p-adic precision and isogeny computation

Applications to cryptography

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

Univ.Rennes 1, TU Berlin, Univ. Pittsburgh, 立教大学

July 5th, 2016

Motivation for isogeny computations

Study of elliptic curves

- Isogenies are "morphisms" between elliptic curves ;

Motivation for isogeny computations

Study of elliptic curves

- Isogenies are "morphisms" between elliptic curves ;
- Relationship between curves: yields point-counting algorithms.

Motivation for isogeny computations

Study of elliptic curves

- Isogenies are "morphisms" between elliptic curves ;
- Relationship between curves: yields point-counting algorithms.

Cryptosytems

- De Fao, Jao and Plût have proposed cryptosystems based on isogenies between elliptic curves

Why should one work with p-adic numbers ?

p-adic methods

- Working in $\mathbb{Q}_p$ instead of $\mathbb{Q}$, one can handle more efficiently the coefficients growth ;

Why should one work with p-adic numbers ?

p-adic methods

- Working in $\mathbb{Q}_p$ instead of $\mathbb{Q}$, one can handle more efficiently the coefficients growth ;

p-adic algorithms

- Going from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$ and then back to $\mathbb{Z}/p\mathbb{Z}$ enables more computation ;

Why should one work with p-adic numbers ?

p-adic methods

- Working in $\mathbb{Q}_p$ instead of $\mathbb{Q}$, one can handle more efficiently the coefficients growth ;

p-adic algorithms

- Going from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$ and then back to $\mathbb{Z}/p\mathbb{Z}$ enables more computation ;
- Counting-points algorithms: Satoh, SEA, Kedlaya, ...

Why should one work with p-adic numbers ?

p-adic methods

- Working in $\mathbb{Q}_p$ instead of $\mathbb{Q}$, one can handle more efficiently the coefficients growth ;

p-adic algorithms

- Going from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$ and then back to $\mathbb{Z}/p\mathbb{Z}$ enables more computation ;
- Counting-points algorithms: Satoh, SEA, Kedlaya, ...

My personal (long-term) motivation

Computing (some) moduli spaces of p-adic Galois representations.

Background

Disclaimer

Background

Disclaimer

- I am **not** an expert in cryptography.

Background

Disclaimer

- I am **not** an expert in cryptography.
- However, one of my goal today is to present **tools** that can be useful for **cryptography** and **computer algebra**: isogenies and p-adic numbers.

Table of contents

What is... an isogeny?

Definition

We can define an isogeny between two elliptic curves E_1 and E_2 to be at the same time:

- a rational map $E_1 \to E_2$;
- a group morphism $E_1 \to E_2$.

Isogeny and quotient

Proposition

*Every isogeny is either **zero** or **surjective**.*

Isogeny and quotient

Proposition

*Every isogeny is either **zero** or **surjective**.*

Remark

All non-zero isogenies corresponds to taking some quotient:

$$E \twoheadrightarrow E/H.$$

Toward point-counting

Why point-counting on elliptic curves?

For the **Elliptic Curve Discrete Logarithm Problem**, some cardinals should be avoided.

Toward point-counting

Why point-counting on elliptic curves?

For the **Elliptic Curve Discrete Logarithm Problem**, some cardinals should be avoided.

Using isogeny for point-counting

If $\Phi : E_1 \to E_2$ is non-zero then:

$$\sharp E_1 = \sharp E_2 + \sharp Ker(\Phi).$$

Further toward point-counting

Isogeny and kernel, Vélu's formula

For $\Phi : E_1 \to E_2$, Φ can be written in affine coordinates as:

$$\Phi(x, y) = \left(\frac{g(x)}{h(x)}, cy \left(\frac{g(x)}{h(x)} \right)' \right),$$

with g, h polynomials, c scalar.

Further toward point-counting

Isogeny and kernel, Vélu's formula

For $\Phi : E_1 \to E_2$, Φ can be written in affine coordinates as:

$$\Phi(x, y) = \left(\frac{g(x)}{h(x)}, cy\left(\frac{g(x)}{h(x)}\right)'\right),$$

with g, h polynomials, c scalar.

Remark

For x-coordinates:

$$Ker(\Phi) = \{\infty\} \cup \{ \text{ zeroes of h } \}$$

Further toward point-counting

Isogeny and kernel, Vélu's formula

For $\Phi : E_1 \to E_2$, Φ can be written in affine coordinates as:

$$\Phi(x, y) = \left(\frac{g(x)}{h(x)}, cy \left(\frac{g(x)}{h(x)} \right)' \right),$$

with g, h polynomials, c scalar.

Remark

For x-coordinates:

$$Ker(\Phi) = \{\infty\} \cup \{ \text{ zeroes of h } \}$$

Point-counting algorithms

Use isogenies between an elliptic curve E and other curves: twist by Frobenius, quotient by l-torsion.

Recall on Diffie-Hellman key exchange

Recall on Diffie-Hellman key exchange

Preparation

Public modulus p and generator g of $\mathbb{Z}/p\mathbb{Z}^\times$.

Recall on Diffie-Hellman key exchange

Preparation

Public modulus p and generator g of $\mathbb{Z}/p\mathbb{Z}^\times$.

Alice

- Choose an integer a.

Bob

- Choose an integer b.

Recall on Diffie-Hellman key exchange

Preparation

Public modulus p and generator g of $\mathbb{Z}/p\mathbb{Z}^\times$.

Alice

- Choose an integer a.
- Sends $A = g^a \mod p$ to Bob.

Bob

- Choose an integer b.
- Sends $B = g^b \mod p$ to Alice.

Recall on Diffie-Hellman key exchange

Preparation

Public modulus p and generator g of $\mathbb{Z}/p\mathbb{Z}^\times$.

Alice

- Choose an integer a.
- Sends $A = g^a \mod p$ to Bob.
- Computes $s = B^a \mod p$.

Bob

- Choose an integer b.
- Sends $B = g^b \mod p$ to Alice.
- Computes $s = A^b \mod p$.

Recall on Diffie-Hellman key exchange

Preparation

Public modulus p and generator g of $\mathbb{Z}/p\mathbb{Z}^\times$.

Alice

- Choose an integer a.
- Sends $A = g^a \mod p$ to Bob.
- Computes $s = B^a \mod p$.

Bob

- Choose an integer b.
- Sends $B = g^b \mod p$ to Alice.
- Computes $s = A^b \mod p$.

Shared information

$s = g^{ab} = B^a = A^b \mod p$.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Alice

- $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, one is inv.

Bob

- $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, one is inv.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Alice

- $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, one is inv.
- $\Phi_A : E_0 \to E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$.

Bob

- $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, one is inv.
- $\Phi_B : E_0 \to E_B = E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Alice

- $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, one is inv.
- $\Phi_A : E_0 \to E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$.
- Sends $E_A, \Phi_A(P_B), \Phi_A(Q_B)$.

Bob

- $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, one is inv.
- $\Phi_B : E_0 \to E_B = E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$.
- Sends $E_B, \Phi_B(P_A), \Phi_B(Q_A)$.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Alice

- $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, one is inv.
- $\Phi_A : E_0 \to E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$.
- Sends $E_A, \Phi_A(P_B), \Phi_A(Q_B)$.
- $\Psi_A : E_B \to E_{AB} = E_B/\langle [m_A]\Phi_B(P_A) + [n_A]\Phi_B(Q_A) \rangle$.

Bob

- $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, one is inv.
- $\Phi_B : E_0 \to E_B = E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$.
- Sends $E_B, \Phi_B(P_A), \Phi_B(Q_A)$.
- $\Psi_B : E_B \to E_{AB} = E_A/\langle [m_B]\Phi_A(P_B) + [n_B]\Phi_A(Q_B) \rangle$.

Elliptic-curve based key exchange: De Feo-Jao-Plût (2011)

Preparation

Elliptic curve $E_0/\mathbb{F}_{p^2}$, generators $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$, $E_0[l_b^{e_B}]$.

Alice

- $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, one is inv.
- $\Phi_A : E_0 \to E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$.
- Sends $E_A, \Phi_A(P_B), \Phi_A(Q_B)$.
- $\Psi_A : E_B \to E_{AB} = E_B/\langle [m_A]\Phi_B(P_A) + [n_A]\Phi_B(Q_A) \rangle$.

Bob

- $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, one is inv.
- $\Phi_B : E_0 \to E_B = E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$.
- Sends $E_B, \Phi_B(P_A), \Phi_B(Q_A)$.
- $\Psi_B : E_B \to E_{AB} = E_A/\langle [m_B]\Phi_A(P_B) + [n_B]\Phi_A(Q_B) \rangle$.

Shared information

$$E_{AB} = \Psi_B(\Phi_A(E_0)) = \Psi_A(\Phi_B(E_0)),$$

and its j-invariant $j(E_{AB})$.

Some remarks

Remark

Not all elliptic curves are safe for this scheme. *e.g.* supersingularity is a requirement in De Feo-Jao-Plût.

Some remarks

Remark

Not all elliptic curves are safe for this scheme. *e.g.* supersingularity is a requirement in De Feo-Jao-Plût.

Remark

Many variants: proof of identity, public-key encryption...

Some remarks

Remark

Not all elliptic curves are safe for this scheme. *e.g.* supersingularity is a requirement in De Feo-Jao-Plût.

Remark

Many variants: proof of identity, public-key encryption...

Remark

Candidate for **Post-Quantum Cryptography**.

How to compute isogenies?

How to compute isogenies?

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

How to compute isogenies?

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

Let us assume that there exists some normalized isogeny I between E and $\tilde{E}$. Then, for some rational fraction U,

$$I(x, y) = (U(x), yU'(x)),$$

How to compute isogenies?

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

Let us assume that there exists some normalized isogeny I between E and $\tilde{E}$. Then, for some rational fraction U,

$$I(x,y) = (U(x), yU'(x)),$$

Writing $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$, we get :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

Change of variable and the differential equation

The differential equation

Let S be such that $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$.

We have the following differential equation for S :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

Change of variable and the differential equation

The differential equation

Let S be such that $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$.
We have the following differential equation for S :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

It rewrites as:

$$y'^2 = g(x)h(y).$$

Change of variable and the differential equation

The differential equation

Let S be such that $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$.
We have the following differential equation for S :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

It rewrites as:

$$y'^2 = g(x)h(y).$$

Solving a differential equation in $\mathbb{Z}/p\mathbb{Z}$???

- **Not easy**:

$$\int X^{p-1} = \frac{1}{p}X^p?$$

Change of variable and the differential equation

The differential equation

Let S be such that $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$.

We have the following differential equation for S :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

It rewrites as:

$$y'^2 = g(x)h(y).$$

Solving a differential equation in $\mathbb{Z}/p\mathbb{Z}$???

- **Not easy**:

$$\int X^{p-1} = \frac{1}{p}X^p?$$

- We would like to be in **zero characteristic**: let's go **p-adic**!

Table of contents

Norms over a field

Definition

A norm over a field K is a mapping $|\cdot| : K \to \mathbb{R}_+, x \mapsto |x|$ such that :

$$(i)\ |x| = 0 \Leftrightarrow x = 0;$$
$$(ii)\ |xy| = |x||y|;$$
$$(iii)\ |x+y| \leqslant |x| + |y|.$$

Norms over a field

Definition

A norm over a field K is a mapping $|\cdot| : K \to \mathbb{R}_+, x \mapsto |x|$ such that :

$$(i)\ |x| = 0 \Leftrightarrow x = 0;$$
$$(ii)\ |xy| = |x||y|;$$
$$(iii)\ |x+y| \leqslant |x| + |y|.$$

It is called **ultrametric** if :

$$(iii')\ |x+y| \leqslant \sup(|x|, |y|).$$

Norms over a vector space

Definition

Let K be a normed field. A norm over a K-vector space E is a mapping $\|\cdot\| : E \to \mathbb{R}_+, x \mapsto \|x\|$ such that :

$$(i)\ \|x\| = 0 \Leftrightarrow x = 0;$$
$$(ii)\ \|\alpha y\| = |\alpha|\|y\|;$$
$$(iii)\ \|x + y\| \leqslant \|x\| + \|y\|.$$

Norms over a vector space

Definition

Let K be a normed field. A norm over a K-vector space E is a mapping $\|\cdot\| : E \to \mathbb{R}_+, x \mapsto \|x\|$ such that :

$$(i)\ \|x\| = 0 \Leftrightarrow x = 0;$$
$$(ii)\ \|\alpha y\| = |\alpha|\|y\|;$$
$$(iii)\ \|x + y\| \leqslant \|x\| + \|y\|.$$

It is called **ultrametric** if :

$$(iii')\ \|x + y\| \leqslant \sup(\|x\|, \|y\|).$$

An ultrametric norm over $\mathbb{Q}$

Definition

For any p prime in $\mathbb{N}$, we define the p-adic valuation $v_p : \mathbb{Q} \to \mathbb{Z}$ by :

An ultrametric norm over $\mathbb{Q}$

Definition

For any p prime in $\mathbb{N}$, we define the p-adic valuation $v_p : \mathbb{Q} \to \mathbb{Z}$ by :

$$v_p(n) = \max \left\{ k \in \mathbb{N},\ p^k \mid n \right\},$$

An ultrametric norm over $\mathbb{Q}$

Definition

For any p prime in $\mathbb{N}$, we define the p-adic valuation $v_p : \mathbb{Q} \to \mathbb{Z}$ by :

$$v_p(n) = \max\left\{k \in \mathbb{N},\ p^k \mid n\right\},$$
$$v_p\left(\frac{a}{b}\right) = v_p(a) - v_p(b).$$

An ultrametric norm over $\mathbb{Q}$

Definition

For any p prime in $\mathbb{N}$, we define the p-adic valuation $v_p : \mathbb{Q} \to \mathbb{Z}$ by :

$$v_p(n) = \max \left\{ k \in \mathbb{N},\ p^k \mid n \right\},$$
$$v_p\left(\frac{a}{b}\right) = v_p(a) - v_p(b).$$

Definition

For any $x \in \mathbb{Q}$,

$$|x|_p = p^{-v_p(x)}.$$

An ultrametric norm over $\mathbb{Q}$

Definition

For any p prime in $\mathbb{N}$, we define the p-adic valuation $v_p : \mathbb{Q} \to \mathbb{Z}$ by :

$$v_p(n) = \max \left\{ k \in \mathbb{N},\ p^k \mid n \right\},$$
$$v_p\left(\frac{a}{b}\right) = v_p(a) - v_p(b).$$

Definition

For any $x \in \mathbb{Q}$,

$$|x|_p = p^{-v_p(x)}.$$

$|\cdot|_p$ is an **ultrametric norm** over $\mathbb{Q}$.

Norms over $\mathbb{Q}$.

Theorem (Ostrowski)

Norms over $\mathbb{Q}$.

Theorem (Ostrowski)

Up to equivalence, the only norms over $\mathbb{Q}$ are $|\cdot|$ and the $|\cdot|_p$.

Norms over $\mathbb{Q}$.

Theorem (Ostrowski)

Up to equivalence, the only norms over $\mathbb{Q}$ are $|\cdot|$ and the $|\cdot|_p$.

Remark

$$|p^k|_p = p^{-k},$$

Norms over $\mathbb{Q}$.

Theorem (Ostrowski)

Up to equivalence, the only norms over $\mathbb{Q}$ are $|\cdot|$ and the $|\cdot|_p$.

Remark

$$|p^k|_p = p^{-k},$$

Therefore,

$$|p^k|_p \to_{k\to+\infty} 0.$$

Norms over $\mathbb{Q}$.

Theorem (Ostrowski)

Up to equivalence, the only norms over $\mathbb{Q}$ are $|\cdot|$ and the $|\cdot|_p$.

Remark

$$|p^k|_p = p^{-k},$$

Therefore,

$$|p^k|_p \rightarrow_{k \rightarrow +\infty} 0.$$

Remark

$\mathbb{Q}$ is complete for none of these norms.

Definition of $\mathbb{Q}_p$

Definition

We define $\mathbb{Q}_p$ as "the" **completion** of $\mathbb{Q}$ for $|\cdot|_p$.

Definition of $\mathbb{Q}_p$

Definition

We define $\mathbb{Q}_p$ as "the" **completion** of $\mathbb{Q}$ for $|\cdot|_p$.
$|\cdot|_p$ and v_p extend to $\mathbb{Q}_p$

Definition of $\mathbb{Q}_p$

Definition

We define $\mathbb{Q}_p$ as "the" **completion** of $\mathbb{Q}$ for $|\cdot|_p$.
$|\cdot|_p$ and v_p extend to $\mathbb{Q}_p$ $|\cdot|_p$ is still **ultrametric**.

Definition

We write :

$$\mathbb{Z}_p = \{x \in \mathbb{Q}_p,\ v_p(x) \geq 0\} = \{x \in \mathbb{Q}_p,\ |x|_p \leq 1\} = B'_{\mathbb{Q}_p}(0,1).$$

Definition of $\mathbb{Q}_p$

Definition

We define $\mathbb{Q}_p$ as "the" **completion** of $\mathbb{Q}$ for $|\cdot|_p$.
$|\cdot|_p$ and v_p extend to $\mathbb{Q}_p$ $|\cdot|_p$ is still **ultrametric**.

Definition

We write :

$$\mathbb{Z}_p = \{x \in \mathbb{Q}_p,\ v_p(x) \geq 0\} = \{x \in \mathbb{Q}_p,\ |x|_p \leq 1\} = B'_{\mathbb{Q}_p}(0,1).$$

$\mathbb{Z}_p$ is a sub-ring of $\mathbb{Q}_p$.

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write

$$x = \sum_{i \geq 0}^{+\infty} a_i p^i .$$

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write
$$x = \sum_{i \geq 0}^{+\infty} a_i p^i.$$
If $x \in \mathbb{Q}_p$, we can write
$$x = \frac{1}{p^k} \sum_{i \geq 0}^{+\infty} a_i p^i.$$

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write
$$x = \sum_{i \geq 0}^{+\infty} a_i p^i.$$

If $x \in \mathbb{Q}_p$, we can write
$$x = \frac{1}{p^k} \sum_{i \geq 0}^{+\infty} a_i p^i.$$

Example

Let $x = \ldots 6666666_7 = \sum_{i \geq 0}^{+\infty} 6 \times 7^i$.

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write

$$x = \sum_{i \geq 0}^{+\infty} a_i p^i.$$

If $x \in \mathbb{Q}_p$, we can write

$$x = \frac{1}{p^k} \sum_{i \geq 0}^{+\infty} a_i p^i.$$

Example

Let $x = \ldots 6666666_7 = \sum_{i \geq 0}^{+\infty} 6 \times 7^i$. Then $x \in \mathbb{Z}_7$. We remark that $x + 1 = 0$.

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write

$$x = \sum_{i \geq 0}^{+\infty} a_i p^i.$$

If $x \in \mathbb{Q}_p$, we can write

$$x = \frac{1}{p^k} \sum_{i \geq 0}^{+\infty} a_i p^i.$$

Example

Let $x = \ldots 6666666_7 = \sum_{i \geq 0}^{+\infty} 6 \times 7^i$. Then $x \in \mathbb{Z}_7$. We remark that $x + 1 = 0$.
We also have :

$$\ldots 44445_7 = \frac{1}{3} \in \mathbb{Z}_7,$$

Working with $\mathbb{Q}_p$

Proposition

If $x \in \mathbb{Z}_p$, we can write
$$x = \sum_{i \geq 0}^{+\infty} a_i p^i.$$
If $x \in \mathbb{Q}_p$, we can write
$$x = \frac{1}{p^k} \sum_{i \geq 0}^{+\infty} a_i p^i.$$

Example

Let $x = \ldots 6666666_7 = \sum_{i \geq 0}^{+\infty} 6 \times 7^i$. Then $x \in \mathbb{Z}_7$. We remark that $x + 1 = 0$.
We also have :
$$\ldots 44445_7 = \frac{1}{3} \in \mathbb{Z}_7,$$
And,
$$\ldots 4444, 6_7 = \frac{4}{21}.$$

Topology and ultrametricity

Proposition

- $\mathbb{Z}_p$ *is* ***both*** *open and closed in* $\mathbb{Q}_p$.

Topology and ultrametricity

Proposition

- $\mathbb{Z}_p$ *is **both** open and closed in* $\mathbb{Q}_p$. $\mathbb{Q}_p$ *is totally discontinuous.*

Topology and ultrametricity

Proposition

- *$\mathbb{Z}_p$ is **both** open and closed in $\mathbb{Q}_p$. $\mathbb{Q}_p$ is totally discontinuous.*
- *$\mathbb{Z}$ is a **dense** subset of $\mathbb{Z}_p$.*

Topology and ultrametricity

Proposition

- $\mathbb{Z}_p$ *is* ***both*** *open and closed in* $\mathbb{Q}_p$*.* $\mathbb{Q}_p$ *is totally discontinuous.*
- $\mathbb{Z}$ *is a* ***dense*** *subset of* $\mathbb{Z}_p$*.* $\mathbb{Q}$ *is a* ***dense*** *subset of* $\mathbb{Q}_p$*.*

Topology and ultrametricity

Proposition

- *$\mathbb{Z}_p$ is **both** open and closed in $\mathbb{Q}_p$. $\mathbb{Q}_p$ is totally discontinuous.*
- *$\mathbb{Z}$ is a **dense** subset of $\mathbb{Z}_p$. $\mathbb{Q}$ is a **dense** subset of $\mathbb{Q}_p$.*

Proposition

*If E is an ultrametric vector space, then **any** point in a ball of E is its **center**.*

Summary and algebraic point of view

Proposition

$$\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}.$$

Summary and algebraic point of view

Proposition

$$\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}.$$

$$\forall k \in \mathbb{N}, \mathbb{Z}_p/p^k\mathbb{Z}_p = \mathbb{Z}/p^k\mathbb{Z}.$$

Summary and algebraic point of view

Proposition

$$\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}.$$

$$\forall k \in \mathbb{N}, \mathbb{Z}_p/p^k\mathbb{Z}_p = \mathbb{Z}/p^k\mathbb{Z}.$$

A first idea

- $\mathbb{Q}_p$ is an extension of $\mathbb{Q}$ where one can perform **calculus**, as simply as over $\mathbb{R}$.

Summary and algebraic point of view

Proposition

$$\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}.$$

$$\forall k \in \mathbb{N}, \mathbb{Z}_p/p^k\mathbb{Z}_p = \mathbb{Z}/p^k\mathbb{Z}.$$

A first idea

- $\mathbb{Q}_p$ is an extension of $\mathbb{Q}$ where one can perform **calculus**, as simply as over $\mathbb{R}$.
- We are **closer to arithmetic** : we can reduce modulo p.

Summary and algebraic point of view

Proposition

$$\mathbb{Z}_p/p\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}.$$

$$\forall k \in \mathbb{N}, \mathbb{Z}_p/p^k\mathbb{Z}_p = \mathbb{Z}/p^k\mathbb{Z}.$$

A first idea

- $\mathbb{Q}_p$ is an extension of $\mathbb{Q}$ where one can perform **calculus**, as simply as over $\mathbb{R}$.
- We are **closer to arithmetic** : we can reduce modulo p.

Remark

$$\begin{tikzcd} \mathbb{Q}_p \arrow[dr, dash] & & \mathbb{R} \arrow[dl, dash] \\ & \mathbb{Q} & \end{tikzcd}$$

$$\begin{tikzcd} \mathbb{Z}_p \arrow[r] \arrow[d] & \mathbb{Z}_p \arrow[d] \\ \mathbb{Z}/p\mathbb{Z} \arrow[r] & \mathbb{Z}/p\mathbb{Z} \end{tikzcd}$$

Table of contents

Definition of the precision

Finite-precision p-adics

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

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

Definition of the precision

Finite-precision p-adics

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

Definition of the precision

Finite-precision p-adics

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

Definition

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

Definition of the precision

Finite-precision p-adics

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

Definition

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

Example

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

p-adic precion *vs* real precision

The quintessential idea of the step-by-step analysis is the following :

Proposition (*p*-adic errors don't add)

Indeed,

$$(a + O(p^k)) + (b + O(p^k)) = a + b + O(p^k).$$

That is to say, if a and b are known up to precision $O(p^k)$, then so is $a + b$.

p-adic precion *vs* real precision

The quintessential idea of the step-by-step analysis is the following :

Proposition (p-adic errors don't add)

Indeed,

$$(a + O(p^{k})) + (b + O(p^{k})) = a + b + O(p^{k}).$$

That is to say, if a and b are known up to precision $O(p^k)$, then so is $a + b$.

p-adic precion *vs* real precision

The quintessential idea of the step-by-step analysis is the following :

Proposition (p-adic errors don't add)

Indeed,

$$(a + O(p^{\boxed{k}})) + (b + O(p^{\boxed{k}})) = a + b + O(p^{\boxed{k}}).$$

That is to say, if a and b are known up to precision $O(p^k)$, then so is $a + b$.

Remark

It is quite the opposite to when dealing with real numbers, because of **Round-off error** :

$$(1 + 5 * 10^{-2}) + (2 + 6 * 10^{-2}) = 3 + 1 * 10^{-1} + 1 * 10^{-2}.$$

That is to say, if a and b are known up to precision 10^{-n}, then $a + b$ is known up to $10^{(-n\ +1\ )}$.

p-adic precion *vs* real precision

The quintessential idea of the step-by-step analysis is the following :

Proposition (p-adic errors don't add)

Indeed,

$$(a + O(p^{\boxed{k}})) + (b + O(p^{\boxed{k}})) = a + b + O(p^{\boxed{k}}).$$

That is to say, if a and b are known up to precision $O(p^k)$, then so is $a + b$.

Remark

It is quite the opposite to when dealing with real numbers, because of **Round-off error** :

$$(1 + 5 * 10^{-2}) + (2 + 6 * 10^{-2}) = 3 + 1 * 10^{-1} + 1 * 10^{-2}.$$

That is to say, if a and b are known up to precision 10^{-n}, then $a + b$ is known up to $10^{(-n \boxed{+1})}$.

Precision formulae

Proposition (addition)

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

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,

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

Table of contents

A little warm-up on computing determinants : question

An example of determinant computation

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

A little warm-up on computing determinants : question

An example of determinant computation

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

Question

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

A little warm-up on computing determinants : question

Another 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 : question

Another example of determinant computation

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

Same question

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

A little warm-up on computing determinants : expansion

An example of determinant computation

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

A little warm-up on computing determinants : expansion

An example of determinant computation

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

Direct expansion

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

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

Direct expansion

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

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

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

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

An example of determinant computation

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

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

An example of determinant computation

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

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

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

Row-echelon form computation

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

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

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

A little warm-up on computing determinants : SNF

An example of determinant computation

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

A little warm-up on computing determinants : SNF

An example of determinant computation

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

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

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

Smith Normal Form (SNF) computation

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

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

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

Summary: precision and p-adic computations

Direct method for precision

Summary: precision and p-adic computations

Direct method for precision

- Has often been enough to get a first view of the problem.

Summary: precision and p-adic computations

Direct method for precision

- Has often been enough to get a first view of the problem.
- Depends heavily on the algorithm chosen for the computation

Summary: precision and p-adic computations

Direct method for precision

- Has often been enough to get a first view of the problem.
- Depends heavily on the algorithm chosen for the computation
- No idea on what is **optimal**.

Table of contents

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

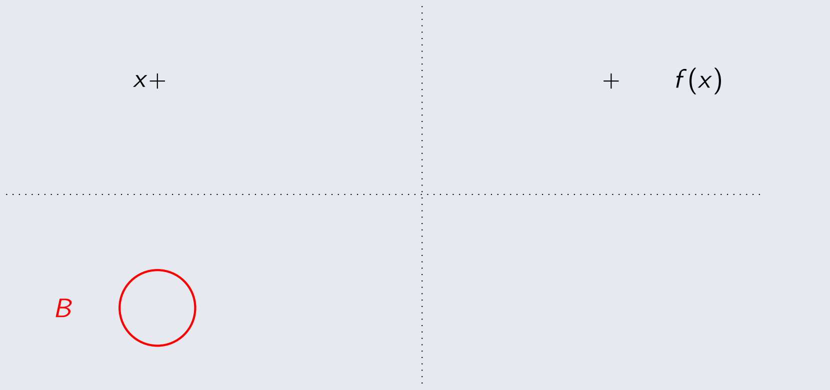
Lemma (CRV14)

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****,*

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

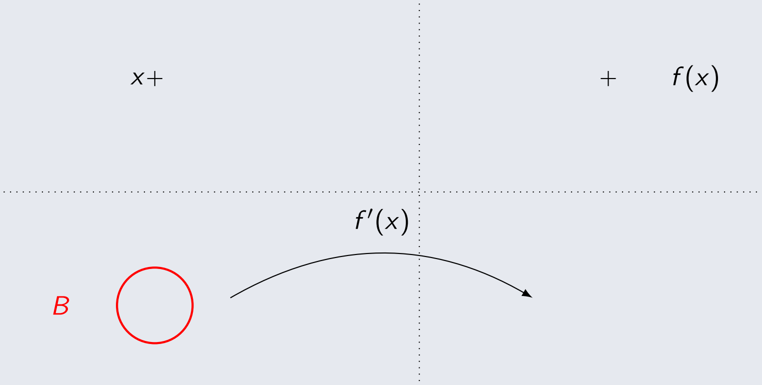
Geometrical meaning

Interpretation



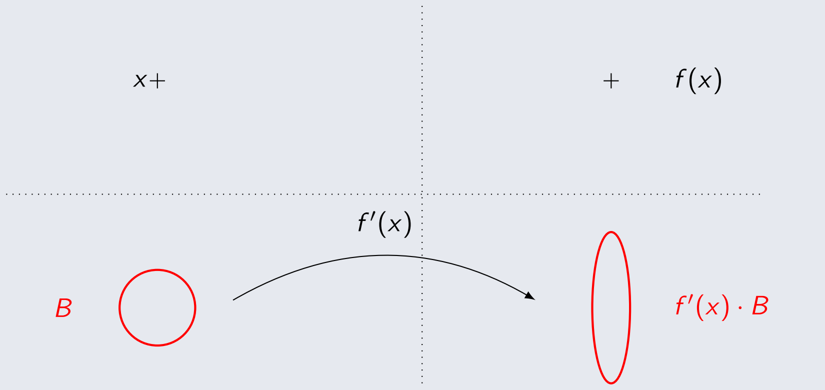
Geometrical meaning

Interpretation



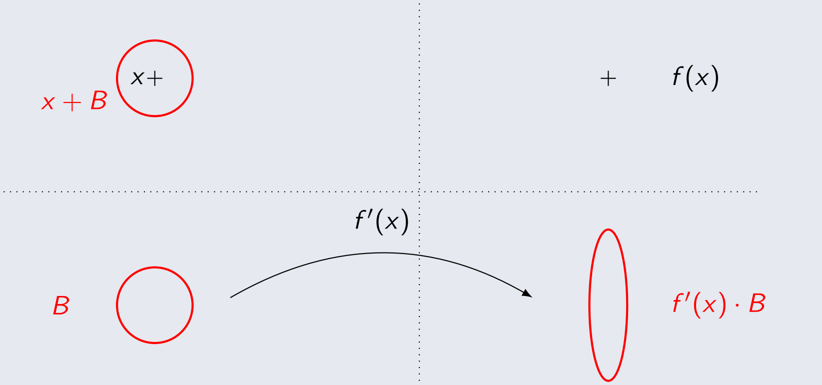
Geometrical meaning

Interpretation



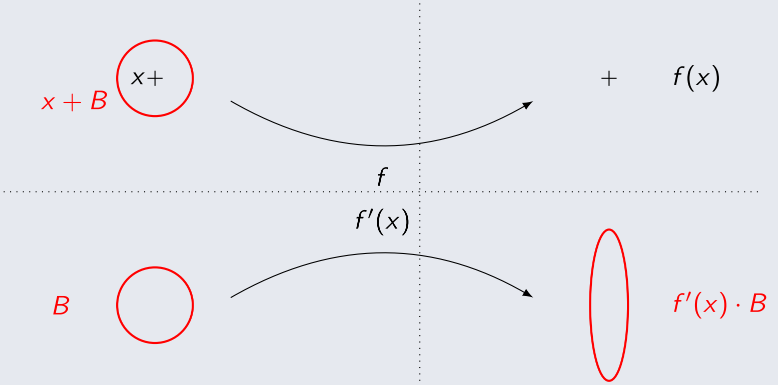
Geometrical meaning

Interpretation



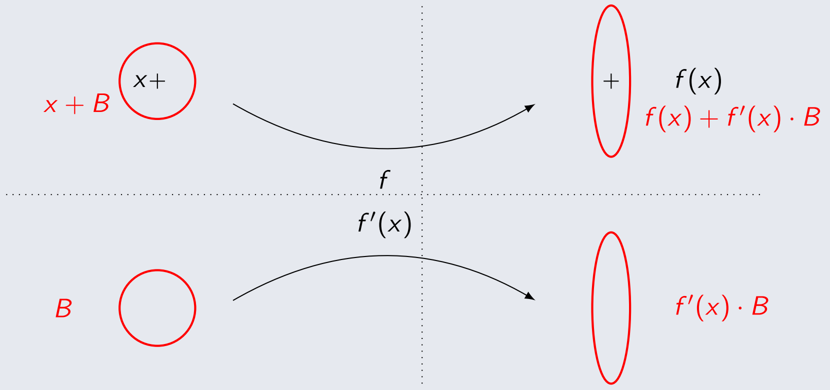
Geometrical meaning

Interpretation



Geometrical meaning

Interpretation



Lattices

Lattices

Lemma

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****,*

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

Lattices

Lemma

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****, for any open* ***lattice*** *$H \subset B$*

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

Lattices

Lemma

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****, for any open* ***lattice*** *$H \subset B$*

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

Remark

This allows more models of precision, like

$$(x, y) = (1 + O(p^{10}), 1 + O(p)).$$

Lattices

Lemma

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****, for any open* ***lattice*** *$H \subset B$*

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

Remark

This allows more models of precision, like

$$(x, y) = (1 + O(p^{10}), 1 + O(p)).$$

Remark

Our framework can be extended to **(complete) ultrametric K-vector spaces** (*e.g.* being $\mathbb{F}_p((X))^n$, $\mathbb{Q}((X))^m$, $\mathbb{R}((\varepsilon))^s$).

Higher differentials

Higher differentials

What is small enough ?

How can we determine when the lemma applies ?

Higher differentials

What is **small enough** ?

How can we determine when the lemma applies ?
When f is locally analytic, it essentially corresponds to

$$\sum_{k=2}^{+\infty} \frac{1}{k!} f^{(k)}(x) \cdot H^k \subset f'(x) \cdot H.$$

Higher differentials

What is **small enough** ?

How can we determine when the lemma applies ?
When f is locally analytic, it essentially corresponds to

$$\sum_{k=2}^{+\infty} \frac{1}{k!} f^{(k)}(x) \cdot H^k \subset f'(x) \cdot H.$$

This can be determined with **Newton-polygon** techniques.

Looking back to the case of the determinant

Differential of the determinant

It is well known:

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

Looking back to the case of the determinant

Differential of the determinant

It is well known:

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

Consequence on precision

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

Looking back to the case of the determinant

Differential of the determinant

It is well known:

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

Consequence on precision

- Loss in precision: coefficient of $\mathrm{Com}(M)$ 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:

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

Consequence on precision

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

Table of contents

An example of p-adic algorithm

An example of p-adic algorithm

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

An example of p-adic algorithm

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

Let us assume that there exists some normalized isogeny I between E and $\tilde{E}$. Then, for some rational fraction U,

$$I(x, y) = (U(x), yU'(x)),$$

An example of p-adic algorithm

Isogeny and Differential equations (*cf* Schoof-Elkies-Atkin algorithm, Bostan-Morain-Salvy-Schost 08, Lercier-Sirvent 08, ...)

Let E and $\tilde{E}$ be two elliptic curves over $\mathbb{Z}/p\mathbb{Z}$:

$$\begin{aligned} E &: y^2 = x^3 + Ax + B, \\ \tilde{E} &: y^2 = x^3 + \tilde{A}x + \tilde{B}. \end{aligned}$$

Let us assume that there exists some normalized isogeny I between E and $\tilde{E}$. Then, for some rational fraction U,

$$I(x, y) = (U(x), yU'(x)),$$

Writing $U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}$, we get :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

Change of variable and the differential equation

The differential equation

Let S be such that

$$U = \frac{1}{S(\frac{1}{\sqrt{x}})^2}.$$

Then if $A, B, \tilde{A}, \tilde{B}$ are in $\mathbb{Z}_p$,

$$S \in \mathbb{Z}_p[[t]]$$

We have the following differential equation for S :

$$(Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6.$$

A p-adic computation of a solution

Computing the isogeny

Given E and $\widetilde{E}$, the goal is to compute the isogeny I *via* the differential equation:

$$\begin{cases} S(0) = 0, \\ (Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6. \end{cases}$$

Going through $\mathbb{Z}_p$

Not easy to solve a differential equation in $\mathbb{Z}/p\mathbb{Z}$.

A p-adic computation of a solution

Computing the isogeny

Given E and $\widetilde{E}$, the goal is to compute the isogeny I *via* the differential equation:

$$\begin{cases} S(0) = 0, \\ (Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6. \end{cases}$$

Going through $\mathbb{Z}_p$

Not easy to solve a differential equation in $\mathbb{Z}/p\mathbb{Z}$. Consequently:

A p-adic computation of a solution

Computing the isogeny

Given E and $\widetilde{E}$, the goal is to compute the isogeny I *via* the differential equation:

$$\begin{cases} S(0) = 0, \\ (Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6. \end{cases}$$

Going through $\mathbb{Z}_p$

Not easy to solve a differential equation in $\mathbb{Z}/p\mathbb{Z}$. Consequently:

1. Lift (consistently) from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$.

A p-adic computation of a solution

Computing the isogeny

Given E and $\widetilde{E}$, the goal is to compute the isogeny I *via* the differential equation:

$$\begin{cases} S(0) = 0, \\ (Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6. \end{cases}$$

Going through $\mathbb{Z}_p$

Not easy to solve a differential equation in $\mathbb{Z}/p\mathbb{Z}$. Consequently:

1. Lift (consistently) from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$.
2. Solve the differential equation in $\mathbb{Z}_p$.

A p-adic computation of a solution

Computing the isogeny

Given E and $\widetilde{E}$, the goal is to compute the isogeny I *via* the differential equation:

$$\begin{cases} S(0) = 0, \\ (Bx^6 + Ax^4 + 1)S'^2 = 1 + \tilde{A}S^4 + \tilde{B}S^6. \end{cases}$$

Going through $\mathbb{Z}_p$

Not easy to solve a differential equation in $\mathbb{Z}/p\mathbb{Z}$. Consequently:

1. Lift (consistently) from $\mathbb{Z}/p\mathbb{Z}$ to $\mathbb{Z}_p$.
2. Solve the differential equation in $\mathbb{Z}_p$.
3. Reduce mod p to get the solution in $\mathbb{Z}/p\mathbb{Z}$.

Table of contents

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Direct analysis

Newton scheme to solve $y' = g \times h(y)$:

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Direct analysis

Newton scheme to solve $y' = g \times h(y)$:

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right).$$

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Direct analysis

Newton scheme to solve $y' = g \times h(y)$:

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right).$$

Remark

$$\int O(p^m) x^k = \frac{O(p^m)}{k+1} x^{k+1}.$$

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Direct analysis

Newton scheme to solve $y' = g \times h(y)$:

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right).$$

Remark

$$\int O(p^m) x^k = \frac{O(p^m)}{k+1} x^{k+1}.$$

One loses $O(N)$ digits at each step, for N the order of truncation.

Change of equation

When $p \neq 2$, we can replace $y'^2 \times G = H(y)$ by $y' = g \times h(y)$ with $g, h \in \mathbb{Z}_p^\times$.

Direct analysis

Newton scheme to solve $y' = g \times h(y)$:

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right).$$

Remark

$$\int O(p^m) x^k = \frac{O(p^m)}{k+1} x^{k+1}.$$

One loses $O(N)$ digits at each step, for N the order of truncation.
To compute $y \mod x^{2^N+1}$, we need an initial precision of $O(N^2)$ digits.

Table of contents

Differential and differential equation

Theorem

Let $\Phi : (g, h) \mapsto y$ *such that* $y(0) = 0$ *and* $y' = gh(y)$*. Then,*

$$\Phi'(g, h) \cdot (\delta g, \delta h) = h(y) \int \delta g + \frac{g \delta h(y)}{h(y)}.$$

Differential and differential equation

Theorem

Let $\Phi : (g,h) \mapsto y$ *such that* $y(0) = 0$ *and* $y' = gh(y)$. *Then,*

$$\Phi'(g,h) \cdot (\delta g, \delta h) = h(y) \int \delta g + \frac{g \delta h(y)}{h(y)}.$$

Proposition

In our case, $p \neq 2$, $y, g, h \in \mathbb{Z}_p[\![x]\!]$, $g(0) = h(0) = 1$. *If* $\delta g = \delta h = O(p^k)$, *then*

Differential and differential equation

Theorem

Let $\Phi : (g,h) \mapsto y$ *such that* $y(0) = 0$ *and* $y' = gh(y)$. *Then,*

$$\Phi'(g,h) \cdot (\delta g, \delta h) = h(y) \int \delta g + \frac{g \delta h(y)}{h(y)}.$$

Proposition

In our case, $p \neq 2$, $y, g, h \in \mathbb{Z}_p[\![x]\!]$, $g(0) = h(0) = 1$. *If* $\delta g = \delta h = O(p^k)$, *then*

$$\Phi'(y) \cdot (\delta g, \delta h) \mod x^{2^N+1} \in \frac{O(p^k)}{p^N} \mathbb{Z}_p[\![x]\!].$$

First conclusion on the application of the lemma

Proposition

$\Phi(g,h) \mod (p, t^{2^n})$ *is determined by* $g, h \mod (p^{1+\log_p 2^n}, t^{2^n})$. *In other words, we have a logarithmic loss in precision.*

What happens in practice ?

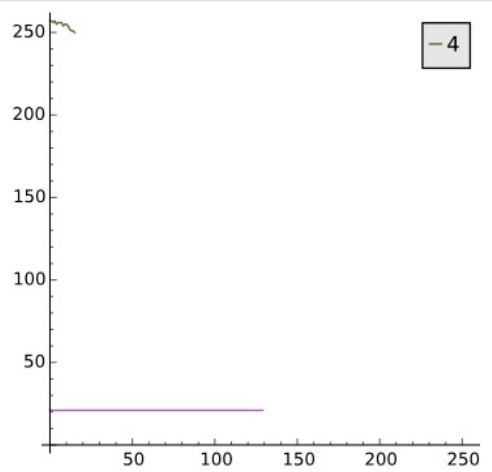


Figure: Precision over the output

What happens in practice ?

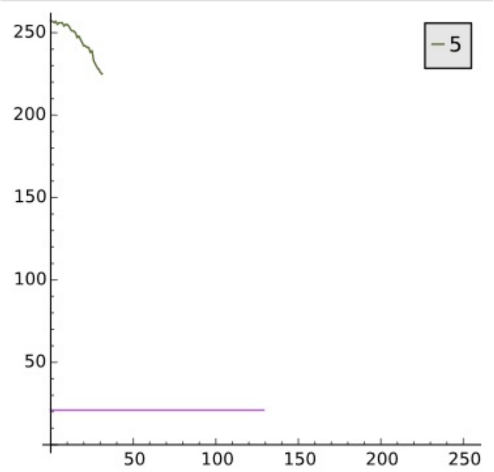


Figure: Precision over the output

What happens in practice ?

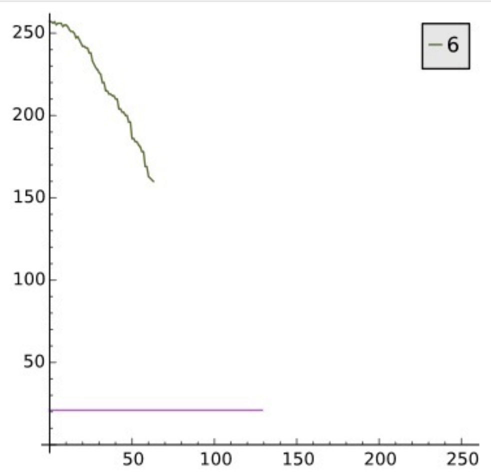


Figure: Precision over the output

What happens in practice ?

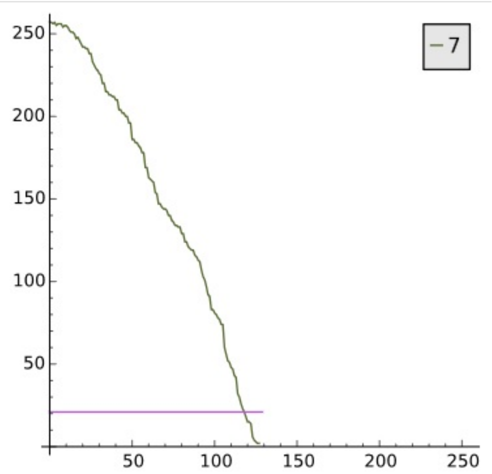


Figure: Precision over the output

Table of contents

Different way of representing the p-adics

Another take on the computation

Different way of representing the p-adics

Another take on the computation

- In the previous computation, we start with some given approximations of g, h, u_0 and try **to follow** the algorithm for the exact counterparts of g, h, u_0.

Different way of representing the p-adics

Another take on the computation

- In the previous computation, we start with some given approximations of g, h, u_0 and try **to follow** the algorithm for the exact counterparts of g, h, u_0. This is somehow **much stronger** than our desire: computing a good approximate solution.

Different way of representing the p-adics

Another take on the computation

- In the previous computation, we start with some given approximations of g, h, u_0 and try **to follow** the algorithm for the exact counterparts of g, h, u_0. This is somehow **much stronger** than our desire: computing a good approximate solution.
- Another way is then to modify the current g, h, u_0 **at each step**, in a consistent way, so as to keep on getting better approximate solutions.

Different way of representing the p-adics

Another take on the computation

- In the previous computation, we start with some given approximations of g, h, u_0 and try **to follow** the algorithm for the exact counterparts of g, h, u_0. This is somehow **much stronger** than our desire: computing a good approximate solution.
- Another way is then to modify the current g, h, u_0 **at each step**, in a consistent way, so as to keep on getting better approximate solutions.
- A third way here will be to work entirely in $\mathbb{Z}/p^\kappa\mathbb{Z}$.

New framework

In this new computation, we consider h as given, and not varying for the lemma.

Lemma

Let $Y : g \mapsto y$ such that $y(0) = 0$ and $y' = gh(y)$. Then,

$$Y'(g) \cdot (\delta g) = h(y) \int \delta g.$$

In concrete terms

A new take on the iteration

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $g_0 = g \mod p^k$

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $u_2' = g_2 h(u_2) \mod (p^k, t^4)$

- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$
- $g_2 = g_1 \mod p^k$

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $u_2' = g_2 h(u_2) \mod (p^k, t^4)$
- $\ldots$

- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$
- $g_2 = g_1 \mod p^k$
- $\ldots$

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $u_2' = g_2 h(u_2) \mod (p^k, t^4)$
- ...
- ...

- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$
- $g_2 = g_1 \mod p^k$
- ...
- ...

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $u_2' = g_2 h(u_2) \mod (p^k, t^4)$
- $\ldots$
- $\ldots$
- $u_l' = g_l h(u_l) \mod (p^k, t^{2^l})$

- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$
- $g_2 = g_1 \mod p^k$
- $\ldots$
- $\ldots$
- $g_l = g_{l-1} \mod p^k$

In concrete terms

A new take on the iteration

- $u_0' = g_0 h(u_0) \mod (p^k, t^1)$
- $u_1' = g_1 h(u_1) \mod (p^k, t^2)$
- $u_2' = g_2 h(u_2) \mod (p^k, t^4)$
- ...
- ...
- $u_l' = g_l h(u_l) \mod (p^k, t^{2^l})$

- $g_0 = g \mod p^k$
- $g_1 = g_0 \mod p^k$
- $g_2 = g_1 \mod p^k$
- ...
- ...
- $g_l = g_{l-1} \mod p^k$

In the end

- $u_l' = g h(u_l) \mod (p^k, t^{2^l})$
- $g_l = g \mod p^k$

Final take on the Newton scheme

As a consequence, we can prove that it is harmless to work in $\mathbb{Z}/p^k\mathbb{Z}$ for our computation.

Proposition

We can obtain the solution $\Phi(g,h) \mod (p, t^{n+1})$ *knowing* $g, h \mod (p^{\lfloor \log_p n \rfloor + 1}, t^{n+1})$ *and applying the following iteration:*

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right),$$

Final take on the Newton scheme

As a consequence, we can prove that it is harmless to work in $\mathbb{Z}/p^k\mathbb{Z}$ for our computation.

Proposition

We can obtain the solution $\Phi(g,h) \mod (p, t^{n+1})$ *knowing* $g, h \mod (p^{\lfloor \log_p n \rfloor + 1}, t^{n+1})$ *and applying the following iteration:*

$$N_{g,h}(u) \leftarrow u - h(u) \int \left(\frac{u'}{h(u)} - g \right),$$

modulo $p^{\lfloor \log_p n \rfloor + 1}$ *and growing order of truncation.*

Timings

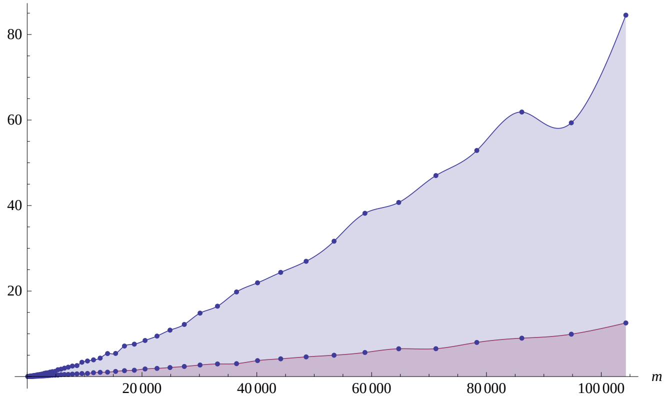


Figure: Timings in seconds, measured on a laptop, of our Algorithm run at precision λ_{old} (upper curve) and λ_{new} (lower curve) in order to compute an approximation modulo $(5, t^{4m+1})$ of some given m-isogenies.

Speedup

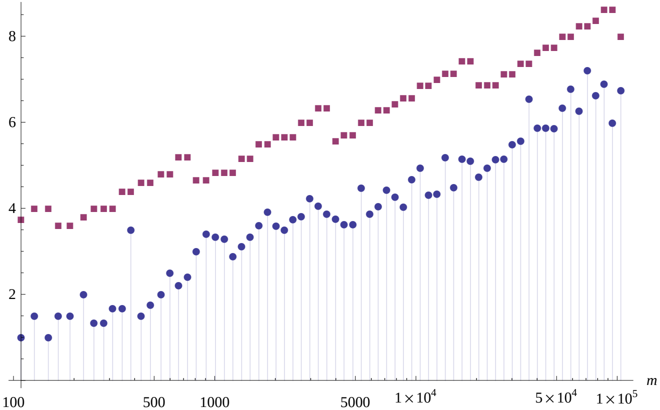


Figure: Practical speedup obtained with the new precision analysis compared with the theoretical improvement (m-axis in logarithmic scale). (■) is the ratio on precisions, (●) is the actual speedup.

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**.
- Can stabilize and attain **optimal** precision, even though naïve computation lose too much 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**.
- Can stabilize and attain **optimal** precision, even though naïve computation lose too much precision.

On differential equations

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**.
- Can stabilize and attain **optimal** precision, even though naïve computation lose too much precision.

On differential equations

- Can attain **optimal loss** in precision for differential equations with separation of variables.

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**.
- Can stabilize and attain **optimal** precision, even though naïve computation lose too much precision.

On differential equations

- Can attain **optimal loss** in precision for differential equations with separation of variables.
- Future works: higher order and $p = 2$.

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 stability in linear algebra, ISSAC 2015.

Differential equations

- Pierre Lairez and Tristan Vaccon On p-adic differential equations with separation of variables, ISSAC 2016.

Thank you for your attention

Thanks