Characteristic polynomials of p-adic matrices.

Xavier Caruso, David Roe, **Tristan Vaccon**

Univ. Bordeaux, MIT, **Université de Limoges**

SMD 2019

A first question

Determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ 2X^6 + O(X^{10}) & 2X + O(X^{10}) & 2X + X^5 + O(X^{10}) \end{bmatrix}$$

A first question

Determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ 2X^6 + O(X^{10}) & 2X + O(X^{10}) & 2X + X^5 + O(X^{10}) \end{bmatrix}$$

Question

What is the **precision on the determinant** ?

A little warm-up on computing determinants : expansion

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ 2X^6 + O(X^{10}) & 2X + O(X^{10}) & 2X + X^5 + O(X^{10}) \end{bmatrix}$$

A little warm-up on computing determinants : expansion

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ 2X^6 + O(X^{10}) & 2X + O(X^{10}) & 2X + X^5 + O(X^{10}) \end{bmatrix}$$

Using Leibniz formula

If we expand directly using the expression of the determinant in terms of the coefficients, we get:

A little warm-up on computing determinants : expansion

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & \boxed{1 + O(X^{10})} & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & \boxed{1 + O(X^{10})} \\ \boxed{2X^6 + O(X^{10})} & 2X + O(X^{10}) & 2X + X^5 + O(X^{10}) \end{bmatrix}$$

Using Leibniz formula

If we expand directly using the expression of the determinant in terms of the coefficients, we get:

$$-2X^9 + O(X^{10}),$$

because of $1 \times 1 \times O(X^{10})$.

A little warm-up on computing determinants : row-echelon form computation

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ O(X^{10}) & O(X^{10}) & -2X^4 + X^5 + O(X^{10}) \end{bmatrix}$$

A little warm-up on computing determinants : row-echelon form computation

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & 1 + O(X^{10}) & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & 1 + O(X^{10}) \\ O(X^{10}) & O(X^{10}) & -2X^4 + X^5 + O(X^{10}) \end{bmatrix}$$

Row-echelon form computation

If we compute **approximate** row-echelon form, we still get:

A little warm-up on computing determinants : row-echelon form computation

An example of determinant computation

$$\begin{bmatrix} X^5 + O(X^{10}) & \boxed{1 + O(X^{10})} & 1 + X^3 + O(X^{10}) \\ O(X^{10}) & 1 + O(X^{10}) & \boxed{1 + O(X^{10})} \\ \boxed{O(X^{10})} & O(X^{10}) & -2X^4 + X^5 + O(X^{10}) \end{bmatrix}$$

Row-echelon form computation

If we compute **approximate** row-echelon form, we still get:

$$-2X^9 + O(X^{10}),$$

because of $1 \times 1 \times O(X^{10})$.

A little warm-up on computing determinants : SNF

An example of determinant computation

$$\begin{bmatrix} 1 + O(X^{10}) & O(X^{10}) & O(X^{10}) \\ O(X^{10}) & X^3 + O(X^{10}) & O(X^{10}) \\ O(X^{10}) & O(X^{10}) & -2X^6 + X^7 + O(X^{10}) \end{bmatrix}$$

A little warm-up on computing determinants : SNF

An example of determinant computation

$$\begin{bmatrix} 1+O(X^{10}) & O(X^{10}) & O(X^{10}) \\ O(X^{10}) & X^3+O(X^{10}) & O(X^{10}) \\ O(X^{10}) & O(X^{10}) & -2X^6+X^7+O(X^{10}) \end{bmatrix}$$

Smith Normal Form (SNF) computation

If we compute **approximate** SNF, we now get:

A little warm-up on computing determinants : SNF

An example of determinant computation

$$\begin{bmatrix} 1+O(X^{10}) & O(X^{10}) & O(X^{10}) \\ O(X^{10}) & X^3+O(X^{10}) & O(X^{10}) \\ O(X^{10}) & O(X^{10}) & -2X^6+X^7+O(X^{10}) \end{bmatrix}$$

Smith Normal Form (SNF) computation

If we compute **approximate** SNF, we now get:

$$-2X^9+X^{10}+O(X^{13}),$$

because of $1 \times X^3 \times O(X^{10}) = O(X^{13})$.

Questions for today

Determinant

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?
- Is there an algorithm to reach this precision?

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?
- Is there an algorithm to reach this precision?

Characteristic polynomial

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?
- Is there an algorithm to reach this precision?

Characteristic polynomial

- Is there an **optimal** precision on the coefficients of the characteristic polynomial?

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?
- Is there an algorithm to reach this precision?

Characteristic polynomial

- Is there an **optimal** precision on the coefficients of the characteristic polynomial?
- Is there an algorithm to reach this precision?

Questions for today

Determinant

- Is there an **optimal** precision on the determinant?
- Is there an algorithm to reach this precision?

Characteristic polynomial

- Is there an **optimal** precision on the coefficients of the characteristic polynomial?
- Is there an algorithm to reach this precision?

Remarque

From now on, we will work over $\mathbb{Q}_p$ instead of $K[\![X]\!]$, but there is no difference in the behaviour regarding to precision.

Motivations and goal

Counting points on curves

- Kedlaya's algorithm to count point on curves.

Motivations and goal

Counting points on curves

- Kedlaya's algorithm to count point on curves.
- One core part of Kedlaya's algorithm is the computation of the **characteristic polynomial** of the linear mapping given by the Frobenius acting on some cohomological p-adic vector space.

Motivations and goal

Counting points on curves

- Kedlaya's algorithm to count point on curves.
- One core part of Kedlaya's algorithm is the computation of the **characteristic polynomial** of the linear mapping given by the Frobenius acting on some cohomological p-adic vector space.

Today's goal

- What is the (optimal) precision on the characteristic polynomial of a matrix with p-adic entries all known at the same precision?
- How can we compute at this precision?

Table of contents

Definition of the precision

Finite-precision p-adics

Elements of $\mathbb{Q}_p$ can be written $\sum_{i=l}^{+\infty} a_i p^i$, with $a_i \in [\![0, p-1]\!]$, $l \in \mathbb{Z}$ and p a prime number.

Working with a computer, we usually only can consider the beginning of this power series expansion: we only consider elements of the form $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$, with $l \in \mathbb{Z}$.

Definition of the precision

Finite-precision p-adics

Elements of $\mathbb{Q}_p$ can be written $\sum_{i=l}^{+\infty} a_i p^i$, with $a_i \in [\![0, p-1]\!]$, $l \in \mathbb{Z}$ and p a prime number.
Working with a computer, we usually only can consider the beginning of this power series expansion: we only consider elements of the form $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$, with $l \in \mathbb{Z}$.

Definition of the precision

Finite-precision p-adics

Elements of $\mathbb{Q}_p$ can be written $\sum_{i=l}^{+\infty} a_i p^i$, with $a_i \in [\![0, p-1]\!]$, $l \in \mathbb{Z}$ and p a prime number.
Working with a computer, we usually only can consider the beginning of this power series expansion: we only consider elements of the form $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$, with $l \in \mathbb{Z}$.

Definition

The **order**, or the **absolute precision** of $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$ is d.

Definition of the precision

Finite-precision p-adics

Elements of $\mathbb{Q}_p$ can be written $\sum_{i=l}^{+\infty} a_i p^i$, with $a_i \in [\![0, p-1]\!]$, $l \in \mathbb{Z}$ and p a prime number.
Working with a computer, we usually only can consider the beginning of this power series expansion: we only consider elements of the form $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$, with $l \in \mathbb{Z}$.

Definition

The **order**, or the **absolute precision** of $\sum_{i=l}^{d-1} a_i p^i + O(p^d)$ is d.

Exemple

The order of $3 * 7^{-1} + 4 * 7^0 + 5 * 7^1 + 6 * 7^2 + O(7^3)$ is 3.

Precision formulae

Proposition (addition)

$$(x_0 + O(p^{k_0})) + (x_1 + O(p^{k_1})) = x_0 + x_1 + O(p^{\min(k_0,k_1)})$$

Precision formulae

Proposition (addition)

$$(x_0 + O(p^{k_0})) + (x_1 + O(p^{k_1})) = x_0 + x_1 + O(p^{\min(k_0, k_1)})$$

Proposition (multiplication)

$$(x_0 + O(p^{k_0})) * (x_1 + O(p^{k_1})) = x_0 * x_1 + O(p^{\min(k_0 + v_p(x_1), k_1 + v_p(x_0))})$$

Precision formulae

Proposition (addition)

$$(x_0 + O(p^{k_0})) + (x_1 + O(p^{k_1})) = x_0 + x_1 + O(p^{\min(k_0,k_1)})$$

Proposition (multiplication)

$$(x_0 + O(p^{k_0})) * (x_1 + O(p^{k_1})) = x_0 * x_1 + O(p^{\min(k_0+v_p(x_1),k_1+v_p(x_0))})$$

Proposition (division)

$$\frac{xp^a + O(p^b)}{yp^c + O(p^d)} = x * y^{-1} p^{a-c} + O(p^{\min(d+a-2c,b-c)})$$

In particular,
$$\frac{1}{p^c y + O(p^d)} = y^{-1} p^{-c} + O(p^{d-2c})$$

The Main lemma of p-adic differential precision

Lemma (CRV14)

Let $f : \mathbb{Q}_p^n \to \mathbb{Q}_p^m$ be a (strictly) ***differentiable*** *mapping.*

The Main lemma of p-adic differential precision

Lemma (CRV14)

Let $f : \mathbb{Q}_p^n \to \mathbb{Q}_p^m$ be a (strictly) ***differentiable*** *mapping.*
Let $x \in \mathbb{Q}_p^n$. We assume that $f'(x)$ is ***surjective****.*

The Main lemma of p-adic differential precision

Lemma (CRV14)

Let $f : \mathbb{Q}_p^n \to \mathbb{Q}_p^m$ be a (strictly) ***differentiable*** *mapping.*
Let $x \in \mathbb{Q}_p^n$. We assume that $f'(x)$ is ***surjective****.*
Then for any ball $B = B(0, r)$ ***small enough****,*

The Main lemma of p-adic differential precision

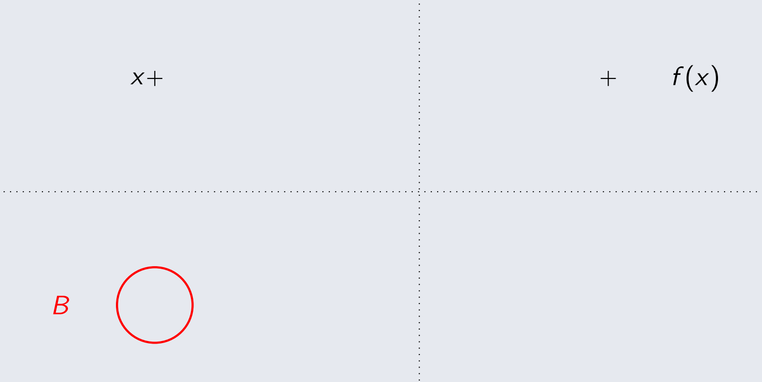
Lemma (CRV14)

Let $f : \mathbb{Q}_p^n \to \mathbb{Q}_p^m$ be a (strictly) ***differentiable*** *mapping.*
Let $x \in \mathbb{Q}_p^n$. We assume that $f'(x)$ is ***surjective****.*
Then for any ball $B = B(0, r)$ ***small enough****,*

$$f(x + B) = f(x) + f'(x) \cdot B.$$

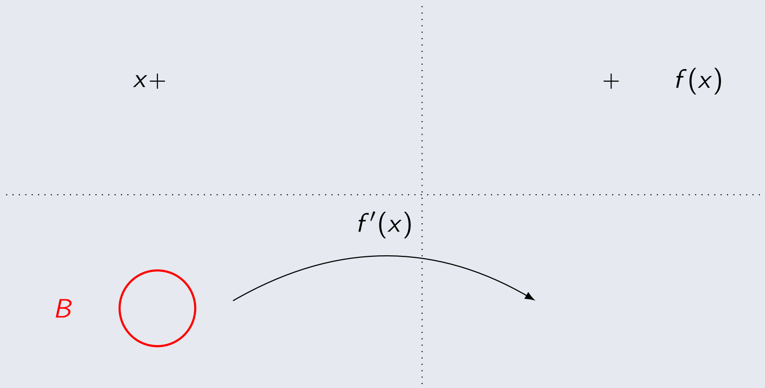
Geometrical meaning

Interpretation



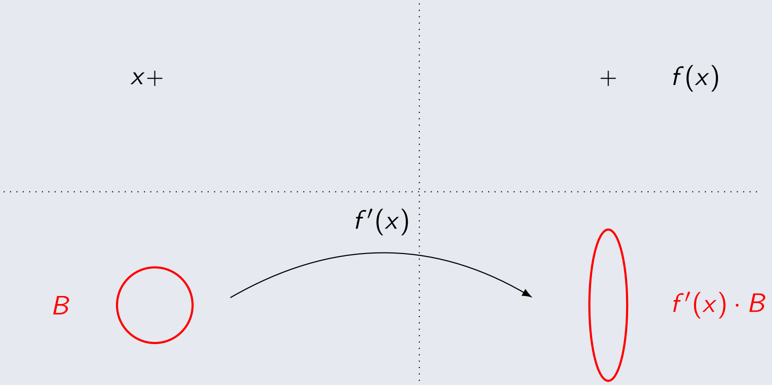
Geometrical meaning

Interpretation



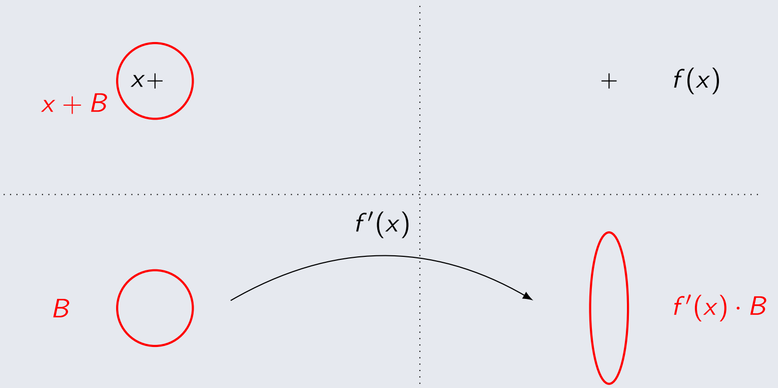
Geometrical meaning

Interpretation



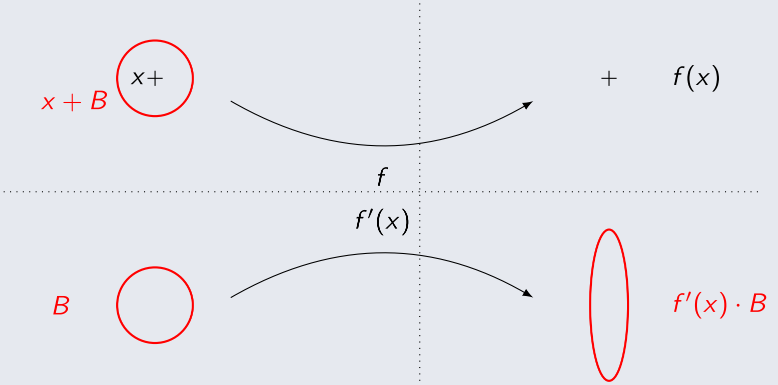
Geometrical meaning

Interpretation



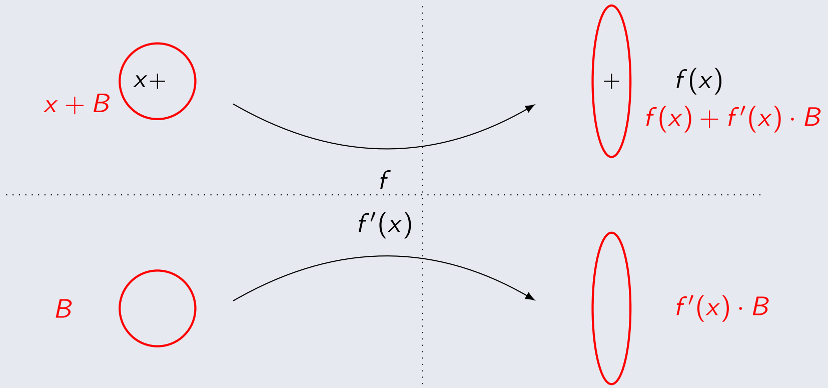
Geometrical meaning

Interpretation



Geometrical meaning

Interpretation



Looking back to the case of the determinant

Differential of the determinant

It is well known:

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Looking back to the case of the determinant

Differential of the determinant

It is well known:

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Consequence on precision

- Loss in precision: coefficient of $\mathrm{Adj}(M)$ with smallest valuation.

Looking back to the case of the determinant

Differential of the determinant

It is well known:

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Consequence on precision

- Loss in precision: coefficient of $\mathrm{Adj}(M)$ with smallest valuation.
- Corresponds to the products of the $n-1$-first invariant factors.

Looking back to the case of the determinant

Differential of the determinant

It is well known:

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Consequence on precision

- Loss in precision: coefficient of $\mathrm{Adj}(M)$ with smallest valuation.
- Corresponds to the products of the $n-1$-first invariant factors.
- **Approximate SNF is optimal.**

Looking back to the case of the determinant

Differential of the determinant

It is well known:

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Consequence on precision

- Loss in precision: coefficient of $\mathrm{Adj}(M)$ with smallest valuation.
- Corresponds to the products of the $n-1$-first invariant factors.
- **Approximate SNF is optimal.**

Linear equations

One can also easily prove that SNF is optimal to solve linear equations.

Table of contents

Classical ways to compute χ_M

Direct Gaussian elimination

Is $\tilde{O}(n^4)$, **with divisions**.

Classical ways to compute χ_M

Direct Gaussian elimination

Is $\tilde{O}(n^4)$, **with divisions**.

Also with divisions

Fadeev-Leverrier and Berlekamp-Massey.

Fastest ways

Deterministic: Storjohann (2001)

Computes Frobenius Normal Form, and hence χ_M.
Is in $O\tilde{}(n^\omega)$, **with divisions**.

Fastest ways

Deterministic: Storjohann (2001)

Computes Frobenius Normal Form, and hence χ_M.
Is in $\tilde{O}(n^\omega)$, **with divisions**.

Non-deterministic: Pernet-Storjohann (2007), field large enough

Is a Las Vegas algorithm to compute Frobenius Normal Form, and hence χ_M. Is in $O(n^\omega)$ in average, **with divisions**.

Fastest ways

Deterministic: Storjohann (2001)

Computes Frobenius Normal Form, and hence χ_M.
Is in $\tilde{O}(n^\omega)$, **with divisions**.

Non-deterministic: Pernet-Storjohann (2007), field large enough

Is a Las Vegas algorithm to compute Frobenius Normal Form, and hence χ_M. Is in $O(n^\omega)$ in average, **with divisions**.

Division-free: Kaltoffen-Villard (2004)

Is in $O(n^{2.7})$.

Fastest ways

Deterministic: Storjohann (2001)

Computes Frobenius Normal Form, and hence χ_M.
Is in $\tilde{O}(n^\omega)$, **with divisions**.

Non-deterministic: Pernet-Storjohann (2007), field large enough

Is a Las Vegas algorithm to compute Frobenius Normal Form, and hence χ_M. Is in $O(n^\omega)$ in average, **with divisions**.

Division-free: Kaltoffen-Villard (2004)

Is in $O(n^{2.7})$.

What is left?

- No division, so precision is saved, can a **gain of precision** be seen?

Fastest ways

Deterministic: Storjohann (2001)

Computes Frobenius Normal Form, and hence χ_M.
Is in $\tilde{O}(n^\omega)$, **with divisions**.

Non-deterministic: Pernet-Storjohann (2007), field large enough

Is a Las Vegas algorithm to compute Frobenius Normal Form, and hence χ_M. Is in $O(n^\omega)$ in average, **with divisions**.

Division-free: Kaltoffen-Villard (2004)

Is in $O(n^{2.7})$.

What is left?

- No division, so precision is saved, can a **gain of precision** be seen?
- If we know the optimal precision. We can perform Kaltoffen-Villard at high-enough precision to get the extra digits.

$\chi'(M)$

Derivative of det

$$\det{}'(M) : dM \mapsto \operatorname{Tr}(\operatorname{Adj}(M) \cdot dM).$$

$\chi'(M)$

Derivative of det

$$\det{}'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(M) \cdot dM).$$

Derivative of χ_M

$$\chi'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

Naïve computations

Formulae

$$\chi'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\mathrm{Adj}(XI_n - M) = \chi_M \times (XI_n - M)^{-1}.$$

Naïve computations

Formulae

$$\chi'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$
$$\mathrm{Adj}(XI_n - M) = \chi_M \times (XI_n - M)^{-1}.$$

First idea

- Compute (approximations of) χ_M and $(XI_n - M)^{-1}$.

Naïve computations

Formulae

$$\chi'(M) : dM \mapsto \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$
$$\mathrm{Adj}(XI_n - M) = \chi_M \times (XI_n - M)^{-1}.$$

First idea

- Compute (approximations of) χ_M and $(XI_n - M)^{-1}$.
- Computing $(XI_n - M)^{-1} \mod X^{n+1}$ is $\tilde{O}(n^4)$ by Gaussian elimination (+ it requires divisions).

Table of contents

Table of contents

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal?

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal? No.

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal? No.
- Frobenius?

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal? No.
- Frobenius? Too complicated?

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal? No.
- Frobenius? Too complicated?
- Hessenberg?

Invariance

Similarity

For $H = PMP^{-1}$,

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - M) \cdot dM).$$

$$\chi'(M) \cdot dM = \mathrm{Tr}(\mathrm{Adj}(XI_n - H) \cdot PdMP^{-1}).$$

Enough for flat precision.

Which form?

- Jordan or trigonal? No.
- Frobenius? Too complicated?
- Hessenberg? Seems a good idea.

Hessenberg form

Hessenberg matrix

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & & m_{1,n-1} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & & m_{2,n} \\ 0 & m_{3,2} & m_{3,3} & m_{3,4} & & & & m_{3,n} \\ 0 & 0 & m_{4,3} & m_{4,4} & & & & m_{4,n} \\ & 0 & 0 & m_{5,4} & & & & \\ & 0 & & 0 & & & & \\ & & & & & m_{n-1,n-2} & m_{n-1,n-1} & m_{n-1,n} \\ 0 & 0 & 0 & 0 & & 0 & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

Remark

A companion matrix is Hessenberg.

Hessenberg form

Hessenberg matrix

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n-1} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ 0 & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ 0 & 0 & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & 0 & 0 & m_{5,4} & & & \\ & 0 & & 0 & & & \\ & & & & m_{n-1,n-2} & m_{n-1,n-1} & m_{n-1,n} \\ 0 & 0 & 0 & 0 & 0 & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

Remark

A companion matrix is Hessenberg. The Frobenius form is Hessenberg.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** and put it on position $(2, 1)$.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** and put it on position $(2, 1)$.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** and put it on position $(2,1)$.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** and put it on position $(2,1)$.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** and put it on position $(2, 1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2, 1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & & & m_{4,n} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & & & m_{3,n} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,2} & m_{n,3} & m_{n,4} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ m_{3,1} & m_{3,4} & m_{3,3} & m_{3,2} & & & m_{3,n} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ m_{3,1} & m_{3,4} & m_{3,3} & m_{3,2} & & & m_{3,n} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ m_{3,1} & m_{3,4} & m_{3,3} & m_{3,2} & & & m_{3,n} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ m_{3,1} & m_{3,4} & m_{3,3} & m_{3,2} & & & m_{3,n} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & m_{1,4} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & m_{4,4} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ m_{2,1} & m_{2,4} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & m_{n,4} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ m_{2,1} & \widetilde{m_{2,4}} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ m_{2,1} & \widetilde{m_{2,4}} & m_{2,3} & m_{2,2} & & & m_{2,n} \\ & & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & & & & & \\ m_{n,1} & \widetilde{m_{n,4}} & m_{n,3} & m_{n,2} & m_{n-1,n} & m_{n,n} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ 0 & \widetilde{m_{n,4}} & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2, 1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ 0 & \widetilde{m_{n,4}} & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ 0 & \widetilde{m_{n,4}} & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ 0 & \widetilde{m_{n,4}} & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2, 1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & m_{1,3} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & m_{4,3} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & \widetilde{m_{2,4}} & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & & & & & \\ 0 & \widetilde{m_{n,4}} & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & 0 & & & & & \\ & 0 & & & & & \\ & & & & & & \\ 0 & 0 & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & 0 & & & & \\ & 0 & & & & \\ 0 & 0 & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2, 1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & 0 & & & & \\ & 0 & & & & \\ 0 & 0 & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & & & & & & \\ 0 & 0 & \widetilde{m_{n,3}} & \widetilde{m_{n,2}} & & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & 0 & 0 & & & \\ & 0 & & & & \\ 0 & 0 & 0 & m_{n,2} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & 0 & 0 & & & \\ & 0 & & & & \\ 0 & 0 & 0 & m_{n,2} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & m_{1,n} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & \widetilde{m_{2,n}} \\ & 0 & 0 & & & \\ & 0 & & & & \\ 0 & 0 & 0 & m_{n,2} & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

We take as pivot the coefficient $m_{i,1}$ on first column with the **smallest valuation** put it on position $(2,1)$. This reflects to the columns. We pivot with the second row. It reflects on the columns. We proceed **recursively**.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix} m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & & m_{1,n-1} & m_{1,n} \\ m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & & m_{4,n} \\ 0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\ 0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\ & 0 & 0 & \widetilde{m_{5,2}} & & & \\ & 0 & & 0 & & & \\ & & & & m_{\widetilde{n-1,n-2}} & m_{\widetilde{n-1,n-1}} & \widetilde{m_{n-1,n}} \\ 0 & 0 & 0 & 0 & 0 & \widetilde{m_{n-1,n}} & \widetilde{m_{n,n}} \end{bmatrix}$$

The result is Hessenberg. It required $O(n^3)$ operations on the base field.

Computation of an Hessenberg form

Hessenberg reduction: modified Gaussian elimination

$$P_* M P_*^{-1} = \begin{bmatrix}
m_{1,1} & \widetilde{m_{1,4}} & \widetilde{m_{1,3}} & m_{1,2} & & m_{1,n-1} & m_{1,n} \\
m_{4,1} & \widetilde{m_{4,4}} & \widetilde{m_{4,3}} & m_{4,2} & & & m_{4,n} \\
0 & \widetilde{m_{3,4}} & \widetilde{m_{3,3}} & \widetilde{m_{3,2}} & & & \widetilde{m_{3,n}} \\
0 & 0 & \widetilde{m_{2,3}} & \widetilde{m_{2,2}} & & & \widetilde{m_{2,n}} \\
 & 0 & 0 & \widetilde{m_{5,2}} & & & \\
 & 0 & & 0 & & & \\
 & & & & m_{\widetilde{n-1,n-2}} & m_{\widetilde{n-1,n-1}} & m_{\widetilde{n-1,n}} \\
0 & 0 & 0 & 0 & 0 & m_{\widetilde{n-1,n}} & \widetilde{m_{n,n}}
\end{bmatrix}$$

The result is Hessenberg. It required $O(n^3)$ operations on the base field.
It is possible to do everything $\bmod p^N$, with no division.

Table of contents

Adjugate of $H = PMP^{-1}$

$XI_n - H$

$$\det(XI_n - H) = \chi_H.$$
$$Adj(XI_n - H) = \chi_M \times (XI_n - H)^{-1}.$$

Adjugate of $H = PMP^{-1}$

$XI_n - H$

$$det(XI_n - H) = \chi_H.$$
$$Adj(XI_n - H) = \chi_M \times (XI_n - H)^{-1}.$$

$I_n - XH$

$$det(I_n - XH) = \chi_H^*, \text{ reciprocal polynomial.}$$
$$\text{Adj}(I_n - XH) = \chi_M \times (I_n - XH)^{-1}.$$

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done mod p^M, X^{n+1}.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & Xh_{1,2} & Xh_{1,3} & & & Xh_{1,n-1} & Xh_{1,n} \\ Xh_{2,1} & 1 - Xh_{2,2} & Xh_{2,3} & & & Xh_{2,n-1} & Xh_{2,n} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & X\widetilde{h_{2,3}} & & X\widetilde{h_{2,n-1}} & X\widetilde{h_{2,n}} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & X\widetilde{h_{2,3}} & & X\widetilde{h_{2,n-1}} & X\widetilde{h_{2,n}} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & X\widetilde{h_{2,3}} & & X\widetilde{h_{2,n-1}} & X\widetilde{h_{2,n}} \\ 0 & Xh_{3,2} & 1 - Xh_{3,3} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & 0 & 0 & 0 & 0 \\ 0 & Xh_{3,2} & 1 - X\widetilde{h_{3,3}} & & & \\ 0 & 0 & Xh_{4,3} & & & \\ 0 & 0 & 0 & & & \\ & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & Xh_{3,2} & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - Xh_{n-1,n-1} & Xh_{n-1,n} \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - Xh_{n,n} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & Xh_{3,2} & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & Xh_{3,2} & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ Xh_{2,1} & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & Xh_{3,2} & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & Xh_{4,3} & & & & \\ 0 & 0 & 0 & & & & \\ & & & & Xh_{n-1,n-2} & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & Xh_{n,n-1} & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ 0 & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & 0 & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & & & & \\ & & & & 0 & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & 0 & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ 0 & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & 0 & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & & & & \\ & & & & 0 & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & 0 & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$. $\det(Id - XH) = \prod_i (1 - X\widetilde{h_{i,i}})$.

Computation

An algorithm for Hessenberg matrices: computation of $(Id - XH)^{-1}$

$$P_*(I_n - XH)Q_* = \begin{bmatrix} 1 - Xh_{1,1} & 0 & 0 & & & 0 & 0 \\ 0 & 1 - X\widetilde{h_{2,2}} & 0 & 0 & & 0 & 0 \\ 0 & 0 & 1 - X\widetilde{h_{3,3}} & & & & \\ 0 & 0 & 0 & & & & \\ 0 & 0 & 0 & & & & \\ & & & & 0 & 1 - X\widetilde{h_{n-1,n-1}} & 0 \\ 0 & 0 & 0 & & 0 & 0 & 1 - X\widetilde{h_{n,n}} \end{bmatrix}$$

Everything done $\bmod p^M, X^{n+1}$. $\det(Id - XH) = \prod_i (1 - X\widetilde{h_{i,i}})$.
$(Id - XH)^{-1}$ obtained from $Q\Delta^{-1}P$, in $\tilde{O}(n^3)$, with no division.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\mathrm{Adj}(I_n - XH)$.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\mathrm{Adj}(I_n - XH)$.

Precision and complexity

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\text{Adj}(I_n - XH)$.

Precision and complexity

- $\text{Adj}(XI_n - H) = \text{Adj}(I_n - XH)^*$.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\text{Adj}(I_n - XH)$.

Precision and complexity

- $\text{Adj}(XI_n - H) = \text{Adj}(I_n - XH)^*$.
- $\text{Adj}(I_n - XH) = P\,\text{Adj}(I_n - XM)P^{-1}$

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\mathrm{Adj}(I_n - XH)$.

Precision and complexity

- $\mathrm{Adj}(XI_n - H) = \mathrm{Adj}(I_n - XH)^*$.
- $\mathrm{Adj}(I_n - XH) = P\,\mathrm{Adj}(I_n - XM)P^{-1}$
- If $P^{-1}dMP = dM$,

$$tr(\mathrm{Adj}(I_n - XM) \cdot dM) = tr(\mathrm{Adj}(I_n - XH) \cdot dM).$$

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\mathrm{Adj}(I_n - XH)$.

Precision and complexity

- $\mathrm{Adj}(XI_n - H) = \mathrm{Adj}(I_n - XH)^*$.
- $\mathrm{Adj}(I_n - XH) = P\,\mathrm{Adj}(I_n - XM)P^{-1}$
- If $P^{-1}dMP = dM$,

$$tr(\mathrm{Adj}(I_n - XM) \cdot dM) = tr(\mathrm{Adj}(I_n - XH) \cdot dM).$$

Precision can directly be read from $\mathrm{Adj}(I_n - XH)$.

Conclusion on flat precision

The case of flat precision

M is given at precision $dM = O(p^N)$ on each coefficient.

1. Compute $H = PMP^{-1}$ an Hessenberg form.
2. Compute $\mathrm{Adj}(I_n - XH)$.

Precision and complexity

- $\mathrm{Adj}(XI_n - H) = \mathrm{Adj}(I_n - XH)^*$.
- $\mathrm{Adj}(I_n - XH) = P\,\mathrm{Adj}(I_n - XM)P^{-1}$
- If $P^{-1}dMP = dM$,

$$tr(\mathrm{Adj}(I_n - XM) \cdot dM) = tr(\mathrm{Adj}(I_n - XH) \cdot dM).$$

Precision can directly be read from $\mathrm{Adj}(I_n - XH)$. All in all in $\tilde{O}(n^3)$, with no division.

Non-flat

Jagged precision

Precision dM on each coefficient of M may differ.

Precision?

No longer $P^{-1}dMP = dM$.

Non-flat

Jagged precision

Precision dM on each coefficient of M may differ.

Precision?

No longer $P^{-1}dMP = dM$.
Computing $P\,\mathrm{Adj}(XI_n - H)P^{-1}$ or $P^{-1}dMP$ is very costly.

Factorization

A classical formula

If $A \in M_n(K)$ is of rank 1,

Factorization

A classical formula

If $A \in M_n(K)$ is of rank 1, there exists $u, v \in K^n$ such that

$$A = u \cdot {}^T v.$$

Factorization

A classical formula

If $A \in M_n(K)$ is of rank 1, there exists $u, v \in K^n$ such that

$$A = u \cdot {}^T v.$$

A new formula for $\mathrm{Adj}(XId - M)$

Under some genericity assumptions,

Factorization

A classical formula

If $A \in M_n(K)$ is of rank 1, there exists $u, v \in K^n$ such that

$$A = u \cdot {}^T v.$$

A new formula for $\mathrm{Adj}(XId - M)$

Under some genericity assumptions, there exists some $U, V \in K[X]_{\leq n}^n$, $f \in K[X]_{\leq n}$ such that:

Factorization

A classical formula

If $A \in M_n(K)$ is of rank 1, there exists $u, v \in K^n$ such that

$$A = u \cdot {}^T v.$$

A new formula for $\mathrm{Adj}(XId - M)$

Under some genericity assumptions, there exists some $U, V \in K[X]^n_{\leq n}$, $f \in K[X]_{\leq n}$ such that:

$$\mathrm{Adj}(XId - M) = f \cdot U \cdot {}^T V \mod \chi_M.$$

Conclusion on jagged precision

An algorithm

Conclusion on jagged precision

An algorithm

- Compute $\mathrm{Adj}(XId - H)$ and approximate χ_M for $M = PHP^{-1}$ with H Hessenberg.

Conclusion on jagged precision

An algorithm

- Compute $\mathrm{Adj}(XId - H)$ and approximate χ_M for $M = PHP^{-1}$ with H Hessenberg.
- Factor $\mathrm{Adj}(XId - H) = f \cdot U \cdot {}^T V \mod \chi_M$.

Conclusion on jagged precision

An algorithm

- Compute $\mathrm{Adj}(XId - H)$ and approximate χ_M for $M = PHP^{-1}$ with H Hessenberg.
- Factor $\mathrm{Adj}(XId - H) = f \cdot U \cdot {}^T V \mod \chi_M$.
- Then $\mathrm{Adj}(XId - M) = f \cdot (PU) \cdot ({}^T V P^{-1}) \mod \chi_M$.

Conclusion on jagged precision

An algorithm

- Compute $\mathrm{Adj}(XId - H)$ and approximate χ_M for $M = PHP^{-1}$ with H Hessenberg.
- Factor $\mathrm{Adj}(XId - H) = f \cdot U \cdot {}^T V \mod \chi_M$.
- Then $\mathrm{Adj}(XId - M) = f \cdot (PU) \cdot ({}^T VP^{-1}) \mod \chi_M$.

Complexity and precision

In $\tilde{O}(n^3)$, but with divisions to compute the factorization (Extended Euclidean Algorithm).

Conclusion on jagged precision

An algorithm

- Compute $\mathrm{Adj}(XId - H)$ and approximate χ_M for $M = PHP^{-1}$ with H Hessenberg.
- Factor $\mathrm{Adj}(XId - H) = f \cdot U \cdot {}^T V \mod \chi_M$.
- Then $\mathrm{Adj}(XId - M) = f \cdot (PU) \cdot ({}^T VP^{-1}) \mod \chi_M$.

Complexity and precision

In $\tilde{O}(n^3)$, but with divisions to compute the factorization (Extended Euclidean Algorithm). Enough for precision on every coefficient.

Table of contents

In practice, is it worth it?

Average precision loss on the characteristic polynomial of a random 9×9 matrix over $\mathbb{Q}_2$— results for a sample of 1000 instances.

	Average loss of accuracy	
	Optimal	Naïve, division-free
X^0 (det.)	3.17	196
X^1	2.98	161
X^2	2.75	129
X^3	2.74	108
X^4	2.57	63.2
X^5	2.29	51.6
X^6	2.07	9.04
X^7	1.64	5.70
X^8 (trace)	0.99	0.99

To sum up

On p-adic precision

To sum up

On p-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.

To sum up

On *p*-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.
- Differential calculus : **intrinsic** and can handle both **gain** and **loss**.

To sum up

On *p*-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.
- Differential calculus : **intrinsic** and can handle both **gain** and **loss**.

On characteristic polynomial: generic case

To sum up

On p-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.
- Differential calculus : **intrinsic** and can handle both **gain** and **loss**.

On characteristic polynomial: generic case

- Can know the **optimal** precision in $\tilde{O}(n^3)$ without division when starting from flat precision.

To sum up

On p-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.
- Differential calculus : **intrinsic** and can handle both **gain** and **loss**.

On characteristic polynomial: generic case

- Can know the **optimal** precision in $\tilde{O}(n^3)$ without division when starting from flat precision.
- Can know the **optimal** precision in $\tilde{O}(n^3)$ with few divisions when starting from jagged precision.

To sum up

On p-adic precision

- Step-by-step analysis : as a first step. Can show differentiability and naïve loss in precision during the computation.
- Differential calculus : **intrinsic** and can handle both **gain** and **loss**.

On characteristic polynomial: generic case

- Can know the **optimal** precision in $\tilde{O}(n^3)$ without division when starting from flat precision.
- Can know the **optimal** precision in $\tilde{O}(n^3)$ with few divisions when starting from jagged precision.
- If one allows (few) divisions, faster methods are possible.

References

Initial article

- Xavier Caruso, David Roe and Tristan Vaccon Tracking p-adic precision, ANTS XI, 2014.

Linear Algebra

- Xavier Caruso, David Roe and Tristan Vaccon p-adic Precision in Linear Algebra, ISSAC 2015.

Characteristic polynomial

- Xavier Caruso, David Roe and Tristan Vaccon Characteristic polynomials of p-adic matrices, ISSAC 2017.

Thank you for your attention

Thanks