Characteristic polynomials of p-adic matrices

Xavier Caruso
Université Rennes 1
xavier.caruso@normalesup.org

David Roe
University of Pittsburg
roed@pitt.edu

Tristan Vaccon
Université de Limoges
tristan.vaccon@unilim.fr

ABSTRACT

We analyze the precision of the characteristic polynomial $\chi(A)$ of an $n \times n$ p-adic matrix A using differential precision methods developed previously. When A is integral with precision $O(p^N)$, we give a criterion (checkable in time $O\tilde{~}(n^\omega)$) for $\chi(A)$ to have precision exactly $O(p^N)$. We also give a $O\tilde{~}(n^3)$ algorithm for determining the optimal precision when the criterion is not satisfied, and give examples when the precision is larger than $O(p^N)$.

CCS Concepts

•**Computing methodologies → Algebraic algorithms;**

Keywords

Algorithms, p-adic precision, characteristic polynomial, eigenvalue

1. INTRODUCTION

The characteristic polynomial is a fundamental invariant of a matrix: its roots give the eigenvalues, and the trace and determinant can be extracted from its coefficients. In fact, the best known division-free algorithm for computing determinants over arbitrary rings [10] does so using the characteristic polynomial. Over p-adic fields, computing the characteristic polynomial is a key ingredient in algorithms for counting points of varieties over finite fields (see [7,8,11].

When computing with p-adic matrices, the lack of infinite memory implies that the entries may only be approximated at some finite precision $O(p^N)$. As a consequence, in designing algorithms for such matrices one must analyze not only the running time of the algorithm but also the accuracy of the result.

Let $M \in M_n(\mathbb{Q}_p)$ be known at precision $O(p^N)$. The simplest approach for computing the characteristic polynomial of M is to compute $\det(X - M)$ either using recursive row expansion or various division free algorithms [10,14]. There are two issues with these methods. First, they are slower than alternatives that allow division, requiring $O(n!)$, $O(n^4)$ and $O\tilde{~}(n^{2+\omega/2})$ operations. Second, while the lack of division implies that the result is accurate modulo p^N as long as $M \in M_n(\mathbb{Z}_p)$, they still do not yield the optimal precision.

A faster approach over a field is to compute the Frobenius normal form of M, which is achievable in running time $O\tilde{~}(n^\omega)$ [15]. However, the fact that it uses division frequently leads to catastrophic losses of precision. In many examples, no precision remains at the end of the calculation.

Instead, we separate the computation of the precision of χ_M from the computation of an approximation to χ_M. Given some precision on M, we use [3, Lem. 3.4] to find the best possible precision for χ_M. The analysis of this precision is the subject of much of this paper. With this precision known, the actual calculation of χ_M may proceed by lifting M to a temporarily higher precision and then using a sufficiently stable algorithm (see Remark 5.3).

One benefit of this approach is that we may account for diffuse precision: precision that is not localized on any single coefficient of χ_M. For example, let $0 \leq \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$, consider a diagonal matrix D with diagonal entries $(p^{\alpha_1}, \ldots, p^{\alpha_n})$, let $P, Q \in \mathrm{GL}_n(\mathbb{Z}_p)$ and set $M = PDQ$. The valuation of the coefficient of X^{n-k} in χ_M will be $\sum_{i=1}^k \alpha_i$, and if $\alpha_{n-1} > 0$ and M is known with precision $O(p^N)$ then the constant term of χ_M will be known with precision larger than $O(p^N)$ (see [4, Prop. 3.2]).

As long as none of the eigenvalues of M are congruent to -1 modulo p, then none of the coefficients of the characteristic polynomial of $1 + M$ will have precision larger than $O(p^N)$. But $\chi_{1+M}(X) = \chi_M(X-1)$, so the precision content of these two polynomials should be equivalent. The solution is that the extra precision in χ_{1+M} is diffuse and not visible on any individual coefficient. We formalize this phenomenon using lattices; see Section 2.1 for further explanation, and [2, §3.2.2] for a specific example of the relationship between χ_M and χ_{1+M}.

Previous contributions.

Since the description of Kedlaya's algorithm in [11], the computation of characteristic polynomials over p-adic numbers has become a crucial ingredient in many counting-points algorithms. For example, [7, 8] use p-adic cohomology and the characteristic polynomial of Frobenius to compute zeta functions of hyperelliptic curves.

In most of these papers, the precision analysis usually deals with great details on how to obtain the matrices (*e.g.* of action of Frobenius) that are involved in the point-counting schemes. However, the computation of their characteristic

ISSAC '17, July 25-28, 2017, Kaiserslautern, Germany
© 2017 ACM. ISBN ***-*-****-****-0/17/07...$15.00
DOI: http://dx.doi.org/10.1145/*******.*******

polynomials is often a little bit less thoroughly studied: some refer to fast algorithms (using division), while others apply division-free algorithms.

In [4], the authors have begun the application of the theory of differential precision of [3] to the stable computation of characteristic polynomials. They have obtained a way to express the optimal precision on the characteristic polynomial, but have not given practical algorithms to attain this optimal precision.

The contribution of this paper.

Thanks to the application the framework of differential precision of [3] in [4], we know that the precision of the characteristic polynomial χ_M of a matrix $M \in M_n(\mathbb{Q}_p)$ is determined by the comatrix $\mathrm{Com}(X - M)$. In this article, we provide:

1. Proposition 2.7: a factorization of $\mathrm{Com}(X - M)$ as a product of two rank-1 matrices (when M has a cyclic vector), computable in $O\tilde{}(n^{\omega})$ operations by Theorem 4.1
2. Corollary 2.4: a simple, $O\tilde{}(n^{\omega})$ criterion to decide whether χ_M is defined at precision higher than the precision of M (when $M \in M_n(\mathbb{Z}_p)$).
3. Theorem 3.11: a $O\tilde{}(n^3)$ algorithm with operations in $\mathbb{Z}_p$ to compute the optimal precision on each coefficient of χ_M (when M is given with uniform precision on its entries).
4. Proposition 5.6: a $O\tilde{}(n^{\omega})$ algorithm to compute the optimal precision on each eigenvalue of M

Organization of the article.

In Section 2, we review the differential theory of precision developed in [3] and apply it to the specific case of the characteristic polynomial, giving conditions under which the differential will be surjective (and thus provide a good measure of precision). We also give a condition based on reduction modulo p that determines whether the characteristic polynomial will have a higher precision that the input matrix, and show that the image of the set of integral matrices has the structure of an $\mathcal{O}_K[X]$-module when M is itself integral. Finally, we give a compact description of $\mathrm{Com}(X - M)$, the main ingredient in the differential.

In Section 3, we develop $O\tilde{}(n^3)$ algorithms to approximate the Hessenberg form of M, and through it to find $\mathrm{Com}(X-M)$ and thus find the precision of the characteristic polynomial of M. In Section 4, we give a $O\tilde{}(n^{\omega})$ algorithm to compute the compact description of $\mathrm{Com}(X - M)$.

Finally, we propose in Section 5 algorithms to compute the optimal coefficient-wise precision for the characteristic polynomial. We also give the results of some experiments demonstrating that these methods can lead to dramatic gains in precision over standard interval arithmetic. We close with results describing the precision associated to eigenvalues of a matrix.

Notation

Throughout the paper, K will refer to a complete, discrete valuation field, $\mathrm{val} : K \twoheadrightarrow \mathbb{Z} \cup \{+\infty\}$ to its valuation, $\mathcal{O}_K$ its ring of integers and π a uniformizer. We will write that $f(n) = O\tilde{}(g(n))$ if there exists some $k \in \mathbb{N}$ such that $f(n) = O(g(n) \log(n)^k)$. We will write M for an $n \times n$ matrix over K, and χ the characteristic polynomial map, $\chi_M \in K[X]$ for the characteristic polynomial of M and $d\chi_M$ for the differential of χ at M, as a linear map from $M_n(K)$ to the space of polynomials of degree less than n. We fix an $\omega \in \mathbb{R}$ such that the multiplication of two matrices over a ring is in $O(n^{\omega})$ operations in the ring. Currently, the smallest known ω is less than 2.3728639 thanks to [13]. We will denote by I_n the identity matrix of rank n in $M_n(K)$. When there is no ambiguity, we will drop this I_n for scalar matrices, *e.g.* for $\lambda \in K$ and $M \in M_n(K)$, $\lambda - M$ denotes $\lambda I_n - M$. Finally, we write $\sigma_1(M), \dots, \sigma_n(M)$ for the elementary divisors of M, sorted in increasing order of valuation.

2. THEORETICAL STUDY

2.1 The theory of p-adic precision

We recall some of the definitions and results of [3] as a foundation for our discussion of the precision for the characteristic polynomial of a matrix. We will be concerned with two K-manifolds in what follows: the space $M_n(K)$ of $n \times n$ matrices with entries in K and the space $K_n[X]$ of monic degree n polynomials over K. Given a matrix $M \in M_n(K)$, the most general kind of precision structure we may attach to M is a *lattice* H in the tangent space at M. However, representing an arbitrary lattice requires n^2 basis vectors, each with n^2 entries. We therefore frequently work with certain classes of lattices, either *jagged* lattices where we specify a precision for each matrix entry or *flat* lattices where every entry is known to a fixed precision $O(p^N)$. Similarly, precision for monic polynomials can be specified by giving a lattice in the tangent space at $f(X) \in K_n[X]$, or restricted to jagged or flat precision in the interest of simplicity.

Let $\chi : M_n(K) \to K_n[X]$ be the characteristic polynomial map. Our analysis of the precision behavior of χ rests upon the computation of its derivative $d\chi$, using [3, Lem. 3.4]. For a matrix $M \in M_n(K)$, we identify the tangent space V at M with $M_n(K)$ itself, and the tangent space W at χ_M with the space $K_{<n}[X]$ of polynomials of degree less than n. Let $\mathrm{Com}(M)$ denote the comatrix of M (when $M \in \mathrm{GL}_n(K)$, we have $\mathrm{Com}(M) = \det(M) M^{-1}$) and $d\chi_M$ the differential at M. Recall [3, Appendix B; 4, §3.3] that $d\chi_M$ is given by

$$d\chi_M : dM \mapsto \mathrm{Tr}(\mathrm{Com}(X - M) \cdot dM). \qquad (1)$$

Proposition 2.1. *For $M \in M_n(K)$, the following conditions are equivalent:*

(i) the differential $d\chi_M$ is surjective

*(ii) the matrix M has a cyclic vector (*i.e. *M is similar to a companion matrix)*

(iii) the eigenspaces of M over the algebraic closure $\bar{K}$ of K all have dimension 1

(iv) the characteristic polynomial of M is equal to the minimal polynomial of M.

PROOF. The equivalence of (ii), (iii), and (iv) is standard; see [9, §7.1] for example. We now show (ii) ⇒ (i) and (i) ⇒ (iii)

For any $A \in \mathrm{GL}_n(K)$, the image of $d\chi$ at M will be the same as the image of $d\chi$ at AMA^{-1}, so we may assume that M is a companion matrix. For a companion matrix, the bottom row of $\mathrm{Com}(X - M)$ consists of $1, X, X^2, \dots, X^{n-1}$ so $d\chi_M$ is surjective.

Now suppose that M has a repeated eigenvalue λ over $\bar{K}$. After conjugating into Jordan normal form over $\bar{K}$, the entries of $\mathrm{Com}(X-M)$ will also be block diagonal, and divisible within each block by the product of $(X-\mu)^{d_\mu}$, where μ, d_μ ranges over the eigenvalues and dimensions of the other Jordan blocks. Since λ occurs in two Jordan blocks, $X-\lambda$ will divide every entry of $\mathrm{Com}(X-M)$ and $d\chi_M$ will not be surjective. □

We also have an analogue of Proposition 2.1 for integral matrices.

Proposition 2.2. *For $M \in M_n(\mathcal{O}_K)$, the following conditions are equivalent:*

(i) the image of $M_n(\mathcal{O}_K)$ under $d\chi_M$ is $\mathcal{O}_K[X] \cap K_{<n}[X]$.

(ii) the reduction of M modulo π has a cyclic vector.

PROOF. The condition (i) is equivalent to the surjectivity of $d\chi_M$ modulo π. The equivalence with (ii) follows the same argument as Proposition 2.1, but over the residue field of K. □

Write $B_V^-(r)$ (resp. $B_V(r)$) for the open (resp. closed) ball of radius r in V, and let $\sigma_1(M), \dots, \sigma_n(M)$ denote the elementary divisors of M.

Proposition 2.3. *Suppose that $M \in M_n(K)$ satisfies one of the conditions in Proposition 2.1, and let*

$$\alpha = \min\left(\prod_{i=1}^{n-1} |\sigma_i(M)|, 1\right).$$

Then, for all $\rho \in (0,1]$ and all $r \in (0, \alpha^{-1} \cdot \rho^{-1})$, any lattice H such that $B_V^-(\rho r) \subset H \subset B_V(r)$ satisfies:

$$\chi(M+H) = \chi(M) + d\chi_M(H). \tag{2}$$

PROOF. Recall [4, Def. 3.3] that the *precision polygon* of M is the lower convex hull of the Newton polygons of the entries of $\mathrm{Com}(X-M)$. By [4, Prop. 3.4], the endpoints of the precision polygon occur at height 0 and $\sum_{i=1}^{n-1} \mathrm{val}(\sigma_i(M))$. By convexity, $B_W(1) \subset d\chi_M(B_V(\alpha^{-1}))$.

Since the coefficients of χ_M are given by polynomials in the entries of M with integral coefficients, [4, Prop. 2.2] implies the conclusion. □

The relationship between precision and the images of lattices under $d\chi_M$ allows us to apply Proposition 2.2 to determine when the precision of the characteristic polynomial is the minimum possible.

Corollary 2.4. *Suppose that $M \in \mathrm{GL}_n(\mathcal{O}_K)$ is known with precision $O(\pi^m)$. Then the characteristic polynomial of M has precision lattice strictly contained in $O(\pi^m)$ if and only if the reduction of M modulo π does not have a cyclic vector.*

Note that this criterion is checkable using $O\tilde{}(n^\omega)$ operations in the residue field [15].

2.2 Stability under multiplication by X

By definition, the codomain of $d\chi_M$ is $K_{<n}[X]$. However, when M is given, $K_{<n}[X]$ is canonically isomorphic to $K[X]/\chi_M(X)$ as a K-vector space. For our purpose, it will often be convenient to view $d\chi_M$ as an K-linear mapping $M_n(K) \to K[X]/\chi_M(X)$.

Proposition 2.5. *Let A be the subring of $K[X]$ consisting of polynomials P for which $P(M) \in M_n(\mathcal{O}_K)$, and $V = d\chi_M\big(M_n(\mathcal{O}_K)\big)$ as a submodule of $K[X]/\chi_M(X)$. Then V is stable under multiplication by A.*

PROOF. Let $C = \mathrm{Com}(X-M)$ and $P \in A$. By (1), V is given by the $\mathcal{O}_K$-span of the entries of C. Using the fact that the product of matrix with its comatrix is the determinant, $(X-M) \cdot C = \chi_M$ and thus $P(X) \cdot C \equiv P(M) \cdot C \pmod{\chi_M(X)}$. The span of the entries of the left hand side is precisely $P(X) \cdot V$, while the span of the entries of the right hand side is contained within V since $P(M) \in M_n(\mathcal{O}_K)$. □

Corollary 2.6. *If $M \in M_n(\mathcal{O}_K)$, then $d\chi_M\big(M_n(\mathcal{O}_K)\big)$ is stable under multiplication by X and hence is a module over $\mathcal{O}_K[X]$.*

2.3 Compact form of $d\chi_M$

Let $\mathscr{C}$ be the companion matrix associated to χ_M:

$$\mathscr{C} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & \cdots & \cdots & -a_{n-1} \end{pmatrix} \tag{3}$$

with $\chi_M = a_0 + a_1 X + \cdots + a_{n-1}X^{n-1} + X^n$. By Proposition 2.1, there exists a matrix $P \in \mathrm{GL}_n(K)$ such that $M = P\mathscr{C}P^{-1}$. Applying the same result to the transpose of M, we find that there exists another invertible matrix $Q \in \mathrm{GL}_n(K)$ such that $M^t = Q\mathscr{C}Q^{-1}$.

Proposition 2.7. *We keep the previous notations and assumptions. Let V be the row vector $(1, X, \dots, X^{n-1})$. Then*

$$\mathrm{Com}(X-M) = \alpha \cdot PV^t \cdot VQ^t \mod \chi_M \tag{4}$$

for some $\alpha \in K[X]$.

PROOF. Write $C = \mathrm{Com}(X-M)$. From $(X-M) \cdot C \equiv 0 \pmod{\chi_M}$, we deduce $(X-\mathscr{C}) \cdot P^{-1}C \equiv 0 \pmod{\chi_M}$. Therefore each column of $P^{-1}C$ lies in the right kernel of $X-\mathscr{C}$ modulo χ_M. On the other hand, a direct computation shows that every column vector W lying in the right kernel of $X-\mathscr{C}$ modulo χ_M can be written as $W = w \cdot V^t$ for some $w \in K[X]/\chi_M$. We deduce that $C \equiv P \cdot V^t B \pmod{\chi_M}$ for some row vector B. Applying the same reasoning with M^t, we find that B can be written $B = \alpha V Q^t$ for some $\alpha \in K[X]/\chi_M$ and we are done. □

Proposition 2.7 shows that $\mathrm{Com}(X-M)$ can be encoded by the datum of the quadruple (α, P, Q, χ_M) whose total size stays within $O(n^2)$: the polynomials α and χ_M are determined by $2n$ coefficients while we need $2n^2$ entries to write down the matrices P and Q. We shall see moreover in Section 4 that interesting information can be read off of this short form (α, P, Q, χ_M).

Remark 2.8. With the previous notation, if $U \in GL_n(K)$, the quadruple for UMU^{-1} is $(\alpha, UP, (U^t)^{-1}Q, \chi_M)$, which can be computed in $O(n^\omega)$ operations in K. This is faster than computing $U\,\mathrm{Com}(X-M)U^{-1}$, which is, at first sight, in $O(n^4)$ operations in K.

3. DIFFERENTIAL VIA HESSENBERG FORM

In this section, we combine the computation of a Hessenberg form of a matrix and the computation of the inverse through the Smith normal form (SNF) over a complete discrete valuation field (CDVF) to compute $\operatorname{Com}(X-M)$ and $d\chi$. If $M \in M_n(\mathcal{O}_K)$, then only division by invertible elements of $\mathcal{O}_K$ will occur.

3.1 Hessenberg form

We begin with the computation of an approximate Hessenberg form.

Definition 3.1. A *Hessenberg matrix* is a matrix $M \in M_n(K)$ with

$$M_{i,j} = 0 \text{ for } j \leq i-2.$$

Given integers $n_{i,j}$, an *approximate Hessenberg matrix* is a matrix $M \in M_n(K)$ with

$$M_{i,j} = O(\pi^{n_{i,j}}) \text{ for } j \leq i-2.$$

If $M \in M_n(K)$ and $H \in M_n(K)$ is an (approximate) Hessenberg matrix similar to M, we say that H is an *(approximate) Hessenberg form* of M.

It is not hard to prove that every matrix over a field admits a Hessenberg form. We prove here that over K, if a matrix is known at finite (jagged) precision, we can compute an approximate Hessenberg form of it. Moreover, we can provide an exact change of basis matrix. It relies on the following algorithm.

```
Algorithm 1: Approximate Hessenberg form computation
Input: a matrix $M$ in $M_n(K)$.
0. $P := I_n$.   $H := M$.
1. for $j = 1, \dots, n-1$ do
2.   swap the row $j+1$ with a row $i_{min}$ ($i_{min} \geq 2$) s.t. val($H_{i_{min},j}$) is minimal.
3.   for $i = j+2, \dots, n$ do
4.     Eliminate the significant digits of $H_{i,j}$ by pivoting with row $j+1$ using a matrix $T$.
5.       $H := H \times T^{-1}$.   $P := T \times P$.
6. Return $H, P$.
```

Proposition 3.2. *Algorithm 1 computes H and P realizing an approximate Hessenberg form of M. P is exact over finite extensions of $\mathbb{Q}_p$ and $k((X))$, and the computation is in $O(n^3)$ operations in K at precision the maximum precision of a coefficent in M.*

Proof. Let us assume that K is a finite extensions of $\mathbb{Q}_p$ or $k((X))$. Inside the nested **for** loop, if we want to eliminate $\pi^{u_y}\varepsilon_y + O(\pi^{n_y})$ with pivot $\pi^{u_x}\varepsilon_x + O(\pi^{n_x})$, with the ε's being units, the corresponding coefficient of the corresponding shear matrix is the lift(in $\mathbb{Z}$, $\mathbb{F}_q[X]$, $\mathbb{Q}[X]$ or adequate extension) of $\pi^{u_y-u_x}\varepsilon_y\varepsilon_x^{-1} \mod \pi^{u_y-u_x \min(n_x-u_x, n_y-u_y)}$. Exactness follows directly. Over other fields, we can not lift, but the computations are still valid. The rest is clear. □

Remark 3.3. From a Hessenberg form of M, it is well known that one can compute the characteristic polynomial of M in $O(n^3)$ operations in K [5, pp. 55–56]. However, this computation involves division, and its precision behavior is not easy to quantify.

3.2 Computation of the inverse

In this section, we prove that to compute the inverse of a matrix over a CDVF K, the Smith normal form is precision-wise optimal in the flat-precision case. We first recall the differential of matrix inversion.

Lemma 3.4. *Let $u : GL_n(K) \to GL_n(K)$, $M \mapsto M^{-1}$. Then for $M \in GL_n(K)$, $du_M(dM) = M^{-1}dMM^{-1}$. It is always surjective.*

We then have the following result about the loss in precision when computing the inverse.

Proposition 3.5. *Let $\operatorname{cond}(M) = \operatorname{val}(\sigma_n(M))$. If dM is a flat precision of $O(\pi^m)$ on M then M^{-1} can be computed at precision $O(\pi^{m-2\operatorname{cond}(M)})$ by a **SNF** computation and this lower-bound is optimal, at least when m is large.*

Proof. The smallest valuation of a coefficient of M^{-1} is $-\operatorname{cond}(M)$. It is $-2\operatorname{cond}(M)$ for M^{-2} and it is then clear that $m-2\operatorname{cond}(M)$ can be obtained as the valuation of a coefficient of $du_M(dM)$ and the smallest that can be achieved this way for dM in a precision lattice of flat precision. Hence the optimality of the bound given, at least when m is large [3, Lem. 3.4].

Now, the computation of the Smith normal form was described in [16]. From M known at flat precision $O(\pi^m)$, we can obtain an exact Δ, and P and Q known at precision at least $O(\pi^{m-\operatorname{cond}(M)})$, with coefficients in $\mathcal{O}_K$ and determinant in $\mathcal{O}_K^\times$ realizing an Smith normal form of M. There is no loss in precision when computing P^{-1} and Q^{-1}. Since the smallest valuation occurring in Δ^{-1} is $-\operatorname{cond}(M)$, we see that $M^{-1} = Q^{-1}\Delta^{-1}P^{-1}$ is known at precision at least $O(\pi^{m-2\operatorname{cond}(M)})$, which concludes the proof. □

3.3 The comatrix of $X-H$

In this section, we compute $\operatorname{Com}(X-H)$ for a Hessenberg matrix H using the Smith normal form computation of the previous section. The entries of $\operatorname{Com}(X-H)$ lie in $K[X]$, which is not a CDVF, so we may not directly apply the methods of the previous section. However, we may relate $\operatorname{Com}(X-H)$ to $\operatorname{Com}(1-XH)$, whose entries lie in the CDVF $K((X))$. In this way, we compute $\operatorname{Com}(X-H)$ using an SNF method, with no division in K.

First, we need a lemma relating comatrices of similar matrices:

Lemma 3.6. *If $M_1, M_2 \in M_n(K)$ and $P \in GL_n(K)$ are such that $M_1 = PM_2P^{-1}$, then:*

$$\operatorname{Com}(X-M_1) = P\operatorname{Com}(X-M_2)P^{-1}.$$

The second ingredient we need is reciprocal polynomials. We extend its definition to matrices of polynomials.

Definition 3.7. Let $d \in \mathbb{N}$ and $P \in K[X]$ of degree at most d. We define the reciprocal polynomial of order d of P as $P^{\mathrm{rec},d} = X^d P(1/X)$. Let $A \in M_n(K[X])$ a matrix of polynomials of degree at most d. We denote by $A^{\mathrm{rec},d}$ the matrix with $(A^{\mathrm{rec},d})_{i,j} = (A_{i,j})^{\mathrm{rec},d}$.

We then have the following result :

Lemma 3.8. *Let $M \in M_n(K)$. Then:*

$$\begin{aligned} \operatorname{Com}(1-XM)^{\mathrm{rec},n-1} &= \operatorname{Com}(X-M), \\ (\chi_M I_n)^{\mathrm{rec},n} &= (1-XM)\operatorname{Com}(1-XM). \end{aligned}$$

Proof. It all comes down to the following result: let $A \in M_d(K[X])$ a matrix of polynomials of degree at most 1, then $\det(A^{\mathrm{rec},1}) = \det(A)^{\mathrm{rec},d}$. Indeed, one can use multilinearity of the determinant on $X^d \det(A(1/X))$ to prove this result. It directly implies the second part of the lemma; the first part follows from the fact that the entries of $\mathrm{Com}(X-M)$ and of $\mathrm{Com}(1-XM)$ are determinants of size $n-1$. □

This lemma allows us to compute $\mathrm{Com}(1-XM)$ instead of $\mathrm{Com}(X-M)$. This has a remarkable advantage: the pivots during the computation of the SNF of $\mathrm{Com}(1-XM)$ are units of $\mathcal{O}_K[\![X]\!]$, and are known in advance to be on the diagonal. This leads to a very smooth precision and complexity behaviour when the input matrix lives in $M_n(\mathcal{O}_K)$.

```
Algorithm 2: Approximate Com(X − H)
Input: an approximate Hessenberg matrix $H$ in $M_n(\mathcal{O}_K)$.
0. $U := 1 - XH$. $U_0 := 1 - XH$.
1. While updating $U$, track $P$ and $Q$ so that $U_0 = PUQ$ is always satisfied.
2. for $i = 1, \dots, n-1$ do
3.   Eliminate, modulo $X^{n+1}$ the coefficients $U_{i,j}$, for $j \geq i+1$ using the invertible pivot $U_{i,i} = 1 + XL_{i,i} \mod X^{n+1}$ (with $L_{i,i} \in \mathcal{O}_K[X]$).
4. for $i = 1, \dots, n-1$ do
5.   Eliminate, modulo $X^{n+1}$ the coefficients $U_{i+1,i}$, using the invertible pivot $U_{i,i}$.
6.   $\psi := \prod_i U_{i,i}$.
7.   Rescale to get $U = I_n \mod X^{n+1}$.
8.   $V := \psi \times P \times Q \mod X^{n+1}$. [1]
9.   Return $V^{\mathrm{rec},n-1}, \psi^{\mathrm{rec},n}$.
```

Theorem 3.9. *Let $H \in M_n(\mathcal{O}_K)$ be an approximate Hessenberg matrix. Then, using Algorithm 2, one can compute* $\mathrm{Com}(X-H)$ *and χ_H in $O\tilde{}(n^3)$ operations in $\mathcal{O}_K$ at the precision given by H.*

Proof. First, the operations of the lines 2 and 3 use $O\tilde{}(n^3)$ operations in $\mathcal{O}_K$ at the precision given by H. Indeed, since H is an approximate Heisenberg matrix, when we use $U_{i,i}$ as pivot the only other nonzero coefficient in its column is $U_{i+1,i}$. As a consequence, when performing this column-pivoting, only two rows (i and $i+1$) lead to operations in $\mathcal{O}_K[\![X]\!]$ other than checking precision. Hence, line 3 costs $O\tilde{}(n^2)$ for the computation of U. Following line 1, the computation of Q is done by operations on rows, starting from the identity matrix. The order in which the entries of U are cleared implies that Q is just filled in as an upper triangular matrix: no additional operations in $\mathcal{O}_K[\![X]\!]$ are required. Thus the total cost for lines 2 and 3 is indeed $O\tilde{}(n^3)$ operations.

For lines 4 and 5, there are only $n-1$ eliminations, resulting in a $O\tilde{}(n^2)$ cost for the computation of U.. Rather than actually construct P, we just track the eliminations performed in order to do the corresponding row operations on Q, since we only need the product $P \times Q$.

Line 6 is in $O\tilde{}(n^2)$ and 7 in $O\tilde{}(n^3)$.

Thanks to the fact that the P only corresponds to the product of $n-1$ shear matrices, the product on line 8 is in $O\tilde{}(n^3)$. We emphasize that no division has been done throughout the algorithm. Line 9 is costless, and the result is then proved. □

Remark 3.10. If $M \in M_n(K)$ does not have coefficients in $\mathcal{O}_K$, we may apply Algorithms 1 and 2 to $p^v M \in M_n(\mathcal{O}_K)$ in $O\tilde{}(n^3)$ operations in $\mathcal{O}_K$, and then divide the coefficient of X^k in the resulting polynomial by p^{kv}.

We will see in Section 5 that for an entry matrix with coefficients known at flat precision, Algorithms 1 and 2 are enough to know the optimal jagged precision on χ_M.

3.4 The comatrix of $X-M$

In this section, we combine Proposition 2.7 with Algorithm 2 to compute the comatrix of $X-M$ when χ_M is squarefree. Note that this condition on χ_M is equivalent to M being diagonalizable under the assumption that $d\chi_M$ is surjective. The result is the following $O\tilde{}(n^3)$ algorithm, where the only divisions are for gcd and modular inverse computations.

```
Algorithm 3: Approximate Com(X−M)
Input: an approx. $M \in M_n(\mathcal{O}_K)$, with $\mathrm{Disc}(\chi_M) \neq 0$.
0. Find $P \in GL_n(\mathcal{O}_K)$ and $H \in M_n(\mathcal{O}_K)$, approximate Hessenberg, such that $M = PHP^{-1}$, using Algorithm 1.
1. Compute $A = \mathrm{Com}(X-H)$ and $\chi_M = \chi_H$ using Algorithm 2.
2. Do $\mathrm{row}(A,1) \leftarrow \mathrm{row}(A,1) + \sum_{i=2}^n \mu_i \,\mathrm{row}(A,i)$, for random $\mu_i \in \mathcal{O}_K$, by doing $T \times A$ for some $T \in GL_n(\mathcal{O}_K)$. Compute $B := TAT^{-1}$.
3. Similarly compute $C := S^{-1}BS$ for $S \in GL_n(\mathcal{O}_K)$ corresponding to adding a random linear combination of the columns of index $j \geq 2$ to the first column of $B$.
4. If $\gcd(C_{1,1}, \chi_M) \neq 1$, then go to 2.
5. Let $F$ be the inverse of $C_{1,1} \mod \chi_M$.
6. Let $U := \mathrm{col}(C,1)$ and $V := F \cdot \mathrm{row}(C,1) \mod \chi_M$.
7. Return $\mathrm{Com}(X-M) := (PT^{-1}SU \times VS^{-1}TP^{-1}) \mod \chi_M$.
```

Theorem 3.11. *For $M \in M_n(\mathcal{O}_K)$ such that* $\mathrm{Disc}(\chi_M) \neq 0$, *Algorithm 3 computes* $\mathrm{Com}(X-M)$ $\pmod{\chi_M}$ *in average complexity $O\tilde{}(n^3)$ operations in K. The only divisions occur in taking gcds and inverses modulo χ_M.*

Proof. As we have already seen, completing Steps 0 and 1 is in $O\tilde{}(n^3)$. Multiplying by T or S or their inverse corresponds to n operations on rows or columns over a matrix with coefficients in $\mathcal{O}_K[X]$ of degree at most n. Thus, it is in $O\tilde{}(n^3)$. Step 5 is in $O\tilde{}(n)$, Step 6 in $O\tilde{}(n^2)$ and Step 7 in $O\tilde{}(n^3)$. All that is to prove is that the set of P and S to avoid is of dimension at most $n-1$. The idea is to work modulo $X-\lambda$ for λ a root of $\chi(M)$ (in an algebraic closure) and then apply Chinese Remainder Theorem. The goal of the Step 2 is to ensure the first row of B contains an invertible entry modulo χ_M. Since $A(\lambda)$ is of rank one, the μ_i's have to avoid an affine hyperplane so that $\mathrm{row}(B,1) \mod (X-\lambda)$ is a non-zero vector. Hence for $\mathrm{row}(B,1) \mod \chi(M)$ to contain an invertible coefficient, a finite union of affine hyperplane is to avoid. Similarly, the goal of Step 3 is to put an invertible coefficient (modulo χ_M) on $C_{1,1}$, and again, only a finite union of affine hyperplane is to avoid. Hence, the set that the μ_i's have to avoid is a finite union of hyperplane, and hence, is of dimension at most $n-1$. Thus, almost any choice of μ_i leads to a matrix C passing the test in Step 4. This concludes the proof. □

[1] The product $P \times Q$ should be implemented by sequential row operations corresponding to the eliminations in Step 5 in order to avoid a product of two matrices in $M_n(\mathcal{O}_K[X])$.

Remark 3.12. As in the previous section, it is possible to scale $M \in M_n(K)$ so as to get coefficients in $\mathcal{O}_K$ and apply the previous algorithm.

Remark 3.13. We refer to [1] for the handling of the precision of gcd and modular inverse computations. In this article, ways to tame the loss of precision coming from divisions are explored, following the methods of [3].

4. DIFFERENTIAL VIA FROBENIUS FORM

The algorithm designed in the previous section computes the differential $d\chi_M$ of χ at a given matrix $M \in M_n(K)$ for a cost of $O(n^3)$ operations in K. This seems to be optimal given that the (naive) size of the $d\chi_M$ is n^3: it is a matrix of size $n \times n^2$. It turns out however that improvements are still possible! Indeed, thanks to Proposition 2.7, the matrix of $d\chi_M$ admits a compact form which can be encoded using only $O(n^2)$ coefficients. The aim of this short section is to design a fast algorithm (with complexity $O\tilde{}(n^{\omega})$) for computing this short form. The price to pay is that divisions in K appear, which can be an issue regarding to precision in particular cases. In this section, we only estimate the number of operations in K and not their behaviour on precision.

From now on, we fix a matrix $M \in M_n(K)$ for which $d\chi_M$ is surjective. Let (α, P, Q, χ_M) be the quadruple encoding the short form of $d\chi_M$; we recall that they are related by the relations:

$$d\chi_M(dM) = \mathrm{Tr}(\mathrm{Com}(X{-}M) \cdot dM)$$
$$\mathrm{Com}(X{-}M) = \alpha \cdot PV^t \cdot VQ^t \mod \chi_M.$$

An approximation to χ_M can be computed in $O\tilde{}(n^{\omega})$ operations in K (*e.g.* as a by-product of [15]).

The matrix P can be computed as follows. Pick $c \in K^n$. Define $c_i = M^i c$ for all $i \geq 1$. The c_i's can be computed in $O\tilde{}(n^{\omega})$ operations in K, *e.g.* using the first algorithm of [12]. Let P_{inv} be the $n \times n$ matrix whose rows are the c_i's for $1 \leq i \leq n$. Remark that P_{inv} is invertible if and only if $(c_0, c_1, \dots, c_{n-1})$ is a basis of K^n if and only if c is a cyclic vector. Moreover after base change to the basis $(c_0, \dots, c_{n-1})$, the matrix M takes the shape (3). In other words, if P_{inv} is invertible, then $P = P_{\mathrm{inv}}^{-1}$ is a solution of $M = P\mathscr{C}P^{-1}$, where $\mathscr{C}$ is the companion matrix similar to M. Moreover, observe that the condition "P_{inv} is invertible" is open for the Zariski topology. It then happens with high probability as soon as it is not empty, that is as soon as M admits a cyclic vector, which holds by assumption.

The characteristic polynomial χ_M can be recovered thanks to the relation $a_0c_0 + a_1c_1 + \cdots + a_{n-1}c_{n-1} = -c_{n-1} \cdot P$.

Now, instead of directly computing Q, we first compute a matrix R with the property that $\mathscr{C}^t = R\mathscr{C}R^{-1}$. To do so, we apply the same strategy as above except that we start with the vector $e = (1, 0, \dots, 0)$ (and not with a random vector). A simple computation shows that, for $1 \leq i \leq n{-}1$, the vector $\mathscr{C}^i e$ has the shape:

$$\mathscr{C}^i e = (0, \dots, 0, -a_0, \star, \dots, \star)$$

with $n{-}i$ starting zeros. Therefore the $\mathscr{C}^i e$'s form a basis of K^n, *i.e.* e is always a cyclic vector of $\mathscr{C}$. Once R has been computed, we recover Q using the relation $Q = P_{\mathrm{inv}}^t R$.

It remains to compute the scaling factor α. For this, we write the relation:

$$\mathrm{Com}(X{-}\mathscr{C}) = \alpha \cdot V^t \cdot VR^t \mod \chi_M \tag{5}$$

which comes from Eq. (4) after multiplication on the left by P^{-1} and multiplication on the right by P. We observe moreover that the first row of R is $(1, 0, \dots, 0)$. Evaluating the top left entry of Eq. (5), we end up with the relation:

$$\alpha = a_1 + a_2X + \cdots + a_{n-1}X^{n-2} + X^{n-1}.$$

No further computation are then needed to derive the value of α. We summarize this section with the following theorem:

Theorem 4.1. *Given $M \in M_n(K)$ such that $d\chi_M$ is surjective, then one can compute (α, P, Q, χ_M) in $K[X]$ such that $\mathrm{Com}(X{-}N) = \alpha \cdot V^t \cdot VR^t \mod \chi_M$ in $O\tilde{}(n^{\omega})$ operations in K.*

5. OPTIMAL JAGGED PRECISION

In the previous Sections, 3 and 4, we have proposed algorithms to obtain the comatrix of $X - M$. Our motivation for these computations is to then be able to understand what is the optimal precision on χ_M. In this section, we provide some answers to this question, along with numerical evidence. We also show that it is then possible to derive optimal precision of eigenvalues of M.

5.1 On the characteristic polynomial

For $0 \leq k < n$, let $\pi_k : K[X] \to K$ be the mapping taking a polynomial to its coefficients in X^k. By applying [3, Lem. 3.4] to the composite $\pi_k \circ \chi_M$, one can figure out the optimal precision on the k-th coefficient of the characteristic polynomial of M (at least if M is given at enough precision).

Let us consider more precisely the case where M is given at jagged precision: the (i,j) entry of M is given at precision $O(\pi^{N_{i,j}})$ for some integers $N_{i,j}$. Lemma 3.4 of [3] then shows that the optimal precision on the k-th coefficient of χ_M is $O(\pi^{N'_k})$ where N'_k is given by the formula:

$$N'_k = \min_{1 \leq i,j \leq n} N_{j,i} + \mathrm{val}(\pi_k(C_{i,j})), \tag{6}$$

where $C_{i,j}$ is the (i,j) entry of the comatrix $\mathrm{Com}(X{-}M)$.

Proposition 5.1. *If $M \in M_n(\mathcal{O}_K)$ is given at (high enough) jagged precision, then we can compute the optimal jagged precision on χ_M in $O\tilde{}(n^3)$ operations in K.*

PROOF. We have seen in §3 and §4 that the computation of the matrix $C = \mathrm{Com}(X{-}M)$ can be carried out within $O\tilde{}(n^3)$ operations in K (either with the Hessenberg method or the Frobenius method). We conclude by applying Eq. (6) which requires no further operation in K (but n^3 evaluations of valuations and n^3 manipulations of integers). □

Remark 5.2. If $M \in M_n(\mathcal{O}_K)$ is given at (high enough) *flat* precision, then we can avoid the final base change step in the Hessenberg method. Indeed, observe that, thanks to Lemma 3.6, we can write:

$$\mathrm{Tr}(\mathrm{Com}(X{-}M) \cdot dM) = \mathrm{Tr}(\mathrm{Com}(X{-}H) \cdot P^{-1}dMP)$$

where P lies in $\mathrm{GL}_n(\mathcal{O}_K)$. Moreover, the latter condition implies that $P^{-1}dMP$ runs over $M_n(\mathcal{O}_K)$ when P runs over $M_n(\mathcal{O}_K)$. As a consequence, the integer N'_k giving the optimal precision on the k-th coefficient of M is also equal to $N + \min_{1 \leq i,j \leq n} \mathrm{val}(\pi_k(C^H_{i,j}))$ where $C^H_{i,j}$ is the (i,j) entry of $\mathrm{Com}(X{-}H)$, where H is the Hessenberg form of M.

Remark 5.3. As a consequence of the previous discussion, once the optimal jagged precision is known, it is possible to lift the entries of M to a sufficiently large precision, rescale them to have entries in O_K and then use Algorithm 2 to compute the characteristic polynomial. The output might then need to be rescaled and truncated at the optimal precision. This requires $O\tilde{~}(n^3)$ operations in O_K and unfortunately, for several instances, may require to increase a lot the precision.

Numerical experiments. We have made numerical experiments in SageMath [6] in order to compare the optimal precision obtained with the methods explained above with the actual precision obtained by the software. For doing so, we picked a sample of 1000 random matrices M in $M_9(\mathbb{Q}_2)$ where all the entries are given at the same *relative precision*. We recall that, in SageMath, random elements $x \in \mathbb{Q}_p$ are generated as follows. We fix an integer N — the so-called relative precision — and generate elements of $\mathbb{Q}_p$ of the shape

$$x = p^v \cdot \big(a + O(p^{N+v_p(a)})\big)$$

where v is a random integer generated according to the distribution:

$$\mathbb{P}[v=0] = \frac{1}{5} \quad ; \quad \mathbb{P}[v=n] = \frac{2}{5\cdot|n|\cdot(|n|+1)} \text{ for } |n| \geq 1$$

and a is an integer in the range $[0, p^N)$, selected uniformly at random.

Once this sample has been generated, we computed, for each $k \in \{0, 1, \dots, 8\}$, the three following quantities:

- the optimal precision on the k-th coefficient of the characteristic polynomial of M given by Eq. (6)
- in the capped relative model[2], the precision gotten on the k-th coefficient of the characteristic polynomial of M computed *via* the call:

  M.charpoly(algorithm="df")
- in the model of floating-point arithmetic (see [2, §2.3]), the number of correct digits of the k-th coefficient of the characteristic polynomial of M.

Remark 5.4. The keyword `algorithm="df"` forces `SageMath` to use the division free algorithm of [14]. It is likely that, proceeding so, we limit the loss of precision.

The table of Figure 1 summarizes the results obtained. It should be read as follows. First, the acronyms CR and FP refers to "capped relative" and "floating-point" respectively. The numbers displayed in the table are the average loss of *relative* precision. More precisely, if N is the relative precision at which the entries of the input random matrix M have been generated and v is the valuation of the k-th coefficient of χ_M, then:

- the column "Optimal" is the average of the quantities $(N'_k - v) - N$ (where N'_k is defined by Eq. (6)): $N'_k - v$ is the optimal *relative* precision, so that the difference $(N'_k - v) - N$ is the loss of relative precision;
- the column "CR" is the average of the quatities $(\mathrm{CR}_k - v) - N$ where CR_k is the computed (absolute) precision on the k-th coefficient of χ_M;

	Average loss of accuracy		
	Optimal	CR	FP
X^0 (det.)	3.17 dev: 1.76	196 dev: 240	189 dev: 226
X^1	2.98 dev: 1.69	161 dev: 204	156 dev: 195
X^2	2.75 dev: 1.57	129 dev: 164	126 dev: 164
X^3	2.74 dev: 1.73	108 dev: 144	105 dev: 143
X^4	2.57 dev: 1.70	63.2 dev: 85.9	60.6 dev: 85.8
X^5	2.29 dev: 1.66	51.6 dev: 75.3	49.7 dev: 74.9
X^6	2.07 dev: 1.70	9.04 dev: 26.9	8.59 dev: 26.4
X^7	1.64 dev: 1.65	5.70 dev: 15.3	5.38 dev: 14.7
X^8 (trace)	0.99 dev: 1.37	0.99 dev: 1.37	0.99 dev: 1.37

Results for a sample of 1000 instances

Figure 1: Average loss of accuracy on the coefficients of the characteristic polynomial of a random 9×9 matrix over $\mathbb{Q}_2$

- the column "FP" is the average of the quatities $(\mathrm{FP}_k - v) - N$ where FP_k is the first position of an incorrect digit on the k-th coefficient of χ_M.

We observe that the loss of relative accuracy stays under control in the "Optimal" column whereas it has a very erratic behavior — very large values and very large deviation as well — in the two other columns. These experiments thus demonstrate the utility of the methods developed in this paper.

5.2 On eigenvalues

Let $M \in M_n(K)$ and $\lambda \in K$ be a *simple* [3] eigenvalue of M. We are interesting in quantifying the optimal precision on λ when M is given with some uncertainty.

To do so, we fix an approximation $M_{\text{app}} \in M_n(K)$ of M and suppose that the uncertainty of M is "jagged" in the sense that each entry of M is given at some precision $O(\pi^{N_{i,j}})$. Let λ_{app} be the relevant eigenvalue of M_{app}. We remark that it is possible to follow the eigenvalue λ_{app} on a small neighborhood $\mathcal{U}$ of M. More precisely, there exists a unique continuous function $f : \mathcal{U} \to K$ such that:

- $f(M_{\text{app}}) = \lambda_{\text{app}}$, and
- $f(M')$ is an eigenvalue of M' for all $M' \in \mathcal{U}$.

Lemma 5.5. *The function f is strictly differentiable on a neighborhood of M_{app}. The differential of f at M is the linear mapping:*

$$dM \mapsto d\lambda = -\frac{\mathrm{Tr}(\mathrm{Com}(\lambda - M)\cdot dM)}{\chi'_M(\lambda)}$$

where χ'_M is the usual derivative of χ_M.

[2] Each coefficient carries its own precision which is updated after each elementary arithmetical operation.

[3] the corresponding generalized eigenspace has dimension 1

Proof. The first assertion follows from the implicit function Theorem. Differentiating the relation $\chi_M(\lambda) = 0$, we get $\chi'_M(\lambda) \cdot d\lambda + \mathrm{Tr}(\mathrm{Com}(X - M) \cdot dM)(\lambda) = 0$, from which the Lemma follows. □

Lemma 3.4 of [3] now implies that, if the $N_{i,j}$'s are large enough and sufficiently well balanced, the optimal precision on the eigenvalue λ is $O(\pi^{N'})$ with:

$$N' = \min_{1\leq i,j\leq n} \big(N_{j,i} + \mathrm{val}(C_{i,j}(\lambda)) - \mathrm{val}(\chi'_M(\lambda))\big)$$

where $C_{i,j}$ denotes as above the (i, j) entry of $\mathrm{Com}(X-M)$. Writing $\mathrm{Com}(X-M) = \alpha \cdot PV^t \cdot VQ^t \bmod \chi_M$ as in Proposition 2.7, we find:

$$\begin{aligned} N' = {} & \mathrm{val}(\alpha(\lambda)) - \mathrm{val}(\chi'_M(\lambda)) \\ & + \min_{1\leq i,j\leq n} \big(N_{j,i} + \mathrm{val}(P_i V(\lambda)^t) + \mathrm{val}(V(\lambda)Q_j^t)\big) \end{aligned} \tag{7}$$

where P_i denotes the i-th row of P and, similarly, Q_j denotes the j-th row of Q. Note moreover that $V(\lambda)$ is the row vector $(1, \lambda, \dots, \lambda^{n-1})$. By the discussion of §4, the exact value of N' can be determined for a cost of $\tilde{O}(n^{\omega})$ operations in K and $O(n^2)$ operations on integers.

When M is given at flat precision, *i.e.* the $N_{i,j}$'s are all equal to some N, the formula for N' may be rewritten:

$$\begin{aligned} N' = {} & N + \mathrm{val}(\alpha(\lambda)) - \mathrm{val}(\chi'_M(\lambda)) \\ & + \min_{1\leq i\leq n} \mathrm{val}(P_i V(\lambda)^t) + \min_{1\leq j\leq n} \mathrm{val}(V(\lambda)Q_j^t)\big) \end{aligned} \tag{8}$$

and can therefore now be evaluated for a cost of $\tilde{O}(n^{\omega})$ operations in K and only $O(n)$ operations with integers.

To conclude, let us briefly discuss the situation where we want to figure out the optimal jagged precision on a tuple $(\lambda_1, \dots, \lambda_s)$ of simple eigenvalues. Applying (7), we find that the optimal precision on λ_k is

$$\begin{aligned} N'_k = {} & \mathrm{val}(\alpha(\lambda_k)) - \mathrm{val}(\chi'_M(\lambda_k)) \\ & + \min_{1\leq i,j\leq n} \big(N_{j,i} + \mathrm{val}(P_i V(\lambda_k)^t) + \mathrm{val}(V(\lambda_k)Q_j^t)\big). \end{aligned}$$

Proposition 5.6. *The N'_k's can be all computed in $\tilde{O}(n^{\omega})$ operations in K and $O(n^2 s)$ operations with integers.*

If the $N_{i,j}$'s are all equal, the above complexity can be lowered to $\tilde{O}(n^{\omega})$ operations in K and $O(ns)$ operations with integers.

Proof. The $\alpha(\lambda_k)$'s and the $\chi'_M(\lambda_k)$'s can be computed for a cost of $\tilde{O}(ns)$ operations in K using fast multipoint evaluation methods (see 10.7 of [17]). On the other hand, we observe that $P_i V(\lambda_k)^t$ is nothing but the (i, k) entry of the matrix:

$$P \cdot \begin{pmatrix} \lambda_1 & \cdots & \lambda_s \\ \lambda_1^2 & \cdots & \lambda_s^2 \\ \vdots & & \vdots \\ \lambda_1^{n-1} & \cdots & \lambda_s^{n-1} \end{pmatrix}.$$

The latter product can be computed in $\tilde{O}(n^{\omega})$ operations in K[4]. Therefore all the $P_i V(\lambda_k)^t$'s (for i and k varying) can be determined with the same complexity. Similarly all the $V(\lambda)Q_j^t$ are computed for the same cost. The first assertion of Proposition 5.6 follows. The second assertion is now proved similarly to the case of a unique eigenvalue. □

[4] It turns out that $\tilde{O}(n^2)$ is also possible because the right factor is a structured matrix (a truncated Vandermonde): computing the above product reduces to evaluating a polynomial at the points $\lambda_1, \dots, \lambda_s$.

References

[1] Xavier Caruso, *Resultants and subresultants of p-adic polynomials*, arxiv:1507.06502 (2015).

[2] ———, *Computations with p-adic numbers*, arxiv:1701.06794 (2017).

[3] Xavier Caruso, David Roe, and Tristan Vaccon, *Tracking p-adic precision*, LMS J. Comput. Math. **17** (2014), no. suppl. A, 274–294. MR3240809

[4] ———, *p-Adic Stability In Linear Algebra*, Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2015, Bath, United Kingdom, July 06 - 09, 2015 (2015), 101–108.

[5] Henri Cohen, *A course in computational algebraic number theory*, Vol. 138, Springer Science & Business Media, 2013.

[6] The Sage Developpers, *Sage Mathematics Software (Version 7.5)*, 2017. `http://www.sagemath.org`.

[7] David Harvey, *Kedlaya's algorithm in larger characteristic*, International Mathematics Research Notices **2007** (2007), rnm095.

[8] ———, *Counting points on hyperelliptic curves in average polynomial time*, Annals of Mathematics **179** (2014), no. 2, 783–803.

[9] Kenneth Hoffman and Ray Kunze, *Linear algebra*, 2nd ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[10] Erich Kaltofen, *On computing determinants of matrices without division*, Proc. 1992 Internat. Symp. Symbolic Algebraic Computation. (ISSAC'92), 1992, pp. 342–349.

[11] Kiran S. Kedlaya, *Counting points on hyperelliptic curves using monsky–washnitzer cohomology*, J. Ramanujan Math. Soc. **16** (2001), 323–338.

[12] Walter Keller-Gehrig, *Fast algorithms for the characteristics polynomial*, Theoretical Computer Science **36** (1985), 309 –317.

[13] François Le Gall, *Powers of tensors and fast matrix multiplication*, Proceedings of the 39th international symposium on symbolic and algebraic computation, 2014, pp. 296–303.

[14] T. R. Seifullin, *Computation of determinants, adjoint matrices, and characteristic polynomials without division*, Cybernetics and Systems Analysis **38** (2002), no. 5, 650–672.

[15] Arne Storjohann, *Deterministic computation of the Frobenius form*, Proceedings of the 42nd IEEE symposium on foundations of computer science, 2001, pp. 368–377.

[16] Tristan Vaccon, *p-adic precision*, Theses, 2015.

[17] Joachim Von Zur Gathen and Jürgen Gerhard, *Modern computer algebra*, Cambridge university press, 2013.