Solving p-adic differential equations in point counting algorithms

Hendrik Hubrechts

Katholieke Universiteit Leuven (Belgium)

Oxford, March 16, 2010

Computing the zeta function of a hyperelliptic curve (HEC)

- Let p be prime, $\mathbb{F}_{p^n}$ a finite field. We suppose here $p \geq 3$.
- A hyperelliptic curve $\bar{\mathcal{C}}/\mathbb{F}_{p^n}$ of genus g of equation
  $$Y^2 = \bar{Q}(X) \quad \text{where} \quad \bar{Q}(X) = X^{2g+1} + a_{2g}X^{2g} + \cdots + a_1 X + a_0.$$
- We can lift this setting to $Y^2 = Q(X)$ over $\mathbb{Z}_{p^n} \subset \mathbb{Q}_{p^n}$.
- (Kedlaya) For computing the zeta function of $\bar{\mathcal{C}}$: suffices to determine matrix F of pth power Frobenius on H^-_{MW}, a $2g$-dimensional vector space over $\mathbb{Q}_{p^n}$ with basis $\{X^i dX/Y^3,\ i = 0, \ldots, 2g-1\}$.
- With $\sigma : \mathbb{Q}_{p^n} \longrightarrow \mathbb{Q}_{p^n}$ the Frobenius automorphism, the matrix $F \cdot F^{\sigma} \cdots F^{\sigma^{n-1}}$ determines the zeta function.

Algorithmic (Kedlaya): can all be done with sufficient precision $p^{n \cdot c}$ in $\tilde{\mathcal{O}}_n(n^3)$ bit operations, and bit space $\mathcal{O}_n(n^3)$.

Deformation of hyperelliptic curves

$$Y^2 = Q(X,t) = X^{2g+1} + \cdots \in \mathbb{Z}_{p^n}[X,t]$$

such that for almost all $\bar{t}_0 \in \mathbb{F}_{p^n}^{\text{alg cl}}$: $\bar{Q}(X,\bar{t}_0)$ is squarefree.

- Goal: Given Frobenius $F(0)$ for $t=0$, find Frobenius $F(t_0)$ for some given root of unity t_0.
- Bad fibers: roots modulo p of $r(t) := \text{Res}_X(Q(X,t);\ \frac{\partial Q(X,t)}{\partial X})$.
- Requirements for $r(t)$:

  $\bar{r}(0) \neq 0$; $\bar{r}(t)$ squarefree;

  $\deg\ r(t) = \deg\ \bar{r}(t)$; $\gcd(r(t), r^\sigma(t^p)) = 1$.
- Idea (Lauder): Take 'relative version' of Kedlaya's construction: $H^-_{MW}(t)$, free module of rank $2g$ over

$$A^\dagger := \left\{ \sum_{i\in\mathbb{Z}} g_i(t) r(t)^i \;\middle|\; \liminf_i \frac{ord(g_i(t))}{|i|} > 0 \right\}.$$

Deformation — differential equation

- We can consider the 'generic matrix of Frobenius' $F(t)$ over $A^\dagger$, on $H^-_{MW}(t)$. We indeed have that $F(t_0)$ is the matrix of Frobenius for the corresponding curve $Y^2 = Q(X, t_0)$.

- Derivation w.r.t. t gives the connection

$$\begin{aligned} \nabla : H^-_{MW}(t) &\to H^-_{MW}(t) \otimes_{A^\dagger} \Omega^1_{A^\dagger}, \\ \varphi &\mapsto \frac{d\varphi}{dt} dt. \end{aligned}$$

- Let $G(t)$ be the matrix of ∇, still w.r.t. the basis $\{X^i dX/Y^3,\ i = 0, \ldots, 2g-1\}$. Note that $G(t)$ can be computed easily.

- From the commutation of Frobenius and the connection follows

$$\frac{dF(t)}{dt} + G(t) \cdot F(t) = pt^{p-1} F(t) \cdot G^\sigma(t^p).$$

More differential equations

- In our setting: $G(t) = \dfrac{H(t)}{r(t)}$, where $H(t)$ consists of polynomials.

  This gives
  $$rr^\sigma \frac{dF}{dt} + r^\sigma HF = pt^{p-1} rFH^\sigma.$$

- We have to work modulo p^{nc}, then
  $$F(t) \equiv \sum_{-nc_1}^{nc_2} f_i(t) \cdot r(t)^i \mod p^{nc}.$$

- Define $K(t) := r(t)^{nc_1} \cdot F(t)$, then $K(t)$ mod p^{nc} consists of polynomials of degree at most nc_3.

- The differential equation for $K(t)$ becomes:
  $$rr^\sigma \frac{dK}{dt} - (nc_1) r^\sigma \frac{dr}{dt} K + r^\sigma HK = pt^{p-1} rKH^\sigma.$$

Solving the equations

$$rr^\sigma \frac{dK}{dt} - (nc_1) r^\sigma \frac{dr}{dt} K + r^\sigma HK = pt^{p-1} rKH^\sigma .$$

- **Assumption.** We know $K_0 = r(0)^{nc} F(0)$.
- Isolating the coefficient of t^{i-1} in the equation for $K(t) = \sum K_i t^i$:

$$K_i \equiv \frac{1}{i \cdot r(0) r^\sigma(0)} \cdot (\text{linear combination of } K_{i-1}, \ldots, K_{i-\zeta})$$

$$\text{with } \zeta = \mathcal{O}_n(1).$$

- Result: $F(t) \equiv \frac{1}{r(t)^{nc}} \left(\sum_{i=0}^{nc_3} K_i t^i \right)$.

Complexity to find $F(t)$ (and also $F(t_0)$ and the zeta function):

We need nc_3 iterations, each one consists of $\zeta \cdot \tilde{\mathcal{O}}_n(n^2)$ bit operations.
Total bit operations: $\tilde{\mathcal{O}}_n(n^3)$, bit space $\mathcal{O}_n(n^3)$.
Note that the bit size cannot be smaller!

Applications

1. Point counting in families defined over $\mathbb{F}_p$: $\tilde{\mathcal{O}}_n(n^2)$.
2. Memory efficient point counting.
3. Time efficient point counting.
4. Fibration method (work in progress!).

The Chudnovsky and Chudnosky trick for recurrence relations.

- Suppose we want to compute a_N for $N >> 0$ (and $N \in \mathbb{Z}^2$) from

$$a_{i+1} := f(i) \cdot a_i, \quad \text{given } a_0,$$

where $f(t) \in \mathbb{Q}(t)$, 'degree f'$\leq \alpha$.

- First compute

$$\varphi(n) := \varphi(n\sqrt{N}) \cdot \varphi(n\sqrt{N}+1) \cdots \varphi(n\sqrt{N}+(\sqrt{N}-1)).$$

Can be done in $\tilde{\mathcal{O}}(\sqrt{N})$ 'operations' via binary product. Note: 'degree φ'$\leq \sqrt{N}\alpha$.

- Then compute

$$a_N = f(0) \cdot f(1) \cdots f(\sqrt{N}-1) \cdot f(\sqrt{N}) \cdots f(N-1)a_0 =$$

$$\varphi(0) \cdot \varphi(1) \cdots \varphi(\sqrt{N})a_0.$$

This can again be done in $\tilde{\mathcal{O}}(\sqrt{N})$ 'operations' via *fast multipoint evaluation* followed by a binary product.