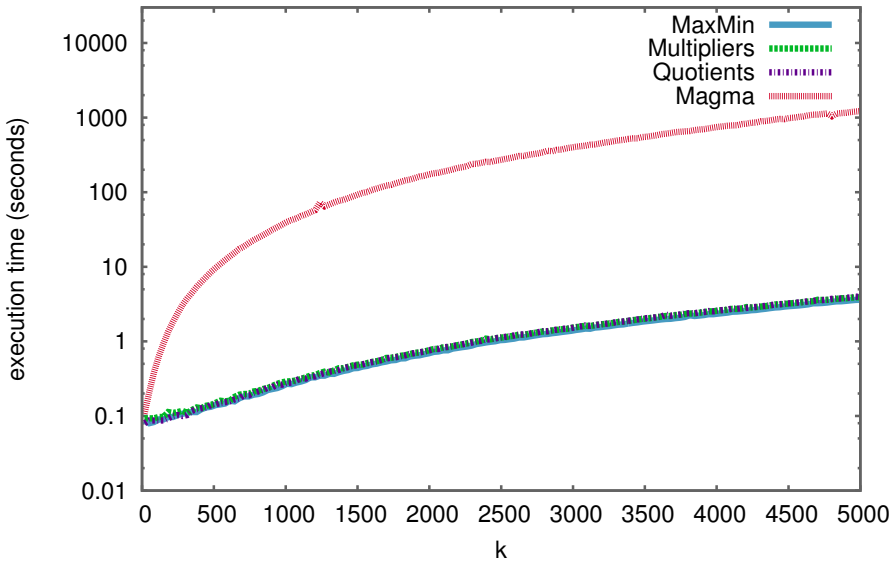


Figure 7.12: Running time for maximal order Hermitian $p(t)$-basis computation defined by polynomials $B_{t^3+2,k}(x)$ with $k \leqslant 5000$.

7.3.2 Multiple prime ideals

Running time vs width

In Figure 7.12, the running times for the computation of the $p(t)$-integral bases of the maximal ideals of the function fields defined by the polynomials $f = B_{p(t),k}(x)$ are presented. In all cases, the prime polynomial is $p(t) = t^3 + 2 \in \mathbb{F}_7[t]$. Similar to the number field case, the small degree of the defining polynomial allows Magma to complete the computation for all values of $k < 5000$ in a reasonable amount of time. The time to convert the bases computed by the OM-based routines to HNF has been included to allow a fair comparison with the routine built into Magma, however as in the number field case, the small size of the basis means that this time is a small fraction of the overall computation.

This figure shows that all four routines require roughly the same amount of time for small k, then the Magma routine presents a rapidly increasing running time as k grows, while the three OM-based routines increase at the same, much slower rate.

The polynomials, $f = C_{p(t),k}(x)$ in $\mathbb{F}_7[t, x]$ define function fields where

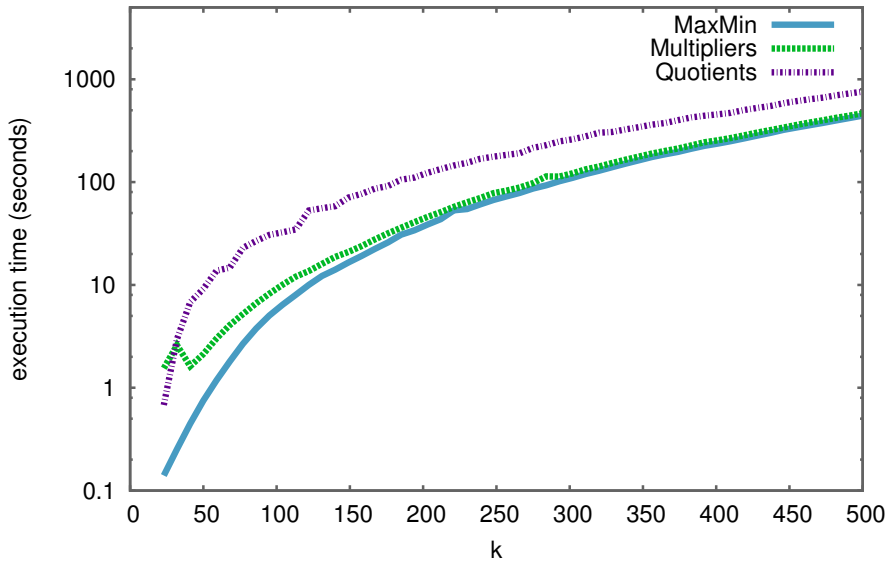


Figure 7.13: Running time for maximal order $p(t)$-basis computation defined by polynomials $C_{t^3+2,k}(x)$ with $k \leqslant 500$.

six prime ideals divide $p(t) = t^3 + 2$. The three OM-based routines have similar running times for this set of defining polynomials, with the Quotients routine being slower across all cases and the MaxMin routine being slightly faster than Quotients. The running times are displayed in Figure 7.13.

Running time vs number of prime factors over $\hat{A}[x]$

As arithmetic operations in functions fields are more costly than in number fields, it is not possible to construct bases for function fields with very large numbers of prime ideals dividing $p(t)$. Therefore we limit this example using defining polynomials $f = A^m_{p(t),n,3}(x)$ with $nm = 64$, where we take $m \in \{2, 4, 8, 32\}$. The prime ideal is $p(t) = t^2 + 4 \in \mathbb{F}_{37}[t]$. The running times are presented in Figure 7.14.

In this example we can see that the Quotients routine increases with the number of factors at a much faster rate than the MaxMin and Multipliers algorithms, although all three methods present linear running times in the number of factors.

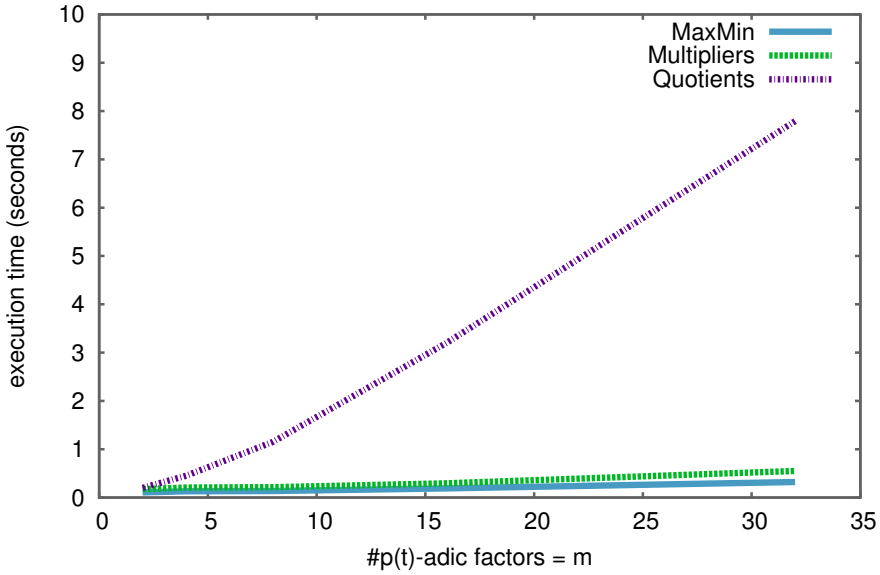


Figure 7.14: Running time for maximal order $p(t)$-basis computation defined by polynomials $A^m_{t^2+4,n,28}(x)$ with $n \cdot m = 64$, $m \in \{2, 4, 8, 32\}$.

Running time vs depth

Figure 7.15 presents the running times for the computation of the $p(t)$-integral bases of the maximal orders of function fields defined by the polynomials $EC_{p(t),j}(x)$ for $1 \leqslant j \leqslant 6$. The prime polynomial $p(t) = t^2+4 \in \mathbb{F}_7[t]$ is divisible by 5 prime ideals, one of which has variable depth j, while the others have fixed depth of 3.

The running times for the MaxMin and Multipliers algorithms are very similar. If we compare this example to that shown in 7.11, it is evident this polynomial defines a much more complex case than that of just $E_{p(t),j}(x)$. This is partly due to additional computation required to perform the Montes algorithm, but can mostly be attributed to the requirement of the single factor lifting algorithm - which accounts for almost all of the computation time in the larger examples.

It is interesting that, even with the large computational requirement imposed by the use of the SFL algorithm, both the MaxMin and the Multipliers routines are considerably faster than the Quotients routine, which does not require SFL at all. The Quotients routine was unable to compute

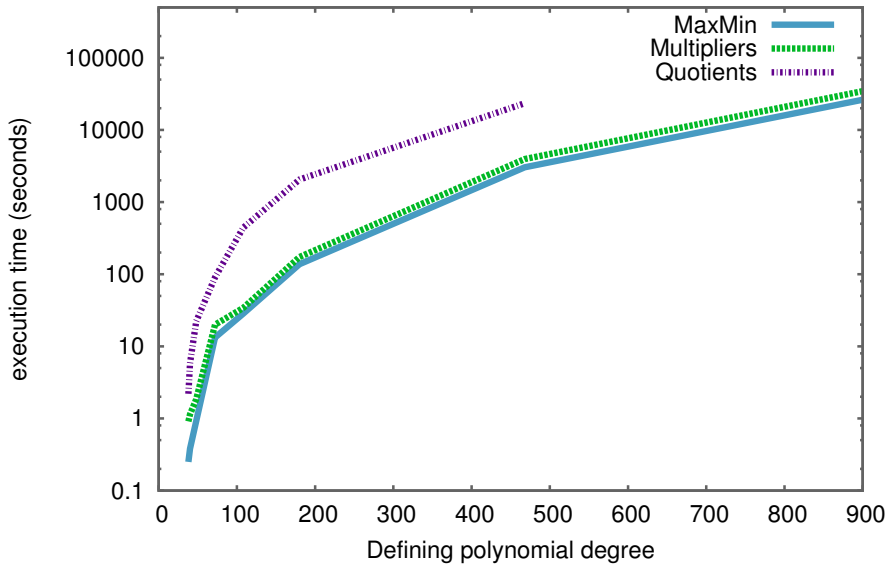


Figure 7.15: Running time for maximal order $p(t)$-basis computation defined by polynomials $EC_{t^2+4,j}(x)$ with $1 \leqslant j \leqslant 8$ of degree 38, 40, 48, 72, 108, 180, 468, 900.

the final basis ($j = 8$) in less than 24 hours.

Running time vs $\deg p(t)$

A characteristic specific to computing $p(t)$-integral bases of the maximal orders of function fields is the degree of the prime polynomial $p(t)$. Figure 7.16 shows the running time for computing the $p(t)$-integral bases of maximal orders of function fields defined by $C_{p(t),23}(x)$, where $p(t) \in \mathbb{F}_{23}[t]$ is of degree $4 \leqslant \deg p(t) \leqslant 200$ incrementing by steps of 2. Specifically, we choose the lexicographically smallest polynomial for each degree.

This example presents approximately linear running time increase in terms of the degree of the prime polynomial $p(t)$ for each of the three OM-based routines. MaxMin runs consistently faster than Multipliers, which is again faster than Quotients.

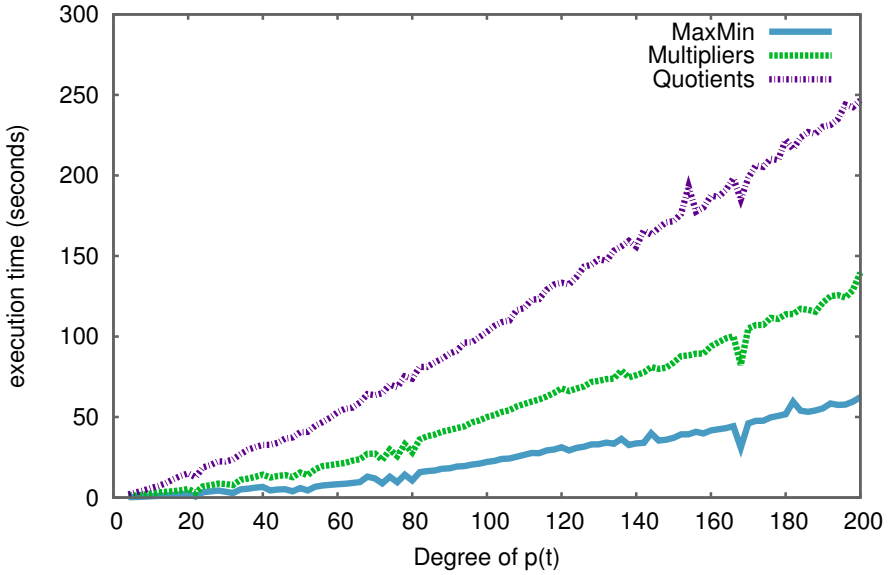


Figure 7.16: Running time for maximal $p(t)$-order basis computation defined by polynomials $C_{p(t),23}(x)$ with $4 \leqslant \deg p(t) \leqslant 200$.

7.4 Fractional ideals

The MaxMin algorithm is also capable of constructing p-bases of fractional ideals, as described in Chapter 5.

We retain the appropriate setting from the previous sections. Now, let $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ be the prime ideals dividing p in the number field case and $p(t)$ in the function field case. Consider a fractional ideal

$$I = \prod_{i=1}^{s} \mathfrak{p}_i^{a_i}, \qquad a_i \in \mathbb{Z}. \tag{7.1}$$

In this section, we will only compare the MaxMin and the Multipliers algorithms, as the Quotients routine cannot compute bases for fractional ideals.

7.4.1 Number Field

Consider the case where all a_i are chosen randomly in the interval $[-30, 30]$. Figure 7.17 shows the required time for the MaxMin and Multipliers algorithms to compute a triangular base for the maximal order compared to a

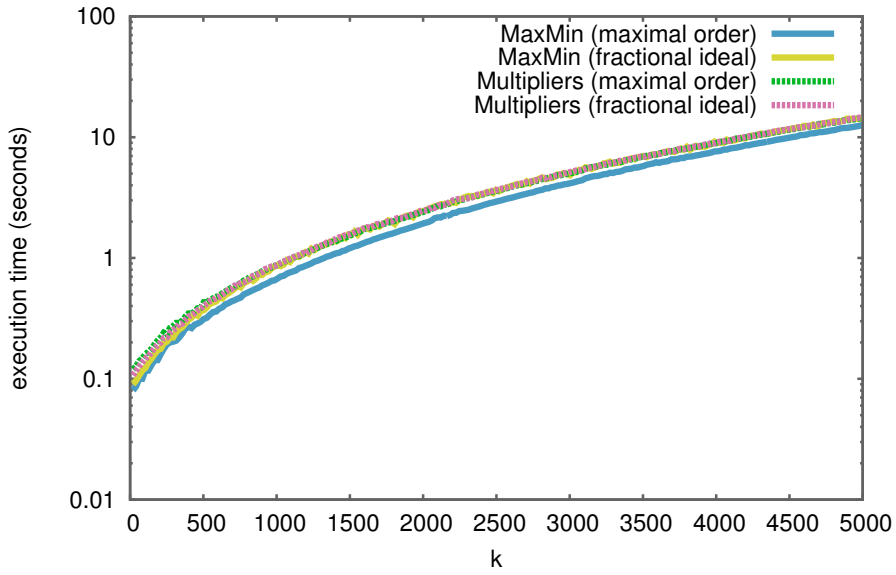


Figure 7.17: Running time for maximal order and fractional ideal p-basis computation defined by polynomials $C_{101,k}(x)$ with $k \leqslant 5000$.

random fractional ideal with these "small" exponents. In both cases, we are working over number fields defined by the polynomials $f = C_{101,k}$ with $k \leqslant 5000$.

It can be seen that constructing a basis of a random fractional ideal requires slightly more time than MaxMin takes for the maximal order. This is because the single factor lifting algorithm is required in the earlier case, whereas in this example it is not required for the maximal order. The Multipliers routine presents time that are much more similar, as it already requires a number of rounds of the SFL algorithm in the maximal order case.

To further explore the time required to construct a basis of a fractional ideal, we consider the case of an ideal $I = \mathfrak{p}_1^{a_1}$ with only one non-zero exponent. In Figure 7.18, the running time is shown as the exponent a_1 is increased.

The running time increases with the difference in exponents. However, since the SFL algorithm approximately doubles the precision of an approximation at each step, the running time increase is logarithmic.

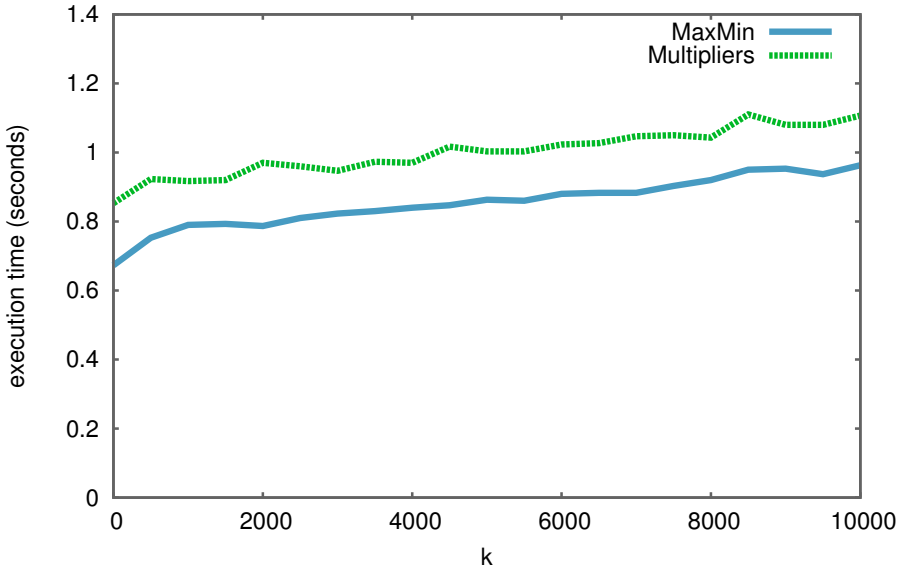


Figure 7.18: Running time for fractional ideal $I = \mathfrak{p}_1^{a_1}$ p-basis computation defined by polynomials $C_{101,1000}(x)$ with exponent $0 \leqslant a_1 \leqslant 10,000$.

7.4.2 Function Field

In Figure 7.19 we compare the running time for computing a basis of the maximal order compared to that for computing a basis of a random fractional ideal for both the MaxMin and Multipliers algorithms. The number field is defined by the polynomials $f = C_{p(t),k}(x)$ with $p(t) = t^3 + 2 \in \mathbb{F}_7[t]$ and $23 \leqslant k \leqslant 500$. The fractional ideal is the product of the 6 prime ideals that divide $p(t)$ each raised to a random exponent in the interval $[-30, 30]$.

We see that the MaxMin algorithm takes less time to compute the maximal order than a fractional ideal. The Multipliers routine takes about that same time to compute a either basis as MaxMin does to compute a basis of a fractional ideal. As in the number field case, this is because the Single Factor Lifting algorithm must be applied in both cases for the Multipliers algorithm, but only in the fractional ideal case for the MaxMin routine.

Figure 7.20 compares the running times of the MaxMin and Multipliers algorithm computing a local basis of a fractional ideal of the form $I = \mathfrak{p}_1^{a_1}$, where $0 \leqslant a_1 \leqslant 2,000$. The underlying function field is the same in all cases, defined by $C_{p(t),100}(x)$ with $p(t) = t^3 + 2 \in \mathbb{F}_7[t]$.

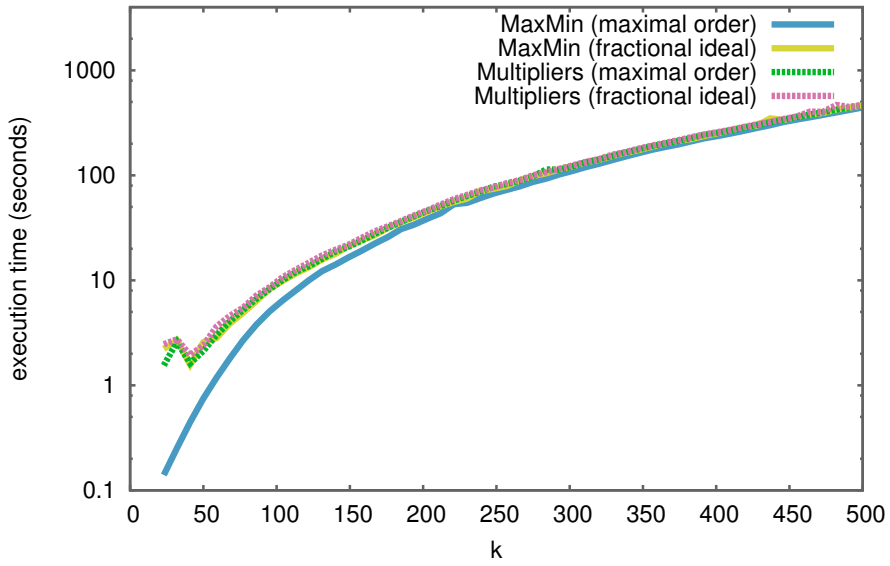


Figure 7.19: Running time for maximal order and fractional ideal $p(t)$-basis computation defined by polynomials $C_{t^3+2,k}(x)$ with $k \leqslant 500$.

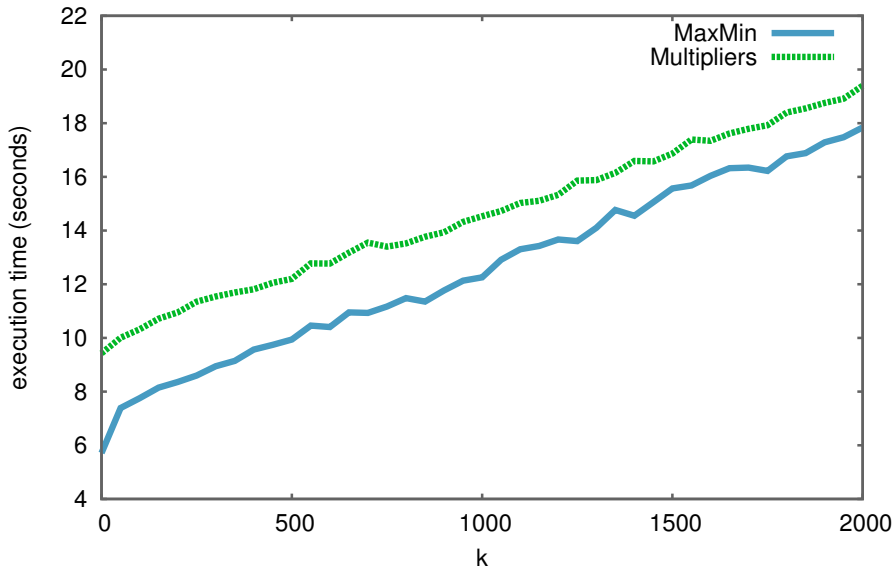


Figure 7.20: Running time for fractional ideal $I = \mathfrak{p}_1^{a_1}$ $p(t)$-basis computation defined by polynomials $C_{p(t),100}(x)$ with $p(t) = t^3+2 \in \mathbb{F}_7$ and exponent $0 \leqslant a_1 \leqslant 2000$.

The running time for both routines is similar, with the MaxMin algorithm taking less time. The time difference between the two methods appears to be decreasing as a_1 rises. This is most likely due to the time required to perform Single Factor Lifting becoming dominant.

7.5 Example polynomials

In this section we show the defining polynomials used throughout this chapter. All polynomials presented here are taken from [GNP12].

The first five polynomials are defined by their parameters,

$$
\begin{aligned}
&A_{p,n,k,r}(x) = (x + 1 + p + p^2 + \cdots + p^r)^n + p^k,\\
&A^m_{p,n,k}(x) = (x^n + 2p^k)((x + 2)^n + 2p^k) \cdots ((x + 2m - 2)^n + 2p^k) + 2p^{mnk},\\
&B_{p,k}(x) = (x^2 - 2x + 4)^3 + p^k,\\
&C_{p,k}(x) = ((x^6 + 4px^3 + 3p^2x^2 + 4p^2)^2 + p^6)^3 + p^k,\\
&D_{\ell,p,n,k}(x) = (x^{\ell-1} + \cdots x + 1) + p^k.
\end{aligned}
$$

The "E" polynomials are explicitly defined for each level,

$$
\begin{aligned}
&E_{p,1}(x) = x^2 + p,\\
&E_{p,2}(x) = E_{p,1}(x)^2 + (p - 1)p^3x,\\
&E_{p,3}(x) = E_{p,2}(x)^3 + p^{11},\\
&E_{p,4}(x) = E_{p,3}(x)^3 + p^{29}xE_{p,2}(x),\\
&E_{p,5}(x) = E_{p,4}(x)^2 + (p - 1)p^{42}xE_{p,1}(x)E_{p,3}(x)^2,\\
&E_{p,6}(x) = E_{p,5}(x)^2 + p^{88}xE_{p,3}(x)E_{p,4}(x),\\
&E_{p,7}(x) = E_{p,6}(x)^3 + p^{295}E_{p,2}(x)E_{p,4}(x)E_{p,5}(x),\\
&E_{p,8}(x) = E_{p,7}(x)^2 + (p - 1)p^{632}xE_{p,1}(x)E_{p,2}(x)^2E_{p,3}(x)^2E_{p,6}(x).
\end{aligned}
$$

Finally, the "EC" polynomials are specified as,

$$EC_{p,j}(x) = E_{p,j}(x) \cdot C_{p,28} + p^{900}.$$

The main characteristics of these polynomials can be found in [GNP12].

A

Catalogue of routines

In this appendix, we will present details of the most important routines in the "+Ideals" package that provide support for constructing triangular bases of integral closures following the MaxMin algorithm. Firstly, the existing routines which we make use of will be described, and then we will proceed to describe the new routines that have been added.

A.1 The +Ideals package

The OM factorisation algorithm presented in Chapter 2 has already been implemented as a package for Magma, the computer algebra system. This is the +Ideals package [GMN10a], which implements the Montes algorithms as well as various related routines for operating on OM representations of ideals.

The package may be downloaded from the web-site listed below, where an in depth list of all sub-routines in the package can also be found.

`http://www-ma4.upc.edu/~guardia/+Ideals.html`

We are primarily concerned with three routines from the +Ideals package.

A.1.1 `Montes(K, p : Basis:=false)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.
- `Basis` determines whether a p-integral basis is computed (default: `false`).

This routine has no explicit output, but it does store data in the structure representing K. Let θ be a root of f, and let $\mathcal{O}_K$ be the ring of integers of K. The following data are computed:

- `K`PrimeIdeals[p]`: A list of OM representations of the prime ideals dividing p.
- `K`LocalIndex[p]`: The p-adic valuation of $(\mathcal{O}_K : \mathbb{Z}[\theta])$.

Additionally, if the parameter `Basis` is set to `true`, then

- `K`pBasis[p]`: A p-integral basis of $\mathcal{O}_K$.

The p-integral basis is computed using the method of the quotients described in Section 7.1.

A.1.2 `pHermiteBasis(K, p)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.

This routine will compute a p-integral basis of K in Hermite Normal Form. It requires a p-integral basis, so if necessary, it will call the `Montes` routine with `Basis` set to `true` to obtain one.

Output

- `K`pHermiteBasis[p]` is a p-integral basis of $\mathcal{O}_K$ in Hermite Normal Form.

A.1.3 `SFL(K, P, slope)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `P` is a prime ideal of `K`.
- `slope` is a positive integer.

Let $F_{\mathtt{P}} \in \mathbb{Z}_p[x]$ be the prime factor of f associated to the prime ideal `P` and let $\phi_{\mathtt{P}} \in \mathbb{Z}[x]$ be the Montes approximation to $F_{\mathtt{P}}$ as a factor of f.

The `SFL` routine performs Single Factor Lifting on $\phi_{\mathtt{P}}$, to improve the quality of the approximation as detailed in Section 2.6. The new polynomial will then be stored in the OM representation of `P` found in the list `K`PrimeIdeals[p]`.

Let r be the Okutsu depth of $F_{\mathfrak{p}}$, so that $r+1$ be the order of the OM representation of `P`. Then, the new approximation $\phi'_{\mathtt{P}}$ will have `P`-valuation

$$w_{\mathtt{P}}\left(\phi'_{\mathtt{P}}(\theta)\right) \geqslant \frac{V_{r+1} + \mathtt{slope}}{e_1 \cdots e_r}.$$

A.2 New routines supporting MaxMin

The routines based on the algorithms presented in Chapter 4 are organised into a subpackage of +Ideals, called IdealsBases.

In this section, we will present the routines made available to the user, as well as the fundamental routines that are used to compute bases of ideals.

Throughout this section, K is a number field defined by a monic irreducible polynomial $f \in \mathbb{Z}[x]$ of degree n, θ is a root of f, and $\mathcal{O}_K$ is the integer ring of K.

A.2.1 `MaxMin(K, p, exp)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.
- `exp` is a sequence of integer exponents for each of the prime ideals of K dividing p.

This routine applies the MaxMin algorithm to the set of prime ideals of K that divide p. If the sequence of exponents `exp` is all zeros, it will compute the data needed to construct the maximal order of K, in the contrary case the data produced will construct a fractional ideal of the form

$$I = \prod_{i=0}^{s} \mathfrak{p}_i^{\texttt{exp[i]}}, \qquad \texttt{exp[i]} \in \mathbb{Z},$$

where s is the number of prime ideals of K dividing p.

The data produced by the routine is the indices of the elements of the Okutsu `p`-bases which are used to construct each element of the p-integral basis, as well as the p-valuation of each of these elements.

Output

- `nums_ind` is sequence of indices used to construct the p-integral basis numerators
- `dens_exp` is a sequence of exponents, such that the p-integral basis denominators are $p^{\texttt{exp[i]}}$ for $0 \leqslant i < n$.

A.2.2 `ComputeNumerators(K, p, nums_ind)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.
- `nums_ind` is a sequence of indices of elements from Okutsu `p`-bases.

The `ComputeNumerators` routine complements the `MaxMin` routine, computing the numerators of the basis from the indices which the `MaxMin` routine outputs. The numerators are elements of K, which computationally, can be thought of as polynomials in θ, the root of f.

The elements of the Okutsu `p`-bases, which are used to compute the final basis elements, are constructed from the ϕ-polynomials held in the OM representations of the prime ideal `p`, stored in the list `K`PrimeIdeals[p]`.

Output

- `nums` is a sequence of polynomials in θ which form the numerators of a p-reduced triangular basis of either a fractional ideal or the maximal order of K, depending on the indices given.

A.2.3 `pTriangularBasis(K, p)`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.

This routine produces a p-integral basis of $\mathcal{O}_K$. The basis is triangular and reduced. The `pTriangularBasis` routine requires OM representations of the prime ideals of K dividing p, so if necessary it will run the `Montes` routine (with `Basis` set to `false`).

Output

- `K`pBasis[p]` is a *reduced*, *triangular* p-integral basis of $\mathcal{O}_K$.

A.2.4 `pTriangularIdealBasis(I, p)`

Input

- `I` is a fractional ideal of the number field K defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.

This routine produces a p-integral basis of I. The basis is triangular and reduced. The `pTriangularIdealBasis` routine requires OM representations of the prime ideals of K dividing p, so if necessary it will run the `Montes` routine (with `Basis` set to `false`).

Output

- A list containing a *reduced*, *triangular* p-integral basis of I.

A.2.5 `pHermiteBasis(K, p : Alg:="MaxMin")`

Input

- `K` is a number field defined by the monic irreducible polynomial $f \in \mathbb{Z}[x]$.
- `p` is a prime number.
- `Alg` determines the algorithm used to create the p-integral basis (default: `"MaxMin"`), valid options are
  - `"MaxMin"` : The `pTriangularBasis` routine will be used, employing the MaxMin algorithm.
  - `"Quotients"` : The quotients method will be used.

In this routine, a p-integral basis of $\mathcal{O}_K$ is computed in Hermite Normal Form. The parameter `Alg` determines the routine used to produce the p-integral basis which will then be converted to Hermite Normal Form.

If `Alg` is set to `"MaxMin"`, then the p-integral basis will be computed using the `pTriangularBasis` routine - thereby starting with a triangular basis.

If the second option, `"Quotients"`, is set then the `Montes` routine will be used with the `Basis` parameter set to `true`. This basis may not be triangular.

The user should note that this routine overwrites the routine of the same name present in the base +Ideals package. Setting `Alg` to `"Quotients"` will use the functionality from the original routine.

Output

- `K`pHermiteBasis[p]` is a p-integral basis of $\mathcal{O}_K$ in Hermite Normal Form.

Bibliography

[Bau14] Jens-Dietrich Bauch. *Lattices over polynomial Rings and Applications to Function Fields.* PhD thesis, Universitat Autònoma de Barcelona, July 2014.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[BNS13] Jens-Dietrich Bauch, Enric Nart, and Hayden D. Stainsby. Complexity of OM factorization of polynomials over local fields. *LMS Journal of Computation and Mathematics*, 16:139–171, July 2013.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics.* Springer-Verlag Berlin Heidelberg New York, 1993.

[FPR02] David Ford, Sebastian Pauli, and Xavier-François Roblot. A fast algorithm for polynomial factorization over $\mathbb{Q}_p$. *Journal de Théorie des Nombres de Bordeaux*, 14(1):151–169, 2002.

[GMN] Jordi Guàrdia, Jesús Montes, and Enric Nart. Higher Newton polygons and integral bases. *to appear in the Journal of Number Theory.*

[GMN10a] Jordi Guàrdia, Jesús Montes, and Enric Nart. Arithmetic in big number fields: the '+Ideals' package. *arXiv.org*, May 2010.

[GMN10b] Jordi Guàrdia, Jesús Montes, and Enric Nart. Okutsu invariants and Newton polygons. *Acta Arithmetica*, 145(1):83–108, 2010.

[GMN13] Jordi Guàrdia, Jesús Montes, and Enric Nart. A new computational approach to ideal theory in number fields. *Foundations of Computational Mathematics*, 13(5):729–762, 2013.

[GN] Jordi Guàrdia and Enric Nart. Genetics of polynomials over local fields. *to appear in the Proceedings of AGCTM, Contemporary Mathematics*.

[GNP12] Jordi Guàrdia, Enric Nart, and Sebastian Pauli. Single-factor lifting and factorization of polynomials over local fields. *Journal of Symbolic Computation*, 47(11):1318–1346, 2012.

[Hal01] Emmanuel Hallouin. Computing local integral closures. *Journal of Symbolic Computation*, 32(3):211–230, 2001.

[Hen08] Kurt Hensel. *Theorie de algebraischen Zahlen*. B. G. Teubner, 1908.

[Mac36a] Saunders MacLane. A construction for absolute values in polynomial rings. *Transactions of the American Mathematical Society*, 40(3):363–395, 1936.

[Mac36b] Saunders MacLane. A construction for prime ideals as absolute values of an algebraic field. *Duke Mathematical Journal*, 2(3):492–510, September 1936.

[Mon99] Jesús Montes. *Polígonos de Newton de orden Superior y Aplicaciones Aritméticas*. PhD thesis, Universitat de Barcelona, July 1999.

[Nar14] Enric Nart. On the equivalence of types. *arXiv.org*, September 2014.

[Oku82a] Kōsaku Okutsu. Construction of integral basis. I. *Proceedings of the Japan Academy, Series A, Mathematical Sciences*, 58(1):47–49, 1982.

[Oku82b] Kōsaku Okutsu. Construction of integral basis. II. *Proceedings of the Japan Academy, Series A, Mathematical Sciences*, 58(2):87–89, 1982.

[Ore23] Öystein Ore. Zur Theorie der Algebraischen Körper. *Acta Mathematica*, 44(1):219–314, 1923.

[Ore25] Öystein Ore. Bestimmung der Diskriminanten Algebraischer Körper. *Acta Mathematica*, 45(1):303–344, 1925.

[Poh93] Michael E. Pohst. *Computational Algebraic Number Theory.* DMV Seminar Band 21. Birkhäuser Verlag, 1993.

[PZ89] Michael E. Pohst and Hans Zassenhaus. *Algorithmic Algebraic Number Theory*, volume Encyclopaedia of mathematics and its applications. Cambridge University Press, Cambridge, 1989.

[Ser68] Jean-Pierre Serre. *Corps locaux.* Hermann, second edition edition, 1968.

[SS71] A Schönhage and V Strassen. Schnelle Multiplikation große Zahlen. *Computing*, 7(3-4):281–292, September 1971.

[vH94] Mark van Hoeij. An algorithm for computing an integral basis in an algebraic function field. *Journal of Symbolic Computation*, 18(4):353–363, 1994.