MATHEMATICS OF COMPUTATION
https://doi.org/10.1090/mcom/3517
Article electronically published on February 18, 2020

A FAST RANDOMIZED GEOMETRIC ALGORITHM FOR COMPUTING RIEMANN-ROCH SPACES

AUDE LE GLUHER AND PIERRE-JEAN SPAENLEHAUER

Abstract. We propose a probabilistic variant of Brill-Noether's algorithm for computing a basis of the Riemann-Roch space $L(D)$ associated to a divisor D on a projective nodal plane curve $\mathbb{C}$ over a sufficiently large perfect field k. Our main result shows that this algorithm requires at most $O(\max(\deg(\mathbb{C})^{2\omega}, \deg(D_+)^{\omega}))$ arithmetic operations in k, where ω is a feasible exponent for matrix multiplication and D_+ is the smallest effective divisor such that $D_+ \geq D$. This improves the best known upper bounds on the complexity of computing Riemann-Roch spaces. Our algorithm may fail, but we show that provided that a few mild assumptions are satisfied, the failure probability is bounded by $O(\max(\deg(\mathbb{C})^4, \deg(D_+)^2)/|\mathcal{E}|)$, where $\mathcal{E}$ is a finite subset of k in which we pick elements uniformly at random. We provide a freely available C++/NTL implementation of the proposed algorithm and we present experimental data. In particular, our implementation enjoys a speedup larger than 6 on many examples (and larger than 200 on some instances over large finite fields) compared to the reference implementation in the Magma computer algebra system. As a by-product, our algorithm also yields a method for computing the group law on the Jacobian of a smooth plane curve of genus g within $O(g^\omega)$ operations in k, which equals the best known complexity for this problem.

1. Introduction

The Riemann-Roch theorem is a fundamental result in algebraic geometry. In its classical version for smooth projective curves, it provides information on the dimension of the linear space of functions with some prescribed zeros and poles. The computation of such Riemann-Roch spaces is a subroutine used in several areas of computer science and computational mathematics. One of its most proeminent applications is the construction of algebraico-geometric error-correcting codes [11]: Such codes are precisely (subspaces of) Riemann-Roch spaces. Another direct application is the computation of the group law on the Jacobian of a smooth curve: representing a point in the Jacobian of a genus-g curve $\mathcal{C}$ as $D - gO$, where D is an effective divisor of degree g and O is a fixed rational point (or more generally, a fixed divisor of degree 1), the sum of the classes of $D_1 - gO$ and $D_2 - gO$ can be computed by finding a function f in the Riemann-Roch space $L(D_1 + D_2 - gO)$. Indeed, by setting $D_3 = D_1 + D_2 - gO + (f)$, the divisor $D_3 - gO$ is linearly equivalent to $(D_1 - gO) + (D_2 - gO)$.

State of the art and related works. In this paper, we focus on the classical geometric approach attributed to Brill and Noether for computing Riemann-Roch

Received by the editor May 16, 2019, and, in revised form, October 8, 2019, and December 6, 2019.

2010 *Mathematics Subject Classification.* Primary 14Q05, 68W30.

spaces. The general algorithmic setting for this approach is described by Goppa in his landmark paper [11, §4]. Given a divisor D on a (not necessarily plane) smooth projective curve $\mathcal{C}$, this method proceeds by finding first a common denominator to all the functions in the Riemann-Roch space $L(D)$. This is done by computing a form h on the curve such that the associated principal effective divisor (h) satisfies $(h) \geq D$. Then the residual divisor $(h) - D$ is computed. From this, a basis of the Riemann-Roch space is found by computing the kernel of a linear map. The correctness of this method is ensured by the residue theorem of Brill and Noether, which works even in the presence of ordinary singularities by using the technique of adjoint curves; see [23, §42], [9, Sec. 8.1]. In its original version [11, §4], Goppa's algorithm works only for finite fields, and some parts of the algorithm use exhaustive search. During the 1990s, several versions of Goppa's algorithm have been proposed, incorporating tools of modern computer algebra. In particular, Huang and Ierardi provide in [15] a deterministic algorithm for computing Riemann-Roch spaces of plane curves $\mathcal{C}$ all singularities of which are ordinary within $O(\deg(\mathcal{C})^6 \deg(D_+)^6)$ arithmetic operations in the base field, where D_+ is the smallest effective divisor such that $D_+ \geq D$. In fact, writing $D_- = D_+ - D$, we can assume without loss of generality that $\deg(D_+) \geq \deg(D_-)$, since $L(D) = L(D_+ - D_-)$ is reduced to zero if $\deg(D) < 0$. Consequently, $\deg(D_+)$ is a relevant measure of the size of the divisor D. Haché [12] proposes the first implementation of Brill-Noether's approach in a computer algebra system, using local desingularizations to handle singularities encountered during the algorithm. For lines of research closely related to this topic, we refer to [13, 18] and the references therein.

A few years later, a breakthrough was achieved by Hess [14]: He provides an arithmetic approach to the Riemann-Roch problem, using fast algorithms for algebraic function fields. Hess' algorithm is now considered as a reference method for computing Riemann-Roch spaces, and it is proved to be polynomial in the input size [14, Remark 6.2].

An important special case of the computation of Riemann-Roch spaces is the computation of the group law on Jacobians of curves. Volcheck [27] describes an algorithm with complexity $O(\max(\deg(\mathcal{C}), g)^7)$ in this context. The best known complexity for computing the group law on Jacobians of general curves is currently achieved by Khuri-Makdisi in [17], where he gives an algorithm which requires $O(g^{\omega+\varepsilon})$ operations in the base field, where ω is a feasible exponent for matrix multiplication and ε is any fixed positive number. Actually, an anonymous reviewer informed us that the ε in this complexity can be removed if the cardinality of the base field grows polynomially in g, which is the case in this paper.

Main results. We propose a probabilistic algorithm for computing Riemann-Roch spaces on plane nodal projective curves $\mathcal{C} \subset \mathbb{P}^2$ defined over sufficiently large perfect fields. We emphasize that any algebraic curve admits such a nodal model up to a birational map if the base field is sufficiently large (see, e.g., [1, Appendix A]), and that computing such a model depends only on the curve and not on the input divisor.

Our main result is that the complexity of the algorithm for computing Riemann-Roch spaces is bounded by $O(\max(\deg(\mathcal{C})^{2\omega}, \deg(D_+)^{\omega}))$ and that, provided that some mild assumptions are satisfied, its failure probability is bounded above by $O(\max(\deg(\mathcal{C})^4, \deg(D_+)^2)/|\mathcal{E}|)$, where $\mathcal{E}$ is a finite subset of the base field k in which we can draw elements uniformly at random. Roughly speaking, these assumptions on the input require that the impact of the singularities during the execution of the algorithm is minimal. In particular, they are always satisfied for smooth curves. If these mild assumptions are not satisfied, then the algorithm always fail. Therefore, we provide at the end of Section 7 a Las Vegas procedure (in the sense of [3, Sec. 0.1]) with complexity $O(\max(\deg(\mathcal{C})^5, \deg(D_+)^{5/2}))$ and probability of failure bounded by $O(\max(\deg(\mathcal{C})^6, \deg(D_+)^3)/|\mathcal{E}|)$ to decide whether these assumptions are satisfied. Combining this verification procedure with our main algorithm turns it into a complete Las Vegas method, at the cost of increasing slightly the complexity and the probability of failure.

We also emphasize that our algorithm is geared towards curves defined over sufficiently large fields k, so that the probability of failure can be made small by choosing a large subset $\mathcal{E} \subset k$. A possible workaround to decrease the probability of failure for curves defined over small finite fields is to do the computations in a field extension, although doing so induces an extra arithmetic cost.

Up to our knowledge, the complexity that we obtain is the best bound for the general problem of computing Riemann-Roch spaces. In the special case of the group law on the Jacobian of plane smooth curves where $\deg(D_+) = O(g)$ and $\deg(\mathcal{C}) = O(\sqrt{g})$ by the genus-degree formula, the complexity becomes $O(g^\omega)$ which equals the best known complexity bound of Khuri-Makdisi's algorithm. Moreover, the algorithm that we propose requires very few assumptions, and its efficiency relies on classical building blocks in modern computer algebra: fast arithmetic of univariate polynomials and fast linear algebra. Consequently, it can be easily made practical by using existing implementations of these building blocks. We have made a C++/NTL implementation of our algorithm which is freely distributed under LGPL-2.1+ license and which is available at `https://gitlab.inria.fr/pspaenle/rrspace`. We also provide experimental data which seem to indicate that our prototype software is competitive with the reference implementation in the Magma computer algebra system [4].

Organization of the paper. Section 2 provides an overview of the main algorithm. Section 3 focuses on the data structures used to represent effective divisors. Algorithms to perform additions and subtractions of divisors with this representation are described in Section 4. Then Section 5 gives the details of the subroutines used in the main algorithm, and their correctness is proved. Section 6 focuses on the complexity of the subroutines and of the main algorithm. Then Section 7 is devoted to the analysis of the failure probability. Finally, Section 8 presents experimental results obtained with our NTL/C++ implementation.

2. Overview of the algorithm

This section is devoted to the description of the general setting of Brill-Noether's method and of the algorithm that we propose, without giving yet all the details on the data structures that we use to represent mathematical objects.

Throughout this paper, k is a perfect field and $\mathcal{C} \subset \mathbb{P}^2$ is an absolutely irreducible projective nodal curve defined over k with r nodes. By nodal curve, we mean that all the singularities of the curve have order 2 and are ordinary. We do not need any assumption about the k-rationality of the slopes of the tangents at the nodes. We emphasize that every algebraic curve admits such a model (up to a field extension if k is a small finite field), which can be for instance obtained by computing the image of a nonsingular projective model of the curve by a generic linear projection to $\mathbb{P}^2$ [1, Appendix A]. We let $\overline{k}$ denote an algebraic closure of k. Also, we use the notation $\widetilde{\mathcal{C}}$ to denote a nonsingular model of $\mathcal{C}$ which projects onto $\mathcal{C}$ (as denoted by X in [9, Ch. 8]). We assume that this implicit projection $\widetilde{\mathcal{C}} \to \mathcal{C}$ is one-to-one on nonsingular points of $\mathcal{C}$ and that it is two-to-one on nodes. By divisor, we always mean a *Weil divisor* on the curve $\widetilde{\mathcal{C}}$, i.e., a formal sum with integer coefficients of closed points of $\widetilde{\mathcal{C}}$. When the support of a divisor D involves only points of $\widetilde{\mathcal{C}}$ which project to nonsingular points of $\mathcal{C}$, we call D a *smooth divisor* of $\mathcal{C}$ by slight abuse of terminology. More generally, we will often identify nonsingular closed points of $\mathcal{C}$ with their corresponding points on $\widetilde{C}$. We will use frequently the *nodal divisor*, denoted by E, which is the effective divisor of degree $2r$ which is the sum of all the closed points of $\widetilde{\mathcal{C}}$ which project to a node of $\mathcal{C}$.

Naming X, Y, Z homogeneous coordinates for $\mathbb{P}^2$, the curve $\mathcal{C} \subset \mathbb{P}^2$ is described by a homogeneous polynomial $Q \in k[X,Y,Z]$ and we let $k[\mathcal{C}] = k[X,Y,Z]/Q(X,Y,Z)$ denote its homogeneous coordinate ring.

Assuming (w.l.o.g. up to linear change of coordinate) that $Q \neq Z$, we let $\mathcal{C}^0 \subset \mathbb{A}^n$ be the affine curve obtained by intersecting $\mathcal{C}$ with the open subset $\{Z \neq 0\} \subset \mathbb{P}^2$. It is described by the bivariate polynomial $q(X,Y) = Q(X,Y,1)$. Closed points of $\mathcal{C}^0$ correspond to maximal ideals in $k[\mathcal{C}^0] = k[X,Y]/q(X,Y)$. We assume (again w.l.o.g.) that all the nodes of the curve belong to its affine subset $\mathcal{C}^0$.

We shall also require that all the divisors that we consider are defined over k, i.e., that they are invariant under the natural action of the Galois group $\mathrm{Gal}(K/k)$ for any extension K of k. In this setting, smooth effective divisors on $\mathcal{C}^0$ can be thought of as nonzero ideals I in $k[\mathcal{C}^0]$ such that $I + \langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$. For two divisors D, D' on $\mathcal{C}$, we write $D \leq D'$ if the valuation of D at any place of $k(\mathcal{C})$ is at most the valuation of D'. If $g \in k[\mathcal{C}]$ is a nonzero form on $\mathcal{C}$, then we let (g) denote the associated effective principal divisor, as defined in [9, Sec. 8.1]. If $g \in k[\mathcal{C}^0]$ is a nonzero regular function on $\mathcal{C}^0$, then by abuse of notation, we overload the notation (g) to denote the effective divisor associated to the form $Z^{\deg(g)} g(X/Z, Y/Z, 1)$. We emphasize that this divisor has no pole and that it is not the principal divisor associated to the function $g(X/Z, Y/Z, 1) \in k(\mathcal{C})$.

If $f \in k(\mathcal{C})$ is a nonzero function on $\mathcal{C}$, i.e., a quotient $f = g/h$ of two nonzero forms $g, h \in k[\mathcal{C}]$ of the same degree, then again by abuse of notation we let (f) denote the associated degree-0 principal divisor. Finally, for a divisor D we let $L(D) = \{f \in k(\mathcal{C}) \setminus \{0\} \mid (f) \geq -D\} \cup \{0\}$ denote the Riemann-Roch space associated to D.

Assumptions on the input divisor. If the curve $\mathcal{C}$ is singular, then we need two mild assumptions on the input divisor D to ensure that our algorithm does not always fail. First, the divisor D should be smooth, and its support should be contained in the affine chart $\mathcal{C}^0$. To describe the second assumption—which is more technical—we need some insight on the data structure that we will use: The input divisor D will be given as a pair of effective divisors (D_+, D_-) such that $D = D_+ - D_-$. Set

$$d = \begin{cases} \lfloor \deg(D_+ + E)/\deg(\mathcal{C}) + (\deg(\mathcal{C}) - 1)/2 \rfloor & \text{if } \binom{\deg(\mathcal{C})+1}{2} \leq \deg(D_+ + E), \\ \lfloor (\sqrt{1 + 8\deg(D_+ + E)} - 1)/2 \rfloor & \text{otherwise.} \end{cases}$$

We will see in what follows that this value of d is in fact the smallest integer which ensures the existence of a nonzero form $h \in \overline{k}[\mathcal{C}]$ of degree d such that $(h) \geq D_+ + E$. Our second assumption is that there exists a form h of degree d such that $(h) \geq D_+ + E$ and $(h) - E$ is a smooth divisor. This is a mild assumption which is satisfied in most cases. In the rare cases where it is not satisfied, a workaround for practical computations—for which we do not prove any theoretical guarantee of success—is to increase slightly the value of d in Algorithm 6 (Interpolate) in order to increase the dimension of the space of such functions h.

Algorithm 1: A bird's eye view of the algorithm.

Function RiemannRochBasis;
Data: A curve $\mathcal{C}$ together with its nodal divisor E, and a divisor $D = D_+ - D_-$ on $\mathcal{C}$ such that D_+ and D_- are smooth effective divisors.
Result: A basis of the Riemann-Roch space $L(D)$.
$h \leftarrow$ Interpolate($\deg(\mathcal{C})$, D_+, E);
$D_h \leftarrow$ CompPrincDiv($\mathcal{C}$, h, E);
$D_{\mathsf{res}} \leftarrow$ SubtractDivisors(D_h, D_+);
$D_{\mathsf{num}} \leftarrow$ AddDivisors(D_-, D_{res});
$B \leftarrow$ NumeratorBasis($\deg(\mathcal{C})$, D_{num}, $\deg(h)$, E);
Return $\{f/h \mid f \in B\}$.

Algorithm 1 gives a bird's eye view of our algorithm for computing Riemman-Roch spaces. We now describe briefly what is done at each step of the algorithm. The routine Interpolate takes as input an effective divisor D_+, and it returns a form h such that $(h) \geq D_+ + E$. Then, CompPrincDiv computes from h a convenient representation of the divisor $(h) - E$. The routines used to perform addition and subtraction of divisors—namely, AddDivisors and SubtractDivisors—will be described in Section 4. Then, NumeratorBasis takes as input the effective divisor D_{num} and the degree of h, and it returns a basis of the vector space of all forms $f \in k[\mathcal{C}]$ of degree $\deg(h)$ such that $(f) \geq D_{\mathsf{num}} + E$. Finally, we divide this basis by the common denominator h in order to obtain a basis of the Riemann-Roch space.

One of the cornerstones of the correctness of Algorithm 1 is the Brill-Noether's residue theorem. This theorem is one of the foundations of the theory of adjoint curves. In the case of nodal plane curves, an adjoint curve is just a curve which goes through all the nodes of $\mathcal{C}$, and E is the adjoint divisor as defined in [9, Sec. 8.1].

Proposition 1 ([9, Sec. 8.1])**.** *Let D, D' be two linearly equivalent effective divisors on $\mathcal{C}$. Let $h \in k[\mathcal{C}]$ be a form such that $(h) = D + E + A$ for some effective divisor A. Then there exists a form $h' \in k[\mathcal{C}]$ of the same degree as h such that $(h') = D' + E + A$.*

We can now prove the general correctness of the main algorithm, assuming that all the subroutines behave correctly.

Theorem 2. *If all the subroutines* Interpolate*,* CompPrincDiv*,* SubtractDivisors*,* AddDivisors*,* NumeratorBasis *are correct, then Algorithm* 1 *is correct: It returns a basis of the space $L(D)$.*

Proof. We first prove that there exists a basis of $L(D)$ such that any basis element f belongs to the vector space spanned by the output of Algorithm 1. To this end, we must prove that f can be written as g/h where h is the output of the subroutine INTERPOLATE and g belongs to the vector space spanned by the output of the subroutine NUMERATORBASIS. Proposition 1 with $D = D_+$, $D' = D_+ + (f)$ and h implies that there exists a form $g \in k[\mathcal{C}]$ such that $(g/h) = (f)$, where g has the same degree as h. Therefore, $f = \lambda g/h$ for some nonzero $\lambda \in k$. It remains to prove that g belongs to the vector space spanned by the output of NUMERATORBASIS. Since $f \in L(D)$, we must have $(g) = (f) + (h) \geq (h) - D = (D_h + E - D_+) + D_- = D_{\mathrm{res}} + D_- + E = D_{\mathrm{num}} + E$. But NUMERATORBASIS returns precisely a basis of the space of forms α of the same degree as h such that $(\alpha) \geq D_{\mathrm{num}} + E$.

Conversely, let f be a function returned by Algorithm 1. Then $f \cdot h$ belongs to B, and hence $(f \cdot h) = (f) + (h) \geq D_{\mathrm{num}} + E = (h) - D$. This implies that $(f) \geq -D$ and hence $f \in L(D)$. □

3. Data structures

Data structure for the curve $\mathcal{C}$. We represent the projective curve $\mathcal{C} \subset \mathbb{P}^2$ by its affine model $\mathcal{C}^0$ in the affine chart $Z \neq 0$ which is described by a bivariate polynomial $q \in k[X, Y]$. We assume that the degree of q in Y equals its total degree. This condition implies that $\mathcal{C}$ is in *projective Noether position* with respect to the projection on the line $Y = 0$, i.e., that the canonical map $k[X, Z] \to k[\mathcal{C}]$ is injective and that it defines an integral ring extension. This also implies that the map $k[X] \to k[\mathcal{C}^0]$ is an integral ring extension. We refer to [10, Sec. 3.1] for more details on the projective Noether position. We emphasize that the projective Noether position is achieved in generic coordinates. Hence this assumption does not lose any generality since it can be enforced by a harmless linear change of coordinates. More precisely, regarding a linear change of coordinate in $\mathbb{P}^2$ as a 3×3 matrix $M = (m_{ij})_{1 \leq i,j \leq 3}$, the invertible matrices which put the curve in projective Noether position are precisely the matrices the 9 coefficients of which do not make a polynomial $P(m_{11}, \ldots, m_{33})$ of degree $\deg(\mathcal{C}) + 3$ vanish. This polynomial is the product of $\det(M)$—which has degree 3—with the coefficient of $Y^{\deg(\mathcal{C})}$ in the new system of coordinates—which has degree $\deg(\mathcal{C})$. Using the Schwartz-Zippel lemma [22, Coro. 1], this implies that the probability that the curve is not in projective Noether position after a linear change of coordinates given by a random matrix whose entries are picked uniformly at random in a finite subset $\mathcal{E} \subset k$ is bounded above by $(\deg(\mathcal{C}) + 3)/|\mathcal{E}|$.

Data structure for forms. We will represent forms on $\mathcal{C}$—namely elements in $k[\mathcal{C}] = k[X, Y, Z]/(Z^{\deg(q)} q(X/Z, Y/Z))$—by their affine counterpart in the affine chart $Z = 1$. Consequently, we shall represent a form $g \in k[\mathcal{C}]$ as an element in $k[X, Y]/q(X, Y)$, given by a representative $\widetilde{g} \in k[X, Y]$ such that $\deg_Y(\widetilde{g}) < \deg(\mathcal{C})$, using the fact that q is monic in Y. This representation is not faithful since it does not encode what happens on the line $Z = 0$ at infinity. In order to encode the behavior on this line and obtain a faithful representation, it is enough to adjoin to $\widetilde{g}$ the degree d of the form g, since g is the class of $Z^d \widetilde{g}(X/Z, Y/Z)$ in $k[\mathcal{C}]$. In the rest of this paper, we do not mention this issue further and we often identify g with $\widetilde{g}$ by slight abuse of notation when the context is clear.

Data structure for smooth divisors on $\mathcal{C}$. For representing divisors which do not involve any node, we use a data structure strongly inspired by the Mumford representation for divisors on hyperelliptic curves and by representations of algebraic sets by primitive elements as in [6]. Our data structure requires a mild assumption on the divisor that we represent: None of the points in the support of the divisor should lie at infinity. In fact, this is not a strong restriction since all points can be brought to an affine chart via a projective change of coordinate. If one does not wish to change the coordinate system, another solution is to maintain three representations, one for each of the three canonical affine charts covering $\mathbb{P}^2$.

We shall represent a smooth divisor D as a pair of smooth effective divisors (D_+, D_-) such that $D = D_+ - D_-$. One crucial point for the representation of effective divisors is that the 0-dimensional algebraic set corresponding to the support (i.e., without considering the multiplicities) of an effective divisor D can be described by a finite étale algebra which is a quotient of $k[\mathcal{C}^0]$ by a nonzero ideal. This étale algebra is isomorphic to a quotient of a univariate polynomial ring if it admits a primitive element. Using primitive elements to represent 0-dimensional algebraic sets is a classical technique in computer algebra; see, e.g., [6, Sec. 2], [10, Sec. 3.2].

Lemma 3. *Let R be a finite étale k-algebra, i.e., a finite product of finite extensions of k. Let $z \in R$ be an element, and let m_z denote the multiplication by z in R, seen as a k-linear endomorphism. The following statements are equivalent:*

(1) *The element z generates R as a k-algebra.*
(2) *The elements $1, z, z^2, \dots, z^{\dim_k(R)-1}$ are linearly independent over k.*
(3) *The characteristic polynomial of m_z equals its minimal polynomial.*
(4) *The characteristic polynomial of m_z is squarefree.*

If z satisfies these four properties, then z is called a primitive element *for R.*

Proof. (2) $\Rightarrow$ (1): By definition, the element z generates R as a k-algebra if and only if its powers generate R as a k-vector space. (1) $\Rightarrow$ (2): Let n_0 be the smallest positive integer such that $1, z, z^2, \dots, z^{n_0}$ are linearly dependent. The integer n_0 must be finite since $\dim_k(R)$ is finite. Write $z^{n_0} = \sum_{i=0}^{n_0-1} a_i z^i$ for some $a_0, \dots, a_{n_0} \in k$. By multiplying this relation by z^{n-n_0} and by induction on n, we obtain that for any $n \geq n_0$, z^n belongs to the vector space generated by $1, z, \dots, z^{n_0-1}$. This implies that the algebra generated by z has dimension n_0 as a k-vector space. By (1), we obtain that $n_0 = \dim_k(R)$. (2) $\Rightarrow$ (3): By (2), the minimal polynomial of m_z has degree at least $\dim_k(R)$, and hence it equals its characteristic polynomial. (3) $\Rightarrow$ (2): The degree of the characteristic polynomial is $\dim_k(R)$. (3) $\Rightarrow$ (4): Let ξ be the squarefree part of the characteristic polynomial of m_z. By (3), $\xi(z)$ must be nilpotent in R. But the only nilpotent element in an étale algebra in 0, so ξ must be a multiple of the minimal polynomial of m_z. Hence, by (3), ξ is the characteristic polynomial of m_z. (4) $\Rightarrow$ (3): This is a consequence of the facts that the characteristic polynomial and the minimal polynomial have the same set of roots, and the minimal polynomial divides the characteristic polynomial. □

We are now ready to define the data structure that we will use to represent smooth effective divisors on the curve. A smooth effective divisor D on $\mathcal{C}$ supported on the affine chart $Z \neq 0$ will be represented as:

- A scalar $\lambda \in k$.

- Three univariate polynomials $\chi, u, v \in k[S]$, such that χ is monic, χ has degree $\deg(D)$, and u, v have degree at most $\deg(D) - 1$

such that

(Div-H1): $q(u(S), v(S)) \equiv 0 \bmod \chi(S)$;
(Div-H2): $\lambda u(S) + v(S) = S$;
(Div-H3): $\mathrm{GCD}(\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S)), \chi(S)) = 1$.

We call the data structure above a *primitive element representation*. An important ingredient of the primitive element representation is that **(Div-H3)** enables us to use Hensel's lemma to encode the multiplicites. More precisely, **(Div-H3)** implies that at each of the closed points in the support of the divisor, the element $\lambda(X - \bar{x}) + (Y - \bar{y})$ is a uniformizing element for the associated discrete valuation ring, where $\bar{x}, \bar{y}$ denote the classes of X, Y in the residue field.

Notice that this representation requires the existence of a primitive element of the form $\lambda X + Y$ which satisfies all the wanted properties. Fortunately, Proposition 4 below shows that such a primitive element exists as soon as k contains more than $\binom{\deg(D)+1}{2}$ elements.

Data structure for the nodal divisor. We shall represent the nodal divisor via an algebraic parametrization by the roots of a univariate polynomial. This algebraic structure is very similar to the representation of smooth divisors, except for a crucial difference: We shall not need to represent multiplicities, so this representation does not need to satisfy condition **(Div-H3)**. More precisely, the nodal divisor E will be represented as:

- A scalar $\lambda_E \in k$.
- Three univariate polynomials $\chi_E, u_E, v_E \in k[S]$, such that χ_E is monic and squarefree, χ_E has degree r, and u, v have degree at most $r - 1$.
- A monic univariate polynomial $T_E \in k[S]$ of degree at most $2r$

such that

$$(\textbf{NodDiv-H1}) \qquad \{(u_E(\zeta), v_E(\zeta)) \mid \zeta \in \overline{k}, \chi_E(\zeta) = 0\} \text{ is the set of nodes of } \mathcal{C}^0,$$

and such that the roots of T_E are the values $\lambda \in \overline{k}$ such that the vector $(1, -\lambda)$ is tangent to $\mathcal{C}^0$ at a node. Notice that the roots of T_E do not record vertical tangents at nodes, so the degree of T_E may be less than $2r$.

Such $(\lambda_E, \chi_E, u_E, v_E)$ satisfying (**NodDiv-H1**) exist as soon as k contains more than $\binom{r}{2}$ elements by Proposition 5. Computing T_E is also an easy task once $(\lambda_E, \chi_E, u_E, v_E)$ are known. This polynomial can be for instance obtained by considering the homogeneous form $Q_2(X, Y, S)$ of degree 2 of the shifted polynomial $q(X + u_E(S), Y + v_E(S))$. Then $T_E(\lambda) = \mathrm{Resultant}_S(Q_2(1, -\lambda, S), \chi_E(S))$ satisfies the desired property. This polynomial T_E will be useful in Algorithm CompPrincDiv. Computing $\lambda_E, \chi_E, u_E, v_E, T_E$ can be thought of as a precomputation since this depends only on the curve $\mathcal{C}$ and not on the input divisor D.

Before going any further, we summarize here the data structures for the input and the output of Algorithm 1 and the properties that they must satisfy.

Input data:

- A bivariate polynomial $q \in k[X, Y]$. This polynomial encodes the curve $\mathcal{C}$.
- Data $(\lambda_E, \chi_E, u_E, v_E, T_E)$ encoding the nodal divisor E.
- A smooth divisor $D = D_+ - D_-$ given by two tuples $(\lambda_+, \chi_+, u_+, v_+)$ and $(\lambda_-, \chi_-, u_-, v_-)$ with $\lambda_\pm \in k$, $\chi_\pm, u_\pm, v_\pm \in k[S]$.

The input data must satisfy the following constraints:

(1) The bivariate polynomial $q \in k[X,Y]$ is absolutely irreducible, and its base field k is perfect.
(2) The total degree of q equals its degree with respect to Y.
(3) The inequalities $\deg(u_\pm) < \deg(\chi_\pm)$, $\deg(v_\pm) < \deg(\chi_\pm)$, $\deg(u_E) < \deg(\chi_E)$, $\deg(v_E) < \deg(\chi_E)$ hold.
(4) The polynomials $\chi_\pm$, χ_E and T_E are monic.
(5) The polynomial χ_E is squarefree.
(6) Both tuples $(\lambda_+, \chi_+, u_+, v_+)$ and $(\lambda_-, \chi_-, u_-, v_-)$ satisfy **(Div-H1)** to **(Div-H3)**.
(7) The tuple $(\lambda_E, \chi_E, u_E, v_E, T_E)$ satisfies (**NodDiv-H1**).
(8) The degree of T_E is at most $2r$.
(9) The roots of the univariate polynomial T_E are the values $\lambda \in \overline{k}$ such that the vector $(1, -\lambda)$ is tangent to $\mathcal{C}^0$ at a node.

Output data:

- A bivariate polynomial $h \in k[X,Y]$.
- A finite set of bivariate polynomials $B \subset k[X,Y]$.

The output data satisfies that the set $\{b/h \mid b \in B\}$ is a basis of the Riemann-Roch space associated to D on $\mathcal{C}$.

The rest of this section is devoted to technical proofs about the primitive element representation. The statements below will be used for proving the correctness of the subalgorithms, but they may be skipped without harming the general understanding of this paper.

The two following propositions (whose proofs are postponed until after Lemma 8) show that primitive representations of smooth effective divisors and of the nodal divisor exist provided that the base field is large enough.

Proposition 4. *Let J be a nonzero ideal of $k[\mathcal{C}^0] = k[X,Y]/q(X,Y)$ such that $J + \langle \frac{\partial q}{\partial X}, \frac{\partial q}{\partial Y} \rangle = k[\mathcal{C}^0]$. Assume that the cardinality of k is larger than $\binom{\dim_k(k[\mathcal{C}^0]/J)+1}{2}$. Then there exist $\lambda \in k$ and polynomials $\chi, u, v \in k[S]$ satisfying* **(Div-H1)** *to* **(Div-H3)** *such that the map $k[\mathcal{C}^0]/J \to k[S]/\chi(S)$ sending X and Y to the classes of u and v is an isomorphism of k-algebras.*

The following proposition states a similar result for radical ideals which do not satisfy the smoothness assumption. This will be useful to represent the nodal divisor.

Proposition 5. *Let J be a nonzero radical ideal of $k[\mathcal{C}^0] = k[X,Y]/q(X,Y)$. Assume that the cardinality of k is larger than $\binom{\dim_k(k[\mathcal{C}^0]/J)}{2}$. Then there exist $\lambda_E \in k$ and polynomials $\chi_E, u_E, v_E \in k[S]$ satisfying* (**NodDiv-H1**) *such that the map $k[\mathcal{C}^0]/J \to k[S]/\chi(S)$ sending X and Y to the classes of u and v is an isomorphism of k-algebras.*

Before proving Propositions 4 and 5, we need some technical lemmas. First, the following lemma generalizes slightly the classical fact that ideals in the coordinate rings of smooth curves admit a unique factorization. Here, we do not assume that $\mathcal{C}^0$ is nonsingular, but the factorization property holds only for ideals of regular functions which do not vanish at any singular point.

Lemma 6. *Let I be a nonzero ideal of $k[\mathcal{C}^0]$ such that $I+\langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$. Then there exists a unique factorization $I = \prod_{i=1}^{\ell} \mathfrak{m}_i^{\alpha_i}$ as a product of maximal ideals of $k[\mathcal{C}^0]$.*

Proof. First, we prove the existence of such a factorization. Let $\mathfrak{m} \subset k[\mathcal{C}^0]$ be an ideal containing I. If $\mathfrak{m}$ is a nonsingular closed point of $\mathcal{C}^0$, then the local ring $k[\mathcal{C}^0]_{\mathfrak{m}}$ is a discrete valuation ring by [9, Sec. 3.2, Thm. 1]. Let $\mathrm{val}_{\mathfrak{m}}(I) \in \mathbb{Z}_{\geq 0}$ denote the integer such that $I = \mathfrak{m}^{\mathrm{val}_{\mathfrak{m}}(I)}$ in this local ring. Let J be the ideal $J = \prod_{\mathfrak{m}\supset I} \mathfrak{m}^{\mathrm{val}_{\mathfrak{m}}(I)}$. By [2, Prop. 9.1], the equality $I = J$ holds if and only if it holds in all the local rings $k[\mathcal{C}^0]_{\mathfrak{m}}$ where $\mathfrak{m} \supset I$. Since the maximal ideals $\mathfrak{m}$ are nonsingular closed points of $\mathcal{C}^0$, this equality holds true because the corresponding local rings are discrete valuation rings, hence the equality of ideals is equivalent to the equality of their $\mathfrak{m}$-valuation.

We now prove the unicity of this factorization: By contradiction, assume that I has two distinct factorizations $\prod_{1\leq i\leq \ell} \mathfrak{m}_i^{\alpha_i}$ and $\prod_{1\leq i\leq \ell'} \mathfrak{m}_i'^{\alpha_i'}$ of the ideal. Without loss of generality, assume that $\mathfrak{m}_1$ does not occur in the second factorization, or that it appears with a different multiplicity. This would lead to a contradiction since it would lead to distinct valuations of the same ideal in the local ring at $\mathfrak{m}_1$. $\square$

An ideal $I \subset k[X, Y]$ in a polynomial ring is called 0-dimensional if the dimension of $k[X, Y]/I$ as a k-vector space is finite. The following lemma identifies values of λ for which $\lambda X + Y$ is not a primitive element for a 0-dimensional algebraic set.

Lemma 7. *Let $I \subset k[X, Y]$ be a radical 0-dimensional ideal, with associated variety $V = \{\alpha_i\}_{1\leq i\leq \dim_k(k[X,Y]/I)} \subset \overline{k}^2$ and let $u, v \in k[X, Y]$ be elements such that for any distinct points $\alpha_i, \alpha_j \in V$, $u(\alpha_i) \neq u(\alpha_j)$ or $v(\alpha_i) \neq v(\alpha_j)$. Then the set of $\lambda \in k$ such that $\lambda u+v$ is not a primitive element for $k[X, Y]/I$ is contained in the set of roots of a nonzero univariate polynomial with coefficients in k of degree $\binom{\dim_k(k[X,Y]/I)}{2}$.*

Proof. Writing $\alpha_i = (\alpha_{i,x}, \alpha_{i,y})$, let $K \subset \overline{k}$ be a field extension of k where I factors as a product of degree-1 maximal ideals $\mathfrak{m}_i = \langle X - \alpha_{i,x}, Y - \alpha_{i,y}\rangle$. This provides an isomorphism of K-algebras between $K[X, Y]/I$ and $K^{\dim_k(k[X_1,X_2]/I)}$ sending polynomials to their evaluations at $\alpha_1, \ldots, \alpha_{\dim_k(k[X,Y]/I)}$. Using this isomorphism, and letting $\mathbf{e}_i$ denote the ith canonical vector in $K^{\dim_k(k[X_1,X_2]/I)}$, we observe that $\mathbf{e}_i$ is an eigenvector of the endomorphism of multiplication by $\lambda u+v$, with associated eigenvalue $\lambda u(\alpha_i) + v(\alpha_i)$. Next, $\lambda u + v$ is a primitive element for $k[X, Y]/I$ if all these eigenvalues are distinct by Lemma 3. This is the case if and only if the discriminant of the characteristic polynomial is nonzero. Since the discriminant is the product of the squared differences of the roots, it equals

$$\prod_{\substack{1\leq i\leq \dim_k(k[X,Y]/I)\\ 1\leq j<i}} (\lambda(u(\alpha_i) - u(\alpha_j)) + v(\alpha_i) - v(\alpha_j))^2.$$

This discriminant is a polynomial in $k[\lambda]$ of degree $2\binom{\dim_k(k[X,Y]/I)}{2}$. It is nonzero because of the assumption that for any distinct α_i, α_j, either $u(\alpha_i) \neq u(\alpha_j)$ or $v(\alpha_i) \neq v(\alpha_j)$. Let $\Delta(\lambda) = \prod_{1\leq j<i}(\lambda(u(\alpha_i) - u(\alpha_j)) + v(\alpha_i) - v(\alpha_j)) \in \overline{k}[\lambda]$ be its squareroot. It remains to prove that the polynomial Δ has coefficients in k. This is due to the fact that automorphisms in $\mathrm{Gal}(K/k)$ permute the points α_i. Therefore

the natural action of $\mathrm{Gal}(K/k)$ acts on Δ by permuting its linear factors, and hence it leaves Δ invariant. $\square$

In the following lemma, the notation $\mathrm{red}(R)$ stands for the quotient of a ring R by its Jacobson radical (i.e., the intersection of its maximal ideals). If I is an ideal of R, we use the notation $\sqrt{I}$ to denote the radical of I. The ring $\mathrm{red}(k[\mathcal{C}^0]/J)$ can be thought of as the coordinate ring of the 0-dimensional algebraic set corresponding to the points in the support of the effective divisor associated to J.

Lemma 8. *Let J be a nonzero ideal of $k[\mathcal{C}^0] = k[X,Y]/q(X,Y)$ such that $J + \langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$. Then there exists a nonzero univariate polynomial Δ with coefficients in k of degree at most $\binom{\dim_k(k[\mathcal{C}^0]/J)+1}{2}$, such that for any λ which is not a root of Δ, the element $\lambda X + Y$ is primitive for $\mathrm{red}(k[\mathcal{C}^0]/J)$ and $\partial q/\partial X - \lambda \partial q/\partial Y$ is invertible in $\mathrm{red}(k[\mathcal{C}^0]/J)$.*

In particular, if k has cardinality larger than $\binom{\dim_k(k[\mathcal{C}^0]/J)+1}{2}$, then there exists a value of λ in k which is not a root of Δ.

Proof. First, notice that $k[\mathcal{C}^0]/J$ is isomorphic to $k[X,Y]/(J + \langle q\rangle)$, by using the classical fact that ideals of a quotient ring R/I correspond to ideals of R containing I. Since q is irreducible, $J + \langle q\rangle \subset k[X,Y]$ is a 0-dimensional ideal, and hence $\mathrm{red}(k[X,Y]/(J+\langle q\rangle)) = k[X,Y]/\sqrt{J+\langle q\rangle}$. Notice that $\dim_k(k[X,Y]/\sqrt{J+\langle q\rangle}) \leq \dim_k(k[\mathcal{C}^0]/J)$. Next, using the fact that two distinct points in the variety have distinct coordinates, Lemma 7 provides a nonzero polynomial Δ_0 of degree at most $\binom{\dim_k(k[\mathcal{C}^0]/J)}{2}$ such that $\lambda X + Y$ is not a primitive element for $\mathrm{red}(k[\mathcal{C}^0]/J)$ only if λ is a root of Δ_0.

Next, we notice that since $\sqrt{J+\langle q\rangle} \subset k[X,Y]$ is a radical 0-dimensional ideal, it can be decomposed as a product $\prod_{1\leq i\leq \ell} \mathfrak{m}_i$ of maximal ideals. Consequently, $\partial q/\partial X - \lambda\partial q/\partial Y$ is invertible in $\mathrm{red}(k[\mathcal{C}^0]/J)$ if and only if $\partial q/\partial X - \lambda \partial q/\partial Y$ does not belong to any of these maximal ideals. Equivalently, $\partial q/\partial X - \lambda\partial q/\partial Y$ must not vanish in any of the residue fields $\kappa_i = k[\mathcal{C}^0]/\mathfrak{m}_i$. Notice that the norm $N_{\kappa_i/k}(\partial q/\partial X - \lambda \partial q/\partial Y)$ is a polynomial Δ_i in λ with coefficients in k. It is nonzero since $J + \langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$ and hence either $\partial q/\partial X$ or $\partial q/\partial Y$ is nonzero in κ_i. Therefore Δ_i is either constant (if $\partial q/\partial Y$ vanishes in κ_i), or it has degree $[\kappa_i : k]$. Finally, the proof is concluded by noticing that $\sum_{1\leq i\leq \ell}[\kappa_i : k] = \dim_k(k[X,Y]/\sqrt{J+\langle q\rangle}) \leq \dim_k(k[\mathcal{C}^0]/J)$, so that the product $\Delta_0 \cdot \prod_{1\leq i\leq \ell}\Delta_i$ has degree at most $\binom{\dim_k(k[\mathcal{C}^0]/J)+1}{2}$ and satisfies all the desired properties. $\square$

We now have all the tools that we need to prove Propositions 4 and 5.

Proof of Proposition 4. First, we assume that $J = \mathfrak{m}^\alpha$ is a power of a maximal ideal in $k[\mathcal{C}^0]$ such that $\mathfrak{m} + \langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$. Then $\mathrm{red}(k[\mathcal{C}^0]/J) = k[\mathcal{C}^0]/\mathfrak{m}$. Let $\lambda \in k$ be an element which is not a root of the polynomial Δ provided by Lemma 8. Such an element exists since the cardinality of k is larger than the degree of Δ. Therefore, $\lambda X + Y$ is a primitive element for $k[\mathcal{C}^0]/\mathfrak{m}$ and hence there exist univariate polynomials $\widetilde{u}, \widetilde{v} \in k[S]$ such that $X = \widetilde{u}(\lambda X + Y)$ and $Y = \widetilde{v}(\lambda X + Y)$ in $k[\mathcal{C}^0]/\mathfrak{m}$. Let $\widetilde{\chi}(S)$ be the minimal polynomial of $\lambda X + Y$ in $k[\mathcal{C}^0]/\mathfrak{m}$, which is irreducible since $k[\mathcal{C}^0]/\mathfrak{m}$ is a field. Notice that the map $k[\mathcal{C}^0]/\mathfrak{m} \to k[S]/\widetilde{\chi}(S)$ sending the classes of X, Y to $\widetilde{u}, \widetilde{v}$ is an isomorphism of k-algebras. Next, set

$\chi(S) = \widetilde{\chi}(S)^\alpha$ and consider the bivariate system

$$\begin{cases} q(X,Y) = 0, \\ \lambda X + Y - S = 0. \end{cases} \tag{1}$$

By construction, this system has solution $(\widetilde{u}, \widetilde{v})$ over $k[S]/\widetilde{\chi}(S)$. The Jacobian of this system is $\frac{\partial q}{\partial X}(X,Y) - \lambda \frac{\partial q}{\partial Y}(X,Y)$, which is invertible in $\mathrm{red}(k[\mathcal{C}^0]/J)$ by Lemma 8, and therefore $\frac{\partial q}{\partial X}(\widetilde{u}(S), \widetilde{v}(S)) - \lambda \frac{\partial q}{\partial Y}(\widetilde{u}(S), \widetilde{v}(S))$ is invertible in $k[S]/\widetilde{\chi}(S)$. By Hensel's lemma, there exist polynomials $u, v \in k[S]_{<\deg(\chi)}$ which are solutions of (1) over $k[S]/\chi(S)$: Indeed, for $i > 1$, if $(\widehat{u}, \widehat{v})$ is a solution of (1) over $k[S]/\widetilde{\chi}(S)^i$, then a Taylor expansion of the system at order 1 shows that

$$\begin{bmatrix} \widehat{u} \\ \widehat{v} \end{bmatrix} - \begin{bmatrix} \partial q/\partial X & \partial q/\partial Y \\ \lambda & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} q(\widehat{u}, \widehat{v}) \\ \lambda \widehat{u} + \widehat{v} - S \end{bmatrix}$$

is a solution of equation (1) over $k[S]/\widetilde{\chi}(S)^{2i}$.

The map $k[\mathcal{C}^0]/J \to k[S]/\chi(S)$ is well-defined because $\mathfrak{m}$ maps to 0 modulo $\widetilde{\chi}$ and hence $J = \mathfrak{m}^\alpha$ maps to 0 modulo $\chi = \widetilde{\chi}^\alpha$. It is an isomorphism because $k[\mathcal{C}^0]/J$ and $k[S]/\chi(S)$ have the same dimension as vector spaces over k and the map $S \mapsto \lambda X + Y$ is the right inverse to the map $(X,Y) \mapsto (u(S), v(S))$. It remains to prove that **(Div-H3)** is satisfied by χ, u, v, which is a direct consequence of the fact that by Lemma 8, $\partial q/\partial X - \lambda \partial q/\partial Y$ does not belong to $\mathfrak{m}$ and hence $\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S))$ is invertible modulo $\chi(S)$.

Next, we consider the general case where J is a nonzero ideal in $k[\mathcal{C}^0]$ such that $J + \langle \partial q/\partial X, \partial q/\partial Y \rangle = k[\mathcal{C}^0]$. Again, let $\lambda \in k$ be an element which is not a root of the polynomial Δ provided by Lemma 8. Lemma 6 implies that J can be written as a product $J = \prod_{i=1}^{\ell} \mathfrak{m}_i^{\alpha_i}$ of powers of maximal ideals. Then for all i, the element $\lambda X + Y$ is primitive for $k[\mathcal{C}^0]/\mathfrak{m}$ and $\partial q/\partial X - \lambda \partial q/\partial Y$ is invertible in $k[\mathcal{C}^0]/\mathfrak{m}$. For each i, using the previous argument, we can construct univariate polynomials $\chi_i, u_i, v_i \in k[S]$ satisfying **(Div-H1)** to **(Div-H3)** with respect to λ such that the maps $k[\mathcal{C}^0]/\mathfrak{m}_i^{\alpha_i} \to k[S]/\chi_i(S)$ sending X, Y to $u_i(S), v_i(S)$ are isomorphisms of k-algebras. Setting $\chi(S) = \prod_{i=1}^{\ell} \chi_i(S)$ and using the CRT, let $u, v \in k[S] < \deg(\chi)$ be such that for all i, we have $u(S) \equiv u_i(S) \bmod \chi_i(S)$ and $v(S) \equiv v_i(S) \bmod \chi_i(S)$. Then the fact that the CRT is a ring morphism allows us to conclude that the map $k[\mathcal{C}^0]/J \to k[S]/\chi(S)$ is an isomorphism and that χ, u, v satisfy **(Div-H1)** to **(Div-H3)**. □

Proof of Proposition 5. The proof is similar to that of Proposition 4, by ignoring the argument about multiplicities. Since the Jacobian $\frac{\partial q}{\partial X}(X,Y) - \lambda \frac{\partial q}{\partial Y}(X,Y)$ need not be invertible in $\mathrm{red}(k[\mathcal{C}^0]/J)$, it is sufficient to choose a value of λ which is not a root of the univariate polynomial constructed in Lemma 7 for J. □

The next lemma shows that any data satisfying **(Div-H1)** to **(Div-H3)** actually encodes a well-defined effective divisor with no singular point in its support.

Lemma 9. *Let (λ, χ, u, v) be such that* **(Div-H1)** *to* **(Div-H3)** *are satisfied, and let $I = \langle X - u(S), Y - v(S), \chi(S) \rangle \cap k[X,Y]$. Then $k[X,Y]/I$ is isomorphic as a k-algebra to $k[\mathcal{C}^0]/J$ where J is a nonzero ideal in $k[\mathcal{C}^0]$. Moreover, $J + \langle \partial q/\partial X, \partial q/\partial Y \rangle = k[\mathcal{C}^0]$, $\lambda X + Y$ is a primitive element for* $\mathrm{red}(k[X,Y]/I)$*, and its minimal polynomial is the squarefree part of χ.*

Proof. Nonzero ideals of $k[\mathcal{C}^0]$ correspond to ideals of $k[X,Y]$ containing properly the principal ideal $\langle q(X,Y)\rangle$. First, notice that **(Div-H1)** implies that $q(X,Y) \in I$. Also, by **(Div-H2)**, we get that $\chi(\lambda X + Y) \in I$. Notice that $\chi(\lambda X + Y)$ factors as a product of polynomials of degree 1 over the algebraic closure of k. Since q is supposed to be absolutely irreducible and to have degree at least 2, this implies that $\chi(\lambda X + Y)$ does not belong to the principal ideal $\langle q(X,Y)\rangle$. Consequently, I contains properly $\langle q(X,Y)\rangle$ and this proves the isomorphism between $k[X,Y]/I$ and $k[\mathcal{C}^0]/J$. In particular, we obtain that $\dim_k(k[\mathcal{C}^0]/J) = \dim_k(k[X,Y]/I) = \deg(\chi)$. Next, **(Div-H3)** implies that $\partial q/\partial X - \lambda \partial q/\partial Y$ is invertible in $k[\mathcal{C}^0]/J$, and hence $k[\mathcal{C}^0] = J + \langle \partial q/\partial X - \lambda \partial q/\partial Y\rangle \subset J + \langle \partial q/\partial X, \partial q/\partial Y\rangle$. Therefore, $J + \langle \partial q/\partial X, \partial q/\partial Y\rangle = k[\mathcal{C}^0]$. Using the isomorphism between $k[\mathcal{C}^0]/J$ and $k[S]/\chi(S)$ in Proposition 4, we obtain that $\mathrm{red}(k[X,Y]/I)$ is isomorphic to $\mathrm{red}(k[S]/\chi(S))$, which is in turn isomorphic to $k[S]/\widetilde{\chi}(S)$, where $\widetilde{\chi}(S)$ is the squarefree part of $\chi(S)$. Finally, the proof is concluded by noticing that S is a primitive element for $k[S]/\widetilde{\chi}(S)$ with minimal polynomial $\widetilde{\chi}(S)$. $\square$

The following lemma explicits the link between the primitive element representation and the ideal vanishing on the 0-dimensional algebraic set that it represents.

Lemma 10. *Let (λ, χ, u, v) be data satisfying* **(Div-H2)**. *Set $I = \langle \chi(S), X - u(S), Y - v(S)\rangle \subset k[X,Y,S]$ and $J = \langle \chi(\lambda X+Y), X - u(\lambda X+Y), Y - v(\lambda X+Y)\rangle$. Then $I \cap k[X,Y] = J$.*

Proof. By **(Div-H2)** and by using the fact that $X - u(S), Y - v(S) \in I$, we deduce that $S - (\lambda X + Y) \in I$. This implies that $I \cap k[X,Y] = \{f(X,Y,\lambda X+Y) \mid f \in I\}$. $\square$

The primitive element representation of an effective divisor is not unique: Two tuples $(\lambda_1, \chi_1, u_1, v_1)$ and $(\lambda_2, \chi_2, u_2, v_2)$ may encode the same effective divisor. The cases where this happens are detailed in the following proposition.

Proposition 11. *Let $(\lambda_1, \chi_1, u_1, v_1), (\lambda_2, \chi_2, u_2, v_2) \in k \times k[S]^3$ be data which satisfy* **(Div-H2)**. *Let $I_1, I_2 \subset k[X,Y,S]$ be the associated ideals $I_1 = \langle \chi_1(S), X - u_1(S), Y - v_1(S)\rangle, I_2 = \langle \chi_2(S), X - u_2(S), Y - v_2(S)\rangle$.*

Then $I_1 \cap k[X,Y] = I_2 \cap k[X,Y]$ if and only if χ_1 is the characteristic polynomial of $\lambda_1 u_2 + v_2$ in $k[S]/\chi_2(S)$, $u_1(\lambda_1 u_2(S) + v_2(S)) \equiv u_2(S) \bmod \chi_2(S)$, and $v_1(\lambda_1 u_2(S) + v_2(S)) \equiv v_2(S) \bmod \chi_2(S)$.

Proof. We first prove the "if" part of the statement. Notice that $k[X,Y]/(I_1 \cap k[X,Y])$ and $k[X,Y]/(I_2 \cap k[X,Y])$ are k-vector spaces of the same finite dimension, since $\deg(\chi_1)$ must equal $\deg(\chi_2)$. Therefore it is enough to show one inclusion to prove the equality. Let $f(X,Y) \in I_1 \cap k[X,Y]$. Using the equalities modulo χ_2, we obtain that $f(X,Y) \equiv f(u_1(\lambda_1 u_2(S) + v_2(S)), v_1(\lambda_1 u_2(S) + v_2(S))) \bmod I_2$, which is divisible by $\chi_1(\lambda_1 u_2(S) + v_2(S))$ because f is in I_1 and by using the Cayley-Hamilton theorem. Finally, we use the fact that $\chi_1(S)$ is the characteristic polynomial of $\lambda_1 u_2(S) + v_2(S)$ and hence χ_2 divides $\chi_1(\lambda_1 u_2(S) + v_2(S))$, which finishes to prove that $f \in I_2$.

Conversely, assume that $I_1 \cap k[X,Y] = I_2 \cap k[X,Y]$. By composing the isomorphisms

$$\begin{array}{ccc} k[S]/\chi_1(S) & \to & k[X,Y]/(I_1 \cap k[X,Y]), \\ S & \mapsto & \lambda_1 X + Y \end{array} \quad \begin{array}{ccc} k[X,Y]/(I_2 \cap k[X,Y]) & \to & k[S]/\chi_2(S) \\ X & \mapsto & u_2(S) \\ Y & \mapsto & v_2(S) \end{array}$$

we obtain that the map $k[S]/\chi_1(S) \to k[S]/\chi_2(S)$ which sends S to $\lambda_1 u_2(S) + v_2(S)$ is an isomorphism. This proves that χ_1 is the characteristic polynomial of $\lambda_1 u_2(S) + v_2(S)$ in $k[S]/\chi_2(S)$. To prove the two congruence relations, we observe that for all $f \in k[X,Y]$, $f(u_1,v_1) \equiv 0 \bmod \chi_1$ if and only if $f(u_2,v_2) \equiv 0 \bmod \chi_2$. In particular, the polynomial $P(X,Y) = u_1(\lambda_1 X + Y) - X$ satisfies $P(u_1,v_1) \equiv 0 \bmod \chi_1$, and hence $P(u_2,v_2) \equiv 0 \bmod \chi_2$. The proof of the last congruence relation is similar. □

4. Divisor arithmetic for smooth divisors

The first step to perform arithmetic operations on smooth divisors given by primitive element representations is to agree on a common primitive element. In order to achieve this, the routine ChangePrimElt (Algorithm 2) performs the necessary change of primitive element by using linear algebra. We will prove in Propositions 21 and 27 that the complexity of this step is the same as the complexity of the subroutine NumeratorBasis in the main algorithm. Therefore, decreasing the complexity of ChangePrimElt would not change the global complexity and hence we make no effort to optimize it, although it might be possible to obtain a better complexity for this step by using a method similar to [10, Algo. 5].

Throughout this paper, for $d > 0$ we let $k[S]_{<d}$ denote the vector space of univariate polynomials with coefficients in k of degree less than d.

Algorithm 2: Changing the primitive element in the representation of a smooth effective divisor.

Function ChangePrimElt;

Data: A scalar $\widetilde{\lambda} \in k$ and a primitive element representation (λ, χ, u, v) of a smooth effective divisor D.

Result: Univariate polynomials $(\widetilde{\chi}, \widetilde{u}, \widetilde{v})$ such that $(\widetilde{\lambda}, \widetilde{\chi}, \widetilde{u}, \widetilde{v})$ is a primitive element representation of D or "fail".

if $\mathrm{GCD}(\frac{\partial q}{\partial X}(u(S),v(S)) - \widetilde{\lambda}\frac{\partial q}{\partial Y}(u(S),v(S)), \chi(S)) \neq 1$ **then**
 ⌊ Return "fail".
$M \leftarrow \deg(\chi) \times \deg(\chi)$ matrix representing the linear map
 $\varphi : k[S]_{<\deg(\chi)} \to k[S]_{<\deg(\chi)}$ such that
 $\varphi(f)(S) \equiv f(S) \cdot (\widetilde{\lambda} u(S) + v(S)) \bmod \chi(S)$;
$\widetilde{\chi} \leftarrow$ CharacteristicPolynomial(M);
$N \leftarrow \deg(\chi) \times \deg(\chi)$ invertible matrix representing the linear map
 $\psi : k[S]_{<\deg(\widetilde{\chi})} \to k[S]_{<\deg(\chi)}$ such that
 $\psi(f)(S) \equiv f(\widetilde{\lambda} u(S) + v(S)) \bmod \chi(S)$;
if *N is not invertible* **then**
 ⌊ Return "fail".
$\widetilde{u} \leftarrow \psi^{-1}(u)$;
$\widetilde{v} \leftarrow \psi^{-1}(v)$;
Return $(\widetilde{\chi}, \widetilde{u}, \widetilde{v})$.

Proposition 12. *Algorithm* 2 (ChangePrimElt) *is correct: If it does not fail, then $(\widetilde{\lambda}, \widetilde{\chi}, \widetilde{u}, \widetilde{v})$ satisfies properties* **(Div-H1)** *to* **(Div-H3)** *and it represents the same effective divisor as (λ, χ, u, v).*

Proof. First, we prove that $(\widetilde{\lambda}, \widetilde{\chi}, \widetilde{u}, \widetilde{v})$ satisfies Properties **(Div-H1)** to **(Div-H3)**. We notice that the map ψ in Algorithm 2 can be extended to an isomorphism Ψ of k-algebras between $k[S]/\widetilde{\chi}(S)$ and $k[S]/\chi(S)$. Property **(Div-H1)** follows from the fact that in $k[S]/\chi(S)$, we have $q(\widetilde{u}, \widetilde{v}) = q(\Psi^{-1}(u), \Psi^{-1}(v)) = \Psi^{-1}(q(u, v)) = 0$. Property **(Div-H2)** follows from the equalities $S = \Psi^{-1}(\Psi(S)) = \Psi^{-1}(\widetilde{\lambda} u + v) = \widetilde{\lambda}\, \psi^{-1}(u(S)) + \psi^{-1}(v(S)) = \widetilde{\lambda}\, \widetilde{u}(S) + \widetilde{v}(S)$ in $k[S]/\widetilde{\chi}(S)$. The fact that the equality $\widetilde{\lambda}\widetilde{u}(S) + \widetilde{v}(S) = S$ also holds in $k[S]$ is a consequence of the degree bounds $\deg(\widetilde{u}), \deg(\widetilde{v}) < \deg(\widetilde{\chi})$. If the first test does not fail, then $\frac{\partial q}{\partial X}(u(S), v(S)) - \widetilde{\lambda} \frac{\partial q}{\partial Y}(u(S), v(S))$ is invertible modulo $\chi(S)$. Applying Ψ^{-1} shows **(Div-H3)**.

Finally, we must prove that both representations encode the same divisor. By Proposition 11, this amounts to showing that

$$\begin{cases} \widetilde{\chi} \text{ is the characteristic polynomial of } \widetilde{\lambda} u(S) + v(S), \\ \widetilde{u}(\widetilde{\lambda} u(S) + v(S)) \equiv u(S) \bmod \chi(S), \text{ and} \\ \widetilde{v}(\widetilde{\lambda} u(S) + v(S)) \equiv v(S) \bmod \chi(S), \end{cases}$$

which is again proved directly by using the isomorphism Ψ^{-1}. □

Algorithm 3: A step of Newton-Hensel's lifting.

Function HenselLiftingStep;

Data: A squarefree bivariate polynomial $q \in k[X, Y]$, (λ, χ, u, v) which satisfies **(Div-H1)** to **(Div-H3)**, and a univariate polynomial $\widehat{\chi}$ which divides χ^2.

Result: Two polynomials $\widehat{u}, \widehat{v} \in k[S]_{<\deg(\widehat{\chi})}$ such that $(\lambda, \widehat{\chi}, \widehat{u}, \widehat{v})$ satisfies **(Div-H1)** to **(Div-H3)**.

$$\widehat{u}(S) \leftarrow \left(u(S) - \frac{q(u(S), v(S)) - (\lambda u(S) + v(S) - S) \frac{\partial q}{\partial Y}(u(S), v(S))}{\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S))} \right) \bmod \widehat{\chi}(S);$$

$$\widehat{v}(S) \leftarrow \left(v(S) - \frac{-\lambda q(u(S), v(S)) + (\lambda u(S) + v(S) - S) \frac{\partial q}{\partial X}(u(S), v(S))}{\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S))} \right) \bmod \widehat{\chi}(S);$$

Return $(\widehat{u}, \widehat{v})$.

Proposition 13. *Algorithm* 3 (HenselLiftingStep) *is correct:* $(\lambda, \widehat{\chi}, \widehat{u}, \widehat{v})$ *satisfies* **(Div-H1)** *to* **(Div-H3)**.

Proof. This is a special case of the Newton-Hensel's lifting. Using Taylor expansion,

$$\begin{aligned} &\begin{bmatrix} q(X, Y) \\ \lambda X + Y - S \end{bmatrix} \\ &= \begin{bmatrix} q(u(S), v(S)) \\ \lambda u(S) + v(S) - S \end{bmatrix} + \begin{bmatrix} \frac{\partial q}{\partial X}(u(S), v(S)) & \frac{\partial q}{\partial Y}(u(S), v(S)) \\ \lambda & 1 \end{bmatrix} \cdot \begin{bmatrix} X - u(S) \\ Y - v(S) \end{bmatrix} \\ &\quad + \varepsilon(X, Y, S), \end{aligned}$$

where ε is such that $\varepsilon(\widetilde{u}(S), \widetilde{v}(S), S) \equiv 0 \bmod \chi(S)^2$ for any polynomials $\widetilde{u}, \widetilde{v} \in k[S]$ such that $\widetilde{u} \equiv u \bmod \chi$ and $\widetilde{v} \equiv v \bmod \chi$. Next, we notice that the denominators in the definitions of $\widehat{u}$ and $\widehat{v}$ are invertible modulo $\chi(S)^2$ because they are invertible modulo $\chi(S)$. The proof of **(Div-H1)** and **(Div-H2)** follows from a direct computation by plugging the values of $\widehat{u}$ and $\widehat{v}$ in the Taylor expansion, and by noticing that $\widehat{u} \equiv u \bmod \chi$ and $\widehat{v} \equiv v \bmod \chi$, so that $\varepsilon(\widehat{u}(S), \widehat{v}(S), S) \equiv 0 \bmod \chi(S)^2$ and hence $\varepsilon(\widehat{u}(S), \widehat{v}(S), S) \equiv 0 \bmod \widehat{\chi}(S)$. Finally, **(Div-H3)** is a direct consequence of the fact that $\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S))$ is invertible modulo $\chi(S)$. $\square$

Algorithm 4: Computing the sum of two smooth effective divisors.

Function AddDivisors;
Data: A polynomial $q \in k[X, Y]$ and two smooth effective divisors D_1, D_2 given by primitive element representations $(\lambda_1, \chi_1, u_1, v_1)$ and $(\lambda_2, \chi_2, u_2, v_2)$.
Result: A primitive element representation of the divisor $D_1 + D_2$ or "fail".
$\widehat{\lambda} \leftarrow \text{Random}(k)$;
$(\widehat{\chi}_1, \widehat{u}_1, \widehat{v}_1) \leftarrow \text{ChangePrimElt}(\widehat{\lambda}, \lambda_1, \chi_1, u_1, v_1)$;
$(\widehat{\chi}_2, \widehat{u}_2, \widehat{v}_2) \leftarrow \text{ChangePrimElt}(\widehat{\lambda}, \lambda_2, \chi_2, u_2, v_2)$;
if $\widehat{u}_1 \not\equiv \widehat{u}_2 \bmod \mathrm{GCD}(\widehat{\chi}_1, \widehat{\chi}_2)$ **then**
  Return "fail"
$\widehat{\chi} \leftarrow \widehat{\chi}_1 \cdot \widehat{\chi}_2$;
$\widetilde{\chi} \leftarrow \mathrm{LCM}(\widehat{\chi}_1, \widehat{\chi}_2)$;
$\widehat{u}_{12} \leftarrow \mathrm{XCRT}((\widehat{\chi}_1, \widehat{\chi}_2), (\widehat{u}_1, \widehat{u}_2)) \in k[S]_{<\deg(\widetilde{\chi})}$;
$\widehat{v}_{12} \leftarrow \mathrm{XCRT}((\widehat{\chi}_1, \widehat{\chi}_2), (\widehat{v}_1, \widehat{v}_2)) \in k[S]_{<\deg(\widetilde{\chi})}$;
$(\widehat{u}, \widehat{v}) \leftarrow \text{HenselLiftingStep}(q, \widetilde{\chi}, \widehat{\lambda}, \widehat{u}_{12}, \widehat{v}_{12}, \widehat{\chi})$;
Return $(\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v})$.

Algorithm 4 uses a variant of the CRT, which we call the *Extended Chinese Remainder Theorem* and which we abbreviate as XCRT. Given four univariate polynomials $u_1, u_2, \chi_1, \chi_2 \in k[S]$ such that $u_1 \equiv u_2 \bmod \mathrm{GCD}(\chi_1, \chi_2)$, it returns a polynomial $u \in k[S]$ of degree less than $\deg(\mathrm{LCM}(\chi_1, \chi_2))$ such that $u \equiv u_1 \bmod \chi_1$ and $u \equiv u_2 \bmod \chi_2$. The main difference with the classical CRT is that we do not require χ_1 and χ_2 to be coprime. A minimal solution to the XCRT problem is given by

$$\mathrm{XCRT}((\chi_1, \chi_2), (u_1, u_2)) = (u_2\, a_1\, (\chi_1/g) + u_1\, a_2\, (\chi_2/g)) \bmod \mathrm{LCM}(\chi_1, \chi_2), \tag{2}$$

where $g = \mathrm{GCD}(\chi_1, \chi_2)$ and $a_1, a_2 \in k[S]$ are Bézout coefficients for χ_1, χ_2, i.e., they satisfy $a_1\chi_1 + a_2\chi_2 = g$. Notice that the XCRT is in fact a k-algebra isomorphism between $k[S]/\mathrm{LCM}(\chi_1(S), \chi_2(S))$ and the subalgebra of $k[S]/\chi_1(S) \times k[S]/\chi_2(S)$ formed by pairs (u_1, u_2) such that $u_1 \equiv u_2 \bmod \mathrm{GCD}(\chi_1, \chi_2)$.

Proposition 14. *Algorithm* 4 (AddDivisors) *is correct: If it does not fail, then it returns a primitive element representation of the smooth effective divisor $D_1 + D_2$.*

Proof. Let I_1, I_2, J denote the three following ideals of $k[\mathcal{C}^0]$:

$$\begin{array}{rcllll} I_1 &=& \langle \chi_1(\lambda_1 X + Y), & X - u_1(\lambda_1 X + Y), & Y - v_1(\lambda_1 X + Y)\rangle; \\ I_2 &=& \langle \chi_2(\lambda_2 X + Y), & X - u_2(\lambda_2 X + Y), & Y - v_2(\lambda_2 X + Y)\rangle; \\ J &=& \langle \widehat{\chi}(\widehat{\lambda} X + Y), & X - \widehat{u}(\widehat{\lambda} X + Y), & Y - \widehat{v}(\widehat{\lambda} X + Y)\rangle. \end{array}$$

Proving that Algorithm 4 is correct amounts to showing that $I_1 \cdot I_2 = J$, and that **(Div-H1)** to **(Div-H3)** are satisfied by $\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v}$. First, let $I'_1, I'_2 \subset k[\mathcal{C}^0]$ be the ideals

$$\begin{array}{rcl} I'_1 & = & \langle \widehat{\chi}_1(\widehat{\lambda}X+Y), X-\widehat{u}_1(\widehat{\lambda}X+Y), Y-\widehat{v}_1(\widehat{\lambda}X+Y)\rangle; \\ I'_2 & = & \langle \widehat{\chi}_2(\widehat{\lambda}X+Y), X-\widehat{u}_2(\widehat{\lambda}X+Y), Y-\widehat{v}_2(\widehat{\lambda}X+Y)\rangle. \end{array}$$

By Proposition 12 and Lemma 10, the equalities $I_1 = I'_1$ and $I_2 = I'_2$ hold.

We start by proving that $\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v}$ satisfy **(Div-H1)** to **(Div-H3)**. For **(Div-H1)** and **(Div-H2)**, Proposition 12 ensures that $q(\widehat{u}_i(S), \widehat{v}_i(S)) \equiv 0 \bmod \widehat{\chi}_i(S)$ for $i \in \{1,2\}$. Since the XCRT is a morphism, we get that $q(\widehat{u}_{12}(S), \widehat{v}_{12}(S)) \equiv 0 \bmod \operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))$ and $\widehat{\lambda}\widehat{u}_{12}(S) + \widehat{v}_{12}(S) \equiv S \bmod \operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))$. Next, Proposition 13 proves the equalities $q(\widetilde{u}(S), \widetilde{v}(S)) \equiv 0 \bmod \operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))^2$ and $\widehat{\lambda}\widetilde{u}(S) + \widetilde{v}(S) \equiv S \bmod \operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))^2$. Using the fact that $\widehat{\chi} = \widehat{\chi}_1 \cdot \widehat{\chi}_2$ divides $\operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))^2$, we get that $q(\widetilde{u}(S), \widetilde{v}(S)) \equiv 0 \bmod \widehat{\chi}$ and $\lambda\widehat{u}(S) + \widehat{v}(S) = S$. For **(Div-H3)**, we observe that the fact that the XCRT is a ring morphism implies that $\frac{\partial q}{\partial X}(\widehat{u}_{12}(S), \widehat{v}_{12}(S)) - \lambda\frac{\partial q}{\partial Y}(\widehat{u}_{12}(S), \widehat{v}_{12}(S))$ is invertible in $k[S]/\operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))$. Consequently, $\frac{\partial q}{\partial X}(\widetilde{u}(S), \widetilde{v}(S)) - \lambda\frac{\partial q}{\partial Y}(\widetilde{u}(S), \widetilde{v}(S))$ is invertible in $k[S]/\operatorname{LCM}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))$, and hence it is also invertible in $k[S]/\widehat{\chi}(S)$.

We prove now that $I'_1 \cdot I'_2 = J$. Using the factorization as a product of maximal ideals given by Lemma 6, it is sufficient to prove that a power $\mathfrak{m}^\ell \subset k[\mathcal{C}^0]$ of a maximal ideal contains $I'_1 \cdot I'_2$ if and only if it contains J. Notice that the powers of maximal ideals which contain I'_1 (resp., I'_2) are of the form $\langle \chi_\mathfrak{m}(\widehat{\lambda}X+Y)^\ell, X-u_{1,\mathfrak{m}^\ell}(\widehat{\lambda}X+Y), Y-v_{1,\mathfrak{m}^\ell}(\widehat{\lambda}X+Y)\rangle$ (resp., $\langle \chi_\mathfrak{m}(\widehat{\lambda}X+Y)^\ell, X-u_{2,\mathfrak{m}^\ell}(\widehat{\lambda}X+Y), Y-v_{2,\mathfrak{m}^\ell}(\widehat{\lambda}X+Y)\rangle$), where $\chi_\mathfrak{m}$ is a prime polynomial such that $\chi_\mathfrak{m}^\ell$ divides $\widehat{\chi}_1$ (resp., $\widehat{\chi}_2$), and $u_{1,\mathfrak{m}^\ell}(S) \equiv \widehat{u}_1(S) \bmod \chi_\mathfrak{m}(S)^\ell$, $v_{1,\mathfrak{m}^\ell}(S) \equiv \widehat{v}_1(S) \bmod \chi_\mathfrak{m}(S)^\ell$ (resp., $u_{2,\mathfrak{m}^\ell}(S) \equiv \widehat{u}_2(S) \bmod \chi_\mathfrak{m}(S)^\ell$, $v_{2,\mathfrak{m}^\ell}(S) \equiv \widehat{v}_2(S) \bmod \chi_\mathfrak{m}(S)^\ell$).

Let $\mathfrak{m}^\ell$ be a power of a maximal ideal which contains $I'_1 \cdot I'_2$. Using the unicity of the factorization in Lemma 6, the powers of maximal ideals which contain $I'_1 \cdot I'_2$ are those $\mathfrak{m}^{\ell_1+\ell_2}$ where $I'_1 \subset \mathfrak{m}^{\ell_1}$ and $I'_2 \subset \mathfrak{m}^{\ell_2}$. This means that $\mathfrak{m}^\ell$ has the form

$$\mathfrak{m}^\ell = \langle \chi_\mathfrak{m}(\widehat{\lambda}X+Y)^{\ell_1+\ell_2}, X-u_{12}(\widehat{\lambda}X+Y), Y-v_{12}(\widehat{\lambda}X+Y)\rangle,$$

where u_{12} (resp., v_{12}) is any polynomial such that $u_{12}(S) \equiv u_{1,\mathfrak{m}^{\ell_1}}(S) \bmod \chi_\mathfrak{m}(S)^{\ell_1}$, $u_{12}(S) \equiv u_{2,\mathfrak{m}^{\ell_2}}(S) \bmod \chi_\mathfrak{m}(S)^{\ell_2}$ (resp., $v_{12}(S) \equiv v_{1,\mathfrak{m}^{\ell_1}}(S) \bmod \chi_\mathfrak{m}(S)^{\ell_1}$, $v_{12}(S) \equiv v_{2,\mathfrak{m}^{\ell_2}}(S) \bmod \chi_\mathfrak{m}(S)^{\ell_2}$). Then we notice that $\widehat{\chi} = \widehat{\chi}_1 \cdot \widehat{\chi}_2$, and therefore $\chi_\mathfrak{m}(S)^{\ell_1+\ell_2}$ divides $\widehat{\chi}(S)$. By using the properties of the XCRT and of the Hensel's lifting, we get that

$$\begin{array}{rcl} \widehat{u}(S) & \equiv & \widehat{u}_1(S) \bmod \widehat{\chi}_1(S); \\ \widehat{u}(S) & \equiv & \widehat{u}_2(S) \bmod \widehat{\chi}_2(S); \\ \widehat{v}(S) & \equiv & \widehat{v}_1(S) \bmod \widehat{\chi}_1(S); \\ \widehat{v}(S) & \equiv & \widehat{v}_2(S) \bmod \widehat{\chi}_2(S). \end{array}$$

This implies that $\mathfrak{m}^\ell = \langle \chi_\mathfrak{m}(\widehat{\lambda}X+Y)^{\ell_1+\ell_2}, X-\widehat{u}(\widehat{\lambda}X+Y), Y-\widehat{v}(\widehat{\lambda}X+Y)\rangle$, and hence $\mathfrak{m}^\ell$ contains J.

The proof that any power of maximal ideal which contains J also contains $I'_1 \cdot I'_2$ is similar. $\square$

Algorithm 5 (SubtractDivisors) provides a method for subtracting effective divisors given by primitive element representations. We emphasize that the divisor

Algorithm 5: Computing the subtraction of smooth effective divisors.

Function SubtractDivisors;
Data: Two smooth effective divisors given by primitive element representations:
$D_1 = (\lambda_1, \chi_1, u_1, v_1)$,
$D_2 = (\lambda_2, \chi_2, u_2, v_2)$.
Result: A primitive element representation of the smooth effective divisor $[D_1 - D_2]_+$ or "fail".
$\widehat{\lambda} \leftarrow \text{Random}(k)$;
$(\widehat{\chi}_1, \widehat{u}_1, \widehat{v}_1) \leftarrow \text{ChangePrimElt}(\widehat{\lambda}, \lambda_1, \chi_1, u_1, v_1)$;
$(\widehat{\chi}_2, \widehat{u}_2, \widehat{v}_2) \leftarrow \text{ChangePrimElt}(\widehat{\lambda}, \lambda_2, \chi_2, u_2, v_2)$;
if $\widehat{u}_1 \not\equiv \widehat{u}_2 \bmod \mathrm{GCD}(\widehat{\chi}_1, \widehat{\chi}_2)$ **then**
  Return "fail"
$\widehat{\chi} \leftarrow \widehat{\chi}_1 / \mathrm{GCD}(\widehat{\chi}_1, \widehat{\chi}_2)$;
$\widehat{u}(S) \leftarrow \widehat{u}_1(S) \bmod \widehat{\chi}(S)$;
$\widehat{v}(S) \leftarrow \widehat{v}_1(S) \bmod \widehat{\chi}(S)$;
Return $(\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v})$.

returned is the subtraction $D_1 - D_2$ only if the result is also effective, i.e., if $D_1 \geq D_2$. If this is not the case, then it returns the positive part of the subtraction.

Proposition 15. *Algorithm* 5 (SubtractDivisors) *is correct: If it does not fail, then it returns a primitive element representation of the smooth effective divisor $[D_1 - D_2]_+$, where the notation $[D]_+$ denotes the positive part of the divisor D, i.e., the smallest effective divisor D' such that $D' \geq D$.*

Proof. Let I_1, I_2, J denote the three following ideals of $k[\mathcal{C}^0]$, using the notation in Algorithm 5:

$$\begin{array}{rcl} I_1 & = & \langle \chi_1(\lambda_1 X + Y), \quad X - u_1(\lambda_1 X + Y), \quad Y - v_1(\lambda_1 X + Y)\rangle; \\ I_2 & = & \langle \chi_2(\lambda_2 X + Y), \quad X - u_2(\lambda_2 X + Y), \quad Y - v_2(\lambda_2 X + Y)\rangle; \\ J & = & \langle \widehat{\chi}(\widehat{\lambda} X + Y), \qquad X - \widehat{u}(\widehat{\lambda} X + Y), \qquad Y - \widehat{v}(\widehat{\lambda} X + Y)\rangle. \end{array}$$

The effective divisor $[D_1 - D_2]_+$ corresponds to the colon ideal $I_1 : I_2 = \{f \in k[\mathcal{C}^0] \mid f \cdot I_2 \subset I_1\}$. Consequently, we must prove that $(\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v})$ satisfies **(Div-H1)** to **(Div-H3)** and that $J = I_1 : I_2$. The equalities **(Div-H1)** to **(Div-H3)** for $\widehat{\lambda}, \widehat{\chi}_1, \widehat{u}_1, \widehat{v}_1$ are satisfied by Proposition 12. Regarding them modulo $\widehat{\chi}$ shows that $(\widehat{\lambda}, \widehat{\chi}, \widehat{u}, \widehat{v})$ satisfies **(Div-H1)** to **(Div-H3)**.

In order to prove that $J = I_1 : I_2$, we proceed as in the proof of Proposition 14, by noticing first that I_1 and I_2 can be rewritten as

$$\begin{array}{rcl} I_1 & = & \langle \widehat{\chi}_1(\widehat{\lambda} X + Y), \quad X - \widehat{u}_1(\widehat{\lambda} X + Y), \quad Y - \widehat{v}_1(\widehat{\lambda} X + Y)\rangle; \\ I_2 & = & \langle \widehat{\chi}_2(\widehat{\lambda} X + Y), \quad X - \widehat{u}_2(\widehat{\lambda} X + Y), \quad Y - \widehat{v}_2(\widehat{\lambda} X + Y)\rangle. \end{array}$$

Using [2, Prop. 9.1] together with the fact that D_1 does not involve any singular point of the curve by **(Div-H3)**, the equality $I_1 : I_2 = J$ holds if and only if the powers of maximal ideals $\mathfrak{m}^\ell \subset k[\mathcal{C}^0]$ which contain $I_1 : I_2$ are exactly those which contain J. Equivalently, this means that if $\mathfrak{m}^{\ell_1}$ is the largest power of $\mathfrak{m}$ which contains I_1 and if $\mathfrak{m}^{\ell_2}$ is the largest power of $\mathfrak{m}$ which contains I_2, then $\mathfrak{m}^{\max(\ell_1 - \ell_2, 0)}$ is the largest power of $\mathfrak{m}$ which contains J. As in the proof of Proposition 14, the maximal ideals $\mathfrak{m} \subset k[\mathcal{C}^0]$ which contain I_1 have the form $\langle \chi_{\mathfrak{m}}(\widehat{\lambda}X+Y), X - u_{\mathfrak{m}}(\widehat{\lambda}X+Y), Y - v_{\mathfrak{m}}(\widehat{\lambda}X+Y)\rangle$, where $u_{\mathfrak{m}} \equiv \widehat{u}_1 \bmod \chi_{\mathfrak{m}}, v_{\mathfrak{m}} \equiv \widehat{v}_1 \bmod \chi_{\mathfrak{m}}$. The proof is concluded by noticing that for any prime factor Φ of $\widehat{\chi}_1$, if Φ^{ℓ_1} is the largest power of Φ which divides $\widehat{\chi}_1$ and Φ^{ℓ_2} is the largest power of Φ which divides $\widehat{\chi}_2$, then the largest power Φ which divides $\widehat{\chi} = \widehat{\chi}_1/\operatorname{GCD}(\widehat{\chi}_1, \widehat{\chi}_2)$ is $\Phi^{\max(\ell_1-\ell_2,0)}$. $\square$

5. Description and correctness of the subroutines

Algorithm 6: Computing a function $h \in k[\mathcal{C}^0]$ of small degree such that $(h) \geq D+E$.

Function Interpolate;
Data: The degree δ of the curve, a smooth effective divisor given by a primitive element representation (λ, χ, u, v), and the nodal divisor given by $(\lambda_E, \chi_E, u_E, v_E, T_E)$.
Result: A polynomial $h \in k[X,Y]$ representing a form in $k[\mathcal{C}]$ such that $(h) \geq D+E$.
if $\binom{\delta+1}{2} \leq \deg(\chi) + \deg(\chi_E)$ **then**
  $d \leftarrow \lfloor (\deg(\chi)+\deg(\chi_E))/\delta + (\delta-1)/2 \rfloor$
else
  $d \leftarrow \lfloor (\sqrt{1+8(\deg(\chi)+\deg(\chi_E))} - 1)/2 \rfloor$
Construct the matrix representing the linear map
$\varphi : \{f \in k[X,Y] \mid \deg(f) \leq d, \deg_Y(f) < \delta\} \to k[S]_{<\deg(\chi)} \times k[S]_{<\deg(\chi_E)}$
defined as
$\varphi(f(X,Y)) = (f(u(S), v(S)) \bmod \chi(S), f(u_E(S), v_E(S)) \bmod \chi_E(S))$;
Compute a basis $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ of the kernel of φ;
$(\mu_1, \ldots, \mu_\ell) \leftarrow$ Random$(k^\ell \setminus \{\mathbf{0}\})$;
Return $h = \sum_{i=1}^{\ell} \mu_i \mathbf{b}_i$.

5.1. **Interpolation.** This section focuses on the following interpolation problem: Given a smooth effective divisor D and the nodal divisor E, find an element $h \in k[\mathcal{C}^0]$ such that its associated principal divisor (h) satisfies $(h) \geq D+E$.

Proposition 16. *Algorithm* 6 (Interpolate) *is correct: The kernel of φ has positive dimension, and its nonzero elements h satisfy* $(h) \geq D+E$.

Proof. The fact that the kernel φ has positive dimension follows from a dimension count, which is postponed to Lemma 17. We now prove the second part of the proposition. First, notice that $\deg_Y(h) < \deg_Y(q)$ for any nonzero h in the kernel of φ, hence h cannot be a multiple of q, which implies that $\langle 0\rangle \subsetneq \langle h\rangle \subset k[\mathcal{C}^0]$. Next, by Lemmas 9 and 10, the ideal $I_{D+E} = \{f \in k[\mathcal{C}^0] \mid (f) \geq D+E\} = \{f \in k[\mathcal{C}^0] \mid (f) \geq D\} \cap \{f \in k[\mathcal{C}^0] \mid (f) \geq E\}$ equals $I_D \cap I_E$, where

$$\begin{aligned} I_D &= \langle \chi(\lambda X+Y), X - u(\lambda X+Y), Y - v(\lambda X+Y)\rangle, \\ I_E &= \langle \chi_E(\lambda_E X+Y), X - u_E(\lambda_E X+Y), Y - v_E(\lambda X+Y)\rangle. \end{aligned}$$

By construction, $h(u(S), v(S)) \equiv 0 \bmod \chi(S)$ and $h(u_E(S), v_E(S)) \equiv 0 \bmod \chi_E(S)$ for any $h \in \ker\varphi$. The proof is concluded by noticing that the polynomials f in I_{D+E} are those which satisfy $f(u(S), v(S)) \equiv 0 \bmod \chi(S)$ and $f(u_E(S), v_E(S)) \equiv 0 \bmod \chi_E(S)$, using the isomorphisms in Propositions 4 and 5. $\square$

The following lemma ensures that Algorithm 6 actually returns a nonzero element, i.e., that the kernel of φ has positive dimension.

Lemma 17. *With the notation in Algorithm* 6*,*

$$\begin{array}{rcl}\deg(\chi)+\deg(\chi_E) & < & \dim_k(\{f\in k[X,Y] \mid \deg(f)\le d, \deg_Y(f)<\delta\})\\ & \le & 3(\deg(\chi)+\deg(\chi_E)).\end{array}$$

Consequently, φ is not injective.

Proof. Set $w=\deg(\chi)+\deg(\chi_E)$. First, a direct dimension count gives

$$\dim_k(\{f\in k[X,Y] \mid \deg(f)\le d, \deg_Y(f)<\delta\}) = \begin{cases}\delta(d-(\delta-3)/2) & \text{if } d\ge\delta,\\ \dbinom{d+2}{2} & \text{otherwise.}\end{cases}$$

On one hand, if $\binom{\delta+1}{2}\le w$, then

$$\begin{array}{rcl} d & = & \lfloor w/\delta+(\delta-1)/2\rfloor\\ & \ge & \left\lfloor \binom{\delta+1}{2}/\delta+(\delta-1)/2\right\rfloor\\ & \ge & \delta,\end{array}$$

and hence

$$\begin{array}{rcl} \delta(d-(\delta-3)/2) & > & \delta(w/\delta+(\delta-1)/2-1-(\delta-3)/2)\\ & = & w\\ \delta(d-(\delta-3)/2) & \le & \delta(w/\delta+(\delta-1)/2-(\delta-3)/2)\\ & \le & w+\delta\\ & \le & w+\binom{\delta+1}{2}\\ & \le & 2w.\end{array}$$

On the other hand, if $\binom{\delta+1}{2}>w$, then

$$\begin{array}{rcl} d & = & \lfloor(\sqrt{1+8w}-1)/2\rfloor\\ & < & \lfloor(\sqrt{1+4\delta(\delta+1)}-1)/2\rfloor\\ & = & \lfloor(\sqrt{(2\delta+1)^2}-1)/2\rfloor\\ & = & \delta.\end{array}$$

Since $\binom{x+2}{2}-w>0$ for any $x>(\sqrt{1+8w}-3)/2$, we get that $w<\binom{d+2}{2}$ as expected. Finally, the last inequality follows from

$$\binom{\lfloor(\sqrt{1+8w}-1)/2\rfloor+2}{2}\le w+(1+\sqrt{1+8w})/2,$$

and direct computations show that $(1+\sqrt{1+8w})/2\le 2w$. □

5.2. **Computing the smooth part of the principal divisor associated to a regular function on the curve.** The section is devoted to the following problem: Given a polynomial $h\in k[\mathcal{C}^0]$ such that $(h)=D_h+E$ where D_h is a smooth divisor on the curve, compute a primitive element representation of D_h.

Let us mention that it may happen that h vanishes at infinity. Therefore, the support of D_h may contain points at infinity, but the primitive element representation only represents points in the affine chart $Z\neq 0$. Ignoring these zeros at infinity may lead to functions having unauthorized poles at infinity in the basis returned by Algorithm 1. As we already mentioned in Section 3, handling what happens at infinity is not a problem: This issue can be solved for instance by doing the computations in three affine spaces which cover $\mathbb{P}^2$, which would multiply the complexity by a constant factor. Notice also that it is easy to detect if h has zeros at infinity: This happens if and only if the degree of the resultant of h and q is strictly less than $\deg(h)\deg(\mathcal{C})$, thanks to the fact that we assumed that $\mathcal{C}$ is in projective Noether position. For simplicity, we will not discuss further this issue in the rest of this paper.

The central element of Algorithm CompPrincDiv is the computation of a resultant and of the associated first subresultant (as defined for instance in [8, Sec. 3]). However, a number of extra steps are required to ensure that this computation satisfies genericity assumptions and returns a correct result. First, a random direction of projection λ is selected for computing the resultant. This direction of projection must satisfy some conditions. In particular, a distinct point in the support of h must project on distinct points. Also, in order to exploit the Poisson formula for the resultant, we also ask that this direction is not a tangent at any node of the curve; see Lemma 19. This condition about the tangents at the node is tested via the evaluation of the univariate polynomial T_E. We also need a representation of the nodal divisor with respect to this λ. This is achieved by using a slightly modified version of Algorithm ChangePrimElt where the first test—which is not relevant for the nodal divisor—is removed. Finally, Algorithm CompPrincDiv must clean out the singular points: this is done by noticing that the roots of the resultant which correspond to the singular points appear with multiplicity at least 2; see Lemma 19 below. Therefore these singular points are removed by dividing out by the square of the univariate polynomial $\widehat{\chi}_E$ whose roots parametrize the coordinates of the singular points.

Proposition 18. *Algorithm* 7 (CompPrincDiv) *is correct: If it does not fail, then it returns a primitive element representation of the smooth part of the principal divisor* (h).

Before proving Proposition 18, we need the following technical lemma, which implies in particular that nodes appear as roots of the resultant with multiplicity at least two.

Lemma 19. *With the notation in Algorithm* 7, *let* $s \in \overline{k}$ *and* $\lambda \in k \setminus \{0\}$. *Let* $R_1, \ldots, R_\ell$ *be the valuation rings in* $\mathrm{Frac}(\overline{k}[X,Y]/q)$ *associated to the points of* $\widetilde{C}$ *above* s, *i.e., the points on* $\mathcal{C}$ *which project to* s *via the projection* $(X,Y) \mapsto \lambda X + Y$. *Assume that the vector* $(1, -\lambda)$ *is not tangent to* $\mathcal{C}^0$ *at any of these points and that the coefficient of* $Y^{\deg(\mathcal{C})}$ *in* $q((S-Y)/\lambda, Y)$ *is nonzero. Let* $m_1, \ldots, m_{\deg(\mathcal{C})}$ *denote the valuations of* h *in these valuation rings. Then* s *is a root of multiplicity* $\sum_{i=1}^{\ell} m_i$ *in* $\mathrm{Resultant}_Y(q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y))$.

Proof. Since we assumed that the curve is nodal, that the coefficient of $Y^{\deg(\mathcal{C})}$ in $q((S-Y)/\lambda, Y)$ is nonzero, and that the vector $(1,-\lambda)$ is not tangent at any point above s, we get that the polynomial $q((S-Y)/\lambda, Y)$ splits over the ring $\overline{k}[[S-s]]$ of power series at s as a product of $\deg(\mathcal{C})$ factors; see, e.g., [20] and the references therein. Notice that this factorization property holds even if some of the points above s are nodes. Let $\widetilde{y}_1, \ldots, \widetilde{y}_{\deg(\mathcal{C})}$ denote its roots in $\overline{k}[[S-s]]$. Using the multiplicativity property of the resultant [16, Sec. 5.7], we get

$$\mathrm{Resultant}_Y(q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y)) = \alpha \prod_{i=1}^{\ell} h((S-\widetilde{y}_i)/\lambda, \widetilde{y}_i),$$

Algorithm 7: Computing a primitive element representation of the smooth part of (h).

Function CompPrincDiv;
Data: A squarefree bivariate $q \in k[X,Y]$ such that $\deg(q) = \deg_Y(q)$, a bivariate polynomial $h \in k[X,Y]$, and a representation $(\lambda_E, \chi_E, u_E, v_E, T_E)$ of the nodal divisor.
Result: A primitive element representation $(\lambda, \chi(S), u(S), v(S))$ of the smooth part of the principal effective divisor (h) or "fail".
$\lambda \leftarrow$ Random(k);
if $\lambda = 0$ *or if the coefficient of* $Y^{\deg(q)}$ *in* $q((S-Y)/\lambda, Y) \in k[S][Y]$ *is* 0 **then**
  Return "fail"
if $T_E(\lambda) = 0$ **then**
  Return "fail";
  `/* Ensures that` $(1, -\lambda)$ `is not tangent to the curve at any node. */`
$(\widehat{\chi}_E, \widehat{u}_E, \widehat{v}_E) \leftarrow$ ChangePrimEltNodal$(\lambda, \lambda_E, \chi_E, u_E, v_E)$;
`/* ChangePrimEltNodal is the same algorithm as ChangePrimElt, but we skip the first test (which would fail on the nodal divisor). */`
$\widetilde{\chi}(S) \leftarrow \mathrm{Resultant}_Y(q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y))$;
$a_0(S) + Ya_1(S) \leftarrow \mathrm{FirstSubRes}_Y(q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y))$;
$\chi \leftarrow \widetilde{\chi}/\widehat{\chi}_E^2$;
if $\mathrm{GCD}(\chi, \widehat{\chi}_E) \neq 1$ **then**
  Return "fail"
if $\mathrm{GCD}(a_1(S), \chi(S)) \neq 1$ **then**
  Return "fail"
$v(S) \leftarrow -a_0(S) \cdot a_1(S)^{-1} \bmod \chi(S)$;
$u(S) \leftarrow (S - v(S))/\lambda$;
if $\mathrm{GCD}(\frac{\partial q}{\partial X}(u(S), v(S)) - \lambda \frac{\partial q}{\partial Y}(u(S), v(S)), \chi(S)) \neq 1$ **then**
  Return "fail"
Return $(\lambda, \chi(S), u(S), v(S))$.

where $\alpha \in k$. The proof is concluded by noticing that $S - s$ is a uniformizing element for all the discrete valuation rings since the vector $(1, -\lambda)$ is not tangent to the curve at any of the points above s, so that m_i precisely corresponds to the largest integer γ such that $(S-s)^\gamma$ divides $h((s - \widetilde{y}_i)/\lambda, \widetilde{y}_i)$. □

Proof of Proposition 18. In order to prove Proposition 18, we must prove that the output (λ, χ, u, v) satisfies **(Div-H1)** to **(Div-H3)** and that the two ideals $\langle h \rangle : I_E^\infty \subset k[\mathcal{C}^0]$ and $\langle \chi(\lambda X + Y), X - u(\lambda X + Y), Y - v(\lambda X + Y)\rangle \subset k[\mathcal{C}^0]$ are equal, where I_E is the radical ideal of $k[\mathcal{C}^0]$ which encodes the algebraic set of the nodes. **(Div-H2)** follows directly from the definitions of $u(S)$ and $v(S)$ in Algorithm 7. To prove **(Div-H1)**, we shall prove that the equality holds modulo $(S-s)^\gamma$ for any root $s \in \overline{k}$ of χ of multiplicity γ. A classical property of the subresultants is that they belong to the ideal generated by the input polynomials. This implies that for any root $s \in \overline{k}$ of $\widetilde{\chi}$ we have

$$a_0(S) + Ya_1(S) \in \langle q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y)\rangle \subset \overline{k}[[S-s]][Y].$$

If the algorithm does not fail, then $a_1(S)$ is invertible modulo $\chi(S)$. Consequently, it is also invertible in $\overline{k}[[S-s]]$ for any root $s \in \overline{k}$ of χ and hence

$$Y + a_0(S)a_1(S)^{-1} \in \langle q((S-Y)/\lambda, Y), h((S-Y)/\lambda, Y)\rangle \subset \overline{k}[[S-s]][Y].$$

Therefore, the GCD of $q((S-Y)/\lambda, Y)$ and $h((S-Y)/\lambda, Y)$ in $\mathrm{Frac}(\overline{k}[[S-s]])[Y]$ divides $Y+a_0(S)a_1(S)^{-1}$. But we also know that this GCD is nonconstant, since s is a root of the resultant $\widetilde{\chi}$. By a degree argument, this GCD equals $Y+a_0(S)a_1(S)^{-1}$ and hence $q((S+a_0(S)a_1(S)^{-1})/\lambda, -a_0(S)a_1(S)^{-1}) = 0$ in $\overline{k}[[S-s]]$. Considering this equation modulo $(S-s)^\gamma$ and using the CRT over all the roots of χ finishes the proof of **(Div-H1)**. Finally, **(Div-H3)** is explicitly tested and hence it must be satisfied if the algorithm does not fail.

It remains to prove the equality of the ideals $\langle h\rangle : I_E^\infty \subset k[\mathcal{C}^0]$ and $\langle \chi(\lambda X + Y), X - u(\lambda X+Y), Y - v(\lambda X+Y)\rangle \subset k[\mathcal{C}^0]$. Using the isomorphism between $k[X,Y]/\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle$ and $k[S]/\chi(S)$ (see Proposition 4), the elements in $\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle$ are precisely the classes of the bivariate polynomials $\psi(X,Y) \in k[X,Y]$ such that $\psi(u(S), v(S)) \equiv 0 \bmod \chi(S)$. Using a proof identical to that of **(Div-H1)** we get that $h(u(S),v(S)) \equiv 0 \bmod \chi(S)$ which proves that $\langle h\rangle \subset \langle \chi(\lambda X+Y), X - u(\lambda X+Y), Y-v(\lambda X+Y)\rangle$. Saturating on both sides, we get that $\langle h\rangle : I_E^\infty \subset \langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle : I_E^\infty = \langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle$, where the last equality comes from the fact that $\mathrm{GCD}(\chi, \widehat{\chi}_E) = 1$. For the other inclusion, we use [2, Prop. 9.1], which implies that $\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle \subset \langle h\rangle : I_E^\infty$ if this inclusion holds in the local ring associated to any maximal ideal $\mathfrak{m} \subset \overline{k}[\mathcal{C}^0]$ which contains $\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle$. Over $\overline{k}$, these maximal ideals have the form $\langle \lambda X+Y-s, X-u(s), Y-v(s)\rangle$, where $s \in \overline{k}$ is a root of χ. The assumption $\mathrm{GCD}(\frac{\partial q}{\partial X}(u(S),v(S)) - \lambda\frac{\partial q}{\partial Y}(u(S),v(S)), \chi(S)) = 1$ ensures that all these maximal ideals correspond to nonsingular points, and hence the associated local rings are discrete valuation rings. For $s \in \overline{k}$ a root of χ, let $y_1,\dots,y_{\deg(\mathcal{C})}$ be the roots of the univariate polynomial $q((s-Y)/\lambda, Y) \in \overline{k}[Y]$. Let m_i denote the intersection multiplicity of h at the point $((s-y_i)/\lambda, y_i)$ of $\mathcal{C}^0$. Since $\mathrm{GCD}(\frac{\partial q}{\partial X}(u(S),v(S)) - \lambda\frac{\partial q}{\partial Y}(u(S),v(S)), \chi(S)) \neq 1$, we obtain that the vector $(1,-\lambda)$ is not tangent to $\mathcal{C}^0$ at any of these points. Lemma 19 then gives that $m_1 + \cdots + m_{\deg(\mathcal{C})} = \alpha$, where α is the multiplicity of the root s in χ. Let k be the integer such that $y_k = v(s)$. Then $m_k \le \alpha$, which shows that we have $\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle \subset \langle h\rangle : I_E^\infty$ in the local ring at the point $(u(s),v(s))$. The statement [2, Prop. 9.1] concludes the proof of the inclusion $\langle \chi(\lambda X+Y), X-u(\lambda X+Y), Y-v(\lambda X+Y)\rangle \subset \langle h\rangle : I_E^\infty$. ☐

5.3. **Computing the linear space of regular functions of bounded degree having prescribed zeros.** The task accomplished by Algorithm NUMERATORBASIS is similar to what Algorithm INTERPOLATE does: It computes a basis of the vector space of regular functions having prescribed zeros. The only difference with Algorithm INTERPOLATE is that Algorithm NUMERATORBASIS returns a basis of this linear space.

Proposition 20. *Algorithm* 8 (NUMERATORBASIS) *is correct: the nonzero elements g in the kernel of φ are not divisible by q and they satisfy $(g) \ge D + E$.*

Proof. The proof is similar to that of Proposition 16. ☐

Algorithm 8: Computing a basis of the vector space of regular functions $g \in k[\mathcal{C}^0]$ of degree δ such that $(g) \geq D + E$.

Function NUMERATORBASIS;

Data: A positive integer δ, a smooth effective divisor given by a primitive element representation $(\lambda, \chi(S), u(S), v(S))$, a positive integer d, and the nodal divisor given by $(\lambda_E, \chi_E, u_E, v_E, T_E)$.

Result: A basis of the space of polynomials $g \in k[X, Y]$ such that $\deg(g) \leq d, \deg_Y(g) < \delta$ and the associated divisor satisfies $(g) \geq D + E$.

Construct the matrix representing the linear map

$\varphi : \{f \in k[X, Y] \mid \deg(f) \leq d, \deg_Y(f) \leq \delta\} \to k[S]_{<\deg(\chi)} \times k[S]_{<\deg(\chi_E)}$

defined as

$\varphi(f(X, Y)) = (f(u(S), v(S)) \bmod \chi(S), f(u_E(S), v_E(S)) \bmod \chi_E(S))$;

Compute and return a basis of the kernel of φ.

6. COMPLEXITY

All complexity bounds count the number of arithmetic operations (additions, subtractions, multiplications, divisions) in k, all at unit cost. We do not include in our complexity bounds the cost of generating random elements, nor the cost of monomial manipulations, nor multiplications by fixed integer constants. In particular, we do not include in our complexity bounds the cost of computing the partial derivatives of a polynomial. We use the classical $O()$ and $\widetilde{O}()$ notation; see, e.g., [28, Sec. 25.7]. The notation $\mathrm{M}(n)$ stands for the number of arithmetic operations required in k to compute the product of two univariate polynomials of degree n with coefficients in k. By [7], $\mathrm{M}(n) = O(n \log n \log \log n)$. In what follows, ω is a feasible exponent for matrix multiplication, i.e., ω is such that there is an algorithm for multiplying two $N \times N$ matrices with entries in k within $O(N^\omega)$ arithmetic operations in k. The best known bound is $\omega < 2.3729$ [19]. In the following, we make the assumption[1] that $\omega > 2$.

Proposition 21. *Algorithm* 2 (CHANGEPRIMELT) *requires at most* $O(\deg(\chi)^\omega)$ *arithmetic operations in* k*.*

Proof. In order to construct the matrix M in Algorithm 2, we must compute the remainders $S^i \cdot (\widetilde{\lambda} u(S) + v(S)) \bmod \chi(S)$ for $i \in \{0, \dots, \deg(\chi) - 1\}$. Each of these computations costs $O(\mathrm{M}(\deg(\chi)))$ arithmetic operations, so the total cost of constructing the matrix M is bounded by $O(\deg(\chi)\,\mathrm{M}(\deg(\chi)))$, which is bounded above by $O(\deg(\chi)^\omega)$. Computing the characteristic polynomial of M can be done within $O(\deg(\chi)^\omega)$ arithmetic operations [21]. We emphasize that in [21], it is assumed that the cardinality of k is at least $2\deg(\chi)^2$, so that the probability of failure is bounded by $1/2$. In fact, using the same algorithm and the same proof as in [21], the assumption on the cardinality of k can be removed but the probability of failure will then only be bounded by $\deg(\chi)^2/|\mathcal{E}|$, where $\mathcal{E} \subset k$ is a finite subset in which we can draw elements uniformly at random. We will incorporate this probability of failure for the computation of the characteristic

[1] If $\omega = 2$, then the $O()$ in Theorem 28 should be replaced by $\widetilde{O}()$.

polynomial in our bound for the probability of failure of the main algorithm; see the proof of Theorem 36.

Constructing the matrix N is done by computing successively the remainders $(\widetilde{\lambda}u(S)+v(S))^i \bmod \chi(S)$ for $i \in \{0,\dots,\deg(\chi)-1\}$. The total cost of this procedure is $O(\deg(\chi)\,\mathrm{M}(\deg(\chi)))$ which is again bounded by $O(\deg(\chi)^\omega)$. Finally, inverting N and applying the inverse linear map can be done using $O(\deg(\chi)^\omega)$ operations in k by using [5]. □

Proposition 22. *Algorithm* 3 (HenselLiftingStep) *requires at most* $O(\deg(q)^2\cdot \mathrm{M}(\deg(\chi)))$ *arithmetic operations in* k*.*

Proof. Algorithm 3 consists of evaluations of q and of its partial derivatives at $(u(S),v(S))$, together with finitely many arithmetic operations in $k[S]/\chi(S)^2$. Each of the arithmetic operations modulo χ^2 costs $O(\mathrm{M}(\deg(\chi)))$ arithmetic operations in k. Evaluating q at $(u(S),v(S))$ modulo $\chi(S)^2$ can be done by computing the remainders $u(S)^iv(S)^j \bmod \chi(S)^2$ for all $(i,j)\in\mathbb{Z}_{\geq 0}$ such that $i+j\leq \deg(q)$, then by multiplying these evaluations by the corresponding coefficients in q and by summing them. Computing all the modular products can be done in $O(\deg(q)^2\,\mathrm{M}(\deg(\chi)))$ operations in k, by considering the pairs (i,j) in increasing lexicographical ordering. Multiplying by the coefficients and summing then costs $O(\deg(q)^2\deg(\chi))$ arithmetic operations in k. Computing the evaluations of the partial derivatives of q is done similarly and it has a similar cost. □

Proposition 23. *Algorithm* 4 (AddDivisors) *requires at most* $O(\deg(q)^2\,\mathrm{M}(\nu)+\nu^\omega)$ *arithmetic operations in* k*, where* $\nu=\max(\deg(\chi_1),\deg(\chi_2))$*.*

Proof. Algorithm 4 starts by two calls to the function ChangePrimElt, with respective costs $O(\deg(\chi_1)^\omega)$ and $O(\deg(\chi_2)^\omega)$ by Proposition 21. The polynomial $\mathrm{GCD}(\widehat{\chi}_1,\widehat{\chi}_2)$ can be computed at cost $O(\mathrm{M}(\nu)\log(\nu))$ using the fast GCD algorithm [28, Coro. 11.9]. The product $\widehat{\chi}$ in Algorithm 4 and the LCM are then also computed at costs $O(\mathrm{M}(\nu))$ and $O(\mathrm{M}(\nu)\log(\nu))$. The XCRT can be computed at cost $O(\mathrm{M}(\nu)\log(\nu))$ by using equation (2) together with the fact that Bézout coefficients can be computed within quasi-linear complexity [28, Coro. 11.9]. Finally, the Hensel lifting step can be achieved at cost $O(\deg(q)^2\,\mathrm{M}(\nu))$ by Proposition 22. □

Proposition 24. *Algorithm* 5 (SubtractDivisors) *requires at most* $O(\nu^\omega)$ *arithmetic operations in* k*, where* $\nu=\max(\deg(\chi_1),\deg(\chi_2))$*.*

Proof. Most of the steps of Algorithm 5 are similar to steps of Algorithm 4, except that Hensel lifting is not required here. The complexity analysis is similar and we refer to the proof of Proposition 23. The only step which does not appear in Algorithm 4 is the exact division of $\widehat{\chi}_1$ by the GCD. The cost of this step does not hinder the global complexity since exact division of polynomials can be done in quasi-linear complexity [28, Thm. 9.1]. □

In practice, if k is sufficiently large, then choosing a global value for λ and using the same value for all the representations of divisors would succeed with large probability. In this case, we do not need to call the function ChangePrimElt within Algorithms AddDivisors and SubtractDivisors. This would decrease significantly the complexities of AddDivisors and SubtractDivisors. In any case, this would not change the global asymptotic complexity of Algorithm 1.

Proposition 25. *Algorithm* 6 (Interpolate) *requires at most* $O((\deg(\chi)+r)^\omega)$ *arithmetic operations in* k *and it returns a polynomial of degree less than* $(\deg(\chi)+r)/\delta+\delta$.

Proof. First, we recall that $\deg(\chi_E)=r$. The computation of the degree d does not cost any arithmetic operations in k. The construction of the matrix representing the linear map φ can be done by computing all the modular products $u(S)^i v(S)^j$ modulo $\chi(S)$ and χ_E for pairs (i,j) such that $i+j\leq d$ and $j<\delta$. Lemma 17 states that the number of such pairs is bounded above by $3(\deg(\chi)+r)$. By considering the pairs (i,j) in increasing lexicographical ordering, computing all these modular products can be done within $O((\deg(\chi)+r)\,\mathrm{M}(\deg(\chi)+r))$ operations in k. Then, since both dimensions of the matrix are in $O(\deg(\chi)+r)$, computing a basis of the kernel can be done at cost $O((\deg(\chi)+r)^\omega)$ (for instance via a row echelon form computation; see [24, Thm. 2.10]).

Next, we show the bound on the degree of the polynomial returned. By construction, the inequality $\deg(h)\leq d$ holds so it suffices to show that $d<(\deg(\chi)+r)/\delta+\delta$. If $\binom{\delta+1}{2}\leq(\deg(\chi)+r)$, we have $d=\lfloor(\deg(\chi)+r)/\delta+(\delta-1)/2\rfloor<(\deg(\chi)+r)/\delta+\delta$. Otherwise, $d=\lfloor(\sqrt{1+8(\deg(\chi)+r)}-1)/2\rfloor<\delta<(\deg(\chi)+r)/\delta+\delta$ by direct computations. In both cases, we have $\deg(h)<(\deg(\chi)+r)/\delta+\delta$. $\square$

Proposition 26. *Algorithm* 7 (CompPrincDiv) *requires at most* $\widetilde{O}(\max(\deg(q),\deg(h))^2\cdot\min(\deg(q),\deg(h)))$ *arithmetic operations in* k.

Proof. The two costly steps in Algorithm 7 are the computations of the resultant and of the subresultant of two bivariate polynomials. This can be done within $\widetilde{O}(\max(\deg(q),\deg(h))^2\cdot\min(\deg(q),\deg(h)))$ operations using [28, Coro. 11.21]. The Bézout bound implies that the degree of the resultant $\widetilde{\chi}(S)$ is bounded above by $\deg(q)\deg(h)$, hence the complexities of all the other steps are quasi-linear in $\deg(q)\deg(h)$, which is negligible compared to the cost of the computation of the resultant and the subresultant. $\square$

We point out that the complexity of computing resultants and subresultants of bivariate polynomials has been recently improved in [25, 26] under some genericity assumptions. However, since the cost in Proposition 26 will be negligible in the global complexity estimate, we make no effort to optimize it further.

Proposition 27. *Algorithm* 8 (NumeratorBasis) *requires at most* $O((\deg(\chi)+r)^\omega)$ *arithmetic operations in* k.

Proof. By Lemma 17, the domain of the map φ has dimension $O(\deg(\chi)+r)$. Using the monomial basis, the matrix representing the map φ can be constructed within $\widetilde{O}((\deg(\chi)+r)^2)$ operations by doing as in the proof of Proposition 25. Similarly to the proof of Proposition 25, a basis of the kernel of this matrix can be obtained by computing first a row echelon form of the matrix within $O((\deg(\chi)+r)^\omega)$ operations [24, Thm. 2.10]. $\square$

All the complexities and the degree estimates computed in this section are summed up in Table 1. For bounding the degree of $D_{\mathsf{num}}=D_{\mathsf{res}}+D_-$, we use the fact that we can assume without loss of generality that $\deg(D_-)\leq\deg(D_+)$, since otherwise $L(D_+-D_-)$ is reduced to 0. Summing all the complexity bounds yields the global complexity bound as follows.

TABLE 1. Complexities of the subroutines in terms of the input size and degrees of divisors.

Divisor	Degree
D_h	$< \deg(\mathcal{C})^2 + \deg(D_+)$
D_{res}	$< \deg(\mathcal{C})^2$
D_{num}	$< \deg(\mathcal{C})^2 + \deg(D_+)$

Subroutine	Complexity
INTERPOLATE	$O((\deg(D_+) + r)^\omega)$
COMPPRINCDIV	$\widetilde{O}(\max(\deg(\mathcal{C})^3, (\deg(D_+) + r)^2/\deg(\mathcal{C})))$
SUBTRACTDIVISORS	$O(\max(\deg(\mathcal{C})^{2\omega}, (\deg(D_+))^\omega))$
ADDDIVISORS	$O(\max(\deg(\mathcal{C})^{2\omega}, (\deg(D_+))^\omega))$
NUMERATORBASIS	$O(\max(\deg(\mathcal{C})^{2\omega}, \deg(D_+)^\omega))$

Theorem 28. *Algorithm* 1 (RIEMANNROCHBASIS) *requires at most*

$$O(\max(\deg(\mathcal{C})^{2\omega}, \deg(D_+)^\omega))$$

arithmetic operations in k.

Proof. A direct consequence of Propositions 23, 24, 25, 26, and 27 is that the complexity of Algorithm 1 is bounded by $O(\max(\deg(\mathcal{C})^{2\omega}, (\deg(D_+) + r)^\omega))$. The proof is concluded by noticing that $r = O(\deg(\mathcal{C})^2)$ since $g = \binom{\deg(\mathcal{C})-1}{2} - r$ is nonnegative. $\square$

7. LOWER BOUNDS ON THE PROBABILITY OF SUCCESS

In this section, we examine all possible sources of failures for the main algorithm. In fact, if the assumptions detailed in Section 2 are satisfied, then failure can only come from a bad choice of an element picked at random. More precisely, we show that these bad choices can be characterized algebraically and that they are included in the set of roots of polynomials. Bounding the degrees of these polynomials provides us with lower bounds on the probability of success if random elements in k are picked uniformly at random in a finite subset $\mathcal{E} \subset k$.

First, we investigate which values of λ make Algorithm 2 (CHANGEPRIMELT) fail: These are the values of λ such that there is a line of equation $\lambda X + Y + \gamma$ for some $\gamma \in \overline{k}$ which goes either through two distinct points in the support of the input divisor, or which is tangent to $\mathcal{C}^0$ at a point in the support of the divisor.

Proposition 29. *Given an effective divisor $D = (\lambda, \chi, u, v)$, the set of $\widetilde{\lambda} \in k$ such that Algorithm* 2 (CHANGEPRIMELT) *with input $D, \widetilde{\lambda}$ fails is contained in the set of roots of a nonzero univariate polynomial with coefficients in k of degree at most $\binom{\deg(\chi)+1}{2}$.*

Proof. There are two possible sources of failures for Algorithm 2: if the vector $(1, -\widetilde{\lambda})$ is tangent to the curve $\mathcal{C}^0$ at one of the points in the support of the effective divisor (first test) or if $\widetilde{\lambda}X + Y$ is not a primitive element (second test).

Let $J = \langle \chi(\lambda X + Y), X - u(\lambda X + Y), Y - v(\lambda X + Y)\rangle \subset k[\mathcal{C}^0]$ be the ideal associated to the effective divisor D, and let Δ be the polynomial given by Lemma 8 for J. By construction, the polynomial Δ satisfies the wanted properties. $\square$

Before investigating Algorithms 4 and 5 (AddDivisors and SubtractDivisors), we need a technical lemma.

Lemma 30. *Let $\phi : R \to S$ be a surjective morphism of finite k-algebras, and let z be a primitive element for R. Then $\phi(z)$ is a primitive element for S.*

Proof. Since ϕ is surjective, any element $y \in S$ equals $\phi(x)$ for some $x \in R$. Since z is primitive, there exists a univariate polynomial $w(S) \in k[S]$ such that $x = w(z)$. Consequently, $y = \phi(w(z)) = w(\phi(z))$. Therefore, $\phi(z)$ is primitive for S. ☐

Proposition 31. *For a given input (q, D_1, D_2) of Algorithm 4 (AddDivisors), the set of values of $\widehat{\lambda}$ which make Algorithm 4 fail is contained in the set of roots of a nonzero univariate polynomial with coefficients in k and of degree bounded by $\binom{\deg(\chi_1)+\deg(\chi_2)+1}{2}$.*

Proof. For $i \in \{1, 2\}$, consider $I_i = \langle U - u_i(S), V - v_i(S), \chi_i(S)\rangle \cap k[U, V]$. Lemma 8 for the ideal $I_1 \cdot I_2 + \langle q(U, V)\rangle$ yields a nonzero polynomial Δ of degree at most $\binom{\deg(\chi_1)+\deg(\chi_2)+1}{2}$. We will prove that this polynomial satisfies the wanted properties. By definition, elements $\widehat{\lambda} \in k$ which are not roots of Δ are such that $\widehat{\lambda} U + V$ is a primitive element for $\mathrm{red}(k[U, V]/(I_1 \cdot I_2))$ and

$$
\begin{aligned}
\mathrm{GCD}\left(\frac{\partial q}{\partial X}(u_1(S), v_1(S)) - \widehat{\lambda}\frac{\partial q}{\partial Y}(u_1(S), v_1(S)), \chi_1(S)\right) &= 1,\\
\mathrm{GCD}\left(\frac{\partial q}{\partial X}(u_2(S), v_2(S)) - \widehat{\lambda}\frac{\partial q}{\partial Y}(u_2(S), v_2(S)), \chi_2(S)\right) &= 1.
\end{aligned} \tag{3}
$$

There are three possible sources of failure for Algorithm 4: the two calls to the algorithm ChangePrimElt, and the conditional test. The fact that the calls to ChangePrimElt succeed is a direct consequence of Lemma 30, using the canonical projections $\mathrm{red}(k[U, V]/(I_1 \cdot I_2)) \to \mathrm{red}(k[U, V]/I_i)$ for $i \in \{1, 2\}$; see the proof of Proposition 29. By Lemma 10 and Proposition 12, we have that for $i \in \{1, 2\}$, $I_i = \langle U - \widehat{u}_i(S), V - \widehat{v}_i(S), \widehat{\chi}_i(S)\rangle \cap k[U, V]$. Then $\widehat{\lambda} U + V$ must be a primitive element for $\mathrm{red}(k[U, V]/I_1)$ and for $\mathrm{red}(k[U, V]/I_2)$ by Lemma 9. Let $\widetilde{\chi}_1$ and $\widetilde{\chi}_2$ denote the minimal polynomials of $\widehat{\lambda} U + V$ in $\mathrm{red}(k[U, V]/I_1)$ and $\mathrm{red}(k[U, V]/I_2)$. Also, set $\xi = \mathrm{LCM}(\widetilde{\chi}_1, \widetilde{\chi}_2)$. Consequently, $\widetilde{\chi}_1(\widehat{\lambda} U+V)\cdot\widetilde{\chi}_2(\widehat{\lambda} U+V) \in \sqrt{I_1 \cdot I_2}$ and ξ is the minimal polynomial of $\widehat{\lambda} U+V$ in $\mathrm{red}(k[U, V]/(I_1\cdot I_2)) = k[U, V]/(\sqrt{I_1}\cap\sqrt{I_2})$. Since $\widehat{\lambda} U + V$ is a primitive element for $\mathrm{red}\,(k[U, V]/(I_1 \cdot I_2))$, then the canonical map

$$\mathrm{red}\,(k[U, V]/(I_1 \cdot I_2)) \to \mathrm{red}(k[U, V]/I_2) \times \mathrm{red}(k[U, V]/I_1)$$

becomes a map

$$k[S]/\xi(S) \to k[S]/\widetilde{\chi}_1(S) \times k[S]/\widetilde{\chi}_2(S).$$

This implies that there exists an element $\widetilde{u}_{12} \in k[S]/\xi(S)$ (which is in fact the class of U in $\mathrm{red}(k[U, V]/(I_1 \cdot I_2)))$ such that $\widetilde{u}_{12} \equiv \widehat{u}_1 \bmod \widetilde{\chi}_1$ and $\widetilde{u}_{12} \equiv \widehat{u}_2 \bmod \widetilde{\chi}_2$. As a consequence, $\widehat{u}_1 \equiv \widehat{u}_2 \bmod \mathrm{GCD}(\widetilde{\chi}_1, \widetilde{\chi}_2)$. Using Hensel's lemma, the property **(Div-H3)**, and the CRT, we obtain that the equation $q(u(S), S - \lambda u(S)) = 0$ has a unique solution s in $k[S]/\mathrm{GCD}(\widehat{\chi}_1(S), \widehat{\chi}_2(S))$ such that $s \equiv \widehat{u}_1 \equiv \widehat{u}_2 \bmod \mathrm{GCD}(\widetilde{\chi}_1, \widetilde{\chi}_2)$. By **(Div-H1)**, both $\widehat{u}_1$ and $\widehat{u}_2$ are solutions, and therefore $\widehat{u}_1 \equiv \widehat{u}_2 \bmod \mathrm{GCD}(\widehat{\chi}_1, \widehat{\chi}_2)$, which shows that the last conditional test succeeds. ☐

Proposition 32. *For a given input (q, D_1, D_2) of Algorithm* 5 (SUBTRACTDIVISORS)*, the set of values of $\widehat{\lambda}$ which make Algorithm* 5 *fail is contained in the set of roots of a nonzero univariate polynomial with coefficients in k and of degree bounded by $\binom{\deg(\chi_1)+\deg(\chi_2)+1}{2}$.*

Proof. The proof is similar to the first part of the proof of Proposition 31. With the same notation as in the proof of Proposition 31, Algorithm 5 fails only if $\widehat{\lambda}U + V$ is not a primitive element for $\mathrm{red}(k[U,V]/(I_1 \cdot I_2))$ or if the vector $(1, -\widetilde{\lambda})$ is tangent to the curve at one of the points in the support of one of the divisors. Using a proof similar to that of Proposition 31, this happens only when $\widehat{\lambda}$ is in the set of roots of the nonzero polynomial of degree at most $\binom{\deg(\chi_1)+\deg(\chi_2)+1}{2}$ provided by Lemma 8 for the ideal $I_1 \cdot I_2 + \langle q(U,V)\rangle$. □

Next, we wish to bound the probability that Algorithm 7 (COMPPRINCDIV) fails. Before stating the next proposition, we recall the second assumption that we have made on the input divisor and which is described in Section 2. It ensures the existence of a form $h \in \overline{k}[\mathcal{C}]$ of given degree such that $(h) \geq D_+ + E$ and $(h) - E$ is smooth. With the notation in the following proposition, this assumption precisely means that $A \neq \ker(\varphi) \otimes_k \overline{k}$.

Proposition 33. *Let $A \subset \ker(\varphi) \otimes_k \overline{k} \subset \overline{k}[X, Y]$ be the subset of all the regular functions h in the kernel of φ in Algorithm* 6 *which are such that $D_h = (h) - E$ is not a smooth divisor. If $A \neq \ker(\varphi)$, then A is contained in the join of at most $2r$ hyperplanes in $\ker(\varphi) \otimes_k \overline{k}$. Consequently, there is a nonzero polynomial in $\overline{k}[Z_1, \ldots, Z_{\dim(\ker(\varphi))}]$ of degree at most $2r$ which vanishes at values $(\mu_1, \ldots, \mu_{\dim(\ker(\varphi))})$ for which the third test in Algorithm* 7 *fails for all $\lambda \in \overline{k}$.*

Proof. If D_h involves a point P of $\widetilde{C}$ which projects to a node, then $(h) \geq E + P$. The set of regular functions h of a given degree which satisfy $(h) \geq E + P$ is a linear space. The set of such h in $\ker(\varphi) \otimes_k \overline{k}$ forms a proper subspace since $A \neq \ker(\varphi) \otimes_k \overline{k}$. Consequently, it is contained in an hyperplane. Such a hyperplane H can be described by a linear form ψ in $\overline{k}[Z_1, \ldots, Z_{\dim(\ker(\varphi))}]$ such that $\psi(\mu_1, \ldots, \mu_{\dim(\ker(\varphi))}) = 0$ if and only if $\sum_{i=1}^{\dim(\ker(\varphi))} \mu_i \mathbf{b}_i \in H$, where $\mathbf{b}_1, \ldots, \mathbf{b}_{\dim(\ker(\varphi))}$ is a basis of $\ker(\varphi)$.

Iterating this argument over all the $2r$ points of the nonsingular model $\widetilde{\mathcal{C}}$ which project to nodes, we obtain that A is contained in the join of $2r$ hyperspaces. Multiplying the $2r$ corresponding linear forms in $\overline{k}[Z_1, \ldots, Z_{\dim(\ker(\varphi))}]$ proves the last sentence of the proposition. □

Proposition 34. *The set of values of λ which make the first test in Algorithm* 7 *fail is contained within the set of roots of a nonzero univariate polynomial with coefficients in k of degree* $\deg(\mathcal{C}) + 1$.

Proof. Writing $\widetilde{q}(S, Y) = q((S - Y)/\lambda, Y)$, the first test fails if $\lambda = 0$ or if the coefficient of the monomial $Y^{\deg(\mathcal{C})}$ in $\widetilde{q}$ vanishes. Writing explicitly the change of variables, we obtain that this coefficient equals $\sum_{i=0}^{\deg(\mathcal{C})} (-1/\lambda)^i q_{i,\deg(\mathcal{C})-i}$, where $q_{i,j}$ stands for the coefficient of $X^i Y^j$ in q. Multiplying by $\lambda^{\deg(\mathcal{C})+1}$ clears the denominator and adds the root 0 to exclude the case $\lambda = 0$; this provides a polynomial satisfying the desired properties. □

Proposition 35. *Let $h \in k[\mathcal{C}^0]$ be a regular function such that the support of $(h) - E$ does not contain any singular point. Then the set of λ which makes Algorithm* 7 (CompPrincDiv) *with input q, h fail is contained in the set of roots of a nonzero univariate polynomial with coefficients in k and of degree bounded by $2\binom{\deg(\mathcal{C})\deg(h)+1}{2} + 2r + \deg(\mathcal{C}) + 1$.*

Proof. First, let Δ_1 be the univariate polynomial constructed in Proposition 34. The first test in Algorithm 7 does not fail only if λ is not a root of Δ_1.

The second test in Algorithm 7 fails only if λ is a root of T_E.

By the Bézout theorem, the effective divisor (h) has degree at most $\deg(\mathcal{C})\deg(h)$. Therefore, Lemma 7 for the ideal $\sqrt{\langle q, h\rangle}$ yields a nonzero polynomial Δ_2 of degree at most $\binom{\deg(\mathcal{C})\deg(h)}{2}$ such that the set of λ such that $\lambda X + Y$ is not a primitive element for $\mathrm{red}(k[\mathcal{C}^0]/\langle h\rangle)$. Since $(h) \geq E$, the fact that $\Delta_2(\lambda) \neq 0$ implies that $\lambda X + Y$ is a primitive element for the k-algebra associated to the nodal divisor, and hence the call to the function ChangePrimEltNodal in Algorithm 7 does not fail. Since by assumption $(h) - E$ is smooth, this also implies that the roots of $\widehat{\chi}_E$ are roots of χ with multiplicity exactly 2 by Lemma 19. Consequently, if λ is not a root of Δ_2, then $\mathrm{GCD}(\chi, \widehat{\chi}_E) = 1$ and therefore the third test in Algorithm 7 must succeed.

Finally, Lemma 8 for the ideal $\sqrt{\langle q, h\rangle} : I_E^\infty \subset k[\mathcal{C}^0]$ yields a nonzero polynomial Δ_3 of degree at most $\binom{\deg(\mathcal{C})\deg(h)+1}{2}$ such that the set of λ such that $\lambda X + Y$ is not a primitive element for $\mathrm{red}(k[\mathcal{C}^0]/\langle h\rangle)$ or such that the last test in Algorithm 7 fails is contained within the set of roots of Δ_3.

We claim that the product $\Delta_1 \cdot \Delta_2 \cdot \Delta_3 \cdot T_E$ satisfies the required properties. To prove this claim, it remains to show that if λ is not a root of $\Delta_1 \cdot \Delta_2 \cdot \Delta_3 \cdot T_E$, then the fourth test succeeds, i.e., $a_1(S)$ is invertible modulo $\chi(S)$.

To this end, we notice that $a_1(S)$ is invertible modulo $\chi(S)$ if and only if $a_1(s)$ is nonzero for any root $s \in \overline{k}$ of $\chi(S)$. By [8, Cor. 5.1], this is equivalent to the fact that the GCD of the polynomials $q((s-Y)/\lambda, Y)$, $h((s-Y)/\lambda, Y)$ has degree 1 for any root s of $\chi(S)$. Next, we note that if λ is not a root of Δ_3, then any common root y of $q((s-Y)/\lambda, Y)$ and $h((s-Y)/\lambda, Y)$ has multiplicity 1 in $q((s-Y)/\lambda, Y)$: Indeed, the vanishing of the derivative $\partial/\partial Y$ of $q((s-Y)/\lambda, Y)$ at $Y = y$ would precisely mean that the vector $(1, -\lambda)$ is tangent to the curve at the intersection point, which is impossible by definition of Δ_3. Consequently, the GCD of the polynomials $q((s-Y)/\lambda, Y)$, $h((s-Y)/\lambda, Y)$ must be squarefree. Finally, let $y_1, y_2 \in \overline{k}$ be two common roots of $q((s-Y)/\lambda, Y)$, $h((s-Y)/\lambda, Y)$. This means that $((s-y_1)/\lambda, y_1)$ and $((s-y_2)/\lambda, y_2)$ are two common zeros of $q(X, Y)$ and $h(X, Y)$. Since λ is not a root of Δ_2, $\lambda X + Y$ is a primitive element for $\mathrm{red}(k[\mathcal{C}^0]/\langle h\rangle) = k[X, Y]/\langle q, h\rangle$, which implies that $\lambda X + Y$ takes distinct values at all points (x, y) in the variety associated to the system $h(X, Y) = q(X, Y) = 0$ (the endomorphism of multiplication by $\lambda X + Y$ must have distinct eigenvalues; see, e.g., the proof of Lemma 7 for more details). In particular, this means that $y_1 = y_2$, since $\lambda X + Y$ takes the same value s at $((s-y_1)/\lambda, y_1)$ and $((s-y_2)/\lambda, y_2)$. Consequently, the GCD of the polynomials $q((s-Y)/\lambda, Y)$, $h((s-Y)/\lambda, Y)$ is a squarefree polynomial with at most one root, hence it has degree at most 1. Since $\mathrm{Resultant}(q((s-Y)/\lambda, Y), h((s-Y)/\lambda, Y))$ vanishes and the coefficient of $Y^{\deg(q)}$ in $q((S-Y)/\lambda, Y)$ is nonzero because λ is not a root of Δ_1, this GCD must have degree at least 1. Therefore, this GCD has degree exactly 1, and hence $a_1(S)$ is invertible modulo $\chi(S)$. □

Finally, we can derive our bound on the probability that the toplevel algorithm fails by summing the probabilities that the subroutines fail.

Theorem 36. *Let $\mathcal{E} \subset k$ be a finite set. Assume that each call to the function* Random(k) *is done by picking an element uniformly at random in $\mathcal{E}$. Then the probability that Algorithm* 1 *fails is bounded above by*

$$O(\max(\deg(\mathcal{C})^4, \deg(D_+)^2)/|\mathcal{E}|).$$

Proof. Propositions 31, 32, together with the fact that the number of roots in k of a univariate polynomial is bounded by its degree, directly imply that the probabilities of failure of Algorithms AddDivisors and SubtractDivisors are bounded by $O(\max(\deg(D_1), \deg(D_2))^2/|\mathcal{E}|)$, if the computation of the characteristic polynomial in Algorithm ChangePrimElt succeeds. Following [21] (see also the remark in the proof of Proposition 21), the probability that the computation of the characteristic polynomial in ChangePrim fails is bounded by $\deg(\chi)^2/|\mathcal{E}|$. Therefore, the probabilities that Algorithms AddDivisors and SubtractDivisors fail are still bounded by $O(\max(\deg(D_1, D_2))^2/|\mathcal{E}|)$ when we take into account the probability that the computations of the characteristic polynomials fail. Notice that our second technical assumption (described in Section 2) on the input divisor ensures that $A \neq \ker(\varphi)$ in Proposition 33. Using Proposition 33, the Schwartz-Zippel lemma [22, Coro. 1], Proposition 35, together with the fact that $r \leq \binom{\deg(\mathcal{C})-1}{2}$, we bound the probability that CompPrincDiv fails by $O(\deg(\mathcal{C})^2 \deg(h)^2/|\mathcal{E}|)$.

The failure probabilities are summed up in Table 2. Next, notice that the probability of failure of Algorithm 1 is bounded by the sum of the probabilities of the subroutines. Finally, the proof is concluded by using the inequality $\deg(h) < (\deg(D_+) + r)/\deg(\mathcal{C}) + \deg(\mathcal{C})$ (Proposition 25) and the degree bounds in Table 1 for the divisors arising in Algorithm 1. □

Table 2. Probabilities of failure.

Algorithm	Failure probability	Statement
ChangePrimElt	$\deg(D)^2/\lvert\mathcal{E}\rvert$	Prop. 29
AddDivisors	$O(\max(\deg(D_1), \deg(D_2))^2/\lvert\mathcal{E}\rvert)$	Prop. 31
SubtractDivisors	$O(\max(\deg(D_1), \deg(D_2))^2/\lvert\mathcal{E}\rvert)$	Prop. 32
CompPrincDiv	$O(\deg(\mathcal{C})^2 \deg(h)^2/\lvert\mathcal{E}\rvert)$	Prop. 33 Prop. 35 Schwartz-Zippel lemma

Deciding whether the assumptions on the input divisor are satisfied. The result in Theorem 36 only holds true if the assumptions on the input divisor described in Section 2 are satisfied. The first assumption—namely, the smoothness of the input divisor—can be easily checked, so we focus here on deciding whether the second assumption is satisfied or not. Namely, this assumption requires the existence of a form $h \in \overline{k}[\mathcal{C}]$ of degree d—where d is the value computed during the execution of Algorithm INTERPOLATE—such that $(h) \geq D_+ + E$ and $(h) - E$ is smooth. In order to have a complete Las Vegas algorithm, we need to be able to check whether this condition is satisfied. To this end, instead of returning only one form during Algorithm INTERPOLATE, we can return a basis $(h_1, \dots, h_\ell)$ of the forms h such that $(h) \geq D_+ + E$. Then, we check if there exists a point above a node which is simultaneously in the support of all the divisors $\{(h_i)-E\}_{i\in\{1,\dots,\ell\}}$. This boils down to computing primitive element representations of the principal divisors $(h_1), \dots, (h_\ell)$, which is done by running on these ℓ forms a modified version of Algorithm COMPPRINCDIV where the last test is removed in order to allow singular points. Each execution of Algorithm COMPPRINCDIV costs $\widetilde{O}(\max(\deg(\mathcal{C})^3, (\deg(D_+)+r)^2/\deg(\mathcal{C})))$ operations in k, and thus—using the fact that $\ell = O(\deg(D_+) + \deg(\mathcal{C})^2)$—the total cost of the procedure is bounded above by $\widetilde{O}(\max(\deg(\mathcal{C})^5, \deg(D_+)^{5/2}))$. Therefore, in theory, running this decision procedure increases the overall complexity stated in Theorem 28 since the best known value of ω is less than $5/2$. However, in practice this does not change the asymptotic complexity since practical algorithms for linear algebra rely on Gauss or Strassen approaches; In this case, $\omega > 5/2$, and hence the cost of this verification procedure is negligible compared to the global complexity of our algorithm. Multiplying the probability of failure of Algorithm COMPPRINCDIV by the number of basis vectors yields the bound $O(\max(\deg(D_+)^3, \deg(\mathcal{C})^6)/|\mathcal{E}|)$ for the probability of failure of this verification procedure.

8. Experimental results

We have implemented Algorithm 1 in C++ for $k = \mathbb{Z}/p\mathbb{Z}$, relying on the NTL library for all operations on univariate polynomials and for linear algebra. We have also implemented the group law on the Jacobian of a curve via Riemann-Roch space computations. Our software `rrspace` is freely available at `https://gitlab.inria.fr/pspaenle/rrspace` and it is distributed under the LGPL-2.1+ license.

All the experiments presented below have been conducted on an Intel(R) Core(TM) i5-6500 CPU@3.20GHz with 16GB RAM. The comparisons with the computer algebra system Magma have been done with its version V2.23-8.

Our first experimental data is generated as follows. We set $k = \mathbb{Z}/65521\mathbb{Z}$. For i from 10 to 100, we consider a curve $\mathcal{C}$ defined by a random bivariate polynomial of degree 10 over k, and we generate i random irreducible k-defined effective divisors $D_1, \dots, D_i$ of degree 10 on $\mathcal{C}$ by using the `RandomPlace()` function in `Magma`. Then we set $D = D_1 + \cdots + D_i$ and we measure the time used for computing a basis of $L(D)$ by using either `Magma` via its function `RiemannRochSpace()` or the software `rrspace`. The experimental results are displayed in the left part of Figure 1. For these parameters, we observe that `rrspace` has a speed-up larger than 7 compared to `Magma`. Since we do not have access to the implementation of the function `RiemannRochSpace()` in `Magma`, we cannot explain the small variations which appear in the `Magma` timings.

Our second experimental data investigate the behavior of our algorithm when the input divisor contains multiplicities. To this end, we generate the input divisor as a multiple of a random place of degree 10 on the curve. The experimental results are displayed in the right part of Figure 1. For these parameters, we observe that `rrspace` has a speed-up larger than 6 compared to `Magma`.

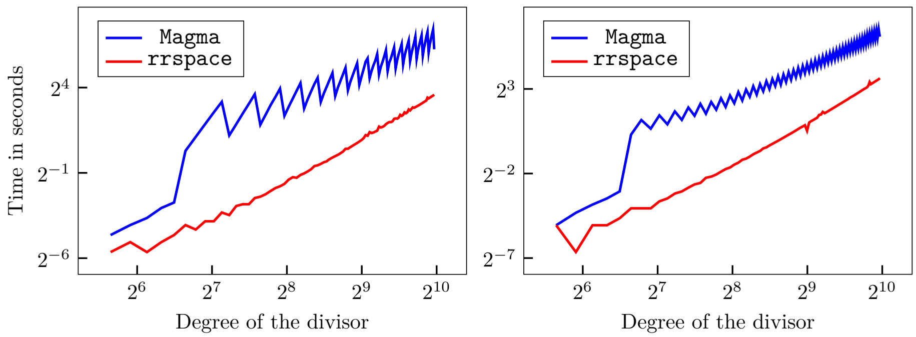

FIGURE 1. Comparison of the time required by `rrspace` and `Magma` to compute a basis of $L(D)$ on a fixed smooth curve of degree 10 over $\mathbb{Z}/65521\mathbb{Z}$. On the left, D is the sum of random irreducible effective divisors of degree 10. On the right, D is a multiple of an irreductible divisor of degree 10. Both axes are in logarithmic scale.

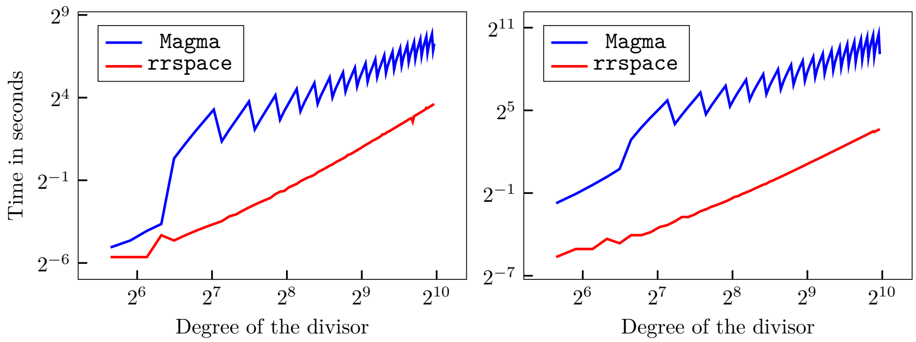

FIGURE 2. Comparison of the time required by `rrspace` and `Magma` to compute a basis of $L(D)$ on a fixed curve of degree 10, where D is the sum of random irreducible effective divisors of degree 10. On the left, the base field is $\mathbb{Z}/65521\mathbb{Z}$ and the curve is nodal. On the right, the base field is $\mathbb{Z}/(2^{32}-5)\mathbb{Z}$ and the curve is smooth. Both axes are in logarithmic scale.

Our third experimental data study the behavior of our algorithm in the presence of nodes. To this end, we fix the following nodal curve defined by the equation

$$Q(X,Y,Z) = -Y^2Z^8 + X^2Z^8 + Y^4Z^6 - X^3Z^7 + X^{10} - 5\,Y^{10} + 3\,X^3Y^7$$

which has a node at the origin and we generate input divisors as for the first experimental data. The experimental results are displayed in the left part of Figure 2. For these parameters, we observe that `rrspace` has a speed-up larger than 10 compared to `Magma`.

Finally, since the timings are very sensible to the efficiency of the linear algebra routines, we study what happens for larger finite fields. The fourth experimental data are generated as for our first experimental data, but we replace the field $\mathbb{Z}/65521\mathbb{Z}$ by the field $\mathbb{Z}/(2^{32}-5)\mathbb{Z}$. Here, the size of the field is out of the range of the highly optimized arithmetic in Magma for small finite fields, and consequently we observe speed-ups larger than 45 (the speed-up goes up to more than 200 for some examples). The experimental results are displayed in the right part of Figure 2.

Acknowledgments

We are grateful to Simon Abelard, Pierrick Gaudry, Emmanuel Thomé, and Paul Zimmermann for useful discussions and for pointing out important references. We thank Pierrick Gaudry for allowing us to use his code for the fast computation of resultants and subresultants of univariate polynomials. We are also grateful to an anonymous referee who helped us improve the paper.

References

[1] E. Arbarello, M. Cornalba, and P. A. Griffiths, *Geometry of Algebraic Curves. Volume II*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 268, Springer, Heidelberg, 2011. With a contribution by Joseph Daniel Harris. MR2807457

[2] M. F. Atiyah and I. G. Macdonald, *Introduction to Commutative Algebra*, Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969. MR0242802

[3] L. Babai, *Monte-Carlo algorithms in graph isomorphism testing*, Université de Montréal Technical Report, DMS (1979), no. 79-10.

[4] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, DOI 10.1006/jsco.1996.0125. Computational algebra and number theory (London, 1993). MR1484478

[5] J. R. Bunch and J. E. Hopcroft, *Triangular factorization and inversion by fast matrix multiplication*, Math. Comp. **28** (1974), 231–236, DOI 10.2307/2005828. MR331751

[6] J. Canny, *Some algebraic and geometric computations in PSPACE*, Proc. of the twentieth annual ACM Symposium on Theory of Computing (STOC), ACM, 1988, pp. 460–467.

[7] D. G. Cantor and E. Kaltofen, *On fast multiplication of polynomials over arbitrary algebras*, Acta Inform. **28** (1991), no. 7, 693–701, DOI 10.1007/BF01178683. MR1129288

[8] M. El Kahoui, *An elementary approach to subresultants theory*, J. Symbolic Comput. **35** (2003), no. 3, 281–292, DOI 10.1016/S0747-7171(02)00135-9. MR1962796

[9] W. Fulton, *Algebraic Curves. An Introduction to Algebraic Geometry*, W. A. Benjamin, Inc., New York-Amsterdam, 1969. Notes written with the collaboration of Richard Weiss; Mathematics Lecture Notes Series. MR0313252

[10] M. Giusti, G. Lecerf, and B. Salvy, *A Gröbner free alternative for polynomial system solving*, J. Complexity **17** (2001), no. 1, 154–211, DOI 10.1006/jcom.2000.0571. MR1817612

[11] V. D. Goppa, *Algebraic-geometric codes* (Russian), Izv. Akad. Nauk SSSR Ser. Mat. **46** (1982), no. 4, 762–781, 896. MR670165

[12] G. Haché, *Computation in algebraic function fields for effective construction of algebraic-geometric codes*, Applied algebra, algebraic algorithms and error-correcting codes (Paris, 1995), Lecture Notes in Comput. Sci., vol. 948, Springer, Berlin, 1995, pp. 262–278, DOI 10.1007/3-540-60114-7_19. MR1448169

[13] G. Haché and D. Le Brigand, *Effective construction of algebraic geometry codes*, IEEE Trans. Inform. Theory **41** (1995), no. 6, 1615–1628, DOI 10.1109/18.476233. Special issue on algebraic geometry codes. MR1391019

[14] F. Hess, *Computing Riemann-Roch spaces in algebraic function fields and related topics*, J. Symbolic Comput. **33** (2002), no. 4, 425–445, DOI 10.1006/jsco.2001.0513. MR1890579

[15] M.-D. Huang and D. Ierardi, *Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve*, J. Symbolic Comput. **18** (1994), no. 6, 519–539, DOI 10.1006/jsco.1994.1063. MR1334660

[16] J.-P. Jouanolou, *Le formalisme du résultant* (French), Adv. Math. **90** (1991), no. 2, 117–263, DOI 10.1016/0001-8708(91)90031-2. MR1142904

[17] K. Khuri-Makdisi, *Asymptotically fast group operations on Jacobians of general curves*, Math. Comp. **76** (2007), no. 260, 2213–2239, DOI 10.1090/S0025-5718-07-01989-8. MR2336292

[18] D. Le Brigand and J.-J. Risler, *Algorithme de Brill-Noether et codes de Goppa* (French, with English summary), Bull. Soc. Math. France **116** (1988), no. 2, 231–253. MR971561
[19] F. Le Gall, *Powers of tensors and fast matrix multiplication*, ISSAC 2014—Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2014, pp. 296–303, DOI 10.1145/2608628.2608664. MR3239939
[20] V. Neiger, J. Rosenkilde, and É. Schost, *Fast computation of the roots of polynomials over the ring of power series*, ISSAC'17—Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2017, pp. 349–356. MR3703706
[21] C. Pernet and A. Storjohann, *Faster algorithms for the characteristic polynomial*, ISSAC 2007, ACM, New York, 2007, pp. 307–314, DOI 10.1145/1277548.1277590. MR2402276
[22] J. T. Schwartz, *Probabilistic algorithms for verification of polynomial identities*, Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979), Lecture Notes in Comput. Sci., vol. 72, Springer, Berlin-New York, 1979, pp. 200–215. MR575691
[23] F. Severi, *Vorlesungen über algebraische Geometrie: Geometrie auf einer Kurve, Riemannsche Flächen, Abelsche Integrale* (German), Berechtigte Deutsche Übersetzung von Eugen Löffler. Mit einem Einführungswort von A. Brill. Begleitwort zum Neudruck von Beniamino Segre. Bibliotheca Mathematica Teubneriana, Band 32, Johnson Reprint Corp., New York-London, 1968. MR0245574
[24] A. Storjohann, *Algorithms for matrix canonical forms*, Ph.D. thesis, ETH Zurich, 2000.
[25] J. V. Der Hoeven and G. Lecerf, *Fast computation of generic bivariate resultants*, Preprint, 2019.
[26] G. Villard, *On computing the resultant of generic bivariate polynomials*, ISSAC'18—Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2018, pp. 391–398. MR3840406
[27] E. J. Volcheck, *Computing in the Jacobian of a plane algebraic curve*, Algorithmic number theory (Ithaca, NY, 1994), Lecture Notes in Comput. Sci., vol. 877, Springer, Berlin, 1994, pp. 221–233, DOI 10.1007/3-540-58691-1_60. MR1322725
[28] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed., Cambridge University Press, Cambridge, 2013. MR3087522

CARAMBA project, Université de Lorraine; and Inria Nancy – Grand Est; and CNRS, UMR 7503, LORIA, Nancy, France
Email address: aude.le-gluher@loria.fr

CARAMBA project, INRIA Nancy – Grand Est; and Université de Lorraine; and CNRS, UMR 7503, LORIA, Nancy, France
Email address: pierre-jean.spaenlehauer@inria.fr