FACTORING POLYNOMIALS OVER FUNCTION FIELDS

JOSÉ FELIPE VOLOCH

Abstract. If K/k is a function field in one variable of positive characteristic, we describe a general algorithm to factor one-variable polynomials with coefficients in K. The algorithm is flexible enough to find factors subject to additional restrictions, e.g., to find all roots that belong to a given finite dimensional k-subspace of K, more efficiently. For bounded characteristic, it runs in polynomial time, relative to factorizations over the constant field k and also provides a deterministic polynomial time irreducibility test. We also discuss applications to places of reducible reduction, when k is a global field, and to list decoding of Reed-Solomon codes.

1. Introduction

Let K/k be a function field in one variable, that is, a finitely generated extension of transcendence degree one with k algebraically closed in K. Let $G(T)$ be a polynomial in one variable over K. The algorithmic problem of finding the irreducible factors of $G(T)$ in $K[T]$ and, in particular its roots in K, is a much-studied problem with many applications, see e.g. [vzGK85, Poh05, BvHKS09] and the references therein. A noteworthy special case is the case where $K = k(x)$, the rational function field, and the coefficients of $G(T)$ are in $k[x]$. This case corresponds to factoring polynomials in two variables with coefficients in k. Some recent papers representing the state of the art of this special case are [Lec10, Wei17].

Many of the applications of the above problem actually require the solution of a more restricted problem. For instance, given $G(T)$ and a finite dimensional k-subspace V of K, find the roots of $G(T)$ in V. An example of an application where this restricted problem suffices is the Guruswami–Sudan list-decoding algorithm. See [GS00] for a comprehensive discussion and [BLQ13, NRS17] for other approaches to this problem. We discuss an instance of this in subsection 5.2 below.

Throughout this paper k has characteristic $p > 0$ and we make the assumption that the polynomial $G(T)$ to be factored is separable. The reduction to this case is a standard first step in all algorithms and is presented in the above cited papers. We also assume $G(0) \neq 0$. This paper describes an algorithm that solves the general problem of factoring such a $G(T)$. Additionally, the algorithm has improved performance when applied to the more restricted problems described above. Indeed, we will describe an algorithm that finds a factor of $G(T)$ of prescribed degree whose coefficients are on prescribed finite dimensional k-subspaces of K. We prove that, assuming the characteristic is bounded, our algorithm provides a deterministic polynomial time absolute irreducibility test and, up to factorization of polynomials in $k[T]$, the algorithm runs in deterministic polynomial time. See Theorem 3.2, Remark 3.3 and Section 4 below for precise statements and discussion. We also discuss some other applications of the ideas in this paper in Section 5. The approach is novel and

2020 *Mathematics Subject Classification.* Primary: 12-08 ; Secondary: 11R09.

Key words and phrases. Polynomial factorization, function fields, irreducibility test, list decoding.

Submitted to Algor. Num. Th. Symp.

relies on casting the problem of finding a factor of $G(T)$ as finding a k-linear dependence relation among some elements of a quotient ring of $K[T]/(G(T))$ and applying a linear independence criterion over k involving Wronskian matrices.

Our algorithm, like most standard algorithms, needs at the beginning, a place of the function field K with a few additional properties. In our presentation of the main part of the algorithm, in Section 3, we just assume its existence. In the subsequent Section 4 we discuss how to find such a place. This seems to, unavoidably, require a search.

Most of the standard algorithms then proceed to compute a complete factorization of the image of the polynomial when specialized to the residue field of the place just discussed (see the description of a generic factorization algorithm in [Poh05]). This is known as the lifting and recombination strategy. The factorization step in the residue field is often easy in practice but can be difficult in certain circumstances (see the beginning of Section 2). These algorithms also often have a bottleneck reconstructing global factorizations from local ones. In contrast, our algorithm does not need to compute this factorization in the residue field at the beginning, nor tries to reconstruct global factorizations from local ones. Instead, it may do a partial factorization during intermediate steps using easy gcd computations. At the end, our algorithm may need to further factor some of these partial factors. If the objective is an irreducibility criterion, it does not require finding such a factorization at all.

There are two algorithms in the literature that use differential operators to factor polynomials and don't follow the lifting and recombination strategy.

The first is the algorithm of [Rup99, Gao03] that uses a certain first order partial differential equation and shares some of the advantages of our approach. It only applies however when K is a rational function field. One application of this approach [Rup99, Theorem pg. 63] is to bound the size of the largest prime p for which an irreducible poynomial in $\mathbb{Z}[x, y]$ factors modulo p. Our approach allows us to obtain similar bounds in full generality when k itself is a global field.

The second is the algorithm of [CSTU02] which uses the linear differential operator annihilating all roots of $G(T)$ (and also uses Wronskians). This algorithm is restricted to characteristic zero as the crucial [CSTU02, Proposition 4.2] does not extend to positive characteristic. Indeed, the proposition there is about the k-dimension of the set of solutions in K of the linear differential operator annihilating roots of $G(T)$ (assumed irreducible). In characteristic p, the set of solutions of a linear differential operator of order $< p$ in a field E/K is a E^p-vector space. But even considering E^p-dimension, the set of solutions in K has the same K^p-dimension as the E^p-dimension of the set of solutions in the splitting field E of $G(T)$, as follows from [Hon81, Lemma 1], as opposed to having dimension 1, which would be the analogue of the characteristic zero result.

2. Preliminaries

Let K/k be a function field in one variable, that is, a finitely generated extension of transcendence degree one with k algebraically closed in K. Factoring polynomials in $K[T]$ includes factoring polynomials in $k[T]$ and the latter could be hard, depending on k. For example, if k is a finite field or a number field, the existence of a polynomial time deterministic factoring algorithm for $k[T]$ is an open problem, although there are polynomial time probabilistic algorithms that also perform well in practice. If k is the field of real numbers,

Submitted to Algor. Num. Th. Symp.

there may be issues with precision. Our goal, therefore, is to present an algorithm to factor polynomials in $K[T]$ relative to factoring polynomials in $k[T]$.

Also, in discussing the running time of the algorithm, the estimates will be in terms of the number of field operations in k or finite extensions thereof.

2.1. **The basic rings.** Let v be a place of K with ring of integers $\mathcal{O}$ and maximal ideal $\mathfrak{m}$. If q is a power of the characteristic p of K, we consider the ring $\mathcal{O}/\mathfrak{m}^q$. We have that the completion of $\mathcal{O}$ is isomorphic to $\ell[[x]]$, where $\ell = \mathcal{O}/\mathfrak{m}$ and x is an uniformiser of v, that is, an element of $\mathfrak{m} \setminus \mathfrak{m}^2$. It follows that $\mathcal{O}/\mathfrak{m}^q$ is isomorphic to $\ell[[x]]/(x^q)$.

Let $G(T) \in \mathcal{O}[T]$ be a monic polynomial such that $G \pmod v$ is separable.

The ring $R_1 = (\mathcal{O}/\mathfrak{m}^q)[T]/(G(T))$ is an Artinian ring and, thus, a direct sum of Artinian local rings. These summands correspond to irreducible factors of $G(T)$ in $(\mathcal{O}/\mathfrak{m}^q)[T]$ which, in turn, correspond to irreducible factors of $G(T) \in (\mathcal{O}/\mathfrak{m})[T]$. The standard factorization version of Hensel's lemma gives an algorithmic way to go from the latter to the former.

2.2. **Gaussian elimination.** In this subsection, we describe a Gaussian elimination procedure on an Artinian ring A which is a sum of finitely many Artinian local rings (such as R_1 above). In addition to the usual row reduction steps (described explicitly below), the algorithm will also involve splitting the ring in a direct sum of two rings and branching the algorithm to each summand. To ease notation, we continue to denote by A any such summand. The final output of the procedure will be a decomposition of the original ring into a direct sum of rings and, for each such direct summand, a matrix with entries in the corresponding subring. Moreover, for each maximal ideal of this subring, the image of the matrix in the quotient field will be row reduced and will have the pivots in the same place. In particular, the rank of the reduction will be independent of the maximal ideal of this subring and will be simply called the rank of the matrix.

The procedure is as follows, we scan each column in turn, looking for an element that is a unit in some summand of A. If none is found, we skip the column. If one is found, we split A as a sum of two rings, one of which is the maximal summand where the element is a unit. In that summand we use the unit as a pivot (and in the other summand we move on to the next entry). Namely, we replace the other rows by the appropriate multiple of the row of the pivot so that the entry in the column of the pivot is 0. Finally, we multiply the row of the pivot by its inverse, so the pivot is replaced by 1. The usual analysis of Gaussian elimination justifies the claimed output. Indeed, we are just doing Gaussian elimination in the various quotient fields simultaneously to the extent possible and decomposing the ring as a direct sum, when it's not possible.

Finally, we note that the direct sum decomposition steps in the above procedure can be done explicitly as follows in the case of R_1. Namely, if the entry in the matrix being inspected is $P(T)$, we compute $H_0(T) = \gcd(G(T), P(T))$ and $E_0(T) = G(T)/H_0(T)$ in $(\mathcal{O}/\mathfrak{m})[T]$. If $E_0(T) \neq 1$, we lift the factorization $G(T) = H_0(T)E_0(T)$ in R_0 to a factorization $G(T) = H(T)E(T)$ in R_1 and decompose R_1 as the direct sum of $(\mathcal{O}/\mathfrak{m}^q)[T]/(H(T))$ and $(\mathcal{O}/\mathfrak{m}^q)[T]/(E(T))$. This works because we assume that $G(T)$ is squarefree in $(\mathcal{O}/\mathfrak{m})[T]$.

2.3. **Hasse derivatives and Wronskians.** We begin by presenting some concepts and results from [Sch39, SV86]. See also [GV87] which proves stronger versions of the main results of [Sch39] and may be more accessible, as well as [Hes02a, Section 6.1] which describes relevant algorithms (particularly algorithms 26 and 28 there). Let K/k be a function field.

Submitted to Algor. Num. Th. Symp.

The usual higher derivatives do not work well in small characteristics. A suitable replacement for higher derivatives that work in general are the Hasse derivatives. We denote by $D^{(i)}, i = 0, 1, \ldots,$ the Hasse derivatives with respect to some separating variable x on K. These are k-linear operators on K with $i!D^{(i)} = (d/dx)^i$ and satisfying:

$$\begin{aligned} D^{(i)} \circ D^{(j)} &= \binom{i+j}{j} D^{(i+j)}, \\ D^{(i)}(uv) &= \sum_{j=0}^{i} D^{(j)}(u) D^{(i-j)}(v). \end{aligned} \tag{2.1}$$

The second formula is an extension of the Leibniz rule. They are defined first in $k[x]$ by setting $D^{(i)}x^n = \binom{n}{i}x^{n-i}$ and extending k-linearly. These operators then extend uniquely to $k(x)$ and any separable extension thereof by requiring that they satisfy the formulas 2.1 above.

By [Sch39, Satz 2], $f_0, \ldots, f_m \in K$ are linearly independent over k if and only if the Wronskian matrix $(D^{(i)}(f_j))$ has maximal rank $m+1$. In this case, there is a (lexicographically) minimal list of integers $0 = \varepsilon_0 < \cdots < \varepsilon_m$ such that the matrix $(D^{(\varepsilon_i)}(f_j))$ has maximal rank $m+1$. We have $\varepsilon_i = i$ if the characteristic is zero or large enough but that is not going to be the situation in this paper. Also, to a place v of K, we can associate a (lexicographically) minimal list of integers $0 = j_0 < \cdots < j_m$ (called the Hermitian invariants and which depend on v) such that the matrix $(D^{(j_i)}(f_j) \pmod v)$ has maximal rank $m+1$. Moreover, if x is a local parameter at v, there exists a linear transformation of the space spanned by the f_i over the residue field of v that transforms f_i into a basis (called an Hermitian basis) g_i with $g_i = x^{j_i} +$ higher order terms ([Sch39, pg. 68]).

If K is the function field of an algebraic curve Y, then the morphism $(f_0 : \ldots : f_m) : Y \to \mathbb{P}^m$ has some degree Δ and $\varepsilon_i \le j_i \le \Delta$ when the f_i are linearly independent ([SV86, Section 2],[Hes02a, Prop. 13]). On the other hand, if the Wronskian matrix has rank m and $a_0, \ldots, a_m \in K$ satisfy $\sum_{j=0}^{m} a_j D^{(i)}(f_j) = 0, i = 0, 1, 2, \ldots$ and $a_0 = 1$, then $a_j \in k, j = 0, 1, \ldots, m$ as follows from the proof of [Sch39, Satz 1].

Consider a monic, separable polynomial

$$G(T) = \sum_{j=0}^{s} a_j T^j, a_j \in K, a_s = 1. \tag{2.2}$$

Let $R = K[T]/G(T)$ and t the image of T in R. We extend the operators $D^{(i)}, i \ge 0$ to R. We need an expression for $D^{(i)}(t)$. We have that

$$0 = D^{(i)}(G(t)) = \sum_{j_1+2j_2+\cdots+ij_i \le i} A_{j_1,\ldots,j_i} (D^{(1)}(t))^{j_1} \cdots (D^{(i)}(t))^{j_i} \tag{2.3}$$

where the $A_{j_1,\ldots,j_i}$ are polynomials in t and, in particular, $A_{0,\ldots,0,1} = G'(t)$ which is invertible in R and this determines $D^{(i)}(t)$ uniquely, by induction. They can be computed more efficiently by the algorithms of [Hes02a].

If v is a place of K with ring of integers $\mathcal{O}$ and maximal ideal $\mathfrak{m} = (x)$ then, for any q power of p, the operators $D^{(i)}, i < q$, defined using x as the separating variable, induce operators on $\mathcal{O}/\mathfrak{m}^q$. Indeed, $D^{(i)}(x^q) = 0, i < q$, so $D^{(i)}(x^q y) = x^q D^{(i)}(y), i < q$ for any $y \in \mathcal{O}$ and it follows that $D^{(i)}, i < q$ preserve $\mathfrak{m}^q$. In our main algorithm, we will choose q

Submitted to Algor. Num. Th. Symp.

large enough so that the non-zero elements of K appearing in the course of the computation have degree smaller than q, so, if the computation produces an element in $\mathfrak{m}^q$, it has to be the zero element.

Now, if $G(T) \in \mathcal{O}[T]$ is as above and is such that $G \pmod v$ is separable, then the operators $D^{(i)}, i < q$ induce operators on $R_1 = (\mathcal{O}/\mathfrak{m}^q)[T]/(G(T))$ by the same formulas. As mentioned before, R_1 is an Artinian ring and, thus, a direct sum of Artinian local rings. For each irreducible factor H_0 of $G \pmod{\mathfrak{m}}$, there is a corresponding factor H of $G \pmod{\mathfrak{m}^q}$ and a corresponding summand $(\mathcal{O}/\mathfrak{m}^q)[T]/(H(T))$ of R_1. If $\mathfrak{m} = (x)$, then $\mathcal{O}/\mathfrak{m}^q$ is isomorphic to $(\mathcal{O}/\mathfrak{m})[X]/(X^q)$ and $(\mathcal{O}/\mathfrak{m}^q)[T]/(H(T))$ is isomorphic to $((\mathcal{O}/\mathfrak{m})[T]/(H_0(T)))[X]/(X^q)$ and we refer to the subring $(\mathcal{O}/\mathfrak{m})[T]/(H_0(T))$ of this latter ring as its constants.

Lemma 2.1. *If $u \in R_1$ satisfies $D^{(i)}(u) = 0, 0 < i < q$, then in each local summand of R_1, u is constant (in the above sense).*

Proof. Each summand is isomorphic to $\ell[X]/(X^q)$ for some field ℓ and $D^{(i)}(x^r) = \binom{r}{i} x^{r-i}$, where x is the image of X, so it is clear that $D^{(i)}(u) = 0, 0 < i < q$ if and only if $u \in \ell$. □

Remark 2.2. *In the process of algorithm 1 below, we will decompose R_1 as a direct sum but we may not go all the way to the full decomposition as a sum of Artinian local rings.*

3. The main algorithm

Our main algorithm is described below as Algorithm 1. As mentioned in the introduction, it takes as input a polynomial $G(T) \in K[T]$ and a collection of finite dimensional k-vector spaces $V_i \subset K, i = 0, \dots r-1$ and decides whether there exists a factor $H(T)$ of $G(T)$ of the form $H(T) = \sum_{i=0}^{r} b_i T^i, b_i \in V_i, i < r, b_r = 1$. Given k-bases $\{h_{ij}\}$ for the V_i, the algorithm proceeds by computing the Wronskian matrix of the functions $h_{ij}T^i$ in a suitable ring where $G(T) = 0$ and doing Gaussian elimination on this matrix, in the sense of Subsection 2.2, identifying a k-linear dependence among the $h_{ij}T^i$.

To obtain a full factorization algorithm (where the V_i are not necessarily given in advance), we prove lemma 3.1 below (a variant of [Poh05, Lemma 4.1]).

Lemma 3.1. *Let $G(T) = \sum_{i=0}^{s} a_i T^i \in K[T]$ with $a_s = 1, a_0 \neq 0$ and let τ be a root of $G(T)$ in some finite extension L/K and $H(T) = \sum_{i=0}^{r} b_i T^i$ be its monic minimal polynomial over K. Then, for any place v of L extending a place of K,*

$$v(\tau) \geq \min_{0,1,\dots,s-1} v(a_i)/(s-i)$$

and, for any place v of K,

$$v(b_j) \geq (r-j) \min_{0,1,\dots,s-1} v(a_i)/(s-i).$$

Proof. Recall that we assume throughout that $G(0) \neq 0$, so $\tau \neq 0$. If $iv(\tau) + v(a_i) > sv(\tau)$ for all $i < s$, then $\infty = v(G(\tau)) = \min\{iv(\tau) + v(a_i)\} = sv(\tau)$, contradiction. This gives the first part of the lemma.

We have that b_j is the $(r-j)$-th elementary symmetric function on the conjugates of τ so the second part follows from the first by extending v to a valuation of the splitting field of H. □

Submitted to Algor. Num. Th. Symp.

Algorithm 1 Find factor of $G(T)$ with restricted coefficients

Input

A function field K/k
A polynomial $G(T) \in K[T]$ monic, separable, of degree s with discriminant $f \neq 0$.
A place v of K/k, with ring of integers $\mathcal{O}$ and maximal ideal $\mathfrak{m}$ with $G(T) \in \mathcal{O}[T]$ and $v(f) = 0$ and an uniformizer of v used to define the $D^{(i)}$.
Finite dimensional k-vector spaces $V_i \subset K, i = 0, \dots r-1$, with $1 \in V_0$, together with a k-basis $\{h_{ij}\}$ for each V_i, where $r < s$. Put $h_{01} = h_{r1} = 1, m = \sum \dim V_i$.
An integer Δ which bounds j_i, ε_i for any map to $\mathbb{P}^m$ obtained by viewing Φ in a quotient field of R in which its entries are linearly independent over k.

Output

Either a proof that $G(T)$ has no monic factor of the form $H(T) = \sum_{i=0}^{r} b_i T^i, b_i \in V_i, i < r, b_r = 1$.
Or a direct summand S of the ring $R_1 = (\mathcal{O}/\mathfrak{m}^q)[T]/(G(T))$ and elements $b_i, i = 0, \dots, r$ of S such that, for each local summand of S, a factor of $G(T)$ of the required form can be obtained (see Theorem 3.2 and Remark 3.3).

1: $R = K[T]/(G(T))$ and t the image of T in R.
2: $\Phi \in R^{m+1}$ the row vector with entries (in some order) $h_{ij}t^i \in R, i = 0, \dots, r$ and for each $i < r, j = 1, \dots, \dim V_i$ and $j = 1$ for $i = r$.
3: Compute q, smallest power of p with $q > \max\{m, \Delta\}$ (so $q \leq p \max\{m, \Delta\}$).
4: $R_1 = (\mathcal{O}/\mathfrak{m}^q)[T]/(G(T)), R_0 = (\mathcal{O}/\mathfrak{m})[T]/(G(T))$.
5: Compute matrix M with rows $D^{(i)}(\Phi), i = 0, \dots, q-1$, working in R_1.
6: Do the Gaussian elimination on M, working in R_1, as described in Subsection 2.2.
7: **for** Each direct summand R' of R_1 returned by the Gaussian elimination step and corresponding matrix M **do**
8: **if** M has full rank $m+1$ **then**
9: **return** $G(T)$ has no factor of the form required for the output in R'.
10: **else if** $M \pmod{\mathfrak{m}}$ has rank m **then**
11: Compute solution u_{ij} in R' of $\sum_{ij} u_{ij} D^{(\ell)}(h_{ij}t^i) = 0, u_{r1} = 1, \ell = 0, 1, \dots, q-1$.
12: **return** $b_i = \sum_j u_{ij} h_{ij}, i = 0, \dots, r$ in R'.
13: **else**
14: Go back to 2, replace the ring R_1 with the subring R', remove an entry from Φ such that the corresponding column of the matrix M has no pivot and set m equal to $m-1$.
15: **end if**
16: **end for**

Lemma 3.1 provides bounds for the valuations of the coefficients of the potential factors of $G(T)$ and these bounds can be used to define spaces V_i such that if $G(T)$ factors, it has factors of the form $H(T) = \sum_{i=0}^{r} b_i T^i, b_i \in V_i, i < r, b_r = 1$ for these V_i. To obtain k-bases for these V_i's (as required in the algorithm) one may apply the results of [Hes02b]. It also follows that we can take

$$\Delta = -r \sum_v \min\{0, \min_{0,1,\dots,s-1} v(a_i)/(s-i)\} \tag{3.1}$$

as a bound for the j_i, ε_i as required by Algorithm 1.

We begin by proving that Algorithm 1 performs as described and runs in polynomial time. We defer a detailed estimate of the running time and a discussion of how to compute the auxiliary place to Section 4.

Theorem 3.2. *Given the above input, Algorithm 1 runs in deterministic polynomial time in p, s, Δ (measured in number of operations in the field $\mathcal{O}/\mathfrak{m}$) and outputs either a certificate that $G(T)$ has no factor of the required form or a decomposition of R_1 as a direct sum of at most s rings. Moreover, for each such summand (R', say), the algorithm outputs elements u_{ij} of R' that are constant in each summand of the decomposition of R' into local rings and from which a factor of $G(T)$ of the required form can be constructed or a certificate that this summand does not yield such a factor. In particular, the algorithm provides a deterministic polynomial time absolute irreducibility test in characteristic p for p polynomially bounded in s, Δ and a general factoring algorithm modulo factoring in $k[T]$ under the same conditions.*

Proof. By induction on m. Assume $m = 1$. Since $1 \in V_0, \Phi = (1, t^r)$ and M has rank 1 or 2. If the rank is 2 in R', it is clear there is no factor of the required form. If the rank is 1 in R', this means that $D^{(i)}(t^r) = 0, i = 1, \ldots, q-1$, so t^r is constant in any local summand of R', the solution of the linear system is $u_{01} = -t, u_{r1} = 1$ and $H(T) = T - t$ is a factor of $G(T)$ of the required form.

As mentioned above, step 3 is dealt with in general by equation 3.1 unless there is a better bound available.

As mentioned above, the operators $D^{(i)}, i < q$ act on R_1 and the computation of the matrix M in step 6 is polynomial in operations in R_1 but $\dim_{\mathcal{O}/\mathfrak{m}} R_1 \leq sq$ giving a bound in terms of the number of operations in $\mathcal{O}/\mathfrak{m}$.

The process of Gaussian elimination has running time polynomial in the size of the matrix just as in the field case.At the end of it, we arrive at a decomposition of R as a sum of at most s rings and the image of M in each of these is put in row echelon form.

Those summands of R_1 where M has full rank $m+1$ yield no factor of $G(T)$ of the required form, since such a factor is a linear relation among the entries of Φ with constant coefficients and, applying $D^{(i)}, i < q$ to this relation shows that the coefficients are in the kernel of M. For other factors R' where M has rank m, we compute the u_{ij} as described in step 13. It follows from the proof of [GV87, Theorem 1] that $D^{(r)}(u_{ij}) = 0, 0 < r < q$ and thus, from Lemma 2.1 the u_{ij} are constant in each local factor of the decomposition of R'. Let S be one such factor. It determines a factor $H(T)$ of $G(T)$ and a place $w|v$ in the function field L obtained by adjoining a root t of $H(T)$ to K. Moreover, if $\hat{u}_{ij}$ are the constants in S corresponding to the u_{ij}, the fact that $\sum u_{ij} h_{ij} t^i = 0$ in S, shows that $\sum \hat{u}_{ij} h_{ij} t^i$ vanishes to order at least q at w. But $q > \Delta$ and Δ is an upper bound for the degree of the morphism associated to Φ, therefore an upper bound for the degree of $\sum \hat{u}_{ij} h_{ij} t^i$ and this implies that $\sum \hat{u}_{ij} h_{ij} t^i = 0$ in L. Hence $\sum \hat{u}_{ij} h_{ij} T^i$ yields a factor of $G(T)$ of the required form, if the $\hat{u}_{ij}$ are elements of k and, otherwise, S does not yield a factor of $G(T)$ of the required form. For the factors R' where M has rank smaller than m, we decrement m, drop an entry of Φ as described, rerun the algorithm and are done by induction.

For the last two claims of the theorem, first notice that when $G(T)$ is absolutely irreducible, the matrices M will have full rank $m + 1$ in every factor ring in play on the algorithm so the algorithm will terminate with a certificate of absolute irreducibility. Finally, a general factoring algorithm follows from Lemma 3.1 and the discussion surrounding it. □

Remark 3.3. *What Algorithm 1 and Theorem 3.2 don't do is to identify the u_{ij} with specific elements of an algebraic extension of k, necessarily. For that, we need to further factor the factor of $G(T) \pmod{\mathfrak{m}}$ corresponding to the ring R' as a product of irreducibles to obtain the full decomposition of R' as a sum of local rings and identify the u_{ij} with constants in each summand. Note that it is possible that $G(T) \pmod{\mathfrak{m}}$ has irreducible factors of degree greater than 1 and, in the local ring corresponding to that factor, $\sum \hat{u}_{ij} h_{ij} T^i$ will not give a factor of $G(T)$ of the required form over k. If such an irreducible factor of $G(T) \pmod{\mathfrak{m}}$ is detected, the corresponding local ring can be discarded. For each linear factor of $G(T) \pmod{\mathfrak{m}}$, once it is identified, we obtain a summand of R' which is a local ring with residue field k and projecting each $\hat{u}_{ij}$ to the residue fields yields the u_{ij}, completing the factorization process.*

As mentioned in the introduction, the first step in most standard factoring algorithms for $K[T]$ is to fully factor $G(T) \in (\mathcal{O}/\mathfrak{m})[T]$ and how it is performed depends on the nature of the field k. In our algorithm, this step may not be required at all (if $G(T)$ has no factors, or a single irreducible factor, of the required form) or it may only be needed for a proper factor of $G(T)$.

On the other hand, it is not claimed that the algorithm, as described, will output all factors of the required form, only that it will output at least one such factor, in case it exists. This can be easily remedied by dividing by the output factors and rerunning the algorithm with the quotient as input.

We will discuss an example but, beforehand, here is a non-example. If $G(T) \in k[T]$, that is, has constant coefficients, then $D^{(i)}(t) = 0, i > 0$, the matrix M has always rank one and the algorithm unravels to the base case $m = 1$. The polynomial $T - t$ is a factor of $G(T)$ for $G(t) = 0$ and we are left with the task of factoring $G(T)$ over the constant field k.

For a more representative example consider

$$G(T) = T^4 + (x+1)T^3 + (x^2+1)T^2 + (x^3+x^2+1)T + (x^2+x) \in \mathbb{F}_2(x)[T].$$

We look for factors of $G(T)$ of the form $T+ax+b$, so $r = 1$, V_0 is spanned by $1, x$, $\Phi = (1, x, t)$ where t is the image of T in $(k[x]/(x^4))[T]/(G(T))$ and $m = 2$. Modulo the ideal $(x+1)$, $G(T)$ reduces to $T^4 + T$. We find, using 2.3 that $D^{(2)}(t) = J(t)/G'(t)^3$, where

$$J(T) = (xT^5 + (x^2+x)T^4 + x^5T + (x^6+x^5)).$$

The gcd of $G(T)$ and $J(T)$ is $H(T) = T^2 + T + x^2 + x$ and, switching to the ring $R_1 = (k[x]/(x+1)^4)[T]/(H(T))$, we let t be the image of T, forcing $D^{(2)}(t) = 0$. Using 2.3 now with H, we find that $D(t) = 1$. The matrix M is therefore

$$M = \begin{pmatrix} 1 & x & t \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Now, we solve the system $t + ax + b = D(t) + a = 0$, so $a = D(t) = 1, b = t + x$. Now, modulo the ideal $(x+1)$, we have $H(T) = T^2 + T = T(T+1)$ and we lift this factorization to $k[x]/(x+1)^4$ and find $H(T) = (T+x)(T+x+1)$. Now, R_1 is a direct sum of two rings from this factorization and $b = 0, 1$ respectively in each of the factors. Consequently, $G(T)$ has the factors $T + x, T + x + 1$ of the required form.

Remark 3.4. *If the characteristic is zero, R_1 is not defined and if the characteristic is large, R_1 is too big. An extension of the algorithm of this paper to those cases would require a replacement for R_1. For theoretical results, such as Theorem 5.1, we can work with $R = \mathcal{O}[T]/(G(T))$ but for algorithmic purposes we need a ring finite dimensional over k. We could work with the ring $(\mathcal{O}/\mathfrak{m}^n)[T]/(G(T))$ for some appropriate choice of n, but the higher derivatives are not typically well-defined as operators on this ring. Instead, it makes sense to look at $D^{(i)} : (\mathcal{O}/\mathfrak{m}^n)[T]/(G(T)) \to (\mathcal{O}/\mathfrak{m}^{n-i})[T]/(G(T))$. We have not worked out the full details of this possibility.*

4. Running time estimates and finding a suitable place

We retain the notation of the previous section and especially of Algorithm 1.

Theorem 4.1. *Given a place v as in the input of Algorithm 1 with $[\mathcal{O}/\mathfrak{m} : k] = d$, the algorithm runs in $O(ds(p\max\{m, \Delta\})^4)$ field operations in k.*

Proof. The ring R_1 is initialized in the algorithm and during its execution it is replaced by its direct summands, of which there are at most s. From [Hes02a, Section 6.1], the computation of the matrix M takes $O(q^2)$ operations in each ring R_1 and Gaussian elimination of the matrix M takes $O(q^3)$ operations in each ring R_1. The latter dominates the former. The initial ring R_1 is a vector space over $\mathcal{O}/\mathfrak{m}$ of dimension sq, so we can bound the running time of these calculations as $O(sq^4)$ operations in $\mathcal{O}/\mathfrak{m}$. Using now that $q \leq p\max\{m, \Delta\}$ we arrive at a running time of $O(s(p\max\{m, \Delta\})^4)$ operations in $\mathcal{O}/\mathfrak{m}$. If we let $d = [\mathcal{O}/\mathfrak{m} : k]$, we finally arrive at $O(ds(p\max\{m, \Delta\})^4)$ operations in k. □

Remark 4.2. *There are faster algorithms in the special cases of most interest, e.g. k finite, $K = k(x)$ [Lec10, Wei17] (e.g. [Lec10, Theorem 1], seems to translate to a running time of $O(ds(\max\{m, \Delta\})^3)$ field operations in k) or finding a root of $G(T)$, again for k finite [GS00]. There does not seem to be an algorithm with the same generality and flexibility with a stated running time for direct comparison.*

It remains to make explicit how to find a suitable place v and compute d. We are given a function field K/k and $f \in K, f \neq 0$, we need to construct a place v of K such that $v(f) = 0$, as well as an uniformizer for this place. Without loss of generality, we assume that k is finitely generated over its prime field. We can also assume that $f \notin k$, for otherwise the condition $v(f) = 0$ is automatic and we can replace f by an element of $K \setminus k$ in order to produce the place v. Under this additional hypothesis, there are only at most $2[K : k(f)]$ places of K with $v(f) \neq 0$ as $[K : k(f)] = \sum_v \max\{0, v(f)\} = -\sum_v \min\{0, v(f)\}$. Hence, to find v it suffices to generate enough places of K until a place with $v(f) = 0$ is found. It seems that such a search is, at the moment, unavoidable. In the case k finite, $K = k(x)$, for example, $f(x) = a(x)/b(x), a(x), b(x) \in k[x]$ and all that is required is an element α (possibly in an extension of k) with $a(\alpha)b(\alpha) \neq 0$ which can be found by a simpleminded search over a suitably large extension of k. However, such a search can be more difficult in the generality we work with and some of the literature seems to gloss over this point. We assume $K/k(f)$ separable. For a general discussion of how to reduce to this case, see [Ste05]. Then $K = k(f, g)$ for some $g \in K$ and there is $P(x, y) \in k[x, y], P(f, g) = 0$. We then find $\alpha \neq 0$ in k or in an extension field such that $P(\alpha, y)$ is separable. There is a place of K corresponding to each irreducible factor of $P(\alpha, y)$ over $k(\alpha)$ and, if $m(x)$ is the minimal polynomial of α over k, then $m(f)$ is an uniformizer for any such place, as $m(f)$

Submitted to Algor. Num. Th. Symp.

is an uniformizer for the corresponding place of $k(f)$ which, by construction, is unramified in K. An easy way of ensuring that $P(\alpha, y)$ is separable is to require that it be irreducible in $k(\alpha)[y]$. This has the advantage that irreducibility tests are easier than factorization and that $m(x) = x - \alpha$ in this case. On the other hand, this makes the degree of the place bigger.

Assume first that k is a finite field. This case is discussed in [GS00, Algorithm 3.2] with some additional assumptions which are relevant there but not here, so we just give a simplified discussion. Basically, one just searches for α in k or an extension thereof until one is found with $P(\alpha, y)$ is irreducible in $k(\alpha)[y]$. Chebotarev's density theorem implies that α exists if $\#k$ is large enough in terms of $[K : k(f)]$ and the genus of K. It is worth pointing out that we may not have a suitable place with residue field k or even an extension of k of small degree as we can see by considering a simple example such as $G(T) = T^p - (x^{p^n} - x)T + 1 \in \mathbb{F}_p(x)[T]$ (which has $f = \pm(x^{p^n} - x)^p$).

The case where k is not algebraic over its prime field can be tackled as follows. Let k_0 be the algebraic closure in k of its prime field. Then K and k are respectively the function field of varieties X, Y over k_0 with a map $X \to Y$ of relative dimension 1. We realize X as a hypersurface of projective space and intersect X with random hypersurfaces. As long as these hypersurfaces intersect X in an irreducible subset that is not a component of the divisor of f and is transversal to the generic fiber of the map $X \to Y$, such a hypersurface will define a place of K/k satisfying our requirements and the equation of the hypersurface is the uniformizer we need. The version of Bertini's theorem over finite fields from [CP16] guarantees that, for high enough degree, most hypersurfaces satisfy our conditions. Checking irreducibility can be done using the main algorithm of this paper (applied to a field of smaller transcendence degree) and the other conditions can be checked directly. Alternatively, one can use the explicit version of Hilbert's irreducibility theorem for function fields from [BSE21] to estimate how far one might need to search.

5. Other applications

5.1. **Arithmetic properties.** As mentioned in the introduction, [Rup99] has a bound on the size of the largest prime p for which an irreducible polynomial in $\mathbb{Z}[x, y]$ factors modulo p. We substantially extend this result. We will consider polynomials in $K[T]$ where K/k is a function field and k itself is a global field. We impose no restriction on the characteristic of k. We can associate a height to elements of k in the usual way ([Lan83, Chapter 3, §3]), namely, for $a \in k$, the height of a is $-\sum_v \min\{0, v(a)\}$, where v runs through the places of k. If we represent K as a finite extension of $k(x)$ for some transcendental element x of K, for an element $a \in K$, there is a polynomial $P(x, y) \in k[x, y], P(x, a) = 0$ of minimal degree. We call the coefficients (in k) of P simply the coefficients of a and use the maximum of their heights as a measure of the complexity of a. Given $G(T) \in K[T]$, for all but finitely primes $\mathfrak{p}$ of k, we can consider the reduction of $G(T)$ modulo $\mathfrak{p}$.

Theorem 5.1. *Let k be a global field, K/k a function field and $G(T) \in K[T]$ an irreducible polynomial, as in equation 2.2 and define Δ as in equation 3.1. Let H be the maximum height of the coefficients of the a_i. The norm $N(\mathfrak{p})$ of the primes $\mathfrak{p}$ of k for which the reduction of $G(T)$ modulo $\mathfrak{p}$ is either undefined or reducible satisfies $N(\mathfrak{p}) = O(H^{(\Delta+1)^3})$, where the implied constant depends on s, Δ.*

Proof. Since $G(T)$ is assumed irreducible, the matrix M in Algorithm 1 has maximal rank and so a maximal minor has non-zero determinant. Note that $\Delta + 1$ is a bound for the

Submitted to Algor. Num. Th. Symp.

number of columns (hence also of rows) of M. We now compute M with entries in R (with no restriction on the characteristic of k). It follows by induction from equation 2.3 that the height of the coefficients of $D^{(i)}(t^j)$ is $O(H^{ij}) = O(H^{(\Delta+1)^2})$, hence the height of the coefficients of the (non-zero) determinant of a maximal $(\Delta+1)\times(\Delta+1)$-minor of M is $O(H^{(\Delta+1)^3})$. But for the reduction of $G(T)$ modulo $\mathfrak{p}$ to be either undefined or reducible, $\mathfrak{p}$ needs to divide the determinant of this minor and the result follows. □

In the case where $k = \mathbb{F}_q(Z)$, the approach of the previous theorem can be used to prove a proximity gap statement (in the sense of [BSCI+20, BLNR22]) for Algebraic Geometry codes.

5.2. **List decoding.** A different application concerns the the Guruswami–Sudan list-decoding algorithm for Reed–Solomon or Algebraic Geometry codes [GS00]. If one has a fixed code, there are fixed finite dimensional k-vector spaces $V_0, W_0, \ldots, W_{s-1} \subset K$ and, for each received message, a polynomial $G(T) \in K[T]$ is constructed with coefficients $a_i \in W_i$ (the precise construction is irrelevant at the moment), for which we want to know its roots in V_0. That is precisely the problem we dealt with above, with $r = 1$.

For an alternative approach to improving the factorization step of the Guruswami–Sudan algorithm, see [AP00]. Additionally, [AP00, Theorem 5] provides an alternative way of finding a suitable place (in the sense of Section 4) to be used in either approach to factorization in the context of the Guruswami–Sudan algorithm.

For a prime power q and an integer $k, 1 \le k \le q$, the Reed-Solomon code $RS_k(q)$ is the subspace of the q-dimensional space of functions $\mathbb{F}_q \to \mathbb{F}_q$ corresponding to the functions represented by a polynomial $f(x) \in \mathbb{F}_q[x], \deg f < k$. Define, as usual for $g_1, g_2 : \mathbb{F}_q \to \mathbb{F}_q$, the Hamming distance $d(g_1, g_2) = \#\{\alpha \in \mathbb{F}_q \mid g_1(\alpha) \neq g_2(\alpha)\}$.

McEliece [RJM03] has shown that, for a fixed Reed-Solomon code, most invocations of the Guruswami–Sudan algorithm output a list of codewords of size 0 or 1. We will strengthen this result and show that, for a certain range of Reed-Solomon codes, for most invocations of the Guruswami–Sudan algorithm, the polynomial $G(T) \in K[T]$ has at most one root in V_0, so we can take full advantage of our factoring algorithm's enhanced perfomance in this situation. Our result is meant to be illustrative and one might expect a similar result in much greater generality.

The following proposition sets the stage for our result and is a special case of Sudan's original precursor [Sud97] to the Guruswami–Sudan algorithm.

Proposition 5.2. *Assume $(q+5)/10 < k \le q/8$ and that $g : \mathbb{F}_q \to \mathbb{F}_q$ is any function. Then there exists a non-zero polynomial $G(x,y) = a_0(x) + a_1(x)y + a_2(x)y^2 + a_3(x)y^3 + a_4(x)y^4 \in \mathbb{F}_q[x,y]$ with $\deg a_i \le (4-i)(k-1), i = 0, \ldots, 4$ such that $\forall \alpha \in \mathbb{F}_q, G(\alpha, g(\alpha)) = 0$. Moreover, if $f(x) \in RS_k(q)$ satisfies $d(f,g) \le [q/2]$ then $G(x, f(x))$ is identically zero.*

Proof. Each condition $G(\alpha, g(\alpha)) = 0$ is a linear equation on the coefficients of the a_i. The total number of coefficients is $\sum_{i=0}^{4}((4-i)(k-1)+1) = 10k-5 > q$, by hypothesis. So there is a G, as prescribed. If $f(x) \in RS_k(q)$ satisfies $d(f,g) \le [q/2]$ then $G(x, f(x))$ is a polynomial with at least $q/2$ zeros but, since $\deg f \le k-1$, we have $\deg G(x, f(x)) \le 4(k-1) < q/2$ by hypothesis, hence $G(x, f(x))$ is identically zero. □

So, as mentioned above, given such G, we need to find its factors of the form $y - f(x), \deg f < k$ to find the elements $f(x) \in RS_k(q), d(f,g) \le [q/2]$ and, by construction, there are at most 4 such. This is the "list" in the list-decoding algorithm. It is easy to see

that, for most choices of g, the list is empty. Indeed, there are q^q possible g's and at most $q^k \sum_{i=0}^{[q/2]} \binom{q}{i}(q-1)^i$ of those satisfy $d(f,g) \leq [q/2]$ for some $f \in RS_k(q)$ and the sum is asymptotic to $q^{k+q/2+o(q)}$, hence bounded above by $q^{5q/8+o(q)}$, much smaller than q^q.

More interestingly, McEliece [RJM03] has shown that, for most choices of g such that there exists some $f \in RS_k(q)$ with $d(f,g) \leq [q/2]$, the f is unique. It follows that $q^{k+q/2+o(q)}$ is the correct order of magnitude for the number of g with $d(f,g) \leq [q/2]$ for some $f \in RS_k(q)$. The following result strengthens this result by showing that, for most choices of g such that there exists some $f \in RS_k(q)$ with $d(f,g) \leq [q/2]$, any polynomial G, output of Proposition 5.2, has only one factor of the form $y - f(x), \deg f < k$. This justifies our claim that a factoring algorithm that performs better when there is only one factor of prescribed type improves the running time of the list decoding algorithm.

Theorem 5.3. *Assume $(q+5)/10 < k < q/8$. Among those $g : \mathbb{F}_q \to \mathbb{F}_q$ such that there exists some $f \in RS_k(q)$ with $d(f,g) \leq q/2$, the number of those for which a polynomial G, output of Proposition 5.2, has more than one factor of the form $y - f(x), \deg f < k$ is at most $q^{5k+o(q)}$.*

Proof. We assume that $d(f_0, g) = \delta \leq [q/2]$, for some f_0 as above. Replacing g by $g - f_0$, where f_0 is one choice, we can assume $f_0 = 0$. By Proposition 5.2, $G(x,y) = yH(x,y)$, for some H. Moreover, $H(\alpha, g(\alpha)) = 0$ for δ values of α. We split the proof in two cases, depending on whether or not there exists $\alpha \in \mathbb{F}_q$ with $H(\alpha, y)$ identically zero.

Suppose first that there is no such α. The number of g's for a given choice of H is at most $\binom{q}{\delta}3^\delta$. Indeed, there are $\binom{q}{\delta}$ choices for a subset of size δ of $\mathbb{F}_q$ and, for each α in this subset, at most 3 choices for a solution of $H(\alpha, y) = 0$ as H is cubic in y (for the α outside of the subset we set $g(\alpha) = 0$). If, in turn, $H(x,y) = (y - f(x))S(x,y)$ with $f \in RS_k(q)$ and

$$S(x,y) = b_0(x) + b_1(x)y + b_2(x)y^2, \deg b_i \leq (2-i)(k-1), i = 0, 1, 2,$$

we can count the number of coefficients, namely k for f and $3k$ for the b_i's. Hence at most q^{4k} possible such H's and at most $\sum_{\delta \leq [q/2]} \binom{q}{\delta} 3^\delta q^{4k}$ possible such g's. If we now allow the initial f_0 to vary (as opposed to be set to $f_0 = 0$) we end up with a total of at most $\sum_{\delta \leq [q/2]} \binom{q}{\delta} 3^\delta q^{5k}$ for the count of the theorem and this is at most $q^{5k+o(q)}$, as desired.

Let's now assume that there are $\alpha \in \mathbb{F}_q$ with $H(\alpha, y)$ identically zero and let $c(x)$ be the monic polynomial that has these α's as roots. Since the coefficient of y^3 in $H(x,y)$ as a polynomial in y is constant, we conclude in this case that it is zero. We can then write H (in the case that G has a second root $f \in RS_k(q)$) as

$$H(x,y) = c(x)(y - f(x))(b_0(x)/c(x) + (b_1(x)/c(x))y).$$

If we fix $\deg c = m$, the number of possible H's is at most $q^m q^k q^{2k-m} q^{k-m} = q^{4k-m}$ by counting the number of possible $c, f, b_0/c, b_1/c$ respectively. The number of g given H is bounded above by $3^{q-m}q^m$ by counting the options for $g(\alpha)$ when $c(\alpha) \neq 0$ or $c(\alpha) = 0$, respectively. So, the count in this case is at most $q^{4k+o(q)}$ and again, allowing m and the initial f_0 to vary (as opposed to be set to $f_0 = 0$) we end up with a total of at most $q^{5k+o(q)}$ g's, completing the proof.

□

We note that $5k < k + q/2$ if $k < q/8$ which justifies the claim made immediately before the theorem. We also note that there are functions, as in the statement of the theorem, for

which G has more than one factor of the required form. For example, if q is even, we can take any g that takes only values $0, 1$ each $q/2$ times. Then $G = y(y+1)$ is an output of Proposition 5.2 for this g.

Finally, we note that Algorithm 1 vastly simplifies in the context of Proposition 5.2. In this case $K = \mathbb{F}_q(x), T = y, r = 1$. The vector Φ is $(1, x, x^2, \ldots, x^{k-1}, y)$, $m = k, \Delta = 4k$ and the matrix M is already in echelon form with rows $(0, 0, 0, \ldots, 0, D^{(\varepsilon)}(y))$ after the first k rows. The only real computation is to express the pivots $D^{(\varepsilon)}(y)$ as elements of R_1 and do the computations producing the splitting of R_1 in subrings from Subsection 2.2. As we want to detect whether y is a polynomial of degree $< k$, it is enough to check whether $D^{(\varepsilon)}(y) = 0$ for $k \leq \varepsilon < q$ and depending on the base-p expansion of k, only a few values of ε need to be checked, e.g. if k is a power of p, then only the powers of p satisfying $k \leq \varepsilon < q$ need to be checked.

Acknowledgements

This research was funded by the Ministry for Business, Innvovation and Employment in New Zealand. I would also like to thank M. Esgin, V. Kuchta, S. Ruj, A. Sakzad and R. Steinfeld of the Trans-Tasman ZK group for questions that motivated part of this research and a number of referees for their very thorough and useful reports.

References

[AP00] Daniel Augot and Lancelot Pecquet, *A Hensel lifting to replace factorization in list-decoding of algebraic-geometric and Reed-Solomon codes*, IEEE Trans. Inform. Theory **46** (2000), no. 7, 2605–2614. ↑11

[BSE21] Lior Bary-Soroker and Alexei Entin, *Explicit Hilbert's irreducibility theorem in function fields*, Abelian varieties and number theory, Contemp. Math., vol. 767, Amer. Math. Soc., Providence, RI, 2021, pp. 125–134. ↑10

[BvHKS09] Karim Belabas, Mark van Hoeij, Jürgen Klüners, and Allan Steel, *Factoring polynomials over global fields*, J. Théor. Nombres Bordeaux **21** (2009), no. 1, 15–39. ↑1

[BSCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf, *Proximity Gaps for Reed-Solomon Codes* (2020). Cryptology ePrint Archive, Report 2020/654. ↑11

[BLQ13] Jérémy Berthomieu, Grégoire Lecerf, and Guillaume Quintin, *Polynomial root finding over local rings and application to error correcting codes*, Appl. Algebra Engrg. Comm. Comput. **24** (2013), no. 6, 413–443. ↑1

[BLNR22] Sarah Bordage, Mathieu Lhotel, Jade Nardi, and Hugues Randriam, *Interactive Oracle Proofs of Proximity to Algebraic Geometry Codes*, 37th Computational Complexity Conference (CCC 2022), 2022, pp. 30:1–30:45. ↑11

[CP16] François Charles and Bjorn Poonen, *Bertini irreducibility theorems over finite fields*, J. Amer. Math. Soc. **29** (2016), no. 1, 81–94. ↑10

[CSTU02] Olivier Cormier, Michael F. Singer, Barry M. Trager, and Felix Ulmer, *Linear differential operators for polynomial equations*, J. Symbolic Comput. **34** (2002), no. 5, 355–398. ↑2

[GS00] Shuhong Gao and M. Amin Shokrollahi, *Computing roots of polynomials over function fields of curves*, Coding theory and cryptography (Annapolis, MD, 1998), Springer, Berlin, 2000, pp. 214–228. ↑1, 9, 10, 11

[Gao03] Shuhong Gao, *Factoring multivariate polynomials via partial differential equations*, Math. Comp. **72** (2003), no. 242, 801–822. ↑2

[GV87] Arnaldo García and José Felipe Voloch, *Wronskians and linear independence in fields of prime characteristic*, Manuscripta Math. **59** (1987), no. 4, 457–469. ↑3, 7

[vzGK85] J. von zur Gathen and E. Kaltofen, *Factorization of multivariate polynomials over finite fields*, Math. Comp. **45** (1985), no. 171, 251–261. ↑1

[Hes02a] Florian Hess, *An algorithm for computing Weierstrass points*, Algorithmic number theory (Sydney, 2002), Lecture Notes in Comput. Sci., vol. 2369, Springer, Berlin, 2002, pp. 357–371. ↑3, 4, 9

[Hes02b] F. Hess, *Computing Riemann-Roch spaces in algebraic function fields and related topics*, J. Symbolic Comput. **33** (2002), no. 4, 425–445. ↑6

[Hon81] Taira Honda, *Algebraic differential equations*, Symposia Mathematica, Vol. XXIV (Sympos., INDAM, Rome, 1979), Academic Press, London-New York, 1981, pp. 169–204. ↑2

[Lan83] Serge Lang, *Fundamentals of Diophantine geometry*, Springer-Verlag, New York, 1983. MR715605 ↑10

[Lec10] Grégoire Lecerf, *New recombination algorithms for bivariate polynomial factorization based on Hensel lifting*, Appl. Algebra Engrg. Comm. Comput. **21** (2010), no. 2, 151–176. ↑1, 9

[RJM03] R. J. McEliece, *On the Average List Size for the Guruswami-Sudan Decoder*, 7th International Symposium on Communications Theory and Applications (ISCTA), 2003. ↑11, 12

[NRS17] Vincent Neiger, Johan Rosenkilde, and Éric 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. ↑1

[Poh05] Michael E. Pohst, *Factoring polynomials over global fields. I*, J. Symbolic Comput. **39** (2005), no. 6, 617–630. ↑1, 2, 5

[Rup99] Wolfgang M. Ruppert, *Reducibility of polynomials $f(x, y)$ modulo p*, J. Number Theory **77** (1999), no. 1, 62–70. ↑2, 10

[Sch39] Friedrich Karl Schmidt, *Die Wronskische Determinante in beliebigen differenzierbaren Funktionenkörpern*, Mathematische Zeitschrift **45** (1939), no. 1, 62–74. ↑3, 4

[Ste05] Allan Steel, *Conquering inseparability: primary decomposition and multivariate factorization over algebraic function fields of positive characteristic*, J. Symbolic Comput. **3** (2005), 1053–1075. ↑9

[SV86] Karl-Otto Stöhr and José Felipe Voloch, *Weierstrass points and curves over finite fields*, Proc. London Math. Soc. (3) **52** (1986), no. 1, 1–19. ↑3, 4

[Sud97] Madhu Sudan, *Decoding of Reed Solomon codes beyond the error-correction bound*, J. Complexity **13** (1997), no. 1, 180–193. ↑11

[Wei17] Martin Weimann, *Bivariate factorization using a critical fiber*, Found. Comput. Math. **17** (2017), no. 5, 1219–1263. ↑1, 9

School of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand

Email address: felipe.voloch@canterbury.ac.nz

URL: http://www.math.canterbury.ac.nz/~f.voloch

Submitted to Algor. Num. Th. Symp.