Algorithmic Number Theory Symposium XVII, Groningen, July 6-10, 2026

COMPUTING SUBMODULES OF POINTS OF GENERAL DRINFELD MODULES OVER FINITE FIELDS

ANTOINE LEUDIÈRE AND RENATE SCHEIDLER

Abstract. We present an algorithm for computing the structure of any submodule of the module of points of a Drinfeld A-module over a finite field, where A is a function ring over $\mathbb{F}_q$. When the function ring is $A = \mathbb{F}_q[T]$, we additionally compute a Frobenius decomposition of the submodule. Our algorithms apply in particular to kernels of isogenies and torsion submodules. They are presented within the frameworks of Frobenius normal forms, presentations of modules, and Fitting ideals. They rely largely on efficient and classical linear algebra methods, combined with fast arithmetic of Ore polynomials. We analyze the complexity of our algorithms, explore optimizations, and provide an implementation in SageMath. Finally, we compute a simple invariant attached to a Drinfeld $\mathbb{F}_q[T]$-module that encodes all the polynomials in $\mathbb{F}_q[T]$ whose associated torsion is rational.

1. Introduction

Drinfeld modules are among the most important tools in function field arithmetic. Rank two Drinfeld modules bear close resemblance to elliptic curves and the theory of complex multiplication, rank one Drinfeld modules are analogous to roots of unity and the Kronecker-Weber theorem, but Drinfeld modules of higher rank have no direct analogue in characteristic zero. Drinfeld modules were introduced in 1977 to solve the Langlands conjectures for GL_r in function fields [Dri77; Laf01], with remarkable success. In the context of more practical applications, Drinfeld modules can be used for state-of-the-art factorization of polynomials over finite fields [DNS21] and computer algebra of linear polynomials [BMZ25]. More recently, Drinfeld modules were discovered to be extremely well suited to applications in coding theory [BDM24; MP26]. In light of the prominent technical and historical role of function fields and algebraic curves in coding theory [Gop83], Drinfeld modules offer a promising approach for significantly advancing the state-of-the-art in algebraic geometry and rank-metric codes.

Despite these successful applications of Drinfeld modules, research on their computational aspects lags far behind its counterpart for elliptic curves. The works of Caranay, Greenberg and Scheidler in 2018 were among the first to explicitly investigate algorithmic aspects of Drinfeld modules, focusing only on ordinary rank two Drinfeld modules [Car18; CGS20]. Musleh and Schost subsequently deployed modern computer algebra techniques to compute the characteristic polynomial of the Frobenius endomorphism for rank two Drinfeld modules over a finite field [MS23]. The rank two assumption made it possible to adapt methods from elliptic curves to Drinfeld modules. This rather restrictive assumption was only lifted as recently as 2023 by Musleh and Schost [MS23] and Caruso and Leudière [CL26] with the introduction of entirely new techniques. Importantly, some of these methods

and tools, such as Anderson motives for computational purposes, have no direct computationally efficient analogue for elliptic curves. Our work herein continues in this direction.

In [CL26], Caruso and Leudière computed norms of isogenies and characteristic polynomials of endomorphisms, the function field equivalent of point counting for elliptic curves. We build on this work and present an efficient general algorithm for computing the module structure, via the invariant factors, of the kernel of a morphism of Drinfeld modules over a finite field. In particular, the special case of the zero morphism recovers the structure of the full module of rational points. Our method takes a different approach from prior work; in particular, it is not rooted in elliptic curve machinery. While relying on classical tools such as Frobenius and Smith normal forms, presentations of modules, and Fitting ideals, it leverages features specific to Drinfeld modules and state-of-the-art techniques in computer algebra. Most importantly, we take advantage of the structure of finitely generated $\mathbb{F}_q$-algebra of the *function ring* A of a Drinfeld module. This has no analogue in the realm of elliptic curves, where the role of the function ring is played by the integers $\mathbb{Z}$. Our algorithm applies to Drinfeld A-modules of arbitrary rank over any finite field for any function ring A (as defined in § 2.3); we refer to this setting as the *general case*.

In the case $A = \mathbb{F}_q[T]$, we further give an algorithm for computing Frobenius decompositions of submodules of the module of points (Algorithm 4) and provide a precise asymptotic complexity analysis. Fixing q relative to other parameters, this algorithm computes the invariant factors and a Frobenius decomposition of the kernel of a morphism for a cost of $\widetilde{O}(dr + dn + d^\omega)$ operations in $\mathbb{F}_q$, where n is the τ-degree of the input morphism, r is the rank of its domain Drinfeld module, d is the $\mathbb{F}_q$-dimension of the base field, and $2 \leqslant \omega \leqslant 3$ is a feasible exponent for matrix multiplication (Proposition 4.2). Other precise complexity statements are also given herein.

Finally, given a Drinfeld $\mathbb{F}_q[T]$-module over a finite field, we use Anderson motives to compute the unique monic element $g \in \mathbb{F}_q[T]$ such that for any $a \in \mathbb{F}_q[T]$, the a-torsion is rational if and only if a divides g. The invariant g is the lcm of all such a, and we compute it without prior knowledge of the invariant factors of the module of rational points. We are unaware of any efficient method to accomplish the same task for elliptic curves.

Our main algorithms are implemented in SageMath [The26; Ayo+23]; the code is accessible on GitHub and is accompanied by a *Jupyter* notebook that readers can readily run in any web browser. All code and instructions are available at `https://github.com/kryzar/research-drinfeld-submodules/`.

To the best of our knowledge, ours is the first work to address the problem of effectively and efficiently computing invariant factors and Frobenius decompositions of submodules of points of a Drinfeld module. For elliptic curves, the best algorithms to compute invariant factors are generic abelian group algorithms. For example, the method of Sutherland [Sut11] proceeds by computing a basis via extracting discrete logarithms on subgroups of prime power order. While discrete logarithms can be readily computed for Drinfeld modules [Sca01], adapting the method of Sutherland to this setting would yield an algorithm that is generally slower than ours, both in terms of asymptotic complexity and implementation. Moreover, our method has the advantages of being deterministic, easily implemented, and close to optimal.

The paper is organized as follows.

(1) Section 2 presents the necessary background. We briefly introduce Ore polynomials (§ 2.2) and Drinfeld modules (§ 2.3). We review the computer algebra of matrices, polynomials and Ore polynomials, with an emphasis on complexity models (§ 2.4). Finally, we present a general framework for computing invariant factors and Frobenius decompositions of finite modules over a Dedekind domain that are uniquely defined by a finite number of matrices (§ 2.5). This framework readily applies to Drinfeld modules. In particular, in § 2.5.1 we show how the invariant factors and a Frobenius decomposition of the kernel of a morphism can be recovered by computing the Frobenius normal form and a change-of-basis matrix of the action of the Drinfeld module on this kernel. This approach does not extend to the general case, where we need to resort to Fitting ideals to recover invariant factors (§ 2.5.2).
(2) Section 3 describes state-of-the-art multipoint evaluation (§ 3.1) and revisits algorithms related to divisibility chains of polynomials (§ 3.2).
(3) Section 4 presents our main algorithms. Once again, we distinguish between the case $A = \mathbb{F}_q[T]$ (§ 4.1) and the general case (§ 4.2). Still in the case $A = \mathbb{F}_q[T]$, we present alternative methods for computing torsion submodules (§ 4.1.2) and discuss the efficiency of our algorithms depending on different low level primitives (§ 4.1.3).
(4) Section 5 presents a simple method for finding all $a \in \mathbb{F}_q[T]$ such that the a-torsion is rational, given a Drinfeld $\mathbb{F}_q[T]$-module over a finite field.

2. Background

2.1. **Invariant factors and Frobenius decompositions.** We begin by defining the quantities that encode the structure of a finite module over a Dedekind domain. Our goal is to compute these quantities for submodules of points of Drinfeld modules over finite fields.

Theorem 2.1. *Let A be a Dedekind domain and W a finite A-module. There exist ideals $\mathfrak{d}_1, \dots, \mathfrak{d}_\ell \subset A$ and elements $x_1, \dots, x_\ell \in W$ such that the following hold.*

(1) *The elements generate the module: $W = Ax_1 \oplus \cdots \oplus Ax_\ell$;*
(2) *The ideals form a divisibility chain: $\mathfrak{d}_1 \mid \cdots \mid \mathfrak{d}_\ell$;*
(3) *For all i, x_i has annihilator $\mathfrak{d}_i$,* i.e. *$Ax_i \simeq A/\mathfrak{d}_i$.*

In particular, $W \simeq \prod_{i=1}^n A/\mathfrak{d}_i$, and the chain $(\mathfrak{d}_1, \dots, \mathfrak{d}_\ell)$ is unique. We call the ideals $\mathfrak{d}_1, \dots, \mathfrak{d}_\ell$ the invariant factors *of W and the elements $x_1, \dots, x_\ell$ a* Frobenius decomposition *of W.*

This theorem is a more explicit version of the classical result on the decomposition of finitely generated modules over a Dedekind domain [Eis95, Section 19.5].

2.2. **Ore polynomials.** Ore polynomials are the main ingredient in the definition of Drinfeld modules and their isogenies. Throughout, let $\mathbb{F}_q$ be a finite field with q elements. Let K be a field extension of $\mathbb{F}_q$, and let τ be the $\mathbb{F}_q$-linear Frobenius endomorphism of $\overline{K}$ defined by $\tau(x) = x^q$.

Definition 2.2. The set

$$K\{\tau\} \coloneqq \left\{ \sum_{i=1}^{n} x_i \tau^i,\ n \in \mathbb{Z}_{\geqslant 0},\ x_1, \dots, x_n \in K \right\}$$

is an $\mathbb{F}_q$-algebra under composition and addition of endomorphisms, called the $\mathbb{F}_q$-algebra of *Ore polynomials* (relative to $\mathbb{F}_q$ and K).

Although multiplication of Ore polynomials is usually noncommutative, Ore polynomials retain many key features of classical polynomials. Specifically, we have the following:

(1) The *τ-degree* of a nonzero Ore polynomial $f = \sum_{i=1}^{n} x_i \tau^i \in K\{\tau\}$, denoted $\deg_\tau(f)$, is the maximal i such that $x_i \neq 0$. The *τ-valuation* of f is the minimal i such that $x_i \neq 0$.
(2) The kernel of $f \in K\{\tau\}$ is an $\mathbb{F}_q$-vector space denoted $\ker f$. If f is nonzero, then $\dim_{\mathbb{F}_q}(\ker(f)) \leqslant \deg_\tau(f)$. We write $\ker_K(f) \coloneqq \ker(f) \cap K$ for the roots of f that belong to K.
(3) An Ore polynomial $f \in K\{\tau\}$ is said to be *separable* if $\dim_{\mathbb{F}_q}(\ker(f)) = \deg_\tau(f)$ and *inseparable* if $\dim_{\mathbb{F}_q}(\ker(f)) < \deg_\tau(f)$.
(4) The algebra $K\{\tau\}$ of Ore polynomial is a right-euclidean domain for the τ-degree. For any $f, g \in K\{\tau\}$, there exist $\alpha, \beta \in K\{\tau\}$ such that $f = \alpha g + \beta$ and $\deg_\tau(\beta) < \deg_\tau(g)$. Therefore, for any $f, g \in K\{\tau\}$ not both zero, we can define $\operatorname{rgcd}(f, g)$ as the unique monic Ore polynomial such that $K\{\tau\}f + K\{\tau\}g = K\{\tau\}\operatorname{rgcd}(f, g)$, and we define $\operatorname{llcm}(f, g)$ to be the unique monic Ore polynomial such that $K\{\tau\}f \cap K\{\tau\}g = K\{\tau\}\operatorname{llcm}(f, g)$.

2.3. **Drinfeld modules.** We now turn to the definition of Drinfeld modules and related objects, including isogenies and modules of points. The reader is referred to [Pap23] or [Arm+25] for a general introduction to the subject. We begin by establishing notation and terminology. Throughout, let C be a smooth projective geometrically connected curve over $\mathbb{F}_q$ with a geometric closed point ∞. Let A be the ring of functions of $\mathbb{F}_q(C)$ that are regular outside ∞. Then A is a Dedekind domain called a *function ring*. Let K be a field extension of $\mathbb{F}_q$, equipped with a morphism of $\mathbb{F}_q$-algebras $\gamma : A \to K$. The pair (K, γ) is called an *A-field*. Let $\overline{K}$ be an algebraic closure of K.

Definition 2.3. A *Drinfeld A-module over* (K, γ) is a morphism of $\mathbb{F}_q$-algebras

$$\begin{array}{rccc} \phi : & A & \to & K\{\tau\} \\ & a & \mapsto & \phi_a, \end{array}$$

such that $\operatorname{Im}(\phi) \not\subset K$ and for each $a \in A$, the constant coefficient of ϕ_a is $\gamma(a)$.

Drinfeld modules, despite their names, are not modules, but $\mathbb{F}_q$-algebra homomorphisms. However, they give rise to module structures on extensions L/K.

Definition 2.4. For any extension L/K, the *module of L-points of a Drinfeld module* ϕ, denoted $\phi(L)$, is the A-module defined by

$$\begin{array}{ccc} A \times L & \to & L \\ (a, z) & \mapsto & \phi_a(z). \end{array}$$

The elements of $\phi(K)$ are called *rational points*.

The underlying set of $\phi(L)$ is the same for all Drinfeld modules, but its structure as an A-module depends on ϕ. Our goal in this paper is to compute the invariant factors of any A-submodule of $\phi(K)$ when K is finite. When $A = \mathbb{F}_q[T]$, we also give a Frobenius decomposition of any submodule of $\phi(L)$ (Algorithm 4). To represent these submodules, we introduce morphisms and isogenies.

Definition 2.5. Let ϕ, ψ be two Drinfeld A-modules over (K, γ). A *morphism from ϕ to ψ* is an Ore polynomial $u \in K\{\tau\}$ such that $u\phi_a = \psi_a u$ for every $a \in A$. An *isogeny* is a nonzero morphism.

The kernel of a morphism u with domain ϕ is a finite A-submodule of $\phi(\overline{K})$. To avoid ambiguity, we write $\phi(\ker(u))$ for $\ker(u)$ when viewed as an A-module. Conversely, every finite A-submodule of $\phi(\overline{K})$ is the kernel of an isogeny starting from ϕ that can be effectively computed [Arm+25, Proposition 3.2]. Therefore, our aim is to compute invariant factors and Frobenius decompositions of kernels of morphisms.

An important invariant attached to every Drinfeld module is its *rank*.

Lemma 2.6. *Let ϕ be a Drinfeld A-module over (K, γ). There exists a unique integer $r \geqslant 1$ such that for all $a \in A$, $\deg_\tau(\phi_a) = r \dim_{\mathbb{F}_q}(A/aA)$. The integer r is called the* rank *of ϕ.*

Remark 2.7. Defining a Drinfeld A-module amounts to defining an injection $A \to K\{\tau\}$. It is not clear how to do so, and such injections may not exist at all [Pap23, Remark 4.3]. On the other hand, the case $A = \mathbb{F}_q[T]$ (corresponding to $C = \mathbb{P}^1_{\mathbb{F}_q}$) is straightforward: a Drinfeld $\mathbb{F}_q[T]$-module ϕ is uniquely defined by the image ϕ_T of T, and its rank is simply $\deg_\tau(\phi)$.

Finally, a particular topic of interest, for example for computing Tate modules, Weil pairings and, more recently, codes in rank metric [BDM24; MP26] is the computation of the kernel of ϕ_a for $a \in A$.

Definition 2.8. Let $a \in A$. The *a-torsion* of ϕ is the kernel of the morphism ϕ_a (from ϕ to itself). It is an A-submodule of $\phi(\overline{K})$, denoted $\phi[a] \coloneqq \phi(\overline{K})[a]$. The a-torsion of ϕ contained in L is $\phi(L)[a] \coloneqq \phi[a] \cap \phi(L)$. The a-torsion of ϕ is said to be *L-rational* if $\phi(L)[a] = \phi[a]$.

2.4. **Computer algebra.** We now specify our complexity model and recall classical computer algebra reductions for polynomials and matrices in § 2.4.1. We particularly emphasize fast primitives for Ore polynomials (§ 2.4.2, § 2.4.3).

2.4.1. *Cost functions for polynomials and matrices.* We start by defining cost functions. Let F be a field. Two polynomials in $F[T]$ with degrees $\leqslant n$ can be multiplied for a cost of $O(\mathbf{PM}(n))$ operations in F. We assume that $\mathbf{PM}$ is *super-additive*, *i.e.* $\mathbf{PM}(x + y) \geqslant \mathbf{PM}(x) + \mathbf{PM}(y)$ for all inputs x, y. When F is a finite field, we can pick $\mathbf{PM}(n) = n \log n \log\log n$ [GG13, § 8.3] (see also [HHL17] for a slightly better complexity). Moreover:

- The Euclidean division of two polynomials in $F[T]$ with respective degrees m and n, $m \geqslant n$, can be performed for a cost of $O(\mathbf{PD}(m, n))$ operations in F. We can pick $\mathbf{PD}(m, n) = O(\mathbf{PM}(m - n) + \mathbf{PM}(n))$ [GG13, § 11].
- The gcd of two polynomials with degrees $\leqslant n$ can be computed for a cost of $O(\mathbf{PG}(n))$ operations in F. We can pick $\mathbf{PG}(n) = \mathbf{PM}(n) \log n$ [GG13, § 11] and assume that $\mathbf{PG}(n) \in O(n^2)$.

Next, we introduce cost functions for matrix arithmetic. The multiplication of two matrices in $F^{d \times d}$ can be performed for a cost of $O(\mathbf{MM}(d))$ operations in F. In particular, we let $2 \leqslant \omega \leqslant 3$ be such that $\mathbf{MM}(d) = d^\omega$.

2.4.2. *Ore polynomials.* Almost all our algorithms rely on low level primitives on Ore polynomials: Euclidean division, llcm and rgcd computation, and multipoint evaluation. Let K be a finite extension of $\mathbb{F}_q$ with $[K:\mathbb{F}_q]=d$. The state of the art for computer algebra of Ore polynomials in $K\{\tau\}$ is due to Caruso and Le Borgne [CL17a] and Puchinger and Wachter-Zeh [PW18]; see also Boucher and Nouetowa [BN25]. These papers use the same computation model, for which one issue is the assumption that all operations in K—including applications of the Frobenius endomorphism (*i.e.* computing x^q for $x\in K$)—can be performed for a cost of $\widetilde{O}(d)$ operations in $\mathbb{F}_q$. This assumption does not always hold (see [MS23] for a discussion on this topic). Therefore, we count arithmetic operations and applications of the Frobenius endomorphism separately.

Definition 2.9. We let

$$\mathbf{SM}=(\mathbf{SM}_{\mathrm{A}},\mathbf{SM}_{\mathrm{F}}):\quad \mathbb{Z}_{>0}^2\quad\to\quad [1,\infty[$$

be a function such that two Ore polynomials with degrees $\leqslant n$ can be multiplied for a cost of $O(\mathbf{SM}_{\mathrm{A}}(n))$ arithmetic operations in K and $O(\mathbf{SM}_{\mathrm{F}}(n))$ Frobenius applications. We refer to this cost as $O(\mathbf{SM}(n))$ *arithmetic and Frobenius operations in* K.

2.4.3. *Divide and conquer algorithms.* Later, we describe the fast multipoint evaluation algorithm of Ore polynomials (§ 3.1) using *divide and conquer* algorithms. We recall how to analyse the cost of such algorithms. In our context, for an input of size ℓ, the cost of a divide and conquer algorithm is given by a recursion of the form

$$T(\ell)\leqslant 2T\left(\frac{\ell}{2}\right)+\alpha\left(\left\lceil\frac{\ell}{2}\right\rceil\right), \tag{2.1}$$

If α is super-additive, we can solve the recursion:

$$T(\ell)=O(\alpha(\ell)\log\ell). \tag{2.2}$$

It is therefore convenient to introduce the following notation:

Proposition 2.10. *For $\alpha:\mathbb{Z}_{>0}\to[1,\infty[$, we define $\alpha^{\geqslant 1}:\mathbb{Z}_{>0}\to[1,\infty[$ via*

$$\alpha^{\geqslant 1}(\ell):=\ell\sup_{1\leqslant\ell'\leqslant\ell}\frac{\alpha(\ell')}{\ell'}.$$

The function $\alpha^{\geqslant 1}$ is super-additive and is the minimal such function exceeding α.

This definition is frequently used for analysing the complexity of recursive algorithms (see *e.g.* [CL17b, § 3]). In our case, we do not assume **SM** to be super-additive, as the value for **SM** given by Caruso and Le Borgne does not satisfy that property [CL17b, p. 83]. Following their methodology, we will instead use $\mathbf{SM}^{\geqslant 1}:=(\mathbf{SM}_{\mathrm{A}}{}^{\geqslant 1},\mathbf{SM}_{\mathrm{F}}{}^{\geqslant 1})$ to solve recursions coming from divide and conquer algorithms.

2.5. **Modules from matrices.** We now present the necessary material from module theory that forms the foundation of our algorithms (§ 4). We first treat the case of modules defined by one endomorphism (§ 2.5.1), followed by the general case of modules defined by multiple endomorphisms (§ 2.5.2). For computational purposes, we represent these endomorphisms by matrices relative to a fixed basis. We later apply these two cases to Drinfeld $\mathbb{F}_q[T]$-modules (§ 4.1), and to general Drinfeld modules (§ 4.2), respectively.

2.5.1. *Modules from one matrix.* Let F be a field, V an F-vector space, and φ an endomorphism of V. This set-up canonically yields an $F[T]$-module structure on V:

$$\begin{array}{rcl} F[T] \times V & \to & V \\ (a(T), x) & \mapsto & a(\varphi)(x). \end{array} \tag{2.3}$$

This new module is denoted $V[\varphi]$.

If V has finite F-dimension, we can endow it with an F-basis $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_d)$, and φ can be represented by a matrix $M \in F^{d\times d}$. The invariant factors of $V[\varphi]$ can be retrieved by computing the *Frobenius normal form* of the matrix M: a block diagonal matrix of the form

$$\mathrm{F}_M = \begin{pmatrix} \boxed{C_{d_1}} & & \\ & \ddots & \\ & & \boxed{C_{d_\ell}} \end{pmatrix} \in F^{d\times d}$$

such that

(1) each C_{d_i} is the companion matrix of a polynomial $d_i \in F[T]$;
(2) the polynomials form a divisibility chain: $d_1 | \cdots | d_\ell$;
(3) the matrices M and F_M are similar over F.

If furthermore S is a change-of-basis matrix such that

$$\mathrm{F}_M = SMS^{-1},$$

then one immediately recovers a Frobenius decomposition of $V[\varphi]$ from S^{-1}. First, we fix notation: write $S^{-1} \in F^{d\times d}$ as a horizontal block matrix in which the i-th block has $\deg(d_i)$ columns, with the first column of the i-th block labeled x_i:

$$S^{-1} = \left(\boxed{x_1 \cdots} \quad \cdots \quad \boxed{x_\ell \cdots} \right) \in F^{d\times d}.$$

Proposition 2.11. *With the above notation, the ideals $(d_1), \ldots, (d_\ell)$ are the invariant factors of $V[\varphi]$, and the elements whose coordinates relative to the F-basis $\mathcal{E}$ of V are the vectors $x_1, \ldots, x_\ell$ form a Frobenius decomposition of $V[\varphi]$.*

Proposition 2.11 can be proved directly using the correspondence between $F[T]$-modules and F-vector spaces endowed with a distinguished F-linear endomorphism. We refer to [Bou03, Chapter 7, § 5.1-3] for details.

By Proposition 2.11, to recover the invariant factors and a Frobenius decomposition of $V[\varphi]$, it suffices to represent φ as a matrix and compute its Frobenius normal form. To compute Frobenius normal forms, one can use an algorithm of Storjohann [Sto00, Chapter 9].

Lemma 2.12. *Let $M \in F^{d\times d}$ be a matrix. The Frobenius normal form of M, along with a unimodular change-of-basis matrix, can be computed for a cost of $O(\mathbf{MM}(d)(\log d)(\log\log d))$ operations in F.*

Remark 2.13. We will use the above framework to compute invariant factors and Frobenius decompositions of kernels of isogenies of Drinfeld $\mathbb{F}_q[T]$-modules (§ 4.1). Let ϕ be a Drinfeld $\mathbb{F}_q[T]$-module over a finite field K and u a morphism whose domain is ϕ. To apply the above framework, we put $F = \mathbb{F}_q$, $V = \ker_K(u)$, $\varphi = \phi_T$

and $V[\varphi] = \phi(\ker_K(u))$. We will compute the invariant factors and a Frobenius decomposition of $\phi(\ker_K(u))$ (an $\mathbb{F}_q[T]$-module) by computing the Frobenius normal form of the matrix of ϕ_T relative to an $\mathbb{F}_q$-basis of $\ker_K(u)$ (an $\mathbb{F}_q$-vector space), and extracting the invariant factors as described above.

2.5.2. *Modules from multiple endomorphisms.* Let F be a field and V an F-vector space of finite dimension d. Let $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_d)$ be an F-basis of V. Here, we do not assume that V comes with a distinguished endomorphism, but rather that it extends to an A-module, where A is an F-algebra with a finite number (but not necessarily only one) of generators $g_1, \dots, g_n \in A$. This A-module structure is denoted W, and we write $V_A \coloneqq A \otimes_F V$.

Remark 2.14. In the case of § 2.5.1, $A = F[T]$ has one generator T, and the $F[T]$-module $V[\varphi]$ satisfies the above assumption by virtue of Equation (2.3).

To handle the difficulty of multiple generators, we use the notion of presentation of modules.

Definition 2.15. Let A be a ring and W an A-module. A *presentation* of W is an exact sequence of A-modules

$$W' \xrightarrow{\alpha} W'' \xrightarrow{\beta} W \longrightarrow 0.$$

In our setting, we have a finite presentation

$$(V_A)^n \xrightarrow{\chi} V_A \xrightarrow{\pi} W \longrightarrow 0,$$

where we define the two maps χ and π by

$$\chi\left((a_k \otimes v_k)_{1 \leqslant k \leqslant n}\right) = \sum_{k=1}^{n} \left(a_k g_k \otimes v_k - a_k \otimes (g_k \cdot v_k)\right),$$
$$\pi(a \otimes v) = a \cdot v.$$

The F-basis $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_d)$ of V yields an A-basis $(1_A \otimes_F \varepsilon_1, \dots, 1_A \otimes_F \varepsilon_d)$ of V_A, which extends to an A-basis of V_A^n whose (i,j)-th vector is $(0, \dots, 0, 1_A \otimes_F \varepsilon_j, 0, \dots, 0)$ for $1 \leqslant j \leqslant d$, where $1_A \otimes_F \varepsilon_j$ appears in the i-th position for $1 \leqslant i \leqslant n$. Then χ is represented by a matrix M_χ which is a block matrix of companion matrices

$$M_\chi = \begin{pmatrix} \boxed{g_1 \operatorname{Id} - M_1} \\ \vdots \\ \boxed{g_n \operatorname{Id} - M_n} \end{pmatrix} \in A^{d \times nd}, \tag{2.4}$$

where each M_i is the matrix of the F-linear endomorphism $x \mapsto g_i x$.

Remark 2.16. In the setting of Remark 2.14 where $A = F[T]$, we see that M_χ is the characteristic matrix $T \operatorname{Id} - M$, and thus $V[\varphi] \simeq A^d / \operatorname{Im}(T \operatorname{Id} - M)$. Therefore, the invariant factors of $V[\varphi]$ are exactly the invariant factors of the characteristic matrix $T \operatorname{Id} - M$, which are the nonzero diagonal elements of its Smith normal form. The Smith normal form can be computed using a straightforward adaptation from

integers to polynomials of an algorithm of Storjohann [Sto00, Section 8]. However, computing the Smith normal form of $T\,\mathrm{Id} - M$ may in practice be more expensive than computing the Frobenius normal form of M. Indeed, a generic algorithm for polynomial matrices would fail to leverage the structure of the characteristic matrix, nor can it avoid potentially costly operations on polynomials.

The invariant factors of W are the invariant factors of the matrix M_χ, which can be retrieved using Fitting ideals [Fit36]. Recall that a k-by-k *minor* of an r-by-s matrix M is the determinant of a k-by-k-submatrix obtained by removing $r-k$ rows and $s-k$ columns from M.

Lemma 2.17. *Let*

$$W' \xrightarrow{\alpha} W'' \xrightarrow{\beta} W \longrightarrow 0$$

be a presentation of W, and assume that α is given (in some basis) by a matrix $M \in A^{r\times s}$ with $r, s \in \mathbb{Z}_{\geqslant 0}$. For $0 \leqslant i \leqslant d-1$, let

$$\mathrm{Fitt}_i(W)$$

be the ideal generated by the $(d-i)$-by-$(d-i)$ minors of M, and let $\mathrm{Fitt}_d(W) := A$; the value of $\mathrm{Fitt}_i(W)$ does not depend on the choice of presentation. For $0 \leqslant i \leqslant d$, we call $\mathrm{Fitt}_i(W)$ the i-th Fitting ideal of W.

A classical result [Sta26, § 15.8] asserts that the Fitting ideals form a divisibility chain

$$\mathrm{Fitt}_d(W) \mid \cdots \mid \mathrm{Fitt}_0(W),$$

and we have the following useful lemma [Sta26, Lemma 15.8.4].

Lemma 2.18. *Let $\mathfrak{d}_1, \ldots, \mathfrak{d}_\ell$ be the invariant factors of W. Then*

$$\mathfrak{d}_i = \frac{\mathrm{Fitt}(d-i)}{\mathrm{Fitt}(d+1-i)}.$$

Remark 2.19. As in Remark 2.13, we explain how to apply the above framework to compute invariant factors and Frobenius decompositions of kernels of isogenies of Drinfeld modules over a general function ring A, where A is an $\mathbb{F}_q$-algebra (§ 4.2). Let ϕ be a Drinfeld A-module over a finite field K and let u be a morphism whose domain is ϕ. To apply the above framework, we put $F = \mathbb{F}_q$, $V = \ker_K(u)$ and $W = \phi(\ker_K(u))$. Let $g_1, \ldots, g_n$ be generators of A as an $\mathbb{F}_q$-algebra. To compute the invariant factors of $\phi(\ker_K(u))$ (an A-module), it therefore suffices to compute the matrices $M_1, \ldots, M_m$ of the actions of $\phi_{g_1}, \ldots, \phi_{g_n}$ on $\ker_K(u)$ (an $\mathbb{F}_q$-vector space), compute the matrices $g_i\,\mathrm{Id} - M_i$ from these matrices and stack them into the matrix M_χ, compute the Fitting ideals of M_χ, and finally apply Lemma 2.18 to obtain the invariant factors of $\phi(\ker_K(u))$.

3. Computer algebra

In § 2.5, we developed the main theoretical tools required to compute the invariant factors of an A-submodule of $\phi(K)$ when K is finite. It remains to analyze the computational complexity of the underlying algorithms. We revisit Ore polynomials (§ 3.1) and divisibility chains of polynomials (§ 3.2).

3.1. **Computer algebra of Ore polynomials.** We present a self-contained and straightforward efficient method for multipoint evaluation of Ore polynomials. This algorithm was already presented by Puchinger and Wachter-Zeh in [PW18]. Their complexity analysis is tighter but relies on more cost functions, while ours only depends on **SM** and thus yields simpler complexity results.

First, we need to assess the cost of fast Euclidean division, llcm and rgcd. Ore polynomial analogues of the classical methods were already described by Caruso and Le Borgne (see [CL17a, § 3.2.1] and [CL17b, § 3.2.2]). We state their complexity to fit with our complexity model.

Lemma 3.1. *Let $f, g \in K\{\tau\}$ be two Ore polynomials with degrees $\leqslant n$. The right-Ore Euclidean division, the rgcd and the llcm of f by g can all be performed for a cost of $O(\mathbf{SM}^{\geqslant 1}(n)\log n)$ arithmetic and Frobenius operations in K.*

Next, we discuss multipoint evaluation. Let f be an Ore polynomial of degree n, and let $X = (x_1, \ldots, x_\ell)$ be elements of K, with $n \leqslant \ell$. We wish to efficiently compute $f(x_1), \ldots, f(x_\ell)$. For any $x \in K$, set $L_x \coloneqq \tau - \frac{\tau(x)}{x} \in K\{\tau\}$. If $R_x \in K$ is the (constant) remainder in the right-division of f by L_x, then $f(x) = R_x x$. In other words, $f(x)$ is the evaluation of R_x, viewed as the constant Ore polynomial $R_x\tau^0$, at x. For elements $x_{i_1}, \ldots, x_{i_{\ell'}} \in X$, we set

$$L_{x_{i_1},\ldots,x_{i_{\ell'}}} \coloneqq \operatorname{llcm}(L_{x_{i_1}}, \ldots, L_{x_{i_{\ell'}}}).$$

Let $R_{x_{i_1},\ldots,x_{i_{\ell'}}}$ be the remainder in the right-division of f by $L_{x_{i_1},\ldots,x_{i_{\ell'}}}$. Then for any x_{i_j} we have

$$f(x_{i_j}) = R_{x_{i_1},\ldots,x_{i_{\ell'}}}(x_{i_j}).$$

With these identities, we can divide up the problem of multipoint evaluation as follows. Assume for clarity of presentation that ℓ is even. We split our inputs into two halves: $X_l = (x_1, \ldots, x_{\frac{\ell}{2}})$ and $X_r = (x_{\frac{\ell}{2}+1}, \ldots, x_\ell)$. Provided that we know L_{X_l} and L_{X_r}, we obtain $f(x_1), \ldots, f(x_\ell)$ by computing R_{X_l} and R_{X_r}, and then $R_{X_l}(x_i)$ for all $x_i \in X_l$ as well as $R_{X_r}(x_i)$ for all $x_i \in X_r$. The cost of this step is two Euclidean divisions of Ore polynomials of degrees $\leqslant \frac{\ell}{2}$ and two recursive calls on an entry of size $\leqslant \frac{\ell}{2}$.

In Algorithm 1, following the classical method, we pre-compute the llcm of X recursively as a binary tree, called the *llcm tree of X*.

Algorithm 1: LlcmTree

Input: A subfamily X' of $X \coloneqq (x_1, \ldots, x_\ell)$
Output: The llcm tree of X'

if $X' = \{x_i\}$ *is a singleton* **then**
  Return the tree whose only leaf is L_{x_i}.
else
  Split X' into two halves X'_l and X'_r.
  Compute and set $T_l = \text{LlcmTree}(X'_l)$.
  Compute and set $T_r = \text{LlcmTree}(X'_r)$.
  Compute the llcm L of the roots of T_l and T_r.
  return *the binary tree rooted at L with subtrees T_l and T_r.*

Let T_X be the tree generated by Algorithm 1, which we assume precomputed. We obtain Algorithm 2 for multipoint evaluation.

Algorithm 2: MULTIPOINTEVALUATION

Input: An Ore polynomial f, elements $x_1, \dots, x_\ell \in K$ and the tree T_X
Output: The evaluations $f(x_1), \dots, f(x_\ell)$

if *If* $\ell = 1$ **then**
  return $f(x_1)$
else
  Split $x_1, \dots, x_\ell$ into two halves X_l and X_r (padding if necessary).
  Let $\mathcal{T}_l$ be the left subtree of T_X, and get L_{X_l} as its root.
  Let $\mathcal{T}_r$ be the right subtree of T_X, and get L_{X_r} as its root.
  Compute R_{X_l} as the Euclidean division of f by L_{X_l}.
  Compute R_{X_r} as the Euclidean division of f by L_{X_r}.
  Compute $f(x_1), \dots, f(x_{\ell/2}) = \text{MULTIPOINTEVALUATION}(R_{X_l}, X_l, \mathcal{T}_l)$.
  Compute $f(x_{\ell/2+1}), \dots, f(x_\ell) = \text{MULTIPOINTEVALUATION}(R_{X_r}, X_r, \mathcal{T}_r)$.
  return $f(x_1), \dots, f(x_\ell)$

Lemma 3.2. *Let $x_1, \dots, x_\ell \in K$ and let f be an Ore polynomial of degree $\leqslant \ell$. The values $f(x_1), \dots, f(x_\ell)$ can be computed for a cost of $O(\mathbf{SM}^{\geqslant 1}(\ell)(\log \ell)^2)$ arithmetic and Frobenius operations in K.*

Proof. The cost of this task is the cost of sequentially running Algorithm 1, followed by Algorithm 2. For both these divide and conquer algorithms, the cost of an input of size ℓ is given by a recursion of the form (2.1), with $\alpha(\ell) = \mathbf{SM}^{\geqslant 1}(\ell) \log \ell$ (Lemma 3.1). We conclude the proof by applying Equation (2.2) and the super-additivity of $x \mapsto \mathbf{SM}^{\geqslant 1}(x) \log x$ (Proposition 2.10). □

Remark 3.3. The cost of Ore polynomial multipoint evaluation is $O(\mathbf{SM}^{\geqslant 1}(\ell)(\log \ell)^2)$ arithmetic and Frobenius operations, whereas the cost of classical multipoint evaluation is $O(\mathbf{PM}(\ell) \log \ell)$ arithmetic operations [GG13, § 10.1]. The extra $(\log \ell)$ factor comes from the fact that we have to use llcm's instead of products.

3.2. **Computer algebra for divisibility chains.** We now turn to efficient computations involving divisibility chains, which we need for computing the invariant factors and a Frobenius decomposition of torsion submodules when we know these quantities for the full module of rational points (§ 4.1.2). Let F be a field, $a \in F[T]$ a polynomial, and $d_1 \mid \cdots \mid d_\ell$ a divisibility chain of ℓ polynomials in $F[T]$. Algorithm 3 efficiently computes the polynomials $\gamma_i \coloneqq \gcd(a, d_i)$ and $\rho_i \coloneqq d_i/\gamma_i$ for all $1 \leqslant i \leqslant \ell$. Later on, we wish to evaluate these polynomials on vectors and matrices. Let $\delta \coloneqq \deg(d_1) + \cdots + \deg(d_\ell)$.

Lemma 3.4. *Algorithm 3 computes $(\gamma_1, \dots, \gamma_\ell)$ and $(\rho_1, \dots, \rho_\ell)$ for a cost of $O(\mathbf{PM}(\deg(a)) + \mathbf{PM}(\delta) \log \delta)$ operations in F, where $\delta = \sum_{i=1}^{\ell} \deg(d_i)$.*

Proof. All operations are arithmetic operations in F. Step 1 costs $O(\mathbf{PM}(\deg(a) - \deg(d_\ell)) + \mathbf{PM}(\deg(a)))$ operations. The two subsequent steps cost $O(\mathbf{PG}(\deg(a') + \deg(d_\ell)) + \mathbf{PD}(\deg(d_\ell) + \deg(\gamma_\ell)))$ operations, which is $O(\mathbf{PG}(\deg(d_\ell)))$. The rest amounts to $\sum_{i=1}^{\ell-1} O(\mathbf{PG}(\deg(d_{i+1})) + \mathbf{PD}(\deg(d_i)))$ operations, which is $O(\mathbf{PG}(d))$ operations. We conclude the proof using the fact that $\mathbf{PG}(\delta) = \mathbf{PM}(\delta) \log \delta$. □

Next, we describe how to evaluate divisibility chains on matrices and vectors. Let M be a d-by-d matrix with entries in F, and let $x_1, \dots, x_\ell$ be vectors of F^d.

Algorithm 3: ChainGcd

Input: The polynomials a and $d_1, \dots, d_\ell$.
Output: The lists $(\gamma_1, \dots, \gamma_\ell)$ and $(\rho_1, \dots, \rho_\ell)$.

```
Compute a' = a mod d_ℓ.
Compute γ_ℓ = gcd(a', d_ℓ).
Compute ρ_ℓ = d_ℓ/γ_ℓ.
for i ranging down from ℓ − 1 to 1 do
    Compute γ_i = gcd(γ_{i+1}, d_i).
    Compute ρ_i = d_i/γ_i.
return (γ_i, ..., γ_ℓ) and (ρ_1, ..., ρ_ℓ).
```

We are interested in computing the vectors $y_i = \rho_i(M)x_i$ for $1 \leqslant i \leqslant \ell$. Once the polynomials $(\rho_i)_{1\leqslant i\leqslant \ell}$ are known (see Lemma 3.4 for the cost of their computation), there are different possible approaches for computing $(y_i)_{1\leqslant i\leqslant \ell}$:

- We can naïvely compute $(y_i)_{1\leqslant i\leqslant \ell}$ for a cost of $O(\delta d^2)$ operations in F.
- In our applications, we have $\delta \leqslant d$. We can thus compute the matrices $(\rho_i(M))_{1\leqslant i\leqslant \ell}$ sequentially, for a total cost of $O(\ell\, \mathbf{MM}(d)(\log d)(\log\log d))$ operations in F, using an algorithm of Storjohann [Sto00, Chapter 9]. Then we compute $(y_i)_{1\leqslant i\leqslant \ell}$ for the same asymptotic cost.

To account for these different approaches for computing $(y_i)_{1\leqslant i\leqslant \ell}$, we introduce a cost function.

Definition 3.5. Let

$$\mathbf{EV}: \quad (\mathbb{Z}_{>0})^3 \quad \to \quad [1, \infty[$$

be a function such that the quantities $d'_1(M)x_1, \dots, d'_{\ell'}(M)x_{\ell'}$ can be computed for a cost of $O(\mathbf{EV}_d(\ell', \delta'))$ operations in F, for any matrix $M \in F^{d'\times d'}$, any set of vectors $x_1, \dots, x_{\ell'} \in F^d$, and any divisibility chain $d'_1 \mid \cdots \mid d'_\ell \in F[T]$, where $\delta' = \sum_{i=1}^{\ell'} \deg(d'_i) \leqslant d'$.

4. Application to Drinfeld modules

We now have all the ingredients for computing the structure of submodules of points of Drinfeld modules over finite fields. More precisely, we let A be a function ring over $\mathbb{F}_q$ and (K, γ) an A-field with $d = \dim_{\mathbb{F}_q}(K)$. We fix an ordered $\mathbb{F}_q$-basis $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_d)$ of K. Let ϕ be a Drinfeld A-module over (K, γ) with rank r. Our goal is to compute the invariant factors of $\phi(\ker_K(u))$ for any morphism u whose domain is ϕ. Note that this includes $\phi(K)$, as this is the kernel of the zero morphism. In the case $A = \mathbb{F}_q[T]$, we also compute a Frobenius decomposition of any $\mathbb{F}_q[T]$-submodule of $\phi(K)$ (presented as the kernel of an isogeny), and give a detailed complexity analysis of our algorithms. This problem is a direct instance of the framework presented in § 2.5. We first consider the case $A = \mathbb{F}_q[T]$ (Remark 2.13), as it can take advantage of very efficient tools, specifically Frobenius normal forms of matrices with entries in a field instead of Fitting ideals of polynomial matrices. Then we turn to the general setting of Drinfeld A-modules where A is a function ring (Remark 2.19).

4.1. **The case $A = \mathbb{F}_q[T]$.** This case falls within the framework presented in § 2.5.1. Reusing the same notation as in § 2.5.1, we have $F = \mathbb{F}_q$, $\varphi = \phi_T$ and $V = \ker_K(u)$. The invariant factors and the Frobenius decomposition of $\phi(\ker_K(u))$ (as an $\mathbb{F}_q[T]$-module) can be obtained by computing the Frobenius normal form of the matrix of ϕ_T acting on $\ker_K(u)$ (as an $\mathbb{F}_q$-vector space). We thus need two ingredients: fast computation of matrices of Ore polynomials and Frobenius normal forms (Lemma 2.12).

4.1.1. *Computing the kernel of an isogeny.*

Lemma 4.1. *Let $f \in K\{\tau\}$ be an Ore polynomial with τ-degree n. Let $x_1, \dots, x_\ell$, with $\ell \leqslant d$, be elements of K. One can compute $f(x_1), \dots, f(x_\ell)$ for a cost of $O(n)$ arithmetic operations in K and $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ arithmetic and Frobenius operations in K. In particular, one can compute the matrix $M_f \in {\mathbb{F}_q}^{d\times d}$ of f relative to $\mathcal{E}$ for the same cost.*

Proof. First, we reduce f modulo $\tau^d - 1$, which can be done by shifting coefficients, for a cost of $O(n)$ arithmetic operations in K. Then the result follows from Lemma 3.2. □

Let u be a morphism of Drinfeld $\mathbb{F}_q[T]$-modules over K with domain ϕ. Algorithm 4 computes the invariant factors and a Frobenius decomposition of $\phi(\ker_K(u))$. Proposition 4.2 gives the asymptotic complexity of this algorithm.

Proposition 4.2. *Let u be a morphism of Drinfeld $\mathbb{F}_q[T]$-modules whose domain is ϕ. Algorithm 4 computes the invariant factors and a Frobenius decomposition of $\phi(\ker(u))$ for a cost of $O(\mathbf{MM}(d)(\log d)(\log\log d))$ arithmetic operations in $\mathbb{F}_q$, $O(r + \deg_\tau(u))$ arithmetic operations in K, and $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ Frobenius and arithmetic operations in K.*

Proof. Step 1, computing M_u, costs $O(\deg(u))$ arithmetic operations in K and $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ arithmetic and Frobenius operations in K, as a direct application of Lemma 4.1. Computing an $\mathbb{F}_q$-basis of $\ker(M_u)$ (Step 2) costs $O(\mathbf{MM}(d))$ operations in $\mathbb{F}_q$.

The next costly step is the computation of the matrix X of ϕ_T as an $\mathbb{F}_q$-linear endomorphism of $\ker_K(u)$ (Step 8). The columns of this matrix are the coordinates of $\phi_T(b_1), \dots, \phi_T(b_{d'})$ relative to the basis $\mathcal{B} = (b_1, \dots, b_{d'})$ computed in Step 2. Still by Lemma 4.1, the computation of $\phi_T(b_1), \dots, \phi_T(b_{d'})$ costs $O(r)$ arithmetic operations in K and $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ arithmetic and Frobenius operations in K. Now X is uniquely determined by the equation $NX = Y$, which can be solved using standard linear algebra techniques, such as LUP decomposition of N followed by forward and backward substitution, for a cost of $O(\mathbf{MM}(d))$ arithmetic operations in $\mathbb{F}_q$ [IMH82].

The next costly step is the computation of the Frobenius normal form F_X of X, together with the unimodular transformation matrix S (Step 9). By Lemma 2.12, this can be done for a cost of $O(\mathbf{MM}(d')(\log d')(\log\log d'))$ operations in $\mathbb{F}_q$, which is contained in $O(\mathbf{MM}(d)(\log d)(\log\log d))$ operations in $\mathbb{F}_q$.

Finally, we need to retrieve the Frobenius decomposition of $\phi(\ker_K(u))$. We cannot extract the $\mathcal{E}$-coordinate vectors of the elements of the Frobenius decomposition from $S^{-1} \in {\mathbb{F}_q}^{d'\times d'}$ (as in Proposition 2.11), as the columns of S^{-1} are the $\mathcal{B}$-coordinate vectors of these elements. The desired $\mathcal{E}$-coordinate vectors are exactly the columns of S' (Step 12). The cost of computing S' is also $O(\mathbf{MM}(d))$ operations in $\mathbb{F}_q$. □

Algorithm 4: MorphismKernelInvariants

Input: A morphism u from a Drinfeld $\mathbb{F}_q[T]$-module ϕ over the finite field K equipped with the $\mathbb{F}_q$-basis $\mathcal{E}$

Output: The invariant factors and a Frobenius decomposition of $\phi(\ker_K(u))$

`// Compute` $\ker_K(u)$

Compute the matrix $M_u \in {\mathbb{F}_q}^{d\times d}$ of u relative to $\mathcal{E}$.

Compute an $\mathbb{F}_q$-basis $\mathcal{B} = (b_1, \dots, b_{d'})$ of $\ker_K M_u$.

if $\ker_K M_u = \{0\}$ **then**

  return $(1 \in \mathbb{F}_q[T])$ *and* $(0 \in K)$

Put the vectors of $\mathcal{B}$ in a matrix $N_u = [b_1, \cdots, b_{d'}] \in {\mathbb{F}_q}^{d\times d'}$.

`// Compute the action of` ϕ_T `on` $\ker_K(u)$

Compute the coordinates y_i of $\phi_T(b_i)$, $1 \leqslant i \leqslant d'$, relative to $\mathcal{E}$.

Put the vectors in a matrix $Y = [y_1, \dots, y_{d'}] \in {\mathbb{F}_q}^{d\times d'}$.

Compute the unique matrix X such that $N_u X = Y$.

`// Compute the Frobenius normal form of the action of` ϕ_T

Compute the Frobenius normal form $\mathrm{F}_X \in {\mathbb{F}_q}^{d'\times d'}$ of X, together with a change-of-basis matrix $S \in {\mathbb{F}_q}^{d'\times d'}$ such that $S\,\mathrm{F}_X\,S^{-1} = X$.

`// Convert the vectors to the basis` $\mathcal{E}$

Compute $S^{-1} \in {\mathbb{F}_q}^{d'\times d'}$.

Write F_X as a block diagonal matrix with entries $C_{d_1}, \dots, C_{d_\ell}$.

Compute the matrix $S' \coloneqq NS^{-1} \in {\mathbb{F}_q}^{d\times d'}$.

`// Record and return the results`

for i *from* 1 *to* ℓ **do**

  Write d_i for the monic polynomials in $\mathbb{F}_q[T]$ corresponding to C_{d_i}.

  Write x_i for the element of K whose F-coordinates (relative to $\mathcal{E}$) is the column with index $\deg(d_1) + \cdots + \deg(d_{i-1})$ in S'.

return $(d_1, \dots, d_\ell)$ *and* $(x_1, \dots, x_\ell)$.

Algorithm 5: ModuleOfPointsInvariants

Input: A Drinfeld $\mathbb{F}_q[T]$-module ϕ over the finite field K

Output: The invariant factors and a Frobenius decomposition of $\phi(K)$

return *MorphismKernelInvariants*(0_ϕ) where 0_ϕ is the zero morphism on ϕ.

Corollary 4.3. *Algorithm 5 computes the invariant factors and a Frobenius decomposition of* $\phi(\ker_K(u))$ *for a cost of* $O(\mathbf{MM}(d)(\log d)(\log\log d))$ *arithmetic operations in* $\mathbb{F}_q$, $O(r)$ *arithmetic operations in* K, *and* $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ *Frobenius and arithmetic operations in* K.

Remark 4.4. Implementing Algorithm 5 (ModuleOfPointsInvariants) without using Algorithm 4 (MorphismKernelInvariants) would only be marginally more efficient. The reason is that in the present description, many intermediate quantities simplify because the basis of $\ker_K(u)$ is $\mathcal{B}$ itself; we are then computing the Frobenius normal form of the matrix of ϕ_T relative to $\mathcal{E}$ itself. The matrix of ϕ_T

can be pre-computed and used for other operations, for a cost of $O(r)$ arithmetic operations in K and $O(\mathbf{SM}^{\geqslant 1}(d)(\log d)^2)$ arithmetic and Frobenius operations in K (Lemma 4.1). The remaining operations cost $O(\mathbf{MM}(d)(\log d)(\log\log d))$ arithmetic operations in K (Lemma 2.12).

4.1.2. *Torsion submodules.* For this special case, where $u = \phi_a$ for some $a \in \mathbb{F}_q[T]$, we can directly reuse the invariant factors $(d_i)_{1\leqslant i\leqslant \ell}$ and the Frobenius decomposition $(x_i)_{1\leqslant i\leqslant \ell}$ of $\phi(K)$, provided they have been computed (see Algorithm 5 and Corollary 4.3). Assuming that x_i corresponds to d_i for each i, and following the notation given prior to Lemma 3.4, we write $\gamma_i \coloneqq \gcd(a, d_i)$ and $\rho_i \coloneqq d_i/g_i$, for $1 \leqslant i \leqslant \ell$. Then we have an isomorphism of $\mathbb{F}_q[T]$-modules

$$\phi(K)[a] \simeq \prod_{i=1}^{\ell} \mathbb{F}_q[T]/(\gamma_i).$$

Furthermore, the elements $x_{a,i} = \phi_{\rho_i}(x_i)$ have order γ_i. Considering only the set I_a of indices i such that $\gamma_i \neq 1$, we have

$$\phi(K)[a] = \bigoplus_{i\in I_a} \mathbb{F}_q[T]x_{a,i}.$$

In other words, $(\gamma_i)_{i\in I_a}$ and $(x_{a,i})_{i\in I_a}$ form the invariants factors and a Frobenius decomposition of $\phi(K)[a]$, respectively. This leads to Algorithm 6.

Algorithm 6: TorsionFromModuleOfPoints

Input: A polynomial $a \in A$, the invariant factors $(d_1, \dots, d_\ell)$ and a Frobenius decomposition $(x_1, \dots, x_\ell)$ of $\phi(K)$, where ϕ is a Drinfeld $\mathbb{F}_q[T]$-module over the finite field K

Output: The invariant factors and a Frobenius decomposition of $\phi(K)[a]$

1 Compute $\gamma_i \coloneqq \gcd(a, d_i)$ and $\rho_i \coloneqq d_i/\gamma_i$, for all $1 \leqslant i \leqslant \ell$, using Algorithm 3.
2 Let I_a be the set of indices $1 \leqslant i \leqslant \ell$ such that $\gamma_i \neq 1$.
3 Compute $x_{a,i} \coloneqq \phi_{\rho_i}(x_i)$ for all $i \in I_a$.
4 **return** $(\gamma_i)_{i\in I_a}$ *and* $(x_{a,i})_{i\in I_a}$

For brevity, write $d_a \coloneqq \dim_{\mathbb{F}_q} \phi(K)[a]$, and $\ell_a \coloneqq \#I_a$ for the number of invariant factors of $\phi(K)[a]$. Recall Definition 3.5 for the definition of $\mathbf{EV}$.

Proposition 4.5. *Given the invariant factors and a Frobenius decomposition of $\phi(K)$, Algorithm 6 computes the invariant factors and a Frobenius decomposition of $\phi(K)[a]$ for a cost of $O(\mathbf{PM}(\deg(a)) + \mathbf{PM}(d)\log d + \mathbf{EV}_d(\ell_a, d_a))$ operations in $\mathbb{F}_q$.*

Proof. The call to Algorithm 3 requires $O(\mathbf{PM}(\deg(a))+\mathbf{PG}(d))$ operations in $\mathbb{F}_q$ by Lemma 3.4. The last step requires $O(\mathbf{EV}_d(\ell_a, d_a))$ operations in K. Asymptotically, $\mathbf{PG}(d) \in O(d^2)$ and $\mathbf{EV}_d(\ell_a, d_a)$ is at most $O(d^2)$, so the total cost of Algorithm 6 is $O(\mathbf{PM}(\deg(a)) + \mathbf{EV}_d(\ell_a, d_a))$. □

4.1.3. *Explicit complexity results.* We determine the cost of some of our algorithms more explicitly, depending on different primitives for Ore polynomials. We start with Algorithm 5. This algorithm essentially has two steps (Remark 4.4): computing the matrix M_{ϕ_T} of ϕ_T relative to the $\mathbb{F}_q$-basis $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_d)$ of K and computing

its Frobenius decomposition. We first ascertain the cost of computing M_{ϕ_T}. We recall that as K is a finite field with $\mathbb{F}_q$-dimension d, an arithmetic operation in K can be performed for a cost of $\widetilde{O}(d \log q)$ bit operations [GG13, Chapter 2].

(1) The most naïve method consists of sequentially evaluating $\phi_T(\varepsilon_1), \dots, \phi_T(\varepsilon_d)$. Counting r Frobenius applications for an evaluation and $O(\log q)$ operations in K for a Frobenius application using square & multiply, we obtain $\widetilde{O}(rd^2(\log q)^2)$ bit operations to compute the matrix of ϕ_T.
(2) If we use multipoint evaluation (Algorithm 2 and Lemma 4.1) with $\mathbf{SM}_{\mathrm{F}}(d) = d^2$ and square & multiply, the cost of computing M_{ϕ_T} becomes $\widetilde{O}(r + d^3(\log q)^2)$. In other words, fast multipoint evaluation is only advantageous with fast primitives for Ore polynomial multiplication.
(3) If $\widetilde{O}(\mathbf{SM}^{\geqslant 1}(d))$ operations in K account for $\widetilde{O}(d^{\frac{9-\omega}{5-\omega}} \log q)$ bit operations as in [CL17b][1], then we can compute M_{ϕ_T} in $\widetilde{O}(rd \log q + d^{\frac{9-\omega}{5-\omega}} (\log q)^2)$ bit operations. If r is in $O(d)$, this is better than the naïve method of item 1; if on the other hand d grows while r remains constant, then the naïve method yields a better complexity in d.
(4) If we use the Euclidean algorithm [LS24, Algorithm 3] in Algorithms 1 and 2, then computing M_{ϕ_T} is asymptotically no faster than the naïve method, as the first llcm computation alone requires a quadratic number of Frobenius and arithmetic operations.
(5) Using the Kedlaya-Umans algorithm [KU11] for Frobenius applications instead of square & multiply in the naïve method (item 1), Lemma 4.1 gives a cost $\widetilde{O}(d(\log q)^2) + (rd^2 \log q)^{1+o(1)}$ bit operations.

This shows that the three most advantageous methods for computing M_{ϕ_T} are the naïve method (item 1), multipoint evaluation using the fast multiplication algorithm (and the complexity model) of Caruso and Le Borgne (item 3), and multipoint evaluation using the Kedlaya-Umans algorithm for applications of the Frobenius endomorphism (item 5).

(1) If we use the naïve method for the computation of M_{ϕ_T} (Item 1), then the cost of Algorithm 5 is $\widetilde{O}(rd^2(\log q)^2 + \mathbf{MM}(d) \log q)$ bit operations.
(3) If we use multipoint evaluation (Algorithm 2) and the model and primitives of Caruso and Le Borgne for computing M_{ϕ_T}, the cost of Algorithm 5 is $\widetilde{O}(rd \log q + d^{\frac{9-\omega}{5-\omega}} (\log q)^2)$ bit operations. With respect to the variable d, the cost of computing M_{ϕ_T} is therefore dominant.
(5) If we use multipoint evaluation (Algorithm 2) and the model and primitives of Kedlaya and Umans (item 5) in Lemma 4.1, the total cost of Algorithm 5 is $\widetilde{O}(d(\log q)^2 + \mathbf{MM}(d) \log q) + (rd^2 \log q)^{1+o(1)}$ bit operations.

More generally, we can always assume that $O(\mathbf{SM}_{\mathrm{A}}{}^{\geqslant 1}(d)) \in O(\mathbf{MM}(d))$ (the naïve method for multiplying two Ore polynomials yields $\mathbf{SM}_{\mathrm{A}}(d) = d^2$). If we fix $\log q$ and use $\widetilde{O}$ for clarity, the total cost of Algorithm 5, as per Corollary 4.3, is $\widetilde{O}(dr + \mathbf{MM}(d))$ arithmetic operations in $\mathbb{F}_q$ and $\widetilde{O}(\mathbf{SM}_{\mathrm{F}}(d))$ applications of the Frobenius endomorphism. In a model where the cost of applying Frobenius is not higher than the cost of other arithmetic operations in K, this is a total of $\widetilde{O}(dr + \mathbf{MM}(d))$ arithmetic operations in $\mathbb{F}_q$, and relative to d, the computation of the Frobenius normal form dominates.

[1]The value of $\mathbf{SM}^{\geqslant 1}$ in the original article contains a typo that was corrected in [CL26, § 1.2.3].

Next, we investigate the cost of computing the invariant factors and a Frobenius decomposition of $\phi(K)[a]$, $a \in \mathbb{F}_q[T]$, knowing the invariant factors and a Frobenius decomposition of $\phi(K)$. As before, write $d_a \coloneqq \dim_{\mathbb{F}_q} \phi(K)[a]$ and $\ell_a \coloneqq \#I_a$ for the number of invariant factors of $\phi(K)[a]$. As discussed in § 3.2, the cost of Algorithm 6 (see Proposition 4.5) is either $O(\mathbf{PM}(\deg(a)) + \mathbf{PM}(d)\log d + d_a d^2)$ operations in $\mathbb{F}_q$ using a direct method, or $O(\mathbf{PM}(\deg(a)) + \mathbf{PM}(d)\log d + \ell_a \mathbf{MM}(d))$ operations in $\mathbb{F}_q$ using the primitive of Storjohann for polynomial-matrix evaluation [Sto00]. The first method is preferred when d_a is small with respect to d; otherwise, the second method should be used. In particular, we see that if the rank is large relative to $\deg(a)$ or d_a is small relative to d, then it is cheaper to compute the invariant factors and Frobenius decomposition of $\phi(K)[a]$ by reusing those of $\phi(K)$.

4.2. **The general case.** For the general case, we apply the results of § 2.5.2. Recall from § 2.3 that C is a smooth projective geometrically connected curve over $\mathbb{F}_q$ with a geometric closed point ∞. Recall that A is defined as the ring of functions of $\mathbb{F}_q(C)$ that are regular outside ∞. Then A is finitely generated as an $\mathbb{F}_q$-algebra. The curve C and the function ring A can be represented in many ways. Here, we assume that the curve equation is given as $P(T_1, \dots, T_n) = 0$ for some $P \in \mathbb{F}_q[T_1, \dots, T_n]$, and A is the quotient $\mathbb{F}_q[T_1, \dots, T_n]/(P)$. We write $g_i \coloneqq T_i \mod P$, $1 \leqslant i \leqslant n$. Let ϕ be a Drinfeld A-module over (K, γ) and u a morphism whose domain is ϕ. Our goal is to compute the invariant factors of $\phi(\ker_K(u))$.

We follow § 2.5.2, with $V = \ker_K(u)$. For each $1 \leqslant i \leqslant n$, let M_i be the matrix (with respect to the fixed F-basis $\mathcal{E}$ of V) of the $\mathbb{F}_q$-linear endomorphism ϕ_{g_i} acting on $\ker_K(u)$. Let $d_u \coloneqq \dim_{\mathbb{F}_q}(\ker_K(u))$. Then we have an isomorphism of A-modules

$$\phi(\ker_K(u)) \simeq A^{d_u}/\mathrm{Im}(M_\chi),$$

where M_χ is the vertical block matrix of blocks $(g_i \operatorname{Id} - M_i)_{1 \leqslant i \leqslant n}$ as in Equation (2.4). Computing the invariant factors of $\phi(\ker_K(u))$ then amounts to a simple sequence of steps:

(1) Computing the matrices $M_1, \dots, M_n$, and the matrix M_χ;
(2) Computing the Fitting ideals $\mathrm{Fitt}_1(M_\chi), \dots, \mathrm{Fitt}_d(M_\chi)$
(3) Computing the ideals $\mathfrak{d}_i \coloneqq \mathrm{Fitt}_{d-i}(M_\chi)/\mathrm{Fitt}_{d+1-i}(M_\chi)$, for $1 \leqslant i \leqslant d$.

By Lemma 2.18, the ideals $(\mathfrak{d}_i)_{1 \leqslant i \leqslant d}$ are the invariant factors of $\phi(\ker_K(u))$. Generators of these ideals can be computed using Gröbner bases techniques for ideal quotients (see [CLO25, Chapter 4, Section 4]).

5. Deciding when the torsion is rational

We conclude with a problem closely related to the computation of rational torsion submodules, namely determining all $a \in \mathbb{F}_q[T]$ for which $\phi[a]$ is rational, where ϕ is a Drinfeld $\mathbb{F}_q[T]$-module over a finite field $K/\mathbb{F}_q$ with $\mathbb{F}_q$-dimension d. While all the points of $\phi(K)$ are torsion points (they are $\chi_\phi(1)$-torsion elements, where χ_ϕ is the *characteristic polynomial of the Frobenius endomorphism of* ϕ), the existence of a non-constant $a \in \mathbb{F}_q[T]$ for which the a-torsion is rational is not guaranteed.

5.1. **Anderson motives.** First, we briefly introduce *Anderson* $\mathbb{F}_q[T]$*-motives*; see [And86; Hei04; Pap23; Arm+25] for a general introduction.

Definition 5.1. The *Anderson* $\mathbb{F}_q[T]$*-motive* attached to a Drinfeld module ϕ over K, denoted $\mathbb{M}(\phi)$, is the $K[T]$-module defined by

$$\begin{array}{rcl} K[T] \times K\{\tau\} & \to & K\{\tau\} \\ \left(\sum_{i=0}^{n} \lambda_i T^i, f(\tau)\right) & \mapsto & \sum_{i=0}^{n} \lambda_i f(\tau) \phi_{T^i}. \end{array}$$

Importantly, $\mathbb{M}(\phi)$ is a free $K[T]$-module with *canonical basis* $(1, \tau, \dots, \tau^{r-1})$. Moreover, $\mathbb{M}$ defines a contravariant functor, as for any morphism of Drinfeld modules $u : \phi \to \psi$, there exists a morphism of $K[T]$-modules $\mathbb{M}(u) : \mathbb{M}(\psi) \to \mathbb{M}(\phi)$, defined by $\mathbb{M}(u)(f) = fu$. The functor $\mathbb{M}$ is fully faithful, and we can therefore represent Ore polynomials by tuples of r coefficients in $K[T]$; this is easily computed, as explained in [CL26, Algorithm 1].

5.2. **Deciding when the torsion is rational.** Current methods for deciding rationality only handle a fixed $a \in \mathbb{F}_q[T]$.

- For any s, write K_s for the degree s extension of K. To determine the minimal field of definition (*i.e.* the splitting field) of the a-torsion $\phi[a]$, one can use Anderson motives (§ 5.1). Indeed, $\phi(K_s)[a] = \phi[a]$ if and only if the Frobenius endomorphism of K_s is the identity on $\phi[a]$. Using the correspondence between Drinfeld modules and Anderson motives via the functor $\mathbb{M}$, this is equivalent to $\mathbb{M}(\tau^{sd})$ being the identity on the quotient module $\mathbb{M}(\phi)/\mathbb{M}(\phi)\phi_a$ (see [Pap23, § 3.6] for details). We can effectively write $\mathbb{M}(\tau^d)$ as a matrix $M \in (\mathbb{F}_q[T]/(a))^{r \times r}$. The minimal s such that $\phi(K_s)[a] = \phi[a]$ is then the minimal s such that M^s is the identity. However, computing s seems computationally hard. For example, when $\mathbb{F}_q[T]/(a)$ is a field, finding the idempotency order of a matrix requires finding the multiplicative order of elements in $\mathbb{F}_q[T]/(a)$. As of now, the best methods to effect this require factoring an integer of the form $q^n - 1$ for some n.
- For a fixed value of a, it is easy to know if the a-torsion is rational. If h is the τ-valuation of ϕ_a, then the a-torsion is rational if and only if ϕ_a right-divides $\tau^h(\tau^d - 1)$.

These two methods determine rationality of the torsion for individual $a \in \mathbb{F}_q[T]$, but are unable to find all $a \in \mathbb{F}_q[T]$ with rational a-torsion. We introduce a method that accomplishes this, beginning with the separable case. Consider the Anderson motive $\mathbb{M}(\phi)$ with its canonical $K[T]$-basis $(1, \dots, \tau^{r-1})$. As K has finite $\mathbb{F}_q$-dimension d, the Anderson motive $\mathbb{M}(\phi)$ is free over $\mathbb{F}_q[T]$ with basis $(\varepsilon_j \tau^i)$, $1 \leqslant j \leqslant d$ and $0 \leqslant i \leqslant r-1$, where $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_d)$ is an ordered $\mathbb{F}_q$-basis of K.

Let $p \in \mathbb{F}_q[T]$ be the monic generator of the kernel of the $\mathbb{F}_q$-algebra morphism $\gamma : \mathbb{F}_q[T] \to K$. Let a be coprime to p, so ϕ_a is separable. Then the a-torsion is K-rational if and only if ϕ_a right-divides $\tau^d - 1$. As the $\mathbb{F}_q[T]$-coordinates of ϕ_a are simply $(a, 0, \dots)$, we have the following proposition.

Proposition 5.2. *Let g_ϕ be the gcd of the $\mathbb{F}_q[T]$-coordinates of $\tau^d - 1 \in \mathbb{M}(\phi)$. If $a \in \mathbb{F}_q[T]$ is coprime to p, then the a-torsion of ϕ is rational if and only if a divides g_ϕ.*

Recall that the $\mathbb{F}_q[T]$-coordinates of any element in $\mathbb{M}(\phi)$ can be computed with [CL26, Algorithm 1], which gives a direct method to compute g_ϕ, knowing only ϕ_T. One can also express g_ϕ as the lcm of all $a \in \mathbb{F}_q[T]$ coprime to p for which the a-torsion is rational. This invariant can also be recovered using the invariant factors

of $\phi(K)$, and the fact that $\phi[a] \simeq (\mathbb{F}_q[T]/(a))^r$, but our direct computation is more efficient.

Note that Proposition 5.2 does not hold if a is a multiple of p. Indeed, ϕ_a is inseparable in this case, and so would be all its multiples. However, $\tau^d - 1$ is separable. To determine the maximum s such that the p^s-torsion is rational, one can however proceed as follows.

(1) Compute the product n_ϕ of the invariant factors of $\phi(K)$. One can obtain n_ϕ directly from the characteristic polynomial of the Frobenius endomorphism χ_ϕ, as $n_\phi = \chi_\phi(1)$ (see [CL26, Appendix A] for a review of methods for computing χ_ϕ). Indeed, any $a \in \mathbb{F}_q[T]$ such that $\phi[a]$ is rational is a divisor of n_ϕ.
(2) Compute the p-adic valuation v of n_ϕ. This can be done for a cost of $O(\mathbf{PM}(\deg(n_\phi)) \log(\deg(n_\phi)/\deg(p)))$ operations in $\mathbb{F}_q$ using the *divide and conquer* method of *radix conversion* [GG13, § 9.2].
(3) The desired integer s is such that $s \leqslant v$. One can find it by trying all possibilities or by implementing a more sophisticated search algorithm.

Remark 5.3. The cost of computing g_ϕ only depends on the size of the input (the Ore polynomial ϕ_T) and not on the size of the output. This cost is polynomial. By contrast, we do not know any efficient method to compute the analogue of g_ϕ for an elliptic curve over a finite field.

Conclusion

All the algorithms presented herein are deterministic and work well in full generality. Nevertheless, there may be room for improvement.

(1) In light of Algorithm 6, one might ask whether it could be profitable to compute the invariant factors and a Frobenius decomposition of $\phi(\ker_K(u))$, for some Drinfeld module ϕ and isogeny u, by reusing the invariant factors and Frobenius decomposition of $\phi(K)$. One would first need to find a matrix representation of u as an $\mathbb{F}_q[T]$-morphism (*e.g.* a presentation matrix), and not simply as an $\mathbb{F}_q$-morphism. The cost of producing such a representation is at least $O(\mathbf{MM}(d))$ operations in d, which is not better (in d) than the estimate of Proposition 4.5. Ultimately, representing isogenies as Ore polynomials is straightforward, but not necessarily the best representation for computations.
(2) In Algorithm 6, we compute quantities of the form $P(M)x$, where P is a polynomial of $\mathbb{F}_q[T]$ of degree $n \leqslant d$, M is a d-by-d matrix with entries in $\mathbb{F}_q$, and x is a vector. The assumption $n \leqslant d$ is critical for obtaining a better complexity (using Storjohann's method [Sto00, Chapter 9]), as without it the best method to compute $P(M)$ is that of Paterson and Stockmeyer [PS73], for a cost of $O(\sqrt{n}\mathbf{MM}(d) + nd^2)$ operations in $\mathbb{F}_q$. The Paterson-Stockmeyer method is optimal in the general case. This begs the question of adapting the Paterson-Stockmeyer method (which is itself a direct adaptation of the Horner method [Hor19]) for computing $P(M)x$ without computing $P(M)$. It seems that a fundamental obstruction is the absence of a relevant vector product. Current workarounds involve a substantial number of precomputations, resulting in a complexity that is identical to or worse than that of Storjohann's method. This raises

the question whether there exists a method for computing $P(M)x$ (in full generality) whose cost is $O(nd + \sqrt{n}d^2)$ (optimal in d) operations in $\mathbb{F}_q$.

(3) Algorithm 6 also requires computations related to divisibility chains. Specifically, we need to compute elements of the form $\phi_{\gamma_i}(x_i)$, where the γ_i form a divisibility chain. We have not found a way to leverage this divisibility chain in a way that significantly improves the worst-case complexity, as for any two i, j with $i < j$, knowing $\phi_{\gamma_i}(x_i)$, γ_j/γ_i and x_j does not *a priori* help to find $\phi_{\gamma_j}(x_j)$. Further investigation into these problems, irrespective of their application to Drinfeld modules, is warranted. Probabilistic methods could help in some cases, *e.g.* when the invariant factors are smooth.

(4) The method of § 4.2 also leaves open questions. The assumption that A is given as a quotient of $\mathbb{F}_q[T_1, \ldots, T_n]$ enables the use of Gröbner bases techniques for the computation of the invariant factors of $\ker_K(u)$. Exploring and implementing this direction may be worthwhile.

Finally, we reflect on the use of finite fields. This is the setting for most practical applications, such as computer algebra [DNS21; BMZ25] and coding theory [BDM24; MP26]). If ϕ is a Drinfeld module over a finite field K, then $\phi(K)$ is finitely generated, both when $A = \mathbb{F}_q[T]$ and for general function fields. We can leverage the structure of A as a finitely generated $\mathbb{F}_q$-algebra and encode the action of the generators using matrices. This is not possible for elliptic curves, where the role of A is played by $\mathbb{Z}$. If the base field K is a function field, then the module $\phi(K)$ is not finitely generated and its non-torsion part is free with infinite rank $\aleph_0$ [Poo95]. The torsion part is known to be finite [Pap23, Theorem 6.4.3]. Anderson motives would likely be an important tool in the computation of these torsion points. Even in the infinite case (but with a suitable change of variables), they facilitate a representation of an isogeny by a finite square polynomial matrix whose cokernel is isomorphic to the kernel of the isogeny. The invariant factors of the cokernel can be recovered by computing the Smith normal form of the matrix. For finite fields, this strategy is often less efficient compared to our approach due to the costly manipulation of polynomial matrices. However, in the general case, Anderson motives simplify the problem by reducing it to classical computer algebra.

Acknowledgments

We warmly thank Delphine Boucher and Xavier Caruso for insightful discussions. We are grateful to the anonymous referees for their very thorough proof-reading and helpful feedback which lead to substantial improvements to our paper. Antoine Leudière is supported in part by the Pacific Institute for the Mathematical Sciences. Renate Scheidler is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, funding RGPIN-2025-03992.

References

[And86] Greg W. Anderson. "t-Motives." In: *Duke Mathematical Journal* 53.2 (1986), pp. 457–502. DOI: 10.1215/S0012-7094-86-05328-7.

[Arm+25] Cécile Armana, Elena Berardini, Xavier Caruso, Antoine Leudière, Jade Nardi, and Fabien Pazuki. "A computational approach to Drinfeld modules." 2025. URL: https://hal.science/hal-05423892.

[Ayo+23] David Ayotte, Xavier Caruso, Antoine Leudière, and Joseph Musleh. "Drinfeld modules in SageMath." In: *ACM Commun. Comput. Algebra* 57.2 (2023), pp. 65–71. DOI: 10.1145/3614408.3614417.

[BDM24] Luca Bastioni, Mohamed O. Darwish, and Giacomo Micheli. *Optimal Rank-Metric Codes with Rank-Locality from Drinfeld Modules*. 2024. DOI: 10.48550/arXiv.2407.06081.

[BMZ25] Luca Bastioni, Giacomo Micheli, and Shujun Zhao. *On the Characteristic Polynomial of Linearized Polynomials*. 2025. DOI: 10.48550/arXiv.2506.16937.

[BN25] Delphine Boucher and Kayodé Epiphane Nouetowa. "A Decoding Algorithm for Skew Cyclic Generalized Skew Reed-Solomon Codes." In: *2025 IEEE International Symposium on Information Theory (ISIT)*. 2025, pp. 1–6. DOI: 10.1109/ISIT63088.2025.11195485.

[Bou03] Nicolas Bourbaki. *Algebra II. Chapters 4–7*. English. Elements of Mathematics (Berlin). Springer-Verlag, Berlin, 2003. DOI: 10.1007/978-3-642-61698-3.

[Car18] Perlas Caranay. "Computing Isogeny Volcanoes of Rank Two Drinfeld Modules." PhD thesis. University of Calgary, 2018.

[CGS20] Perlas Caranay, Matthew Greenberg, and Renate Scheidler. "Computing modular polynomials and isogenies of rank two Drinfeld modules over finite fields." In: *Contemporary Mathematics* 754 (2020). Ed. by Susanne Brenner, Igor Shparlinski, Chi-Wang Shu, and Daniel Szyld, pp. 283–313. DOI: 10.1090/conm/754/15148.

[CL17a] Xavier Caruso and Jérémy Le Borgne. "A new faster algorithm for factoring skew polynomials over finite fields." In: *Journal of Symbolic Computation* 79 (2017), pp. 411–443. DOI: 10.1016/j.jsc.2016.02.016.

[CL17b] Xavier Caruso and Jérémy Le Borgne. "Fast Multiplication for Skew Polynomials." In: *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*. ISSAC '17. Association for Computing Machinery, 2017, pp. 77–84. DOI: 10.1145/3087604.3087617.

[CL26] Xavier Caruso and Antoine Leudière. "Algorithms for computing norms and characteristic polynomials on general Drinfeld modules." In: *Mathematics of Computation* 95.357 (2026). DOI: 10.1090/mcom/4052.

[CLO25] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms—an introduction to computational algebraic geometry and commutative algebra*. Fifth. Undergraduate Texts in Mathematics. Springer, Cham, 2025, pp. xvi+656. ISBN: 978-3-031-91840-7; 978-3-031-91841-4. DOI: 10.1007/978-3-031-91841-4.

[DNS21] Javad Doliskani, Anand Kumar Narayanan, and Éric Schost. "Drinfeld modules with complex multiplication, Hasse invariants and factoring polynomials over finite fields." In: *Journal of Symbolic Computation*. MICA 2016 105 (2021), pp. 199–213. DOI: 10.1016/j.jsc.2020.06.007.

[Dri77] Vladimir G. Drinfel'd. "Commutative subrings of certain noncommutative rings." In: *Functional Analysis and Its Applications* 11.1 (1977), pp. 9–12. DOI: 10.1007/BF01135527.

[Eis95] David Eisenbud. *Commutative algebra*. Vol. 150. Graduate Texts in Mathematics. With a view toward algebraic geometry. Springer-Verlag, New York, 1995. DOI: 10.1007/978-1-4612-5350-1.

[Fit36] Hans Fitting. "Die Determinantenideale eines Moduls." In: *Jahresbericht der Deutschen Mathematiker-Vereinigung* 46 (1936), pp. 195–228. URL: http://eudml.org/doc/146122.

[GG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. 3rd ed. Cambridge University Press, 2013. DOI: 10.1017/CBO9781139856065.

[Gop83] Valerii Denisovich Goppa. "Algebraico-geometric codes." In: *Mathematics of the USSR-Izvestiya* 21.1 (1983), pp. 75–91. DOI: 10.1070/IM1983v021n01ABEH001641.

[Hei04] Gert-Jan van der Heiden. "Weil Pairing for Drinfeld Modules." In: *Monatshefte für Mathematik* 143.2 (2004), pp. 115–143. DOI: 10.1007/s00605-004-0261-4.

[HHL17] David Harvey, Joris Van Der Hoeven, and Grégoire Lecerf. "Faster Polynomial Multiplication over Finite Fields." In: *J. ACM* 63.6 (Jan. 2017). DOI: 10.1145/3005344.

[Hor19] William George Horner. "A new method of solving numerical equations of all orders, by continuous approximation." In: *Philosophical Transactions of the Royal Society of London* 109 (Dec. 1819), pp. 308–335. DOI: 10.1098/rstl.1819.0023.

[IMH82] Oscar H Ibarra, Shlomo Moran, and Roger Hui. "A generalization of the fast LUP matrix decomposition algorithm and applications." In: *Journal of Algorithms* 3.1 (1982), pp. 45–56. DOI: 10.1016/0196-6774(82)90007-4.

[KU11] Kiran S. Kedlaya and Christopher Umans. "Fast Polynomial Factorization and Modular Composition." In: *SIAM Journal on Computing* 40.6 (2011), pp. 1767–1802. DOI: 10.1137/08073408X.

[Laf01] Laurent Lafforgue. "Chtoucas de Drinfeld et correspondance de Langlands." In: *Inventiones mathematicae* 147 (2001), pp. 1–241. DOI: 10.1007/s002220100174.

[LS24] Antoine Leudière and Pierre-Jean Spaenlehauer. "Computing a group action from the class field theory of imaginary hyperelliptic function fields." In: *J. Symbolic Comput.* 125 (2024), Paper No. 102311, 19. DOI: 10.1016/j.jsc.2024.102311.

[MP26] Giacomo Micheli and Mihran Papikian. *Rank metric codes from Drinfeld modules*. 2026. URL: https://arxiv.org/abs/2601.03653v1.

[MS23] Yossef Musleh and Éric Schost. "Computing the Characteristic Polynomial of Endomorphisms of a finite Drinfeld Module using Crystalline Cohomology." In: *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*. ISSAC '23. Association for Computing Machinery, 2023, pp. 461–469. DOI: 10.1145/3597066.3597080.

[Pap23] Mihran Papikian. *Drinfeld Modules*. Vol. 296. Graduate Texts in Mathematics. Springer International Publishing, 2023. DOI: 10.1007/978-3-031-19707-9.

[Poo95] Bjorn Poonen. "Local height functions and the Mordell-Weil theorem for Drinfeld modules." In: *Compositio Mathematica* 97.3 (1995), pp. 349–368.

[PS73] Michael S. Paterson and Larry J. Stockmeyer. "On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials." In: *SIAM Journal on Computing* (1973). Publisher: Society for Industrial and Applied Mathematics. DOI: 10.1137/0202007.

[PW18] Sven Puchinger and Antonia Wachter-Zeh. "Fast operations on linearized polynomials and their applications in coding theory." In: 89 (2018), pp. 194–215. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2017.11.012.

[Sca01] Thomas Scanlon. "Public Key Cryptosystems Based on Drinfeld Modules Are Insecure." In: *Journal of Cryptology* 14.4 (2001), pp. 225–230. DOI: 10.1007/s00145-001-0004-9.

[Sta26] The Stacks project authors. *The Stacks project*. https://stacks.math.columbia.edu. 2026.

[Sto00] Arne Storjohann. "Algorithms for matrix canonical forms." PhD thesis. ETH Zurich, 2000. URL: https://doi.org/10.3929/ethz-a-004141007.

[Sut11] Andrew Sutherland. "Structure computation and discrete logarithms in finite abelian p-groups." In: *Mathematics of Computation* 80.273 (2011), pp. 477–500. DOI: 10.1090/S0025-5718-10-02356-2.

[The26] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.8)*. https://www.sagemath.org. 2026.

Department of Mathematics and Statistics, University of Calgary, 2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

Email address: {antoine.leudiere, rscheidl}@ucalgary.ca