Computing the Characteristic Polynomial of Endomorphisms of a finite Drinfeld Module using Crystalline Cohomology

Yossef Musleh
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
ymusleh@uwaterloo.ca

Éric Schost
Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
eschost@uwaterloo.ca

Abstract

We present a new algorithm for computing the characteristic polynomial of an arbitrary endomorphism of a finite Drinfeld module using its associated crystalline cohomology. Our approach takes inspiration from Kedlaya's p-adic algorithm for computing the characteristic polynomial of the Frobenius endomorphism on a hyperelliptic curve using Monsky-Washnitzer cohomology. The method is specialized using a baby-step giant-step algorithm for the particular case of the Frobenius endomorphism, and in this case we include a complexity analysis that demonstrates asymptotic gains over previously existing approaches.

CCS Concepts

• **Computing methodologies** → **Symbolic and algebraic algorithms**.

Keywords

Drinfeld module; algorithms; complexity.

ACM Reference Format:
Yossef Musleh and Éric Schost. 2023. Computing the Characteristic Polynomial of Endomorphisms of a finite Drinfeld Module using Crystalline Cohomology. In *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Drinfeld modules were first introduced by Vladimir Drinfel'd in order to prove the Langlands conjecture for GL_n over a global function field [11]. Since then, Drinfeld modules have attracted attention due to the well established correspondence between elliptic curves and the rank two case. Moreover, the rank one case, often referred to as *Carlitz modules*, provides a function field analogy of cyclotomic extensions; the role played in class field theory over number fields by elliptic curves with complex multiplication shows strong parallels with that of Drinfeld modules of rank two for the function field setting. This has motivated efforts to translate constructions and algorithms for elliptic curves, including modular polynomials [6], isogenies [6], and endomorphism rings [13, 27].

Naturally, cryptographic applications of Drinfeld modules have also been explored [28], but were long anticipated to be vulnerable for public key cryptography based on isogenies [23, 36]. This question was finally put to rest by Wesolowski who showed that isogenies between Drinfeld modules of any rank could be computed in polynomial time [38].

Drinfeld modules of rank $r > 2$ do not have such a clear parallel, although an analogy exists between abelian surfaces and so called t-modules [1]. Owing to this discrepancy, rank two Drinfeld modules have been studied far more closely than the case of more general ranks.

The main goal of this work is to study a Drinfeld module analogue of p-adic techniques such as Kedlaya's algorithm [25] for computing the characteristic polynomial of the Frobenius endomorphism acting on an elliptic or hyperelliptic curve over a finite field. Algorithms for elliptic curves compute the action of the Frobenius on a basis of a particular subspace of the de Rham cohomology of a characteristic 0 lift of the curve, with coefficients in $\mathbb{Q}_p$. Our approach follows a very similar outline, but turns out to be remarkably simpler to describe, resting crucially on a suitable version of crystalline cohomology for Drinfeld modules due Gekeler and Anglès [2].

More generally, the approach we present can be used to compute the characteristic polynomial of any Drinfeld module endomorphism.

2 Background and Main result

2.1 Basic Preliminaries

Let R be any ring, $r \in R$, and $\sigma : R \to R'$ a ring homomorphism. We will follow the notational convention that writes $\sigma(r) = \sigma_r = r^\sigma$ throughout this work. If R is a polynomial ring and σ acts on its coefficient ring, r^σ denotes coefficient-wise application.

Let q be a prime power, and let $\mathbb{F}_q$ denote a finite field of order q, fixed throughout. We also fix a field extension $\mathbb{L}$ of $\mathbb{F}_q$ such that $[\mathbb{L} : \mathbb{F}_q] = n$. Explicitly, $\mathbb{L}$ is defined as $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t))$ for some degree n irreducible $\ell(t) \in \mathbb{F}_q[t]$, so elements of $\mathbb{L}$ are represented as polynomials in $\mathbb{F}_q[t]$ of degree less than n. We will discuss below an alternative representation, better suited for some computations.

2.2 Drinfeld Modules

In general, Drinfeld modules can be defined over a ring A consisting of the functions of a projective curve over $\mathbb{F}_q$ that are regular

ISSAC, 2023,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

outside of a fixed place at infinity. For our purposes, we will restrict ourselves to the consideration of Drinfeld modules defined over the regular function ring of $\mathbb{P}^1 - \{\infty\}$; that is $A = \mathbb{F}_q[x]$.

We fix a ring homomorphism $\gamma : A \to \mathbb{L}$ and let $\mathfrak{p} \in A$ the monic irreducible generator of $\ker \gamma$. Then $\mathbb{F}_\mathfrak{p} = \mathbb{F}_q[x]/(\mathfrak{p})$ is isomorphic to a subfield of $\mathbb{L}$; we let $m = \deg(\mathfrak{p})$, so that m divides n. This gives us an isomorphism $\mathbb{L} \simeq \mathbb{F}_q[x,t]/(\mathfrak{p}(x), g(x,t))$, with g monic of degree n/m in t. It will on occasion be convenient to switch from the representation of elements of $\mathbb{L}$ as univariate polynomials in t to the corresponding bivariate representation in x, t; in that case, for instance, γ_x is simply the residue class of x modulo $(\mathfrak{p}(x), g(x,t))$. We assume that $\mathfrak{p}$ and g are given as part of the input.

To define Drinfeld modules, we also have to introduce the ring $\mathbb{L}\{\tau\}$ of skew polynomials, namely

$$\mathbb{L}\{\tau\} = \{U = u_0 + u_1\tau + \cdots + u_s\tau^s \mid s \in \mathbb{N}, u_0, \ldots, u_s \in \mathbb{L}\},$$

where multiplication is induced by the relation $\tau u = u^q \tau$, for all u in $\mathbb{L}$.

Definition 1. *A* Drinfeld A-module *of rank r over $(\mathbb{L}, \gamma)$ is a ring homomorphism $\phi : A \to \mathbb{L}\{\tau\}$ such that*

$$\phi_x = \gamma_x + \Delta_1\tau^1 + \ldots + \Delta_r\tau^r$$

with Δ_i in $\mathbb{L}$ for all i and $\Delta_r \neq 0$.

For readers interested in the more general setting under which Drinfeld modules are typically defined, we recommend the survey by Deligne and Husemöller in [9].

A Drinfeld module is defined over the *prime field* when $\mathbb{L} \cong \mathbb{F}_\mathfrak{p}$ (that is, $m = n$). Algorithms for Drinfeld modules in the prime field case tend to be algorithmically simpler, and we will often highlight the distinction with the more general case.

Example 1. *Let $\mathbb{F}_q = \mathbb{Z}/5\mathbb{Z}$ and $n = 4$. Set $\ell(t) = t^4 + 2$ and $\mathbb{L} = \mathbb{F}_5[t]/(\ell(t))$. Let $\gamma_x = t \bmod \ell(t)$, in which case $\mathbb{L} = \mathbb{F}_\mathfrak{p}$. A rank two Drinfeld module is given by $\phi_x = \tau^2 + \tau + t$.*

We may instead take $\gamma_x = t^3 + t^2 + t + 3 \bmod \ell(t)$ in which case $\mathfrak{p} = x^2 + 4x + 2$ and $\mathbb{F}_\mathfrak{p} \cong \mathbb{F}_{25}$. The field $\mathbb{L}$ admits the representations

$$\mathbb{L} = \mathbb{F}_5[t]/(\ell(t)) \simeq \mathbb{F}_5[x,t]/(\mathfrak{p}(x), g(x,t)),$$

with $g(x,t) = t^2 + 4tx + 3t + x$. A rank three Drinfeld module is given by $\phi_x = \tau^3 + (t^3+1)\tau^2 + t\tau + t^3 + t^2 + t + 3$.

Given Drinfeld A-modules ϕ, ψ defined over $(\mathbb{L}, \gamma)$, an $\mathbb{L}$-morphism $u : \phi \to \psi$ is a $u \in \mathbb{L}\{\tau\}$ such that $u\phi_a = \psi_a u$ for all $a \in A$. The set $\mathrm{End}_\mathbb{L}(\phi)$ is the set of $\mathbb{L}$-morphisms $\phi \to \phi$; it is therefore the centralizer of ϕ_x in $\mathbb{L}\{\tau\}$. It admits a natural ring structure, and contains the *Frobenius endomorphism* τ^n. The degree of an $\mathbb{L}$-morphism u is the τ-degree of the underlying skew polynomial in $\mathbb{L}\{\tau\}$.

2.3 Characteristic Polynomials

The characteristic polynomial of an endomorphism $u \in \mathrm{End}_\mathbb{L}(\phi)$ can be defined through several points of view.

Through the action of ϕ, $A = \mathbb{F}_q[x]$ and its fraction field $K = \mathbb{F}_q(x)$ can be seen as a subring, resp. subfield of the skew field of fractions $\mathbb{L}(\tau)$ of $\mathbb{L}\{\tau\}$. Then, $\mathrm{End}^0_\mathbb{L}(\phi) = \mathrm{End}_\mathbb{L}(\phi) \otimes_A K$ is the centralizer of ϕ_x in $\mathbb{L}(\tau)$; this is a division ring that contains K in its center.

Definition 2. *The characteristic polynomial* $\mathrm{CharPoly}(u)$ *of $u \in \mathrm{End}_\mathbb{L}(\phi)$ is its reduced characteristic polynomial, relative to the subfield K of $\mathrm{End}^0_\mathbb{L}(\phi)$ [35, Section 9.13].*

The characteristic polynomial of u has degree r and coefficients in $A \subset K$, so that it belongs to $A[Z]$. More precisely, if $\deg(u) = d$, $\mathrm{CharPoly}(u)$ has coefficients $a_0, \ldots, a_{r-1} \in A$ with $\deg(a_i) \leq d(r-i)/r$ for all i [27, Prop. 4.3] and satisfies

$$u^r + \sum_{i=0}^{r-1} \phi_{a_i} u^i = 0. \tag{1}$$

Another definition of $\mathrm{CharPoly}(u)$ follows from the introduction of the *Tate modules* of ϕ. The Drinfeld module ϕ induces an A-module structure on the algebraic closure $\overline{\mathbb{L}}$ of $\mathbb{L}$ by setting $a * c = \phi_a(c)$ for $a \in A$, $c \in \overline{\mathbb{L}}$ (defining $\tau^i(c) = c^{q^i}$). Then, for $\mathfrak{I} \in A$, the $\mathfrak{I}$-torsion module of ϕ is defined as $\phi[\mathfrak{I}] = \{c \in \overline{\mathbb{L}} \mid \mathfrak{I} * c = 0\}$. Setting $\mathfrak{I}$ to be any irreducible element of A different from $\mathfrak{p}$, we can define the $\mathfrak{I}$-adic Tate module as $T_\mathfrak{I}(\phi) = \varprojlim \phi[\mathfrak{I}^i]$.

Letting $A_\mathfrak{I}$ be the $\mathfrak{I}$-adic completion of A, $T_\mathfrak{I}(\phi)$ becomes a free $A_\mathfrak{I}$-module of rank r and elements of $\mathrm{End}_\mathbb{L}(\phi)$ induce endomorphisms on $T_\mathfrak{I}(\phi)$. Then, for $u \in \mathrm{End}_\mathbb{L}(\phi)$, the characteristic polynomial $\mathrm{CharPoly}_{A_\mathfrak{I}}(u)$ of the induced endomorphism $u \in \mathrm{End}_{A_\mathfrak{I}}(T_\mathfrak{I}(\phi))$ agrees with $\mathrm{CharPoly}(u)$ [2, 17].

Example 2. *Let $\mathbb{F}_q, \mathbb{L}$ be as in the context of example 1, and $\gamma_x = t^3 + 4t^2 + t + 1 \bmod \ell(t)$. A rank 5 Drinfeld module is given by $\phi_x = (4t^3 + t^2 + 2)\tau^5 + (t^3 + 3t^2 + t + 1)\tau^4 + (4t+3)\tau^3 + (3t^2 + 4t + 4)\tau^2 + (4t^3 + 4t^2 + 4t)\tau + \gamma_x$.*

The characteristic polynomial of τ^n on ϕ is $Z^5 + 3Z^4 + (x^3 + 4x^2 + x)Z^3 + (2x^2 + 4x + 3)Z^2 + (x^3 + 2x^2 + 4x + 2)Z + 2x^4 + 3x^2 + 4x + 2$

The results in this paper are based on another interpretation of $\mathrm{CharPoly}(u)$, as the characteristic polynomial of the endomorphism induced by u in a certain *crystalline cohomology* module, due to Gekeler and Anglès [2]. Our first main result is an algorithm for computing the characteristic polynomial of the Frobenius endomorphism.

Here, ω is a real number such that two $s \times s$ matrices over a ring R can be multiplied in $O(s^\omega)$ ring operations in R; the current best value is $\omega \leq 2.372$ [12]. We will also let λ denote an exponent such that the characteristic polynomial of an $s \times s$ matrix over a ring R can be computed in $O(s^\lambda)$ ring operations in R. When R is a field, this can be done at the cost of matrix multiplication and therefore $\lambda = \omega$ [32]. For more general rings, the best known value to date is $\lambda \approx 2.7$ [24].

Theorem 1. *Let ϕ be a rank r Drinfeld module over $(\mathbb{L}, \gamma)$. There is a deterministic algorithm to compute the characteristic polynomial of the Frobenius endomorphism τ^n with bit complexity*

- $(r^\omega n^{1.5} \log q + n \log^2 q)^{1+o(1)}$ *for the prime field case ($m = n$)*
- $((r^\lambda/m + r^\omega/\sqrt{m}) n^2 \log q + n \log^2 q)^{1+o(1)}$ *for the general case $m < n$.*

When r and q are fixed, the runtime in the theorem is thus essentially linear in $n^2/\sqrt{m}$, which is $n^{1.5}$ in the prime field case and gets progressively closer to n^2 as m decreases. The best prior results [30] were limited to the case $r = 2$, with runtimes essentially linear in $n^{1.5}$ in the prime field case and n^2 otherwise (for fixed q).

This first algorithm builds upon techniques for linear recurrences originating from [10], which are so far limited to the particular case of the Frobenius endomorphism.

We also obtain two algorithms that can be applied to any $u \in \mathrm{End}_{\mathbb{L}}(\phi)$. The complexity in this case partly depends on the bit cost of multiplication and Euclidean division in $\mathbb{L}\{\tau\}$, which we will denote $\mathsf{SM}(d, n, q)$ and which will be discussed in more detail in Section 3.

Theorem 2. *With assumptions as in Theorem 1, there are deterministic algorithms to compute the characteristic polynomial of an endomorphism u of degree d with bit complexities*

- $\left(\frac{\min(dr^2,(d+r)r^{\omega-1})}{m}(d+m)n\log q + r^\lambda n(d+m)/m\log q + n\log^2 q\right)^{1+o(1)}$
- $(r\mathsf{SM}(d+r,n,q) + r^\lambda n(d+m)/m\log q + n\log^2 q)^{1+o(1)}$.

Again, it is worth considering the situation with r and q fixed. In this case, the runtimes we obtain are, respectively, essentially linear in $d(d+m)n/m$ and $\mathsf{SM}(d,n,q)$. In the next section, we review known values for SM; for the best known value of ω, and fixed q, it is $(d^{1.69}n)^{1+o(1)}$ for $d \le n^{0.76}$, and $(dn^{1.52})^{1+o(1)}$ otherwise. In the case $d = \Theta(n)$, the runtimes are thus essentially linear in n^3/m and $n^{2.53}$, respectively (so which is the better algorithm depends on the value of m). For $u = \tau^n$, the algorithm in the previous theorem is of course superior.

3 Computational Preliminaries

The key element in our complexity analyses is the cost of the following operations in $\mathbb{L}$: addition/subtraction, multiplication, inverse and (iterated) Frobenius application.

Some of the algorithms we use below (multiplication and Euclidean division in $\mathbb{L}\{\tau\}$ from [7, 34]) assume that all these operations can be done using $\tilde{O}(n)$ operations in $\mathbb{F}_q$. For the representation of $\mathbb{L}$ we use, this is however not known to be the case; Couveignes and Lercier proved the existence of "elliptic bases" that satisfy these requirements [8], but conversion to our representation does not appear to be obvious.

This explains why in our main result, we do not count operations in $\mathbb{F}_q$, but bit operations instead (our complexity model is a standard RAM); we explain below how this allows us to bypass the constraints above.

Using FFT based algorithms, polynomials of degree at most n with coefficients in $\mathbb{F}_q$ can be multiplied in boolean time $(n\log q)^{1+o(1)}$ [5, 20]. It follows that elementary field operations (addition, multiplication, inversion) in $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t))$ can be done with the same asymptotic cost.

Conversions between univariate and bivariate representations for elements of $\mathbb{L}$ take the same asymptotic runtime. Denote by α the isomorphism $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t)) \to \mathbb{F}_q[x,t]/(\mathfrak{p}(x), g(x,t))$; then, given f of degree less than n in $\mathbb{F}_q[t]$, we can compute the image $\alpha(f \bmod \ell(t))$ in $(n\log q)^{1+o(1)}$ bit operations; the same holds for α^{-1} [22, 33].

The last important operation is the application of the q-power Frobenius in $\mathbb{L}$. Recall that given polynomials $f, g, h \in \mathbb{F}_q[x]$ of degree at most n, *modular composition* is the operation that computes $f(g) \bmod h$. As showed in [15], for c in $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t))$, c^q can be computed in the same asymptotic time (up to logarithmic factors) as degree n modular composition, following a one-time precomputation that takes $(n\log^2 q)^{1+o(1)}$ bit operations. This then extends to arbitrary powers (positive and negative) of the Frobenius. We should point out that modular composition techniques also underlie the algorithms for switching between the two representations of the elements in $\mathbb{L}$ mentioned above.

In [26], Kedlaya and Umans proved that modular composition in degree n can be computed in $(n\log q)^{1+o(1)}$ bit operations (see also the refinement due to van der Hoeven and Lecerf [22]), whence a similar cost for (iterated) Frobenius in $\mathbb{L}$. Here, the fact that we work in a boolean model is crucial: Kedlaya and Umans' algorithm is not known to admit a description in terms of $\mathbb{F}_q$-operations.

From this, we can directly adapt the cost analyses in [7, 34] to our boolean model. In particular, following the latter reference (which did so in an algebraic cost model), we let $\mathsf{SM}(d,n,q)$ be a function such that

- degree d multiplication and right Euclidean division in $\mathbb{L}\{\tau\}$ can be done in $O(\mathsf{SM}(d,n,q))$ bit operations
- for n, q fixed, $d \mapsto \mathsf{SM}(d,n,q)/d$ is non-decreasing.

The latter condition is similar to the super-linearity of multiplication functions used in [14], and will allow us to streamline some cost analyses. Unfortunately, there is no simple expression for $\mathsf{SM}(d,n,q)$: on the basis of the algorithms in [7, 34], the analysis done in [7] gives the following upper bounds:

- for $d \le n^{(5-\omega)/2}$, we can take $\mathsf{SM}(d,n,q)$ in $(d^{(\omega+1)/2}n\log q)^{1+o(1)}$
- else, we can take $\mathsf{SM}(d,n,q)$ in $(dn^{4/(5-\omega)}\log q)^{1+o(1)}$

For instance, with $d = n$, this is $(n^{(9-\omega)/(5-\omega)}\log q)^{1+o(1)}$.

With $\omega = 2.37$, the cost is $(d^{1.69}n\log q)^{1+o(1)}$ for $d \le n^{0.76}$, and $(dn^{1.52}\log q)^{1+o(1)}$ otherwise; the exponent for $d = n$ is 2.53. For completeness, we point out that these algorithms heavily rely on Frobenius applications, and as such, require spending the one-time cost $(n\log^2 q)^{1+o(1)}$ mentioned previously.

One should also keep in mind that these asymptotic cost analyses are not expected to reflect practical runtimes. To the authors' knowledge, software implementations of the Kedlaya-Umans algorithm achieving its theoretical complexity, or of matrix multiplication with exponent close to 2.37, do not currently exist. For practical purposes, implementations of modular composition use an algorithm due to Brent and Kung [4], with an algebraic complexity of $O(n^{(\omega+1)/2})$ operations in $\mathbb{F}_q$. Revisiting skew polynomial algorithms and their analyses on such a basis is work that remains to be done.

Finally, we will note that an instance of the characteristic polynomial computation consists of the following inputs:

- the finite fields $\mathbb{L}, \mathbb{F}_q$
- the coefficients of a Drinfeld module of rank r over $\mathbb{L}$
- a degree d endomorphism.

The fields can be specified using $O(n\log q)$ bits, and the Drinfeld module itself costs $O(nr\log q)$ bits to encode. The endomorphism itself costs $O(dn\log q)$ to write down in general, and costs $O(1)$ for the Frobenius.

4 Prior Work

The question of computing the characteristic polynomial, particularly of the Frobenius endomorphism, was studied in detail in [18] for the rank two case only.

The most general approach constructs a linear system based on the degree constraints of the coefficients $a_i = \sum_{j=0}^{n(r-i)/r} a_{i,j}x^j$. Evaluating the characteristic polynomial at the Frobenius element and equating coefficients gives a linear system based on

$$\tau^{nr} + \sum_{i=0}^{r-1} \sum_{j=0}^{\frac{n(r-i)}{r}} \sum_{k=0}^{n(r-i)} a_{i,j} f_{j,k} \tau^{k+ni} = 0, \tag{2}$$

with $f_{j,k}$ the coefficient of τ^k in ϕ_{x^j}. Letting $\mathrm{MinPoly}(\tau^n)$ denote the minimal polynomial of τ^n (as an element of the division algebra $\mathrm{End}^0_{\mathbb{L}}(\phi)$ over the field $K = \mathbb{F}_q(x)$), the solution of the preceding system is unique and yields the characteristic polynomial generically, and only if $\mathrm{MinPoly}(\tau^n) = \mathrm{CharPoly}(\tau^n)$.

Garai and Papikian gave an algorithm for computing the characteristic polynomial [13, §5.1] valid for the prime field case only. As with the previous approach, this relies on the explicit computation of ϕ_{x^i}, which is the dominant computational step. This can be done by $O(n^2)$ evaluations of the recurrence

$$f_{i+1,j} = \gamma_x^{q^j} f_{i,j} + \sum_{t=1}^{r} \Delta_t^{q^{j-t}} f_{i,j-t}.$$

Thus the bit complexity of computing all of $\phi_x, \phi_{x^2}, \ldots, \phi_{x^n}$ is $(rn^3 \log(q))^{1+o(1)}$.

Further study of algorithms for the specific case of the Frobenius endomorphism in rank $r = 2$ was done in [31] and [30]. The latter focused on the complexity of the algorithms and used the same computational model that will be used here. As we reported after Theorem 1, the best known runtime to date was quadratic in n, except in the case where $\mathrm{MinPoly}(\tau^n) = \mathrm{CharPoly}(\tau^n)$, or in the prime field case where a bit cost of $(n^{1.5}\log q + n\log^2 q)^{1+o(1)}$ is possible [10]. To our knowledge, no previous analysis is available for an arbitrary endomorphism u.

In the context of elliptic curves, Kedlaya's algorithm [25] computes the characteristic polynomial of a matrix representation of the lift of the Frobenius action to a subspace of the Monsky-Washnitzer cohomology, up to some finite precision. Our algorithm follows the same high-level approach: we compute a matrix for the endomorphism acting on the crystalline cohomology with coefficients in a power series ring analogue to Witt vectors. The induced endomorphism turns out to be quite simple to describe in terms of skew-polynomial multiplication, which eliminates the need for a complicated lifting step.

5 Crystalline Cohomology

In this section, we first review the construction of the crystalline cohomology of a Drinfeld module and its main properties; this can be found in [2], where the definition is credited to unpublished work of Gekeler. Then, we introduce truncated versions of these objects, which reduce the computation of characteristic polynomials of endomorphisms of a Drinfeld module to characteristic polynomial computations of matrices over truncated power series rings.

5.1 Definition

The contents of this subsection is from [2, 16]. The set of *derivations* $D(\phi, \mathbb{L})$ of a Drinfeld module ϕ is the set of $\mathbb{F}_q$-linear maps $\eta : A \to \mathbb{L}\{\tau\}\tau$ satisfying the relation

$$\eta_{ab} = \gamma_a \eta_b + \eta_a \phi_b, \quad a, b \in A$$

Let then y be a new variable. The set $D(\phi, \mathbb{L})$ can be made into an $\mathbb{L}[y]$-module in the following manner.

Definition 3. *[2, Section 2] The set $D(\phi, \mathbb{L})$ is an $\mathbb{L}[y]$-module under $(cy^i * \eta)_a = c\eta_a \phi_{x^i}$, for η in $D(\phi, \mathbb{L})$, c in $\mathbb{L}$, $i \geq 0$ and a in A.*

Let further I be the ideal of $\mathbb{L}[y]$ generated by $y - \gamma_x$; for $k \geq 1$, we set

$$W_k = \mathbb{L}[y]/I^k$$

and

$$W = \varprojlim W_k \cong \mathbb{L}[[y - \gamma_x]].$$

Thus W comes equipped with projections $\pi_k : W \to W_k$ obtained by truncation of a power series, written as sum of powers of $(y-\gamma_x)$, in degree k. We have canonical ring homomorphisms $\iota_k : A \to W_k$ given by $\iota_k(x) = y \bmod I^k$. They lift to an inclusion $\iota : A \to W$, simultaneously commuting with each π_k, which represents elements of A via their I-adic expansion.

The *crystalline cohomology* $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$ of ϕ is the W-module $W \otimes_{\mathbb{L}[y]} D(\phi, \mathbb{L})$, that is, the completion of $D(\phi, \mathbb{L})$ at the ideal $I = (y - \gamma_x)$ of $\mathbb{L}[y]$.

Gekeler proved that $D(\phi, \mathbb{L})$ is a projective, hence free, $\mathbb{L}[y]$-module of rank r [16], with canonical basis $\hat{\eta}^{(i)}$ such that $\hat{\eta}^{(i)}(x) = \tau^i$ for $1 \leq i \leq r$. From this, it follows that $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$ is a free W-module of rank r as well, as pointed out in [2].

Remark 1. *In that reference, A is not necessarily a polynomial ring, and $\mathbb{L}[y]$ is replaced by $A_{\mathbb{L}} := \mathbb{L} \otimes_{\mathbb{F}_q} A$. In this case, $D(\phi, \mathbb{L})$ is a projective $A_{\mathbb{L}}$-module of rank r, the definition of ideal I changes, but it remains maximal in $A_{\mathbb{L}}$, so the completion W of $A_{\mathbb{L}}$ at I is still a local ring and $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$ is still free of rank r over W.*

An endomorphism u of ϕ induces an $\mathbb{L}[y]$-endomorphism u^* of $D(\phi, \mathbb{L})$, defined as $(u^*(\eta))_x = \eta_x u$, for η in $D(\phi, \mathbb{L})$; the same holds for the completion $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$. Following [2], using the fact that $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$ is free over W, one can then define the characteristic polynomial $\mathrm{CharPoly}_W(u^*)$ in the usual manner.

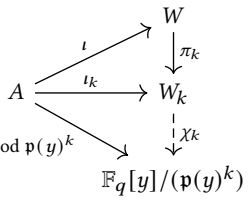
Recall now that $\mathrm{CharPoly}(u)$ denotes the characteristic polynomial of u, as defined in Section 2.3. The following theorem due to Anglès [2, Thm. 3.2] relates this characteristic polynomial to that of the induced endomorphism on $H^*_{\mathrm{crys}}(\phi, \mathbb{L})$, where ι below acts coefficient-wise.

Theorem 3. *For u in $\mathrm{End}_{\mathbb{L}}(\phi)$, $\mathrm{CharPoly}(u)^\iota = \mathrm{CharPoly}_W(u^*)$.*

Remark 2. *Recall that we have restricted ourselves to the case where A is the ring of functions on $\mathbb{P}^1$ regular outside of the point at infinity. However, theorem 3 holds for any Drinfeld module, and it is likely that the algorithms presented here can be generalized.*

5.2 Truncated Cohomology

Recall now that $\mathfrak{p} \in A$ is the minimal polynomial of $\gamma_x \in \mathbb{L}$ over $\mathbb{F}_q$. For $k \geq 1$, we are going to define an $\mathbb{F}_q$-linear homomorphism χ_k such that the following diagram commutes:



There exists an isomorphism

$$T_k : \mathbb{F}_q[x,y]/(\mathfrak{p}(x),(y-x)^k) \to \mathbb{F}_q[y]/(\mathfrak{p}(y)^k);$$

see e.g. [29, Lemma 13]. On the other hand, recall that $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t))$ is isomorphic to

$$\mathbb{F}_q[x,t]/(\mathfrak{p}(x), g(x,t)),$$

for some g in $\mathbb{F}_q[x,t]$, monic of degree n/m in t; in this representation of $\mathbb{L}$, γ_x is simply (the residue class of) x. As a result, we get

$$\begin{aligned} W_k &= \mathbb{F}_q[t,y]/(\ell(t),(y-\gamma_x)^k) \\ &\simeq \mathbb{F}_q[x,t,y]/(\mathfrak{p}(x), g(x,t),(y-x)^k) \\ &\simeq \mathbb{F}_q[y,t]/(\mathfrak{p}(y)^k, G_k(y,t)), \end{aligned} \tag{3}$$

for a certain polynomial $G_k \in \mathbb{F}_q[y,t]$, monic of degree n/m in t.

We can then define $\chi_k : W_k \to \mathbb{F}_q[y]/(\mathfrak{p}(y)^k)$ by

$$\chi_k : \sum_{0 \le i < n/m} c_i t^i \mapsto c_0,$$

and we verify that it satisfies our claim. The details of how to compute this homomorphism are discussed in Section 6.

For $k \ge 1$, we further define the *precision k* cohomology space $H^*_k(\phi,\mathbb{L})$ as the W_k-module

$$D(\phi,\mathbb{L})/I^k\, D(\phi,\mathbb{L}) \simeq H^*_{\mathrm{crys}}(\phi,\mathbb{L})/I^k\, H^*_{\mathrm{crys}}(\phi,\mathbb{L}).$$

It is thus free of rank r, and an endomorphism u of ϕ induces a W_k-linear endomorphism u^*_k of $H^*_k(\phi,\mathbb{L})$.

Remark 3. *In [16], Gekeler introduced* de Rham cohomology *of Drinfeld modules; this is the case $k = 1$ in this construction (in which case $W_k = \mathbb{L}$).*

In the following claim, recall that for a polynomial P and for any map χ acting on its coefficient ring, we let P^χ denote coefficient-wise application of χ to P.

Corollary 4. *For u in $\mathrm{End}_{\mathbb{L}}(\phi)$ and $k \ge 1$, $\mathrm{CharPoly}(u)^{\theta_k} = \mathrm{CharPoly}_{W_k}(u^*_k)^{\chi_k}$.*

Proof. Apply $\chi_k \circ \pi_k$ coefficient-wise to the equality in Theorem 3. □

If u has degree d in τ, we know that all coefficients of $\mathrm{CharPoly}(u)$ have degree at most d, so they can be recovered from their reductions modulo $\mathfrak{p}^k$ for $k = \lceil \frac{d+1}{m} \rceil \in O((d+m)/m)$. In the prime field case, where $m = n$, and for the special case $u = \tau^n$, the above formula gives $k = 2$, but we can take $k = 1$ instead; this is discussed in Section 6.4.

Note also that if we take $k = d+1$, there is no need to consider the map χ_k: on the representation of W_{d+1} as

$$W_{d+1} = \mathbb{F}_q[x,t,y]/(\mathfrak{p}(x), g(x,t),(y-x)^{d+1}),$$

for f of degree up to d, $\iota_k(f)$ is simply the polynomial $f(y)$, so we can recover f from $\iota_k(f)$ for free. We will however refrain from doing so, as it causes k to increase.

6 Main Algorithms

We will now see how the former discussion can be made more concrete, by rephrasing it in terms of skew polynomials only. The evaluation map $\eta \mapsto \eta_x$ gives an additive bijection $D(\phi,\mathbb{L}) \to \mathbb{L}\{\tau\}\tau$. This allows us to transport the $\mathbb{L}[y]$-module structure on $D(\phi,\mathbb{L})$ to $\mathbb{L}\{\tau\}\tau$: one verifies that it is given by $(cy^i * \eta) = c\eta\phi_{x^i}$, for η in $\mathbb{L}\{\tau\}\tau$, c in $\mathbb{L}$ and $i \ge 0$, and that $\mathcal{B} = (\tau,\ldots,\tau^r)$ is a basis of $\mathbb{L}\{\tau\}\tau$ over $\mathbb{L}[y]$.

Further, an endomorphism $u \in \mathrm{End}_{\mathbb{L}}(\phi)$ now induces an $\mathbb{L}[y]$-linear endomorphism $u^\star : \mathbb{L}\{\tau\}\tau \to \mathbb{L}\{\tau\}\tau$ simply given by $u^\star(v) = vu$ for v in $\mathbb{L}\{\tau\}\tau$. Reducing modulo the ideal $I^k \subset \mathbb{L}[y]$, we denote by $u^\star_k$ the corresponding W_k-linear endomorphism on the quotient module $\mathbb{L}\{\tau\}\tau/I^k_{\mathbb{L}}\mathbb{L}\{\tau\}\tau \simeq H^*_k(\phi,\mathbb{L})$.

We can then outline the algorithm referenced in Theorems 1 and 2; its correctness follows directly from Corollary 4 and the bound on k given previously.

1. Set $k = \lceil \frac{d+1}{m} \rceil$, with $d = \deg_\tau(u)$, except if $n = m$ and $u = \tau^n$ (in which case we can take $k = 1$)
2. Compute the coefficients $u_{i,1},\ldots,u_{i,r} \in W_k$ of $\tau^i u \bmod I^k$ on the basis $\mathcal{B}$, for $i = 1,\ldots,r$
3. Using the coefficients computed in step 2, construct the matrix for $u^\star_k$ acting on $\mathbb{L}\{\tau\}\tau/I^k_{\mathbb{L}}\mathbb{L}\{\tau\}\tau$ and compute its characteristic polynomial $\mathrm{CharPoly}_{W_k}(u^\star_k) \in W_k[Z]$
4. Apply the map χ_k to the coefficients of $\mathrm{CharPoly}_{W_k}(u^\star_k)$ to recover $\mathrm{CharPoly}(u)^{\theta_k}$, and thus $\mathrm{CharPoly}(u)$.

In Subsections 6.1 to 6.3, we discuss how to complete Step 2: we give two solutions for the case of an arbitrary endomorphism u, and a dedicated, more efficient one, for $u = \tau^n$. We freely use the following notation:

- for c in $\mathbb{L}$ and $t \in \mathbb{Z}$, let $c^{[t]}$ denote the value of the tth power Frobenius applied to c, that is, $c^{[t]} = c^{q^t}$
- for f in $\mathbb{L}[y]$, $f^{[t]} \in \mathbb{L}[y]$ is obtained by applying the former operator coefficient-wise, so $\deg(f) = \deg(f^{[t]})$
- for $M = (m_{i,j})_{1\le i\le u, 1\le j\le v}$ in $\mathbb{L}[y]^{u\times v}$, $M^{[t]}$ is the matrix with entries $(m^{[t]}_{i,j})_{1\le i\le u,1\le j\le v}$.

Finally, we define $\mu = (y-\gamma_x)^k \in \mathbb{L}[y]$ (with the value of k defined above); it generates the ideal I^k in $\mathbb{L}[y]$.

6.1 Using a Recurrence Relation

The following lemma is a generalization of a recurrence noted by Gekeler [19, Section 5] for $r = 2$. Recall that we write $\phi_x = \gamma_x + \Delta_1\tau^1 + \ldots + \Delta_r\tau^r$, with all Δ_i's in $\mathbb{L}$; in the expressions below, we write $\Delta_0 = \gamma_x$.

Lemma 1. *For any $t \ge 1$, the following relation holds in the $\mathbb{L}[y]$-module $\mathbb{L}\{\tau\}\tau$:*

$$\sum_{i=0}^{r} \Delta_i^{[t]}\tau^{t+i} = y * \tau^t. \tag{4}$$

Proof. This follows directly from the module action of $\mathbb{L}[y]$ on $\mathbb{L}\{\tau\}\tau$, by commuting τ^t across the defining coefficients Δ_i of ϕ:

$$y * \tau^t = \tau^t \phi_x = \tau^t \sum_{i=0}^{r} \Delta_i \tau^i = \sum_{i=0}^{r} \Delta_i^{[t]} \tau^{t+i}. \quad \square$$

For $i = 0, \ldots, r-1$, let $\Lambda_i = -\frac{\Delta_i}{\Delta_r}$ and define the order t companion matrix for the recurrence, $\mathcal{A}_t \in \mathbb{L}[y]^{r\times r}$, as

$$\mathcal{A}_t = \begin{bmatrix} \Lambda_{r-1}^{[t]} & \Lambda_{r-2}^{[t]} & \ldots & \Lambda_1^{[t]} & \Lambda_0^{[t]} + \frac{y}{\Delta_r^{[t]}} \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{5}$$

For $t \geq 1$, let $\kappa_t \in \mathbb{L}[y]^{1\times r}$ denote the coefficient vector of τ^t with respect to the standard basis $\mathcal{B}$. Then, we have the following relation between $r \times r$ matrices over $\mathbb{L}[y]$:

$$\begin{bmatrix} \kappa_{t+r} \\ \kappa_{t+r-1} \\ \vdots \\ \kappa_{t+1} \end{bmatrix} = \mathcal{A}_t \begin{bmatrix} \kappa_{t+r-1} \\ \kappa_{t+r-2} \\ \vdots \\ \kappa_t \end{bmatrix} \tag{6}$$

For $k \geq 1$, these relations can be taken modulo μ, to give equalities over $W_k = \mathbb{L}[y]/\mu$; below, we will write $\bar{\kappa}_t = \kappa_t \bmod \mu \in W_k^{1\times r}$.

Starting from $\bar{\kappa}_t, \ldots, \bar{\kappa}_{t+r-1}$, we obtain $\bar{\kappa}_{t+r-1}$ using $O(r)$ operations (divisions, Frobenius) in $\mathbb{L}$ to obtain the coefficients appearing on the first row of $\mathcal{A}_t$, followed by $O(kr)$ operations in $\mathbb{L}$ to deduce the entries of $\bar{\kappa}_{t+r}$.

Below, we will need $\bar{\kappa}_1, \ldots, \bar{\kappa}_{d+r}$. Altogether, computing them takes $((d+r)krn \log q)^{1+o(1)}$ bit operations; with our chosen value of k, this is also

$$((d+r)(d+m)rn/m \log q+)^{1+o(1)}.$$

Let us then write $u = u_0 + \cdots + u_d\tau^d$. For $i = 1, \ldots, r$, we have

$$\tau^i u = u_0^{[i]}\tau_i + \cdots + u_d^{[i]}\tau^{d+i},$$

so the coefficient vector $[u_{i,1} \cdots u_{i,r}] \in W_k$ of $\tau^i u \bmod I^k$ on the basis $\mathcal{B}$ is given by the product

$$[u_0^{[i]} \ \cdots \ u_d^{[i]}] \begin{bmatrix} \bar{\kappa}_i \\ \bar{\kappa}_{i+1} \\ \vdots \\ \bar{\kappa}_{i+d} \end{bmatrix} \in W_k^{1\times r}.$$

Each such operation takes $O(dkrn)$ operations in $\mathbb{L}$, for a total of $(d(d+m)r^2n/m \log q)^{1+o(1)}$ bit operations if done independently of one another (this is the dominant cost in the algorithm).

In cases when d is not small compared to r, we can reduce the cost slightly using matrix arithmetic, since all coefficient vectors we want can be read off an $r \times (d+r) \times r$ matrix product,

$$\begin{bmatrix} u_0^{[1]} & \cdots & u_d^{[1]} & 0 & \cdots & \cdots & 0 \\ 0 & u_0^{[2]} & \cdots & u_d^{[1]} & 0 & \cdots & 0 \\ & & \ddots & & \ddots & & \\ 0 & \cdots & \cdots & 0 & u_0^{[r]} & \cdots & u_d^{[r]} \end{bmatrix} \begin{bmatrix} \bar{\kappa}_1 \\ \bar{\kappa}_{i+1} \\ \vdots \\ \bar{\kappa}_{d+r} \end{bmatrix} \in W_k^{r\times r}.$$

This takes $((d+r)(d+m)r^{\omega-1}n/m \log q)^{1+o(1)}$ bit operations.

6.2 Using Euclidean Division

This section describes an alternative approach to computing the coefficients of an endomorphism u on the canonical basis $\mathcal{B}$. Computations are done in $\mathbb{L}[y]$ rather than $W_k = \mathbb{L}[y]/\mu$ (we are not able to take reduction modulo μ into account in the main recursive process).

The algorithm is inspired by a well-known analogue for commutative polynomials [14, Section 9.2]: for a fixed $a \in \mathbb{L}[y]$ of degree r, we can rewrite any f in $\mathbb{L}[y]$ as $f = \sum_{0\leq i<r} f_i(a)y^i$, for some coefficients $f_0, \ldots, f_{r-1}$ in $\mathbb{L}[y]$. This is done in a divide-and-conquer manner.

This approach carries over to the non-commutative setting. We start by showing how f of degree d in $\mathbb{L}\{\tau\}$ can be rewritten as

$$f = \sum_i f_i \phi_x^i,$$

for some f_i of degree less than r in $\mathbb{L}\{\tau\}$. If we let K be such that $d < Kr \leq 2d$, with K a power of 2, index i in the sum above ranges from 0 to $K-1$.

If $K = 1$, we are done. Else set $K' = K/2$, and compute the quotient g and remainder h in the right Euclidean division of f by $\phi_x^{K'}$, so that $f = g\phi_x^{K'} + h$. Recursively, we compute $g_0 \ldots, g_{K'-1}$ and $h_0, \ldots, h_{K'-1}$, such that

$$g = \sum_{0\leq i<K'} g_i \phi_x^i \quad \text{and} \quad h = \sum_{0\leq i<K'} h_i \phi_x^i.$$

Then, we return $h_0, \ldots, h_{K'-1}, g_0, \ldots, g_{K'-1}$. The runtime of the whole procedure is $O^{\tilde{}}(\mathsf{SM}(d, n, q))$ bit operations, with SM as defined in Section 3 (the analysis is the same as the one done in the commutative case in [14], and uses the super-linearity of SM with respect to d).

From there, we are able to compute the coefficients of $f \in \mathbb{L}\{\tau\}\tau$ on the monomial basis $\mathcal{B}$. This essentially boils down to using the procedure above, taking care of the fact that f is a multiple of τ. Factor τ on the left, writing f as τg: if $f = F\tau$, $g = F^{[-1]}$. Apply the previous procedure, to write $g = \sum_{0\leq i\leq s} g_i\phi_x^i$, with all g_i' of degree less than r and $s \leq d/r$.

This gives $f = \tau g = \sum_{0\leq i\leq s}(g_i^{[1]}\tau)\phi_x^i$, with all coefficients $g_i^{[1]}\tau$ supported on $\tau, \ldots, \tau^r$. Extracting coefficients of $\tau, \ldots, \tau^r$, we obtain polynomials $G_1, \ldots, G_r$ of degree at most s in $\mathbb{L}[\tau]$ such that $f = \sum_{1\leq i\leq r} G_i * \tau^i$.

The cost of left-factoring τ in f, and of multiplying all coefficients of g back by τ, is $(dn \log q)^{1+o(1)}$, so the dominant cost is $O^{\tilde{}}(\mathsf{SM}(d, n, q))$ bit operations from the divide-and-conquer process. To obtain the matrix of an endomorphism u of degree d, we apply r times this operation, to the terms $\tau^i u$, $i = 1, \ldots, r$. The runtime is then dominated by $O^{\tilde{}}(r\mathsf{SM}(d+r, n, q))$. Finally, reducing the entries of the matrix modulo $\mu = (y - \gamma_x)^k$ takes softly linear time in the size of these entries, so can be neglected.

6.3 Special Case of the Frobenius Endomorphism

In the particular case where $u = \tau^n$, we may speed up the computation using a baby-step giant-step procedure, based on the approach

used in [10]. As a first remark, note that for $u = \tau^n$, $d = n$ and k in $O(n/m)$.

In this case, it is enough to compute the vectors $\bar{\kappa}_{n+1}, \dots, \bar{\kappa}_{n+r}$. They are given by

$$\begin{bmatrix} \bar{\kappa}_{n+r} \\ \bar{\kappa}_{n+r-1} \\ \vdots \\ \bar{\kappa}_{n+1} \end{bmatrix} = \bar{\mathcal{A}}_n \dots \bar{\mathcal{A}}_1, \tag{7}$$

with $\bar{\mathcal{A}}_t$ the image of $\mathcal{A}_t$ modulo $\mu = (y - \gamma_x)^k$ for all t. To compute the matrix product $\bar{\mathcal{A}} = \bar{\mathcal{A}}_n \dots \bar{\mathcal{A}}_1$, we slightly extend the approach used in [10] (which dealt with the case $k = 1$). Consider the following element of $\mathbb{L}[y]^{r\times r}$:

$$\mathcal{B} = \begin{bmatrix} \Lambda_{r-1} & \Lambda_{r-2} & \dots & \Lambda_1 & \Lambda_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \dots & \Delta_r^{-1} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} y. \tag{8}$$

It follows in particular that for $t \geq 1$,

$$\mathcal{A}_t = \mathcal{B}^{[t]} \quad \text{and} \quad \bar{\mathcal{A}}_t = \mathcal{B}^{[t]} \bmod \mu,$$

with reduction applied coefficient-wise.

Write $n^* = \lceil \sqrt{nk} \rceil \in O(n/\sqrt{m})$, and let n be written as $n = n^* n_1 + n_0$ with $0 \leq n_0 < n^*$, so that $n_1 \leq \sqrt{n/k}$. Setting

$$C = \mathcal{B}^{[n^*+n_0]} \dots \mathcal{B}^{[n_0+1]}$$

and

$$C_0 = \mathcal{B}^{[n_0]} \dots \mathcal{B}^{[1]},$$

the matrix $\mathcal{A}$ is the product

$$\mathcal{A} = C^{[(n_1-1)n^*]} \dots C^{[n^*]} C C_0.$$

Our goal is to compute $\bar{\mathcal{A}} = \mathcal{A} \bmod \mu$, without computing $\mathcal{A}$ itself.

Any Frobenius application (of positive or negative index) in $\mathbb{L}$ takes $(n \log q)^{1+o(1)}$ bit operations. In particular, computing all matrices $\mathcal{B}^{[i]}$ that arise in the definitions of C and C_0 takes $(rn^2/\sqrt{m} \log q)^{1+o(1)}$ bit operations.

Once they are known, the next stage of the algorithm computes C and C_0 in $\mathbb{L}[y]$. This is done using a matrix subproduct-tree algorithm [14, Chapter 10], using a number of operations in $\mathbb{L}$ softly linear in $r^\omega n^*$. This is $(r^\omega n^2/\sqrt{m} \log q)^{1+o(1)}$ bit operations.

To deduce the shifted matrices

$$C^{[(n_1-1)n^*]} \bmod \mu, \dots, C^{[n^*]} \bmod \mu,$$

we use the following lemma.

Lemma 2. *For f in $\mathbb{L}[y]$ and $t \geq 0$,*

$$f^{[t]} \bmod \mu = (f \bmod \mu^{[-t]})^{[t]}$$

Proof. Let $g = f \bmod \mu^{[-t]}$, so that we have an equality of the form $f = a\mu^{[-t]} + g$ in $\mathbb{L}[y]$. We raise this to the power q^t coefficient-wise; this gives $f^{[t]} = a^{[t]}\mu + g^{[t]}$. Since g, and thus $g^{[t]}$, have degree less than k, this shows that $g^{[t]} = f^{[t]} \bmod \mu$. □

1: **procedure** CharPolyFrobenius
2: **Input** A field extension $\mathbb{L}$ of degree n over $\mathbb{F}_q$, $(\Delta_1, \dots, \Delta_r) \in \mathbb{L}^r$ representing a rank r Drinfeld module ϕ over $(\mathbb{L}, \gamma)$.
3: **Output** $a_i \in A$ such that the characteristic polynomial of the Frobenius is $X^r + \sum_{i=0}^{r-1} a_i X^i$.
4: $n^*, n_1, n_0 \leftarrow \lceil \sqrt{nk} \rceil, \lfloor n/n^* \rfloor, n \bmod n^*$.
5: $\mathcal{B}$ as in (8)
6: $C \leftarrow \mathcal{B}^{[n^*+n_0]} \dots \mathcal{B}^{[n_0+1]}$.
7: $\bar{C}_0 \leftarrow \mathcal{B}^{[n_0]} \dots \mathcal{B}^{[1]} \bmod \mu$
8: $\bar{C}^{[in^*]} \leftarrow (C \bmod \mu^{[-in^*]})^{[in^*]}$ for $0 \leq i < n_1$.
9: $\bar{\mathcal{A}} \leftarrow \left(\prod_{i=0}^{n_1-1} \bar{C}^{[in^*]} \right) \bar{C}_0$
10: $\bar{a}_i \leftarrow$ coefficient of Z^i in $\det(\bar{\mathcal{A}} - ZI)$
11: **return** $a_i = \chi_k(\bar{a}_i)$ for $0 \leq i < r$

Applying this entry-wise, we compute $C^{[in^*]} \bmod \mu$ by reducing all entries of C modulo $\mu^{[-in^*]}$, then raising all coefficients in the result to the power q^{in^*}, for $i = 1, \dots, (n_1 - 1)$.

Matrix C has degree $O(n/\sqrt{m})$, and the sum of the degrees of the moduli $\mu^{[-t]}$ is kn_1, which is $O(n/\sqrt{m})$ as well. Altogether, this takes $O(r^2 n/\sqrt{m})$ applications of Frobenius in $\mathbb{L}$, together with $O(r^2 n/\sqrt{m})$ arithmetic operations in $\mathbb{L}$ to perform all Euclidean divisions [14, Chapter 10]. Thus, the runtime is $(r^2 n^2/\sqrt{m} \log q)^{1+o(1)}$ bit operations.

Finally, we multiply all matrices $C^{[in^*]} \bmod \mu$ and $C_0 \bmod \mu$. This takes $(r^\omega n^2/\sqrt{m} \log q)^{1+o(1)}$ bit operations.

6.4 Other Operations

Once the coefficients of the skew polynomials $\tau^i u$ on the basis $\mathcal{B}$ are known modulo μ, we compute the characteristic polynomial of the matrix formed from these coefficients. This can be done with a bit cost of $(r^\lambda kn \log q)^{1+o(1)}$ when the matrix has entries in W_k, with λ the exponent defined in Section 2.3.

At this stage, we have all coefficients of $\mathrm{CharPoly}_{W_k}(u_k^\star)$ in W_k. It remains to apply the map χ_k to each of them to recover $\mathrm{CharPoly}(u)$.

Elements of $W_k = \mathbb{F}_q[t, y]/(\ell(t), (y - \gamma_x)^k)$ are written as bivariate polynomials in t, y, with degree less than n in t and less than k in y. To compute their image through χ_k, we first apply the isomorphisms

$$W_k = \mathbb{F}_q[t, y]/(\ell(t), (y - \gamma_x)^k) \xrightarrow{A_k} \mathbb{F}_q[x, t, y]/(\mathfrak{p}(x), g(x, t), (y - x)^k) \xrightarrow{B_k} \mathbb{F}_q[y, t]/(\mathfrak{p}(y)^k, G_k(y, t))$$

from (3), with $\mathfrak{p}(y)^k$ of degree km and G_k of degree n/m in t.

We mentioned in Section 3 that for c in $\mathbb{L} = \mathbb{F}_q[t]/(\ell(t))$, we can compute its image $\alpha(c)$ in $\mathbb{F}_q[x, t]/(\mathfrak{p}(x), g(x, t)$ using $(n \log q)^{1+o(1)}$ bit operations. Proceedings coefficient-wise with respect to y, this shows that for C in W_k, we can compute $A_k(C)$ in $(kn \log q)^{1+o(1)}$ bit operations.

The *tangling* map of [21, §4.5] provides an algorithm for computing the isomorphism $\mathbb{F}_q[x, y]/(\mathfrak{p}(x), (y - x)^k) \to \mathbb{F}_q[y]/(\mathfrak{p}(y)^k)$ in $(km \log q)^{1+o(1)}$ bit operations (this could also be done through modular composition, with a similar asymptotic runtime, but the

algorithm in [21] is simpler and faster). Applying it coefficient-wise with respect to t, this allows us to compute $B_k(A_k(C))$ in $(kn \log q)^{1+o(1)}$ bit operations again. At this stage, the mapping χ_k is simply extraction of the degree-0 coefficient in t.

We apply this procedure r times, for a total cost of $(rkn \log q)^{1+o(1)}$ bit operations. This can be neglected in the runtime analysis.

When using precision $k = 1$ for the prime field case, for $u = \tau^n$, it is necessary to compute the constant coefficient a_0 separately. This is done using the formula $a_0 = (-1)^{n(r+1)+r} N_{\mathbb{L}/\mathbb{F}_q}(\gamma_{\Delta_r})^{-1}\mathfrak{p}$ from [13] and takes $(n \log q)^{1+o(1)}$ bit operations.

Summing the costs seen so far for the various steps of the algorithm finishes the proof of our main theorems.

6.5 Example

Let $\mathbb{F}_q = \mathbb{Z}/2\mathbb{Z}$, $n = 3$ and set $\ell(t) = t^3 + t + 1$ and $\mathbb{L} = \mathbb{F}_2[t]/(\ell(t))$. Let $\gamma_x = t + 1 \bmod \ell(t)$, so that

$$\mathfrak{p} = x^3 + x^2 + 1 = \ell(x + 1),$$

and $\mathbb{L} \cong \mathbb{F}_\mathfrak{p} = \mathbb{F}_q[x]/(\mathfrak{p}(x))$, with the isomorphism given by $f(t) \mapsto f(x + 1)$. Consider the rank 4 Drinfeld module $\phi_x = t\tau^4 + (t^2 + t)\tau^3 + \tau^2 + t^2\tau + t + 1$. We proceed to compute the characteristic polynomial using the de Rham cohomology, that is, crystalline cohomology truncated in degree $k = 1$. In other words, all computations are done over $\mathbb{L}$

The recurrence of equation (4) becomes $\tau^{k+4} = (t+1)^{2^k}\tau^{(k+3)} + (t^2+1)^{2^k}\tau^{k+2} + t^{2^k}\tau^{k+1} + (1 + t^{1-2^k})\tau^k$. Running the recurrence for $n = 3$ iterations gives:

- $\tau^5 = (t^2+1)\tau^4 + (t^2+t+1)\tau^3 + t^2\tau^2 + t^2\tau^1$
- $\tau^6 = (t^2+1)\tau^4 + (t^2+1)\tau^3 + (t^2+t)\tau^2 + \tau^1$
- $\tau^7 = \tau^4 + t\tau^3 + (t+1)\tau^2 + \tau^1$

A matrix for the Frobenius endomorphism can be inferred to be

$$\begin{bmatrix} 1 & t & t+1 & 1 \\ t^2+1 & t^2+1 & t^2+t & 1 \\ t^2+1 & t^2+t+1 & t^2 & t^2 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

It has characteristic polynomial $Z^4 + (t+1)Z^2 + (t+1)Z$. Using the expression for a_0 which is valid in the prime field case, the Frobenius norm can be inferred to be $a_0 = x^3 + x^2 + 1$.

To recover the final coefficients, observe that $t \mapsto x + 1$ gives the required map $\chi_1 : W_1 = \mathbb{L} \to \mathbb{F}_\mathfrak{p}$. Finally, we conclude that the characteristic polynomial of τ^n is $Z^4 + xZ^2 + xZ + x^3 + x^2 + 1$.

7 Experimental Results

An implementation of the algorithm of section (6.3) was created in SageMath [37] and is publicly available at https://github.com/ymusleh/drinfeld-module. An implementation in MAGMA [3] is also publicly available at https://github.com/ymusleh/drinfeld-magma and was used to generate the experimental results included in this work. Our implementation differs from our theoretical version in a few ways.

- The Kedlaya-Umans algorithm is most likely not used by MAGMA for computing Frobenius mappings of elements of $\mathbb{L}$.
- To compute the images of coefficients under the map χ_k, we leverage a simpler procedure using reduction modulo bivariate Gröbner bases, rather than the tangling map of van der Hoeven and Lecerf. In any case, this does not impact the run times presented.

Run Times for $m = 10$ $q = 25$ in seconds							
	$n = 100$	$n = 150$	$n = 200$	$n = 300$	$n = 400$	$n = 500$	$n = 600$
$r = 5$	0.400	2.260	42.190	86.830	269.760	635.170	1099.110
$r = 9$	0.790	4.210	78.860	157.100	481.090	1129.670	
$r = 12$	1.170	6.080	104.630	220.430	658.950	1531.580	
$r = 18$	2.300	11.360	170.790	366.690	1074.840	2451.530	
$r = 23$	3.820	17.580	240.100	525.670	1518.370		

Acknowledgments

We thank Xavier Caruso, Antoine Leudière and Pierre-Jean Spaenlehauer for interesting discussions. Schost is supported by an NSERC Discovery Grant. We would also like to thank the reviewers for their comments.

References

[1] G. W. Anderson. 1986. t-Motives. *Duke Mathematical Journal* 53, 2 (1986), 457 – 502. https://doi.org/10.1215/S0012-7094-86-05328-7

[2] B. Anglès. 1997. On some characteristic polynomials attached to finite Drinfeld modules. *manuscripta mathematica* 93, 1 (01 Aug 1997), 369–379. https://doi.org/10.1007/BF02677478

[3] W. Bosma, J. Cannon, and C. Playoust. 1997. The Magma algebra system. I. The user language. *J. Symbolic Comput.* 24, 3-4 (1997), 235–265. https://doi.org/10.1006/jsco.1996.0125 Computational algebra and number theory (London, 1993).

[4] R. P. Brent and H. T. Kung. 1978. Fast Algorithms for Manipulating Formal Power Series. *J. ACM* 25, 4 (1978), 581–595.

[5] D. G. Cantor and E. Kaltofen. 1991. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica* 28 (1991), 693–701.

[6] P. Caranay, M. Greenberg, and R. Scheidler. 2020. Computing modular polynomials and isogenies of rank two Drinfeld modules over finite fields. *75 Years of Mathematics of Computation* (2020).

[7] X. Caruso and J. Le Borgne. 2017. Fast multiplication for skew polynomials. In *ISSAC'17*. ACM, 77–84.

[8] J.-M. Couveignes and R. Lercier. 2009. Elliptic periods for finite fields. *Finite Fields Their Appl.* 15, 1 (2009), 1–22.

[9] P. Deligne and D. Husemoller. 1987. Survey of Drinfel'd modules. In *Current trends in arithmetical algebraic geometry (Arcata, Calif., 1985)*. Contemp. Math., Vol. 67. Amer. Math. Soc., Providence, RI, 25–91. https://doi.org/10.1090/conm/067/902591

[10] J. Doliskani, A. K. Narayanan, and É. Schost. 2021. Drinfeld modules with complex multiplication, Hasse invariants and factoring polynomials over finite fields. *Journal of Symbolic Computation* 105 (2021), 199–213. https://doi.org/10.1016/j.jsc.2020.06.007 MICA 2016.

[11] V. G. Drinfel'd. 1974. Elliptic modules. *Matematicheskii Sbornik* 94, 23 (1974), 561–593.

[12] R. Duan, H. Wu, and R. Zhou. 2022. Faster Matrix Multiplication via Asymmetric Hashing. https://arxiv.org/abs/2210.10173

[13] S. Garai and M. Papikian. 2018. Endomorphism rings of reductions of Drinfeld modules. *Journal of Number Theory* 212 (04 2018). https://doi.org/10.1016/j.jnt.2019.02.008

[14] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra* (third ed.). Cambridge University Press, Cambridge.

[15] J. von zur Gathen and V. Shoup. 1992. Computing Frobenius maps and factoring polynomials. *Computational Complexity* 2, 3 (1992), 187–224.

[16] E.-U. Gekeler. 1988/89. De Rham cohomology for Drinfeld modules. *Séminaire de Théorie des Nombres de Paris* (1988/89), 57–85.

[17] E.-U. Gekeler. 1991. On finite Drinfeld modules. *J. Algebra* 141, 1 (1991), 187–203.

[18] E.-U. Gekeler. 2008. Frobenius Distributions of Drinfeld Modules over Finite Fields. *Trans. Amer. Math. Soc.* 360, 4 (2008), 1695–1721. http://www.jstor.org/stable/20161942

[19] E.-U. Gekeler. 2011. Frobenius actions on the de Rham cohomology of Drinfeld modules. *Trans. Amer. Math. Soc.* 363 (06 2011). https://doi.org/10.1090/S0002-9947-2011-05422-X

[20] D. Harvey, J. van der Hoeven, and G. Lecerf. 2017. Faster Polynomial Multiplication over Finite Fields. *J. ACM* 63, 6, Article 52 (2017). https://doi.org/10.1145/3005344

[21] J. van der Hoeven and G. Lecerf. 2017. Composition Modulo Powers of Polynomials. In *ISSAC'17*. ACM, 445–452.

[22] J. van der Hoeven and G. Lecerf. 2020. Fast multivariate multi-point evaluation revisited. *Journal of Complexity* 56 (2020), 101405. https://doi.org/10.1016/j.jco.2019.04.001
[23] A. Joux and A. K. Narayanan. 2019. Drinfeld modules are not for isogeny based cryptography. *IACR Cryptol. ePrint Arch.* 2019 (2019), 1329.
[24] E. Kaltofen and G. Villard. 2004. On the complexity of computing determinants. *Computational Complexity* 13, 3-4 (2004), 91–130.
[25] K. Kedlaya. 2001. Counting Points on Hyperelliptic Curves using Monsky-Washnitzer Cohomology. *J. Ramanujan Math. Soc.* 16 (06 2001).
[26] K. Kedlaya and C. Umans. 2008. Fast Polynomial Factorization and Modular Composition. *SIAM J. Comput.* 40 (01 2008). https://doi.org/10.1137/08073408X
[27] N. Kuhn and R. Pink. 2016. Finding Endomorphisms of Drinfeld modules. *Journal of Number Theory* 232 (08 2016). https://doi.org/10.1016/j.jnt.2021.02.013
[28] A. Leudière and P.-J. Spaenlehauer. 2022. Computing a Group Action from the Class Field Theory of Imaginary Hyperelliptic Function Fields. https://doi.org/10.48550/ARXIV.2203.06970
[29] E. Mehrabi and É. Schost. 2016. A softly optimal Monte Carlo algorithm for solving bivariate polynomial systems over the integers. *J. Complexity* 34 (2016), 78–128.
[30] Y. Musleh and É. Schost. 2019. Computing the Characteristic Polynomial of a Finite Rank Two Drinfeld Module. In *ISSAC'19*. ACM Press.
[31] A. K. Narayanan. 2018. Polynomial factorization over finite fields by computing Euler-Poincaré characteristics of Drinfeld modules. *Finite Fields Appl.* 54 (2018), 335–365.
[32] V. Neiger and C. Pernet. 2020. Deterministic computation of the characteristic polynomial in the time of matrix multiplication. https://doi.org/10.48550/ARXIV.2010.04662
[33] A. Poteaux and É. Schost. 2013. Modular Composition Modulo Triangular Sets and Applications. *Computational Complexity* 22, 3 (2013), 463–516.
[34] S. Puchinger and A. Wachter-Zeh. 2017. Fast operations on linearized polynomials and their applications in coding theory. *J. Symb. Comput.* (2017).
[35] I. Reiner. 2003. *Maximal Orders*. Clarendon Press.
[36] T. Scanlon. 2001. Public Key cryptosystems based on Drinfeld modules Are insecure. *Journal of Cryptology* 14, 4 (2001), 225–230.
[37] The Sage Developers. 2020. *SageMath, the Sage Mathematics Software System (Version 9.2)*. `https://www.sagemath.org`.
[38] B. Wesolowski. 2022. Computing isogenies between finite Drinfeld modules. Cryptology ePrint Archive, Paper 2022/438. https://eprint.iacr.org/2022/438 https://eprint.iacr.org/2022/438.