Point Counting on Quasi-Diagonal Hypersurfaces

Henri Cohen
(Talk given by Atsuko Miyaji)

ECC, Osaka, November 2018

LFANT INRIA, IMB, Université de Bordeaux

Quasi-diagonal Hypersurfaces I

Let $\mathbb{F}_q$ be a finite field with $q = p^f$ elements for some prime p. A quasi-diagonal hypersurface V in $\mathbb{P}^{m-1}$ is a variety given by a projective equation

$$\sum_{1 \le i \le m} a_i x_i^m - b \prod_{1 \le i \le m} x_i = 0 ,$$

with $a_i \in \mathbb{F}_q^*$ and $b \in \mathbb{F}_q$ (note: the number of variables is equal to the degree). We want to compute $|V(\mathbb{F}_q)|$, its number of (projective) points over $\mathbb{F}_q$.

Important tool: we denote by ω a generator of the group of characters of $\mathbb{F}_q^*$ (with values in some algebraically closed field): recall that $\mathbb{F}_q^*$ is a cyclic group, so ω exists and can be defined by $\omega(g) = \zeta_{q-1}$ for g a generator of $\mathbb{F}_q^*$ and ζ_{q-1} a primitive $(q-1)$-th root of unity.

Quasi-diagonal Hypersurfaces I

Let $\mathbb{F}_q$ be a finite field with $q = p^f$ elements for some prime p. A quasi-diagonal hypersurface V in $\mathbb{P}^{m-1}$ is a variety given by a projective equation

$$\sum_{1 \le i \le m} a_i x_i^m - b \prod_{1 \le i \le m} x_i = 0 ,$$

with $a_i \in \mathbb{F}_q^*$ and $b \in \mathbb{F}_q$ (note: the number of variables is equal to the degree). We want to compute $|V(\mathbb{F}_q)|$, its number of (projective) points over $\mathbb{F}_q$.

Important tool: we denote by ω a generator of the group of characters of $\mathbb{F}_q^*$ (with values in some algebraically closed field): recall that $\mathbb{F}_q^*$ is a cyclic group, so ω exists and can be defined by $\omega(g) = \zeta_{q-1}$ for g a generator of $\mathbb{F}_q^*$ and ζ_{q-1} a primitive $(q-1)$-th root of unity.

Quasi-diagonal Hypersurfaces II

Then theorem: if $\gcd(m, q-1) = 1$ and $b \neq 0$, set $B = \prod_{1 \le i \le m}(a_i/b)$. We have$|V(\mathbb{F}_q)| = (A(\mathbb{F}_q) - 1)/(q-1)$ with

$$A(\mathbb{F}_q) = (-1)^{m-1} + \sum_{0 \le n \le q-2} \omega^{-n}(B) J_m(\omega^n, \dots, \omega^n) ,$$

J_m is an m-variable Jacobi sum defined as follows:

$$J_m(\chi_1, \dots, \chi_m) = \sum_{x_1 + \cdots + x_m = 1} \chi_1(x_1) \cdots \chi_m(x_m)$$

for characters χ_i of $\mathbb{F}_q^*$. The proof is not difficult.

Quasi-diagonal Hypersurfaces II

Then theorem: if $\gcd(m, q-1) = 1$ and $b \neq 0$, set $B = \prod_{1 \le i \le m}(a_i/b)$. We have $|V(\mathbb{F}_q)| = (A(\mathbb{F}_q) - 1)/(q-1)$ with

$$A(\mathbb{F}_q) = (-1)^{m-1} + \sum_{0 \le n \le q-2} \omega^{-n}(B) J_m(\omega^n, \dots, \omega^n) ,$$

J_m is an m-variable Jacobi sum defined as follows:

$$J_m(\chi_1, \dots, \chi_m) = \sum_{x_1 + \cdots + x_m = 1} \chi_1(x_1) \cdots \chi_m(x_m)$$

for characters χ_i of $\mathbb{F}_q^*$. The proof is not difficult.

Gauss and Jacobi Sums I

Thus, we need preliminaries on Gauss and Jacobi sums. Let χ be a character of $\mathbb{F}_q^*$. The Gauss sum is defined by

$$\mathfrak{g}(\chi) = \sum_{x \in \mathbb{F}_q^*} \chi(x) \exp(2\pi i \operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(x)/p)$$

($\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}$ denotes the trace from $\mathbb{F}_q$ to $\mathbb{F}_p$). If χ is the trivial character ε, we have $\mathfrak{g}(\chi) = -1$, otherwise it is easy to prove that $|\mathfrak{g}(\chi)| = q^{1/2}$.

Gauss–Jacobi sum relation: if $\chi_i \neq \varepsilon$ for all i and $\prod_i \chi_i \neq \varepsilon$, then

$$J_m(\chi_1, \dots, \chi_m) = \frac{\mathfrak{g}(\chi_1) \cdots \mathfrak{g}(\chi_m)}{\mathfrak{g}(\chi_1 \cdots \chi_m)} .$$

If some $\chi_i = \varepsilon$ or $\prod_i \chi_i = \varepsilon$, there are other, simpler, formulas, for instance, $J_m(\varepsilon, \dots, \varepsilon) = q^{m-1}$.

Gauss and Jacobi Sums I

Thus, we need preliminaries on Gauss and Jacobi sums. Let χ be a character of $\mathbb{F}_q^*$. The Gauss sum is defined by

$$\mathfrak{g}(\chi) = \sum_{x \in \mathbb{F}_q^*} \chi(x) \exp(2\pi i \operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(x)/p)$$

($\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}$ denotes the trace from $\mathbb{F}_q$ to $\mathbb{F}_p$). If χ is the trivial character ε, we have $\mathfrak{g}(\chi) = -1$, otherwise it is easy to prove that $|\mathfrak{g}(\chi)| = q^{1/2}$.

Gauss–Jacobi sum relation: if $\chi_i \neq \varepsilon$ for all i and $\prod_i \chi_i \neq \varepsilon$, then

$$J_m(\chi_1, \dots, \chi_m) = \frac{\mathfrak{g}(\chi_1) \cdots \mathfrak{g}(\chi_m)}{\mathfrak{g}(\chi_1 \cdots \chi_m)} .$$

If some $\chi_i = \varepsilon$ or $\prod_i \chi_i = \varepsilon$, there are other, simpler, formulas, for instance, $J_m(\varepsilon, \dots, \varepsilon) = q^{m-1}$.

Gauss and Jacobi Sums II

So if everything is different from the trivial character ε, we have an immediate recursion

$$J_m(\chi_1, \dots, \chi_m) = J_{m-1}(\chi_1, \dots, \chi_{m-1}) J_2(\psi, \chi_m)$$

with $\psi = \chi_1 \cdots \chi_{m-1}$ (and even simpler recursions if some χ_i or ψ is equal to ε).

Naive computation of J_m requires summing over $(x_1, \dots, x_m)$ such that $x_1 + \cdots + x_m = 1$, so q^{m-1} operations.
Use of the recursion requires $(m-1)$ times computation of J_2, hence essentially $(m-1)q$ operations, so much faster.

We write

$$J(\chi_1, \chi_2) := J_2(\chi_1, \chi_2) = \sum_{x \in \mathbb{F}_q} \chi_1(x) \chi_2(1-x) .$$

Gauss and Jacobi Sums II

So if everything is different from the trivial character ε, we have an immediate recursion

$$J_m(\chi_1,\dots,\chi_m)=J_{m-1}(\chi_1,\dots,\chi_{m-1})J_2(\psi,\chi_m)$$

with $\psi=\chi_1\cdots\chi_{m-1}$ (and even simpler recursions if some χ_i or ψ is equal to ε).

Naive computation of J_m requires summing over $(x_1,\dots,x_m)$ such that $x_1+\cdots+x_m=1$, so q^{m-1} operations.
Use of the recursion requires $(m-1)$ times computation of J_2, hence essentially $(m-1)q$ operations, so much faster.

We write

$$J(\chi_1,\chi_2):=J_2(\chi_1,\chi_2)=\sum_{x\in\mathbb{F}_q}\chi_1(x)\chi_2(1-x)\;.$$

Introduction
The Four Methods

Gauss and Jacobi Sums II

So if everything is different from the trivial character ε, we have an immediate recursion

$$J_m(\chi_1,\dots,\chi_m)=J_{m-1}(\chi_1,\dots,\chi_{m-1})J_2(\psi,\chi_m)$$

with $\psi=\chi_1\cdots\chi_{m-1}$ (and even simpler recursions if some χ_i or ψ is equal to ε).

Naive computation of J_m requires summing over $(x_1,\dots,x_m)$ such that $x_1+\cdots+x_m=1$, so q^{m-1} operations.
Use of the recursion requires $(m-1)$ times computation of J_2, hence essentially $(m-1)q$ operations, so much faster.

We write

$$J(\chi_1,\chi_2):=J_2(\chi_1,\chi_2)=\sum_{x\in\mathbb{F}_q}\chi_1(x)\chi_2(1-x)\;.$$

Henri Cohen (Talk given by Atsuko Miyaji) Point Counting on Quasi-Diagonal Hypersurfaces

A Complete Example I

Recall that ω is a generator of the group of characters of $\mathbb{F}_q^*$.
We write for simplicity

$$J_m(n_1,\dots,n_m):=J_m(\omega^{n_1},\dots,\omega^{n_m})$$

(since any character is a power of ω, this is the general Jacobi sum).

We consider the quasi-diagonal hypersurface V as above, i.e. with projective equation $\sum_{1\le i\le m}a_ix_i^m-b\prod_{1\le i\le m}x_i=0$. By the above theorem, if $\gcd(m,q-1)=1$ and $b\neq0$, the number of projective points $V(\mathbb{F}_q)$ is equal to $(A(\mathbb{F}_q)-1)/(q-1)$, where

$$A(\mathbb{F}_q)=(-1)^{m-1}+S(q;B)\quad\text{with }B=\prod_{1\le i\le m}(a_i/b)\quad\text{and}$$

$$S(q;B)=\sum_{0\le n\le q-2}\omega^{-n}(B)J_m(n,\dots,n)\;.$$

Introduction
The Four Methods

A Complete Example I

Recall that ω is a generator of the group of characters of $\mathbb{F}_q^*$.
We write for simplicity

$$J_m(n_1,\dots,n_m):=J_m(\omega^{n_1},\dots,\omega^{n_m})$$

(since any character is a power of ω, this is the general Jacobi sum).

We consider the quasi-diagonal hypersurface V as above, i.e. with projective equation $\sum_{1\le i\le m}a_ix_i^m-b\prod_{1\le i\le m}x_i=0$. By the above theorem, if $\gcd(m,q-1)=1$ and $b\neq0$, the number of projective points $V(\mathbb{F}_q)$ is equal to $(A(\mathbb{F}_q)-1)/(q-1)$, where

$$A(\mathbb{F}_q)=(-1)^{m-1}+S(q;B)\quad\text{with }B=\prod_{1\le i\le m}(a_i/b)\quad\text{and}$$

$$S(q;B)=\sum_{0\le n\le q-2}\omega^{-n}(B)J_m(n,\dots,n)\;.$$

Henri Cohen (Talk given by Atsuko Miyaji) Point Counting on Quasi-Diagonal Hypersurfaces

A Complete Example II

We choose the reasonably nontrivial example $m = 5$, and we will study several methods for computing

$$S(q;B) = \sum_{0\le n\le q-2} \omega^{-n}(B) J_5(n,n,n,n,n)\ :$$

1. A direct method using the definition of $J(n,n)$.
2. Using the fact that all the character values are in the cyclotomic ring $\mathbb{Z}[\zeta_{q-1}]$, and in fact in the ring with zero divisors $R = \mathbb{Z}[X]/(X^{q-1}-1)$, we can work with simple polynomials.
3. Using theta functions.
4. Using Morita's p-adic gamma function and the Gross–Koblitz formula.

This last method is the most sophisticated, but by far the best, so I will spend some time describing it in detail.

The Direct Method

In this method we compute each $J_5(n,n,n,n,n)$ independently. Recall that generically (in this case, exactly when $(q-1)\nmid 5n$) we have $J_5(n,n,n,n,n) = J(n,n)J(2n,n)J(3n,n)J(4n,n)$, which thus require approximately $4q$ operations, so essentially $4q^2$ to compute $S(q;B)$. No need to give the exact formula since this is the slowest method.

Sample timings (all timings given in this talk are with a standard Intel 2.4 Ghz Core i7 processor using the `Pari/GP` library): for q of the order of 10^k with $k = 2, 3, 4$, requires 0.03, 1.46, 149 seconds respectively, compatible with $O(q^2)$ time. Prohibitive.

The Direct Method

In this method we compute each $J_5(n,n,n,n,n)$ independently. Recall that generically (in this case, exactly when $(q-1)\nmid 5n$) we have $J_5(n,n,n,n,n) = J(n,n)J(2n,n)J(3n,n)J(4n,n)$, which thus require approximately $4q$ operations, so essentially $4q^2$ to compute $S(q;B)$. No need to give the exact formula since this is the slowest method.

Sample timings (all timings given in this talk are with a standard Intel 2.4 Ghz Core i7 processor using the `Pari/GP` library): for q of the order of 10^k with $k = 2, 3, 4$, requires 0.03, 1.46, 149 seconds respectively, compatible with $O(q^2)$ time. Prohibitive.

Working with Polynomials I

Recall that $\omega(x) \in \mathbb{Z}[\zeta_{q-1}]$, where ζ_{q-1} is a primitive $(q-1)$-th root of unity, so all operations can be done in this ring. However, slightly expensive. More efficient: work in $R = \mathbb{Z}[X]/(X^{q-1}-1)$, with the natural surjective map from R to $\mathbb{Z}[\zeta_{q-1}]$ given by $X \mapsto \zeta_{q-1}$. The ring R has zero divisors, but no problem.

Let g be the unique generator of $\mathbb{F}_q^*$ such that $\omega(g) = \zeta_{q-1}$. Generically, we have

$$J(n,an) = \sum_{1\le u\le q-2} \omega^n(g^u)\omega^{an}(1-g^u) = \sum_{1\le u\le q-2} \zeta_{q-1}^{nu+na\log_g(1-g^u)}\ ,$$

where $\log_g$ is the discrete logarithm ($g^{\log_g(x)} = x$) modulo $q-1$.

Working with Polynomials I

Recall that $\omega(x) \in \mathbb{Z}[\zeta_{q-1}]$, where ζ_{q-1} is a primitive $(q-1)$-th root of unity, so all operations can be done in this ring. However, slightly expensive. More efficient: work in $R = \mathbb{Z}[X]/(X^{q-1}-1)$, with the natural surjective map from R to $\mathbb{Z}[\zeta_{q-1}]$ given by $X \mapsto \zeta_{q-1}$. The ring R has zero divisors, but no problem.

Let g be the unique generator of $\mathbb{F}_q^*$ such that $\omega(g) = \zeta_{q-1}$. Generically, we have

$$J(n, an) = \sum_{1 \le u \le q-2} \omega^n(g^u)\omega^{an}(1-g^u) = \sum_{1 \le u \le q-2} \zeta_{q-1}^{nu + na\log_g(1-g^u)}\;,$$

where $\log_g$ is the discrete logarithm ($g^{\log_g(x)} = x$) modulo $q-1$.

Working with Polynomials II

Thus for $1 \le a \le 4$ we define the polynomials of degree $q-2$

$$P_a(X) = \sum_{1 \le u \le q-2} X^{(u + a\log_g(1-g^u)) \bmod q-1} \in \mathbb{Z}[X]\;,$$

so $J(n, an) = P_a(\zeta_{q-1}^n)$ when $(q-1) \nmid an$, and more generally

$$J(n, an) = P_a(\zeta_{q-1}^n) + \begin{cases} 0 & \text{if } (q-1) \nmid an\;, \\ 1 & \text{if } (q-1) \mid an \text{ but } (q-1) \nmid n\;, \\ 2 & \text{if } (q-1) \mid n\;. \end{cases}$$

Since generically we have $J_5(n, n, n, n, n) = J(n, n)J(n, 2n)J(n, 3n)J(n, 4n)$, it follows that if we set $P(X) = P_1(X)P_2(X)P_3(X)P_4(X)$ we have (generically) $J_5(n, n, n, n, n) = P(\zeta_{q-1}^n)$.

Working with Polynomials II

Thus for $1 \le a \le 4$ we define the polynomials of degree $q-2$

$$P_a(X) = \sum_{1 \le u \le q-2} X^{(u + a\log_g(1-g^u)) \bmod q-1} \in \mathbb{Z}[X]\;,$$

so $J(n, an) = P_a(\zeta_{q-1}^n)$ when $(q-1) \nmid an$, and more generally

$$J(n, an) = P_a(\zeta_{q-1}^n) + \begin{cases} 0 & \text{if } (q-1) \nmid an\;, \\ 1 & \text{if } (q-1) \mid an \text{ but } (q-1) \nmid n\;, \\ 2 & \text{if } (q-1) \mid n\;. \end{cases}$$

Since generically we have $J_5(n, n, n, n, n) = J(n, n)J(n, 2n)J(n, 3n)J(n, 4n)$, it follows that if we set $P(X) = P_1(X)P_2(X)P_3(X)P_4(X)$ we have (generically) $J_5(n, n, n, n, n) = P(\zeta_{q-1}^n)$.

Working with Polynomials II

Write $P(X) = \sum_{0 \le j \le q-2} a_j X^j$, and set $\ell = \log_g(B)$. We have

$$\omega^{-n}(B)J_5(n, n, n, n, n) = \zeta_{q-1}^{-n\ell} \sum_{0 \le j \le q-2} a_j \zeta_{q-1}^{nj} = \sum_{0 \le j \le q-2} a_j \zeta_{q-1}^{n(j-\ell)}\;.$$

The whole point of this method is that when we sum on n the expression $\sum_{0 \le n \le q-2} \zeta_{q-1}^{n(j-\ell)}$ almost always vanishes, more precisely it vanishes if $j \ne \ell$ and otherwise it equals $q-1$. Thus (if all terms were generic) we would have $S(q; B) = (q-1)a_\ell$, so instead of computing the $J_5(n, n, n, n, n)$ individually, we immediately have the sum.

Working with Polynomials II

Write $P(X)=\sum_{0\le j\le q-2}a_jX^j$, and set $\ell=\log_g(B)$. We have

$$\omega^{-n}(B)J_5(n,n,n,n,n)=\zeta_{q-1}^{-n\ell}\sum_{0\le j\le q-2}a_j\zeta_{q-1}^{nj}=\sum_{0\le j\le q-2}a_j\zeta_{q-1}^{n(j-\ell)}\;.$$

The whole point of this method is that when we sum on n the expression $\sum_{0\le n\le q-2}\zeta_{q-1}^{n(j-\ell)}$ almost always vanishes, more precisely it vanishes if $j\ne\ell$ and otherwise it equals $q-1$. Thus (if all terms were generic) we would have $S(q;B)=(q-1)a_\ell$, so instead of computing the $J_5(n,n,n,n,n)$ individually, we immediately have the sum.

Working with Polynomials II

We must take care of the nongeneric terms, but this is simple bookkeeping. The final result is the following (same notation a_j and ℓ):

$$S(q;B)=(q-1)(a_\ell+T_1+T_2+T_3+T_4)$$

with $T_m=0$ if $m\nmid(q-1)$, and otherwise

$$T_1=8(q^2-2q+2)\;,\quad T_2=\chi_2(B)(q+1)\;,$$
$$T_3=2\Re(\chi_3^{-1}(B)J(\chi_3,\chi_3)^2)\;,\quad T_4=2\Re(\chi_4^{-1}(B)J(\chi_4,\chi_4)^2)\;,$$

where χ_m is any character of $\mathbb{F}_q^*$ of order exactly m. Note that $J(\chi_m,\chi_m)$ can be computed in a special very fast way.

Working with Polynomials III

This requires essentially $O(q)$ time, much faster than the direct method. Main drawback of this method: although $O(q)$ time, it has also $O(q)$ storage, so useless if $q>10^8$, say. For many applications, it is sufficient.

Sample timings: for q of the order 10^k with $k=2,3,4,5,6,7$, requires 0.002, 0.02, 0.08, 0.85, 9.9, 123 seconds respectively, compatible with $O(q)$ time and of course much faster than the direct method; however already needs several gigabytes of storage for $q\approx 10^7$.

Working with Polynomials III

This requires essentially $O(q)$ time, much faster than the direct method. Main drawback of this method: although $O(q)$ time, it has also $O(q)$ storage, so useless if $q>10^8$, say. For many applications, it is sufficient.

Sample timings: for q of the order 10^k with $k=2,3,4,5,6,7$, requires 0.002, 0.02, 0.08, 0.85, 9.9, 123 seconds respectively, compatible with $O(q)$ time and of course much faster than the direct method; however already needs several gigabytes of storage for $q\approx 10^7$.

Using Theta Functions I

Assume that $q = p$. For χ a character on $\mathbb{F}_p^*$ and $t > 0$ we define the theta function

$$\Theta(\chi, t) = 2 \sum_{m \geq 1} m^e \chi(m) e^{-\pi m^2 t/p} ,$$

where $e = 0$ or 1 is the parity of χ ($\chi(-1) = (-1)^e$).

Main properties: first, it is very rapidly convergent (essentially $O(p^{1/2})$ terms to compute numerical values). Second and most importantly, it has a functional equation for $t \in \mathbb{R}_{>0}$

$$\Theta(\chi, 1/t) = W(\chi) t^{1/2+e} \overline{\Theta(\chi, t)} , \quad \text{where}$$
$$W(\chi) = \mathfrak{g}(\chi)/(i^e p^{1/2}) .$$

Introduction
The Four Methods

Using Theta Functions I

Assume that $q = p$. For χ a character on $\mathbb{F}_p^*$ and $t > 0$ we define the theta function

$$\Theta(\chi, t) = 2 \sum_{m \geq 1} m^e \chi(m) e^{-\pi m^2 t/p} ,$$

where $e = 0$ or 1 is the parity of χ ($\chi(-1) = (-1)^e$).

Main properties: first, it is very rapidly convergent (essentially $O(p^{1/2})$ terms to compute numerical values). Second and most importantly, it has a functional equation for $t \in \mathbb{R}_{>0}$

$$\Theta(\chi, 1/t) = W(\chi) t^{1/2+e} \overline{\Theta(\chi, t)} , \quad \text{where}$$
$$W(\chi) = \mathfrak{g}(\chi)/(i^e p^{1/2}) .$$

Henri Cohen (Talk given by Atsuko Miyaji) Point Counting on Quasi-Diagonal Hypersurfaces

Using Theta Functions II

Thus, if for instance $\Theta(\chi, 1) \neq 0$ (otherwise use $t \neq 1$ or apply L'Hospital's rule) we have

$$\mathfrak{g}(\chi) = i^e p^{1/2} \Theta(\chi, 1)/\overline{\Theta(\chi, 1)} .$$

Thus (for $q = p$) the Gauss sum $\mathfrak{g}(\chi)$ can be computed in time essentially $O(q^{1/2})$ (more precisely $O(q^{1/2+\varepsilon})$ for all $\varepsilon > 0$, but we ignore ε). Since Jacobi sums can be expressed in terms of products of Gauss sums, it follows that they also can be computed in $O(q^{1/2})$. Much faster than the direct method which requires $O(q)$.

Using Theta Functions II

Using this, we can compute $S(q; B)$ (for $q = p$) in time $O(q^{3/2})$. Slower than the polynomial version above which was in $O(q)$, but big advantage: essentially no storage. For $q > 10^8$, much too slow however.

Sample timings: for $q = p$ of the order 10^k with $k = 2, 3, 4, 5$, requires 0.02, 0.4, 16.2, 663 seconds, compatible with $O(q^{3/2})$ time. Much slower than the polynomial method, but very little storage.

Using Theta Functions II

Using this, we can compute $S(q;B)$ (for $q=p$) in time $O(q^{3/2})$. Slower than the polynomial version above which was in $O(q)$, but big advantage: essentially no storage. For $q>10^8$, much too slow however.

Sample timings: for $q=p$ of the order 10^k with $k=2,3,4,5$, requires 0.02, 0.4, 16.2, 663 seconds, compatible with $O(q^{3/2})$ time. Much slower than the polynomial method, but very little storage.

Morita's p-adic Gamma Function I

We now come to the most efficient method, but also the most sophisticated method to compute $S(q;B)$. Since behind the scenes there are variants of crystalline cohomology theories, this is a distant cousin of Kedlaya's algorithm for counting points on hyperelliptic curves. But don't be afraid of these dirty words, you will see that at the end of the day everything is completely elementary.

We assume some familiarity with p-adic numbers: recall simply that in the p-adic topology $p^j\to 0$ when $j\to\infty$. We denote by $\mathbb{Z}_p$ the ring of p-adic integers $s=a_0+a_1p+a_2p^2+\cdots$ with $0\le a_i\le p-1$.

Morita's p-adic Gamma Function I

We now come to the most efficient method, but also the most sophisticated method to compute $S(q;B)$. Since behind the scenes there are variants of crystalline cohomology theories, this is a distant cousin of Kedlaya's algorithm for counting points on hyperelliptic curves. But don't be afraid of these dirty words, you will see that at the end of the day everything is completely elementary.

We assume some familiarity with p-adic numbers: recall simply that in the p-adic topology $p^j\to 0$ when $j\to\infty$. We denote by $\mathbb{Z}_p$ the ring of p-adic integers $s=a_0+a_1p+a_2p^2+\cdots$ with $0\le a_i\le p-1$.

Morita's p-adic Gamma Function II

We need to define the p-adic analogue of the ordinary gamma function, called Morita's p-adic gamma function and denoted Γ_p. Its definition is very simple (all limits p-adic):

$$\Gamma_p(s)=\lim_{\substack{m\to s\\ m\in\mathbb{Z}_{>0}}}(-1)^m\prod_{\substack{0\le k<m\\ p\nmid k}}k=\lim_{\substack{m\to s-1\\ m\in\mathbb{Z}_{>0}}}(-1)^{m+1}\frac{m!}{p^{\lfloor m/p\rfloor}(\lfloor m/p\rfloor)!}\;.$$

Observe this definition: eliminating terms k such that $p\mid k$ is natural. But why the $(-1)^m$? this is due to Wilson's theorem $(p-1)!\equiv -1\pmod{p}$. Need to show convergence: immediately follows from the following lemma (exercise!):

$$\prod_{\substack{m\le k<m+a\cdot p^N\\ p\nmid k}}k\equiv(-1)^{a\cdot p^N}\pmod{p^N}\;.$$

Morita's p-adic Gamma Function II

We need to define the p-adic analogue of the ordinary gamma function, called Morita's p-adic gamma function and denoted Γ_p. Its definition is very simple (all limits p-adic):

$$\Gamma_p(s)=\lim_{\substack{m\to s\\ m\in\mathbb{Z}_{>0}}}(-1)^m\prod_{\substack{0\le k<m\\ p\nmid k}}k=\lim_{\substack{m\to s-1\\ m\in\mathbb{Z}_{>0}}}(-1)^{m+1}\frac{m!}{p^{\lfloor m/p\rfloor}(\lfloor m/p\rfloor)!}\;.$$

Observe this definition: eliminating terms k such that $p\mid k$ is natural. But why the $(-1)^m$? this is due to Wilson's theorem $(p-1)!\equiv -1\pmod{p}$. Need to show convergence: immediately follows from the following lemma (exercise!):

$$\prod_{\substack{m\le k<m+a\cdot p^N\\ p\nmid k}}k\equiv(-1)^{a\cdot p^N}\pmod{p^N}\;.$$

Morita's p-adic Gamma Function III

Properties completely analogous (but slightly different from) the ordinary gamma function $\Gamma(s)$ (expression of $\Gamma_p(m)$ in terms of factorials for $m\in\mathbb{Z}$, recursion formula giving $\Gamma_p(s+1)$ in terms of $\Gamma_p(s)$, reflection formula giving $\Gamma_p(1-s)$ in terms of $\Gamma_p(s)$, duplication and more generally distribution formula giving $\prod_{0\le j<N}\Gamma_p(s+j/N)$, explicit expression for $\Gamma_p(1/2)$, explicit power series expansion of $\log_p(\Gamma_p(s+1))$, Raabe's formula). As a consequence: easy algorithms for computing it implemented in most computer algebra systems.

There is a more sophisticated formula for the ordinary gamma function called the Lerch, Chowla–Selberg formula which I will not state. The p-adic analogue is what concerns us here, called the Gross–Koblitz formula.

Morita's p-adic Gamma Function III

Properties completely analogous (but slightly different from) the ordinary gamma function $\Gamma(s)$ (expression of $\Gamma_p(m)$ in terms of factorials for $m\in\mathbb{Z}$, recursion formula giving $\Gamma_p(s+1)$ in terms of $\Gamma_p(s)$, reflection formula giving $\Gamma_p(1-s)$ in terms of $\Gamma_p(s)$, duplication and more generally distribution formula giving $\prod_{0\le j<N}\Gamma_p(s+j/N)$, explicit expression for $\Gamma_p(1/2)$, explicit power series expansion of $\log_p(\Gamma_p(s+1))$, Raabe's formula). As a consequence: easy algorithms for computing it implemented in most computer algebra systems.

There is a more sophisticated formula for the ordinary gamma function called the Lerch, Chowla–Selberg formula which I will not state. The p-adic analogue is what concerns us here, called the Gross–Koblitz formula.

The Gross–Koblitz Formula I

We have a surprise: some natural values are algebraic: for example one computes that

$$\Gamma_5(1/4)=\sqrt{-2+\sqrt{-1}}$$

for suitable signs of square roots. This is totally different from the ordinary gamma ($\Gamma(1/4)$ is known to be transcendental), but is a special case of the Gross-Koblitz formula. More generally, $\Gamma_p(r/(p-1))$ is an algebraic number.

The general Gross–Koblitz formula says in rough terms that: **Any Gauss sum over $\mathbb{F}_{p^f}$ is equal to an (explicit) product of f values of $\Gamma_p(s_i)$ at rational arguments** $(s_i)_{1\le i\le f}$, up to a known sign and rational power of p.

The Gross–Koblitz Formula I

We have a surprise: some natural values are algebraic: for example one computes that

$$\Gamma_5(1/4) = \sqrt{-2+\sqrt{-1}}$$

for suitable signs of square roots. This is totally different from the ordinary gamma ($\Gamma(1/4)$ is known to be transcendental), but is a special case of the Gross-Koblitz formula. More generally, $\Gamma_p(r/(p-1))$ is an algebraic number.

The general Gross–Koblitz formula says in rough terms that: **Any Gauss sum over $\mathbb{F}_{p^f}$ is equal to an (explicit) product of f values of $\Gamma_p(s_i)$ at rational arguments $(s_i)_{1\le i\le f}$**, up to a known sign and rational power of p.

The Gross–Koblitz Formula II

Consequence: to compute Gauss sums (and Jacobi sums, since they can be expressed in terms of Gauss sums), it is sufficient to be able to compute $\Gamma_p(s)$.

As mentioned, there exist efficient algorithms for this (see my book Springer GTM 240). Can now forget about p-adic numbers: one can obtain the result modulo p, or modulo p^2, etc... This is sufficient because of the Deligne–Weil bounds.

The Gross–Koblitz Formula II

Consequence: to compute Gauss sums (and Jacobi sums, since they can be expressed in terms of Gauss sums), it is sufficient to be able to compute $\Gamma_p(s)$.

As mentioned, there exist efficient algorithms for this (see my book Springer GTM 240). Can now forget about p-adic numbers: one can obtain the result modulo p, or modulo p^2, etc... This is sufficient because of the Deligne–Weil bounds.

A Sample Pari/GP Session

```
? gamma(1/4+O(5^12))
% = 1 + 4*5 + 3*5^4 + 5^6 + 5^7 + 4*5^9 + 5^10 + O(5^12)
? algdep(%,4)
% = x^4 + 4*x^2 + 5 /* algebraic number */
? gamma(1/3+O(7^20))
% = 4 + 3*7 + 5*7^2 + 7^3 + 7^4 + 2*7^5 + 7^6 + 5*7^7 + ...
? algdep(%,6)
% = x^6 - x^3 + 7 /* algebraic number */
? gamma(1/6+O(7^20))
% =  1 + 4*7 + 4*7^2 + 6*7^3 + 6*7^4 + 7^5 + 4*7^6 + ...
? algdep(%,6)
% = x^6 + 13*x^3 + 49 /* algebraic number */
? gamma(1/6+O(7^1000));
time = 96 ms.  /* Very fast, even for 1000 p-adic digits. *
```

The Method using Gross–Koblitz I

Using the Gross–Koblitz formula, it is easy to prove the following result for our problem: let H_n be the nth harmonic sum $H_n = \sum_{1\le j\le n} 1/j$. Then

$$S(p;B) \equiv \sum_{0<r\le(p-1)/5} \frac{(5r)!}{r!^5}(1+5pr(H_{5r}-H_r))B^{pr}$$
$$-p \sum_{(p-1)/5<r\le 2(p-1)/5} \frac{(5r-(p-1))!}{r!^5}B^r \pmod{p^2}\ .$$

Note that this requires only $O(p)$ operations and essentially no storage.

The Method using Gross–Koblitz II

Now we have seen above that $(p-1) \mid S(p;B)$. Thus the above congruence determines $S(p;B)$ modulo $p^2(p-1)$. On the other hand, the Weil conjectures (more precisely the Riemann hypothesis for varieties, Deligne's theorem) tells us that $|S(p;B)-p^4| < 4p^{5/2}$. A small computation shows that for $p \ge 67$ the congruence modulo $(p-1)p^2$ determines completely $S(p;B)$.

The Method using Gross–Koblitz II

Thus this is a $O(p)$ method, much faster than everything else (even the implicit constant in the $O()$ is very small), and not needing much storage. Example in 2 seconds we obtain

$$S(10^6+3;2) = 1000012000056356142712140\ .$$

Sample timings: for $q = p$ of the order 10^k with $k = 2, 3, 4, 5, 6, 7, 8$ requires $0.001, 0.01, 0.03, 0.21, 2.13, 21.9, 229$ seconds respectively, compatible with time $O(q)$ (and 5 to 6 times faster than the polynomial method), but requiring essentially no storage.
Therefore it is the best available method.

The Method using Gross–Koblitz II

Thus this is a $O(p)$ method, much faster than everything else (even the implicit constant in the $O()$ is very small), and not needing much storage. Example in 2 seconds we obtain

$$S(10^6+3;2) = 1000012000056356142712140\ .$$

Sample timings: for $q = p$ of the order 10^k with $k = 2, 3, 4, 5, 6, 7, 8$ requires $0.001, 0.01, 0.03, 0.21, 2.13, 21.9, 229$ seconds respectively, compatible with time $O(q)$ (and 5 to 6 times faster than the polynomial method), but requiring essentially no storage.
Therefore it is the best available method.

Conclusion I

We have presented four algorithms for computing the number of points of a quasi-diagonal hypersurface. A summary of the timings (in seconds) for a prime q of the order of 10^k is given in the following table, where $\ast$ means that I have not been patient enough for the program to terminate:

k	2	3	4	5	6	7	8
Direct	0.03	1.56	149.	$\ast$	$\ast$	$\ast$	$\ast$
Theta	0.02	0.40	16.2	663.	$\ast$	$\ast$	$\ast$
Mod $X^{q-1}-1$	0.00	0.02	0.08	0.85	9.90	123	$\ast$
Gross–Koblitz	0.00	0.01	0.03	0.21	2.13	21.9	229.

Conclusion II

The definite conclusion is that the method using the Gross–Koblitz formula is both by far the best in terms of speed, but also in terms of storage since it does not need much.

Two additional remarks. First, note that this method can be used in point-counting for much more general varieties than quasi-diagonal hypersurfaces, for instance for varieties coming from hypergeometric motives.

Second, computing $|V(\mathbb{F}_q)|$ for all small prime powers q allows the construction of the global L-function attached to the variety V, and in particular permits the experimental testing of numerous conjectures (generalizing the Taniyama–Weil conjecture, i.e., Wiles's theorem).

Conclusion II

The definite conclusion is that the method using the Gross–Koblitz formula is both by far the best in terms of speed, but also in terms of storage since it does not need much.

Two additional remarks. First, note that this method can be used in point-counting for much more general varieties than quasi-diagonal hypersurfaces, for instance for varieties coming from hypergeometric motives.

Second, computing $|V(\mathbb{F}_q)|$ for all small prime powers q allows the construction of the global L-function attached to the variety V, and in particular permits the experimental testing of numerous conjectures (generalizing the Taniyama–Weil conjecture, i.e., Wiles's theorem).

Conclusion II

The definite conclusion is that the method using the Gross–Koblitz formula is both by far the best in terms of speed, but also in terms of storage since it does not need much.

Two additional remarks. First, note that this method can be used in point-counting for much more general varieties than quasi-diagonal hypersurfaces, for instance for varieties coming from hypergeometric motives.

Second, computing $|V(\mathbb{F}_q)|$ for all small prime powers q allows the construction of the global L-function attached to the variety V, and in particular permits the experimental testing of numerous conjectures (generalizing the Taniyama–Weil conjecture, i.e., Wiles's theorem).

Thank you for your attention.