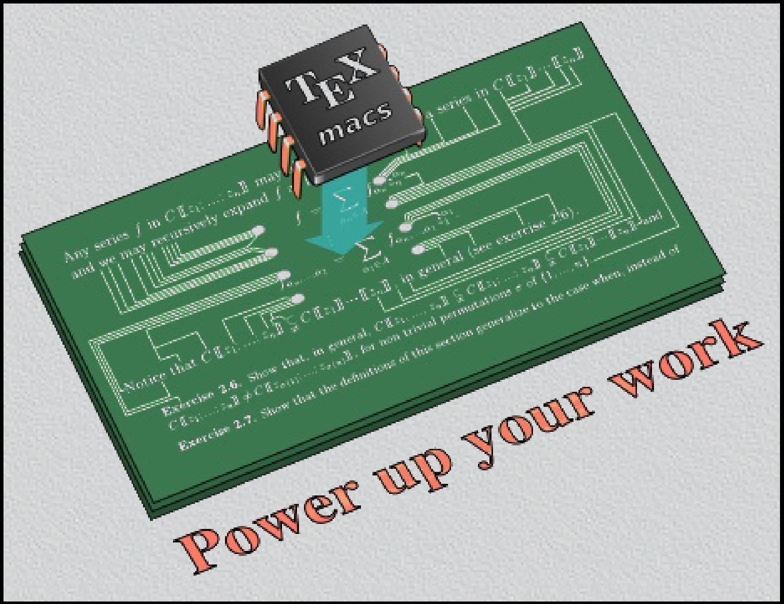
Sparse polynomial interpolation II

Joris van der Hoeven

CNRS, visiting professor at PIMS and SFU

Joint work with Grégoire Lecerf

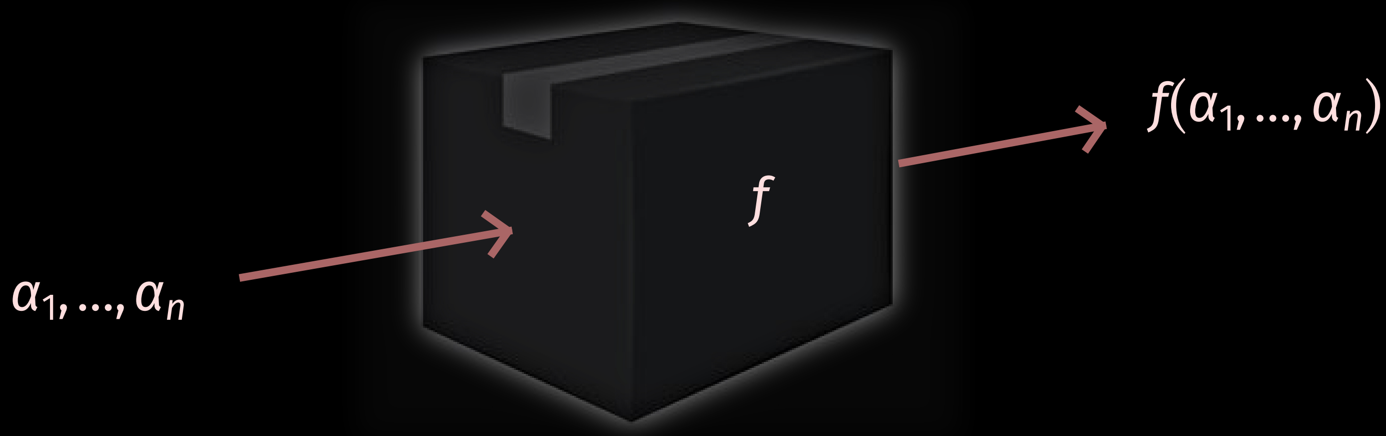


SFU, Vancouver **January 14, 2020**

Part I

Statement of the problem

Input



Output

$$f(x_1, \ldots, x_n) = c_1 x_1^{e_{1,1}} \cdots x_n^{e_{1,n}} + \cdots + c_t x_1^{e_{t,1}} \cdots x_n^{e_{t,n}}$$

Variants

Coefficients K

- A field from analysis such as $K = \mathbb{C}$.
- A discrete field such as $K = \mathbb{Q}$ or a finite field $K = \mathbb{F}_q$.
- Roots of unity ω of large smooth order in K?

Complexity model

- Algebraic *versus* bit complexity.
- Deterministic (needs bounds) *versus* probabilistic.
- Theoretic (asymptotic) *versus* practical complexity.
- Divisions in K allowed for evaluation of f?
- Allow evaluations at points in A^n for extension $A \supseteq K$?

How sparse?

- Weakly sparse: total degrees d of the order $O(\log t)$.
- Normally sparse: total degrees d of the order $t^{O(1)}$.
- Super sparse: total degrees of order d with $\log t = o(\log d)$.

Part II

Generalities

Generalities

Reductions

- Sparse interpolation $\longrightarrow$ Sparse interpolation with bounds $T \geqslant t$, $D \geqslant d$
- Sparse interpolation $\longrightarrow$ Approximate sparse interpolation

Roots of unity in finite fields

$$n \text{ is smooth} \Rightarrow q^n - 1 \text{ is supersmooth}$$

$$2^{60} - 1 = 3^2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 31 \cdot 41 \cdot 61 \cdot 151 \cdot 331 \cdot 1321$$

Part III

The cyclic extension approach

(Univariate case)

Idea

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t}$$

Main idea

For pairwise coprime $r = r_1, r_2, \ldots$, evaluate f at $\bar{x} \in K[x]/(x^r - 1)$, which yields

$$f \operatorname{rem} (x^r - 1) = c_1 x^{e_1 \operatorname{rem} r} + \cdots + c_t x^{e_t \operatorname{rem} r}$$

Match corresponding terms and reconstruct f using Chinese remaindering

Diversification

Several ways to "match corresponding terms"

Easiest approach: assume that $c_1, \ldots, c_t$ are (almost all) pairwise distinct

α is random and $|K|$ large $\Rightarrow f(\alpha x)$ is diversified with high probability

Complexity analysis

- L: number of operations needed to evaluate f
- $M(n) = O^\flat(n \log n)$: cost to multiply two polynomials of degree $\leqslant n$
- Cost of one evaluation $f(x) \operatorname{rem} (x^r - 1)$ is $O(L\,\mathrm{M}(r))$
- Expected number of correct terms: $\mathrm{e}^{-t/r}\,t$
- Cost per correct term proportional to $r\,\mathrm{e}^{t/r}$
- Optimum obtained by taking $r_1 \approx \cdots \approx r_l \approx t$

Proposition (modulo heuristic hypothesis)

Given $0 < \eta < 1$ and a diversified polynomial $f \in \mathbb{F}_q[x]$ of degree $d \leqslant D$ and with $t \leqslant T$ terms, there exists a Monte Carlo probabilistic algorithm which computes at least $(1 - \eta)\,t$ terms of f in time

$$O^\flat(L\,T \log D \log(q\,T)).$$

Part IV

The geometric progression approach

(Univariate case)

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned} f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\ f(\omega^1) &= c_1 \omega^{1e_1} + \cdots + c_t \omega^{1e_t} \\ f(\omega^2) &= c_1 \omega^{2e_1} + \cdots + c_t \omega^{2e_t} \\ &\vdots \end{aligned}$$

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1)\, z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2)\, z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\vdots
\end{aligned}$$

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1)\, z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2)\, z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\vdots \\
\sum_{k=0}^{\infty} f(\omega^k)\, z^k &= \frac{c_1}{1 - \omega^{e_1} z} + \cdots + \frac{c_t}{1 - \omega^{e_t} z} = \frac{N(z)}{\Lambda(z)}
\end{aligned}$$

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1)\, z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2)\, z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\vdots \\
\sum_{k=0}^{\infty} f(\omega^k)\, z^k &= \frac{c_1}{1 - \omega^{e_1} z} + \cdots + \frac{c_t}{1 - \omega^{e_t} z} = \frac{N(z)}{\Lambda(z)}
\end{aligned}$$

- Recover N and Λ from the first $2t-1$ evaluations

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1)\, z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2)\, z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\vdots \\
\sum_{k=0}^{\infty} f(\omega^k)\, z^k &= \frac{c_1}{1 - \omega^{e_1} z} + \cdots + \frac{c_t}{1 - \omega^{e_t} z} = \frac{N(z)}{\Lambda(z)}
\end{aligned}$$

- Recover N and Λ from the first $2t-1$ evaluations
- Determine the roots ω^{-e_i} of Λ

The method

$$f = c_1 x^{e_1} + \cdots + c_t x^{e_t} \in \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1) z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2) z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\vdots \\
\sum_{k=0}^{\infty} f(\omega^k) z^k &= \frac{c_1}{1-\omega^{e_1} z} + \cdots + \frac{c_t}{1-\omega^{e_t} z} = \frac{N(z)}{\Lambda(z)}
\end{aligned}$$

- Recover N and Λ from the first $2t-1$ evaluations
- Determine the roots ω^{-e_i} of Λ
- Compute the discrete logarithms e_i of ω^{e_i} w.r.t. ω

The method

$$f \;=\; c_1 x^{e_1} + \cdots + c_t x^{e_t} \;\in\; \mathbb{F}_q[x]$$

For some number $\omega \in K$ of high multiplicative order, compute

$$\begin{aligned}
f(\omega^0) &= c_1 \omega^{0e_1} + \cdots + c_t \omega^{0e_t} \\
f(\omega^1)\, z &= c_1 \omega^{1e_1} z + \cdots + c_t \omega^{1e_t} z \\
f(\omega^2)\, z^2 &= c_1 \omega^{2e_1} z^2 + \cdots + c_t \omega^{2e_t} z^2 \\
&\;\vdots \\
\sum_{k=0}^{\infty} f(\omega^k)\, z^k &= \frac{c_1}{1-\omega^{e_1} z} + \cdots + \frac{c_t}{1-\omega^{e_t} z} \;=\; \frac{N(z)}{\Lambda(z)}
\end{aligned}$$

- Recover N and Λ from the first $2t-1$ evaluations
- Determine the roots ω^{-e_i} of Λ
- Compute the discrete logarithms e_i of ω^{e_i} w.r.t. ω
- Compute the coefficients c_i using linear algebra

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ $\qquad O^\flat(L\,T\log q)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ $O^\flat(L\,T \log q)$
- Recover N and Λ
  - Half-gcd $O^\flat(M_q(T) \log T)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ $\quad O^\flat(L\,T\log q)$
- Recover N and Λ
  - Half-gcd $\quad O^\flat(T(\log T)^2\log q)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ $O^\flat(L\,T\log q)$
- Recover N and Λ
  - Half-gcd $O^\flat(T(\log T)^2\log q)$
- Determine the roots ω^{-e_i} of Λ
  - Cantor–Zassenhaus $O^\flat(T(\log T)^2(\log q)^2)$
  - Graeffe + $q-1$ large smooth factor $O^\flat(T(\log T)^3\log q)$
  - Tangent-Graeffe + $q-1$ large smooth factor $O^\flat(T(\log T)^2\log q)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ — $O^\flat(L\,T \log q)$
- Recover N and Λ
  - Half-gcd — $O^\flat(T(\log T)^2 \log q)$
- Determine the roots ω^{-e_i} of Λ
  - Cantor–Zassenhaus — $O^\flat(T(\log T)^2 (\log q)^2)$
  - Graeffe + $q-1$ large smooth factor — $O^\flat(T(\log T)^3 \log q)$
  - Tangent-Graeffe + $q-1$ large smooth factor — $O^\flat(T(\log T)^2 \log q)$
- Compute the discrete logarithms e_i of ω^{e_i} w.r.t. ω
  - Pohlig-Helmann + $q-1$ large smooth factor — $O^\flat(T \log T \log q)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ $O^\flat(L\,T\log q)$
- Recover N and Λ
  - Half-gcd $O^\flat(T(\log T)^2\log q)$
- Determine the roots ω^{-e_i} of Λ
  - Cantor–Zassenhaus $O^\flat(T(\log T)^2(\log q)^2)$
  - Graeffe + $q-1$ large smooth factor $O^\flat(T(\log T)^3\log q)$
  - Tangent-Graeffe + $q-1$ large smooth factor $O^\flat(T(\log T)^2\log q)$
- Compute the discrete logarithms e_i of ω^{e_i} w.r.t. ω
  - Pohlig-Helmann + $q-1$ large smooth factor $O^\flat(T\log T\log q)$
- Compute the coefficients c_i using linear algebra
  - Transposed fast multi-point interpolation $O^\flat(T(\log T)^2\log q)$

Complexity analysis

- Evaluate $f(\omega^0), f(\omega^1), \ldots, f(\omega^{2T-1})$ — $O^\flat(L\,T\log q)$
- Recover N and Λ
  - Half-gcd — $O^\flat(T(\log T)^2\log q)$
- Determine the roots ω^{-e_i} of Λ
  - Cantor–Zassenhaus — $O^\flat(T(\log T)^2(\log q)^2)$
  - Graeffe + $q-1$ large smooth factor — $O^\flat(T(\log T)^3\log q)$
  - Tangent-Graeffe + $q-1$ large smooth factor — $O^\flat(T(\log T)^2\log q)$
- Compute the discrete logarithms e_i of ω^{e_i} w.r.t. ω
  - Pohlig-Helmann + $q-1$ large smooth factor — $O^\flat(T\log T\log q)$
- Compute the coefficients c_i using linear algebra
  - Transposed fast multi-point interpolation — $O^\flat(T(\log T)^2\log q)$

Total — $O((L+(\log T)^3)\,T\log q)$

Cantor–Zassenhaus (probabilistic)

$$\Lambda = (x - \alpha_1) \cdots (x - \alpha_t) \in \mathbb{F}_q[x]$$

Cantor–Zassenhaus (probabilistic)

$$\Lambda = (x-\alpha_1)\cdots(x-\alpha_t) \in \mathbb{F}_q[x]$$

$$x \operatorname{rem} \Lambda \stackrel{\sim}{=} (\alpha_1, \ldots, \alpha_t) \in \mathbb{F}_q[x]/(x-\alpha_1) \times \cdots \times \mathbb{F}_q[x]/(x-\alpha_t)$$

$$\downarrow$$

$$x^2 \operatorname{rem} \Lambda \stackrel{\sim}{=} (\alpha_1^2, \ldots, \alpha_t^2) \in \mathbb{F}_q[x]/(x-\alpha_1) \times \cdots \times \mathbb{F}_q[x]/(x-\alpha_t)$$

$$\downarrow$$

$$\vdots$$

$$\downarrow$$

$$R := x^{\frac{q-1}{2}} \operatorname{rem} \Lambda \stackrel{\sim}{=} \left(\alpha_1^{\frac{q-1}{2}}, \ldots, \alpha_t^{\frac{q-1}{2}}\right) \in \mathbb{F}_q[x]/(x-\alpha_1) \times \cdots \times \mathbb{F}_q[x]/(x-\alpha_t)$$

Digression on root finding

Cantor–Zassenhaus (probabilistic)

$$\Lambda = (x-\alpha_1)\cdots(x-\alpha_t) \in \mathbb{F}_q[x]$$

$$\begin{array}{ccl}
x \operatorname{rem} \Lambda & \stackrel{\sim}{=} & (\alpha_1,\dots,\alpha_t) \in \mathbb{F}_q[x]/(x-\alpha_1)\times\cdots\times\mathbb{F}_q[x]/(x-\alpha_t) \\
\downarrow & & \\
x^2 \operatorname{rem} \Lambda & \stackrel{\sim}{=} & (\alpha_1^2,\dots,\alpha_t^2) \in \mathbb{F}_q[x]/(x-\alpha_1)\times\cdots\times\mathbb{F}_q[x]/(x-\alpha_t) \\
\downarrow & & \\
\vdots & & \\
\downarrow & & \\
R := x^{\frac{q-1}{2}} \operatorname{rem} \Lambda & \stackrel{\sim}{=} & \left(\alpha_1^{\frac{q-1}{2}},\dots,\alpha_t^{\frac{q-1}{2}}\right) \in \mathbb{F}_q[x]/(x-\alpha_1)\times\cdots\times\mathbb{F}_q[x]/(x-\alpha_t)
\end{array}$$

$$\gcd(R-1,\Lambda) = \prod_{\alpha_i^{\frac{q-1}{2}}=1}(x-\alpha_i) \qquad \gcd(R+1,\Lambda) = \prod_{\alpha_i^{\frac{q-1}{2}}-1}(x-\alpha_i)$$

Graeffe (deterministic)

Assume $q-1=s2^k$, $s\approx t$ (or even $s\approx t\log t$)

$$\begin{array}{rcl}
\Lambda & = & (x-\alpha_1)\cdots(x-\alpha_t) \\
& \downarrow & \\
G_2(\Lambda) & = & (x-\alpha_1^2)\cdots(x-\alpha_t^2) \\
& \downarrow & \\
& \vdots & \\
& \downarrow & \\
G_{2^k}(\Lambda) & = & (x-\alpha_1^{2^k})\cdots(x-\alpha_t^{2^k})
\end{array}$$

Graeffe (deterministic)

Assume $q-1=s\,2^k$, $s\approx t$ (or even $s\approx t\log t$)

$$\begin{array}{rcl}
\Lambda & = & (x-\alpha_1)\cdots(x-\alpha_t) \\
 & \downarrow & \\
G_2(\Lambda) & = & (x-\alpha_1^2)\cdots(x-\alpha_t^2) \\
 & \downarrow & \\
 & \vdots & \\
 & \downarrow & \\
G_{2^k}(\Lambda) & = & (x-\alpha_1^{2^k})\cdots(x-\alpha_t^{2^k}) \quad \xrightarrow{\mathrm{FFT}_s} \quad \alpha_1^{2^k},\ldots,\alpha_t^{2^k}
\end{array}$$

Digression on root finding

Graeffe (deterministic)

Assume $q-1=s\,2^k$, $s\approx t$ (or even $s\approx t\log t$)

$$
\begin{array}{rcl@{\qquad}c@{\qquad}c}
\Lambda & = & (x-\alpha_1)\cdots(x-\alpha_t) & & \alpha_1,\ldots,\alpha_t \\
 & & \downarrow & & \uparrow \\
G_2(\Lambda) & = & (x-\alpha_1^2)\cdots(x-\alpha_t^2) & & \alpha_1^2,\ldots,\alpha_t^2 \\
 & & \downarrow & & \uparrow \\
 & & \vdots & & \vdots \\
 & & \downarrow & & \uparrow \\
G_{2^k}(\Lambda) & = & (x-\alpha_1^{2^k})\cdots(x-\alpha_t^{2^k}) & \xrightarrow{\mathrm{FFT}_s} & \alpha_1^{2^k},\ldots,\alpha_t^{2^k}
\end{array}
$$

Graeffe (deterministic)

Assume $q-1=s\,2^k$, $s\approx t$ (or even $s\approx t\log t$)

$$\begin{array}{rclcc}
\Lambda & = & (x-\alpha_1)\cdots(x-\alpha_t) & & \alpha_1,\ldots,\alpha_t \\
 & & \downarrow & & \uparrow \\
G_2(\Lambda) & = & (x-\alpha_1^2)\cdots(x-\alpha_t^2) & & \alpha_1^2,\ldots,\alpha_t^2 \\
 & & \downarrow & & \uparrow \\
 & & \vdots & & \vdots \\
 & & \downarrow & & \uparrow \\
G_{2^k}(\Lambda) & = & (x-\alpha_1^{2^k})\cdots(x-\alpha_t^{2^k}) & \xrightarrow{\mathrm{FFT}_s} & \alpha_1^{2^k},\ldots,\alpha_t^{2^k}
\end{array}$$

Complexity

$$\mathsf{F}_{\mathrm{CZ}}(t) \asymp \mathsf{F}_{\mathrm{Gr}}(t) = O^\flat(t(\log t)^2(\log q)^2)$$

Digression on root finding

Graeffe (deterministic)

Assume $q-1=s\,2^k$, $s\approx t$ (or even $s\approx t\log t$)

$$\begin{array}{rcl}
\Lambda & = & (x-\alpha_1)\cdots(x-\alpha_t) & & \alpha_1,\ldots,\alpha_t \\
& \downarrow & & & \uparrow \\
G_2(\Lambda) & = & (x-\alpha_1^2)\cdots(x-\alpha_t^2) & & \alpha_1^2,\ldots,\alpha_t^2 \\
& \downarrow & & & \uparrow \\
& \vdots & & & \vdots \\
& \downarrow & & & \uparrow \\
G_{2^k}(\Lambda) & = & (x-\alpha_1^{2^k})\cdots(x-\alpha_t^{2^k}) & \xrightarrow{\mathrm{FFT}_s} & \alpha_1^{2^k},\ldots,\alpha_t^{2^k}
\end{array}$$

Complexity

$$\mathsf{F}_{\mathrm{CZ}}(t) \asymp \mathsf{F}_{\mathrm{Gr}}(t) = O^\flat(t(\log t)^3\log q), \qquad \omega^r = 1, \quad r \leqslant t^{O(1)}$$

Tangent numbers

$$\mathbb{F}_q[\epsilon]/(\epsilon^2) = \{a + b\epsilon : a, b \in \mathbb{F}_q, \epsilon^2 = 0\}$$

Digression on root finding

Tangent numbers

$$\mathbb{F}_q[\epsilon]/(\epsilon^2) \;=\; \{a + b\,\epsilon : a, b \in \mathbb{F}_q, \epsilon^2 = 0\}$$

Tangent Graeffe (probabilistic)

$$\Lambda(x) \;=\; (x-\alpha_1)\cdots(x-\alpha_t) \;\in\; \mathbb{F}_q[x]$$

$$\tilde{\Lambda}(x) \;:=\; \Lambda(x-\epsilon) \;=\; (x-(\alpha_1+\epsilon))\cdots(x-(\alpha_t+\epsilon)) \;\in\; (\mathbb{F}_q[\epsilon]/(\epsilon^2))[x]$$

Digression on root finding

Tangent numbers

$$\mathbb{F}_q[\epsilon]\,/\,(\epsilon^2) \;=\; \{a + b\,\epsilon : a, b \in \mathbb{F}_q, \epsilon^2 = 0\}$$

Tangent Graeffe (probabilistic)

$$\Lambda(x) \;=\; (x - \alpha_1)\cdots(x - \alpha_t) \;\in\; \mathbb{F}_q[x]$$

$$\tilde{\Lambda}(x) \;:=\; \Lambda(x - \epsilon) \;=\; (x - (\alpha_1 + \epsilon))\cdots(x - (\alpha_t + \epsilon)) \;\in\; (\mathbb{F}_q[\epsilon]\,/\,(\epsilon^2))[x]$$

$$G_{2^k}(\tilde{\Lambda}) \;=\; \left(x - (\alpha_1 + \epsilon)^{2^k}\right)\cdots\left(x - (\alpha_t + \epsilon)^{2^k}\right)$$

Tangent numbers

$$\mathbb{F}_q[\epsilon]/(\epsilon^2) = \{a + b\,\epsilon : a, b \in \mathbb{F}_q, \epsilon^2 = 0\}$$

Tangent Graeffe (probabilistic)

$$\Lambda(x) = (x-\alpha_1)\cdots(x-\alpha_t) \in \mathbb{F}_q[x]$$

$$\tilde{\Lambda}(x) := \Lambda(x-\epsilon) = (x-(\alpha_1+\epsilon))\cdots(x-(\alpha_t+\epsilon)) \in (\mathbb{F}_q[\epsilon]/(\epsilon^2))[x]$$

$$\begin{aligned} G_{2^k}(\tilde{\Lambda}) &= (x-(\alpha_1+\epsilon)^{2^k})\cdots(x-(\alpha_t+\epsilon)^{2^k}) \\ &= (x-(\alpha_1^{2^k}+2^k\alpha_1^{2^k-1}\epsilon))\cdots(x-(\alpha_t^{2^k}+2^k\alpha_t^{2^k-1}\epsilon)) \end{aligned}$$

Tangent numbers

$$\mathbb{F}_q[\epsilon]/(\epsilon^2) \;=\; \{a + b\,\epsilon : a, b \in \mathbb{F}_q, \epsilon^2 = 0\}$$

Tangent Graeffe (probabilistic)

$$\Lambda(x) \;=\; (x - \alpha_1) \cdots (x - \alpha_t) \;\in\; \mathbb{F}_q[x]$$

$$\tilde{\Lambda}(x) \;:=\; \Lambda(x - \epsilon) \;=\; (x - (\alpha_1 + \epsilon)) \cdots (x - (\alpha_t + \epsilon)) \;\in\; (\mathbb{F}_q[\epsilon]/(\epsilon^2))[x]$$

$$\begin{aligned} G_{2^k}(\tilde{\Lambda}) &= (x - (\alpha_1 + \epsilon)^{2^k}) \cdots (x - (\alpha_t + \epsilon)^{2^k}) \\ &= (x - (\alpha_1^{2^k} + 2^k \alpha_1^{2^k - 1} \epsilon)) \cdots (x - (\alpha_t^{2^k} + 2^k \alpha_t^{2^k - 1} \epsilon)) \end{aligned}$$

$$\alpha_i^{2^k} + 2^k \alpha_i^{2^k - 1} \epsilon \quad \rightsquigarrow \quad \alpha_i \;=\; 2^k \frac{\alpha_i^{2^k}}{2^k \alpha_i^{2^k - 1}} \qquad (\text{single root } \alpha_i^{2^k})$$

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3\log(q\,T)\right).$$

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3\log(q\,T)\right).$$

- If $q<2\,T$, then we need to replace K by $\mathbb{F}_{q^s}$ for $s\geqslant\left\lceil\frac{\log T}{\log q}\right\rceil$
- Significantly smaller hidden constant in $O^\flat$

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3\log(q\,T)\right).$$

- If $q<2T$, then we need to replace K by $\mathbb{F}_{q^s}$ for $s\geqslant\left\lceil\frac{\log T}{\log q}\right\rceil$
- Significantly smaller hidden constant in $O^\flat$
- If $L\geqslant(\log T)^2$ and $D\leqslant T^{O(1)}$, then use geometric progression approach

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3 \log(q\,T)\right).$$

- If $q<2T$, then we need to replace K by $\mathbb{F}_{q^s}$ for $s \geqslant \left\lceil \frac{\log T}{\log q} \right\rceil$
- Significantly smaller hidden constant in $O^\flat$
- If $L \geqslant (\log T)^2$ and $D \leqslant T^{O(1)}$, then use geometric progression approach
- If $L < (\log T)^2$ or $D > T^{O(1)}$, then use cyclic extension approach

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3\log(q\,T)\right).$$

- If $q<2\,T$, then we need to replace K by $\mathbb{F}_{q^s}$ for $s\geqslant\left\lceil\frac{\log T}{\log q}\right\rceil$
- Significantly smaller hidden constant in $O^\flat$
- If $L\geqslant(\log T)^2$ and $D\leqslant T^{O(1)}$, then use geometric progression approach
- If $L<(\log T)^2$ or $D>T^{O(1)}$, then use cyclic extension approach
- For $n\times n$ symbolic determinant: $L=n^3$ and $t=n!$

Conclusion so far

Proposition (modulo suitable smoothness assumptions)

We can compute the sparse interpolate of f in time

$$O^\flat\left((L+(\log T)^3)\,T\left(\frac{\log D}{\log T}\right)^3\log(q\,T)\right).$$

- If $q<2T$, then we need to replace K by $\mathbb{F}_{q^s}$ for $s\geqslant\left\lceil\frac{\log T}{\log q}\right\rceil$
- Significantly smaller hidden constant in $O^\flat$
- If $L\geqslant(\log T)^2$ and $D\leqslant T^{O(1)}$, then use geometric progression approach
- If $L<(\log T)^2$ or $D>T^{O(1)}$, then use cyclic extension approach
- For $n\times n$ symbolic determinant: $L=n^3$ and $t=n!$

Problems

- Can we reduce the $(\log T)^3$ factor ?
- If q is small, then can we avoid paying the extension factor $s\geqslant\left\lceil\frac{\log T}{\log q}\right\rceil$?

Part V

FFT-based approach

(Univariate case)

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x]/(x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x] / (x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Most favorable case

- $r \mid (q - 1)$ and $r \approx T$, so that $x^r - 1 = (x - 1)(x - \omega) \cdots (x - \omega^{r-1})$ for some $\omega \in \mathbb{F}_q$

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x]/(x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Most favorable case

- $r \mid (q-1)$ and $r \approx T$, so that $x^r - 1 = (x-1)(x-\omega)\cdots(x-\omega^{r-1})$ for some $\omega \in \mathbb{F}_q$

$$f \operatorname{rem} (x^r - 1) \underset{\text{Inverse FFT}}{\overset{\text{FFT}}{\rightleftarrows}} (f(1), f(\omega), \ldots, f(\omega^{r-1}))$$

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x]/(x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Most favorable case

- $r \mid (q-1)$ and $r \approx T$, so that $x^r - 1 = (x-1)(x-\omega)\cdots(x-\omega^{r-1})$ for some $\omega \in \mathbb{F}_q$

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x]/(x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Most favorable case

- $r \mid (q-1)$ and $r \approx T$, so that $x^r - 1 = (x-1)(x-\omega)\cdots(x-\omega^{r-1})$ for some $\omega \in \mathbb{F}_q$

Next favorable case

- $r \mid (q^s - 1)$ and $r \approx T$ for a small s
- $x^r - 1$ factors into polynomials of small degrees over $\mathbb{F}_q$

Main idea

- Computation of $f \operatorname{rem} (x^r - 1)$ In cyclic extension method:

  Evaluate f over $K[x]/(x^r - 1)$ $\longrightarrow$ Use FFT (or Frobenius FFT)

Most favorable case

- $r \mid (q-1)$ and $r \approx T$, so that $x^r - 1 = (x-1)(x-\omega)\cdots(x-\omega^{r-1})$ for some $\omega \in \mathbb{F}_q$

Next favorable case

- $r \mid (q^s - 1)$ and $r \approx T$ for a small s
- $x^r - 1$ factors into polynomials of small degrees over $\mathbb{F}_q$

Frobenius FFT

- If we need to evaluate $f \in \mathbb{F}_q[x]$ over $\mathbb{F}_{q^s}$ with $s > 1$, then
  - $f(\alpha^q) = f(\alpha)^q$ for all $\alpha \in \mathbb{F}_{q^s}$
  - Compute only one of the values $f(\omega^i), f(\omega^{qi}), \ldots, f(\omega^{q^{s-1}i})$ for each i
  - Use inverse Frobenius FFT to recover $f \operatorname{rem} (x^r - 1)$

Most favorable case only…

- We can pick sufficiently many coprime $r_1, \ldots, r_l \approx T$

Complexity analysis

Most favorable case only…

- We can pick sufficiently many coprime $r_1, \ldots, r_l \approx T$

Proposition (modulo many provisos)

We can compute the sparse interpolation of $f \in \mathbb{F}_q[x]$ in time

$$O^\flat\left((L + \log T)\, T \left(\frac{\log D}{\log T}\right)^2 \log(q\, T)\right)$$

Complexity analysis

Most favorable case only…

- We can pick sufficiently many coprime $r_1, \ldots, r_l \approx T$

Proposition (modulo many provisos)

We can compute the sparse interpolation of $f \in \mathbb{F}_q[x]$ in time

$$O^\flat\left((L + \log T)\, T \left(\frac{\log D}{\log T}\right)^2 \log(q\, T)\right)$$

About constant factors

- The geometric progression method uses $2T - 1$ evaluations
- The FFT-based method uses T evaluations with success rate e^{-1}

Do good orders r exist?

Example for $q = 2^{30}$, $T = 10^6$, and $D = 10^{18}$

$$
\begin{array}{lcl lclcl lcl}
s_1 & = & 1 & r_1 & = & 1549411 & = & 31\cdot 151\cdot 331 & \Lambda_1 & \approx & 1.5\cdot 10^{6} \\
s_2 & = & 2 & r_2 & = & 1047553 & = & 13\cdot 61\cdot 1321 & \Lambda_2 & \approx & 1.6\cdot 10^{12} \\
s_3 & = & 3 & r_3 & = & 1701703 & = & 73\cdot 23311 & \Lambda_3 & \approx & 2.8\cdot 10^{18} \\
s_4 & = & 3 & r_4 & = & 1186911 & = & 3^2\cdot 11\cdot 19\cdot 631 & \Lambda_4 & \approx & 3.2\cdot 10^{24} \\
s_5 & = & 4 & r_5 & = & 1048577 & = & 17\cdot 61681 & \Lambda_5 & \approx & 3.4\cdot 10^{30} \\
s_6 & = & 4 & r_6 & = & 1729175 & = & 5^2\cdot 7\cdot 41\cdot 241 & \Lambda_6 & \approx & 5.9\cdot 10^{36} \\
s_7 & = & 5 & r_7 & = & 1016801 & = & 251\cdot 4051 & \Lambda_7 & \approx & 6.0\cdot 10^{42} \\
s_8 & = & 5 & r_8 & = & 1082401 & = & 601\cdot 1801 & \Lambda_8 & \approx & 6.5\cdot 10^{48} \\
s_9 & = & 5 & r_9 & = & 1108811 & = & 11\cdot 100801 & \Lambda_9 & \approx & 6.6\cdot 10^{53} \\
s_{10} & = & 6 & r_{10} & = & 1134021 & = & 3\cdot 7\cdot 54001 & \Lambda_{10} & \approx & 3.6\cdot 10^{58}
\end{array}
$$

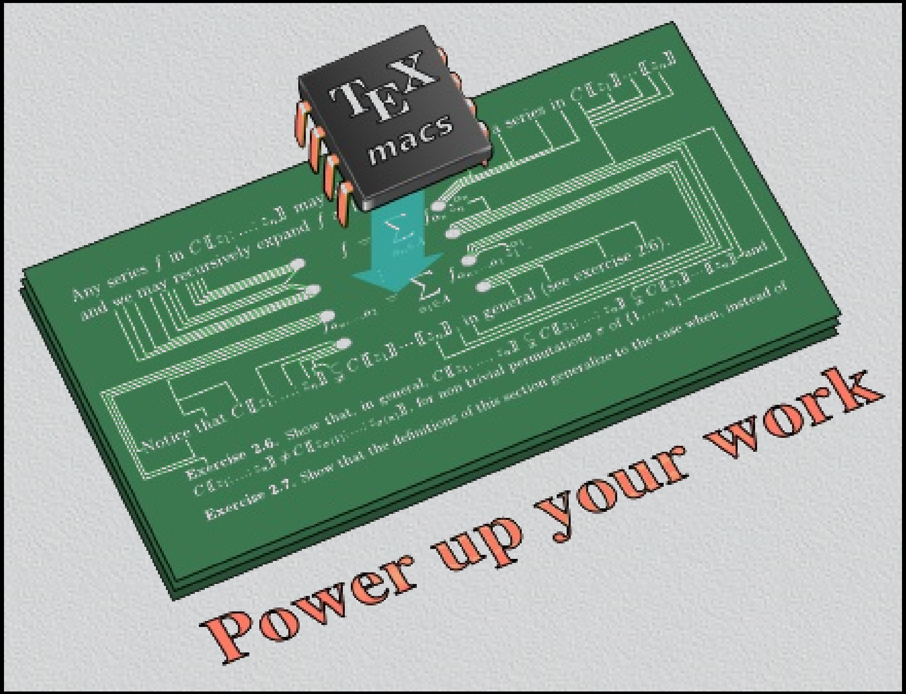
$$\Lambda_i := \operatorname{lcm}(r_1, \ldots, r_i), \qquad D^{\mathrm{e}} \approx 8.5\cdot 10^{48}$$

Do good orders r exist?

Example for (prime) $q = 1299743$, $T = 10^6$, and $D = 10^{18}$

$$
\begin{array}{lll}
s_1 = 1 & r_1 = 1299742 = 2 \cdot 649871 & \Lambda_1 \approx 1.3 \cdot 10^6 \\
s_2 = 2 & r_2 = 1299744 = 2^5 \cdot 3^2 \cdot 4513 & \Lambda_2 \approx 8.4 \cdot 10^{11} \\
s_3 = 4 & r_3 = 1006325 = 5^2 \cdot 40253 & \Lambda_3 \approx 8.5 \cdot 10^{17} \\
s_4 = 4 & r_4 = 1678714 = 2 \cdot 193 \cdot 4349 & \Lambda_4 \approx 7.1 \cdot 10^{23} \\
s_5 = 5 & r_5 = 1690111 = 701 \cdot 2411 & \Lambda_5 \approx 1.2 \cdot 10^{30} \\
s_6 = 6 & r_6 = 1119937 = 7 \cdot 13 \cdot 31 \cdot 397 & \Lambda_6 \approx 1.4 \cdot 10^{36} \\
s_7 = 8 & r_7 = 1196324 = 2^2 \cdot 17 \cdot 73 \cdot 241 & \Lambda_7 \approx 4.0 \cdot 10^{41} \\
s_8 = 9 & r_8 = 1185702 = 2 \cdot 3 \cdot 7^2 \cdot 37 \cdot 109 & \Lambda_8 \approx 1.1 \cdot 10^{46} \\
s_9 = 10 & r_9 = 1376122 = 2 \cdot 11 \cdot 71 \cdot 881 & \Lambda_9 \approx 7.8 \cdot 10^{51} \\
s_{10} = 11 & r_{10} = 3423619 = 23 \cdot 148853 & \Lambda_{10} \approx 2.7 \cdot 10^{58}
\end{array}
$$

Thank you !



http://www.TEXMACS.org