On the use of Frobenius map to accelerate polynomial multiplication with Cantor FFT

Chen-Mou Cheng
chenmou.cheng@gmail.com

Dept. Electrical Engineering
National Taiwan University

Graduate School of Engineering
Osaka University

June 29, 2018

Acknowledgment

- W.-D. Li, M.-S. Chen, P.-C. Kuo, C.-M. Cheng, and B.-Y. Yang. "Frobenius additive fast Fourier transform." In *ISSAC 2018*, New York, NY, USA. July 2018.

Notation

- Throughout this talk:
  - p will always denote a prime number
  - q will always denote a power of a prime number
    - That is, $q = p^d$ for p prime and d positive integer
- We will consider $\mathbb{F}_p$, $\mathbb{F}_q$, *and* $\mathbb{F}_{p^q}$

Notation

- Throughout this talk:
  - p will always denote a prime number
  - q will always denote a power of a prime number
    - ★ That is, $q = p^d$ for p prime and d positive integer
- We will consider $\mathbb{F}_p$, $\mathbb{F}_q$, *and* $\mathbb{F}_{p^q}$
- When in doubt, $p = 2$:-)

The Fourier transform

- The Fourier transform of $f \in \mathbb{F}_q[t]$ is the evaluation of f in some zero set $Z = \{\zeta_1, \ldots, \zeta_n\}$ of $\mathbb{F}_q$: $f(Z) = (f(\zeta_1), \ldots, f(\zeta_n))$
- Let $P(t) = \prod_{z \in Z}(t - z)$ be the vanishing polynomial on Z; then $\mathbb{F}_q^Z \simeq \mathbb{F}_q[t]/(P)$
- If Z has some "nice" (group) structure, then often there are fast algorithms for computing $f(Z)$, e.g.:
  - For $Z = \langle \xi \rangle$, where ξ a primitive n-th root of unity: $P(t) = t^n - 1$
  - For $Z = \mathbb{F}_q$ (as an additive group): $P(t) = t^q - t$
- This way we can turn (polynomial) multiplication in $\mathbb{F}_q[t]/(P)$ into pointwise multiplication in $\mathbb{F}_q^Z$
  - Doesn't matter what P is if the degree of the product is $< n$
- See Dan's paper: "Multidigit multiplication for mathematicians" (and engineers!) for more detail

The Kronecker segmentation

- For $q = p^d$, to multiply $f, g \in \mathbb{F}_p[t]$ such that $\deg fg < n$, write

$$\begin{cases} f(t) = & f_0(t) + f_1(t)T + \cdots + f_{2n/d-1}(t)T^{2n/d-1} = F(T) \\ g(t) = & g_0(t) + g_1(t)T + \cdots + g_{2n/d-1}(t)T^{2n/d-1} = G(T), \end{cases}$$

  where $T = t^{d/2}$ and $\deg f_i, \deg g_i < d/2$
- Interpret f_i, g_i as elements in $\mathbb{F}_q \simeq \mathbb{F}_p[t]/(P)$ for some irreducible P
  - Again doesn't matter what P is, as $\deg f_i g_j < d$
  - Now we can multiply F and G using, e.g., (fast) Fourier transform
  - Need to "carry" to get back $f(t)g(t)$ from $F(T)G(T)$

The Frobenius Fourier transform

- In ISSAC'17, van der Hoeven and Larrieu showed how to avoid the factor-of-two loss using Frobenius map $\phi(x) = x^p$ to accelerate computing $f(Z)$ for $Z \subset \mathbb{F}_q$ and $f \in \mathbb{F}_p[t]$
  - Partitioning Z into a disjoint union of orbits of elements in Z under the action of the Galois group $\mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p)$
  - Choose a representative in each orbit to form a *cross section* Σ; thus

$$Z = \bigcup_{\sigma \in \Sigma} \mathrm{Gal}(\mathbb{F}_q/\mathbb{F}_p) \cdot \sigma = \bigcup_{\sigma \in \Sigma} \{\sigma, \phi(\sigma), \phi^{\circ 2}(\sigma), \dots\}$$

  - Compute $f(\Sigma)$ and get the rest of $f(Z)$ via $f(\phi(\sigma)) = \phi(f(\sigma))$
- Main result: for $q = p^d$, computing $f(Z)$ for $f \in \mathbb{F}_p[t]$ is roughly d times faster than computing $g(Z)$ for $g \in \mathbb{F}_q[t]$, as $|\Sigma| \approx |Z|/n$

Cantor's "FFT" and its derivatives

- Cantor showed how to compute $f(Z)$ for some additive subgroup Z of $\mathbb{F}_{p^q}$ in $O(n(\log n)^2)$ time for $n = |Z|$ via what he called "an analogue of the fast Fourier transform"
  - Based on a tower $\mathbb{F}_p, \mathbb{F}_{p^p}, \mathbb{F}_{p^{p^2}}, \ldots$ of Artin-Schreier extensions of $\mathbb{F}_p$
- Gao and Mateer gave an $O(n \log n \log\log n)$ Cooley-Tukey-style algorithm, a.k.a. *really* fast Fourier transform, when $p = 2$ and $Z = \mathbb{F}_{p^q}$

Cantor's "FFT" and its derivatives

- Cantor showed how to compute $f(Z)$ for some additive subgroup Z of $\mathbb{F}_{p^q}$ in $O(n(\log n)^2)$ time for $n = |Z|$ via what he called "an analogue of the fast Fourier transform"
  - Based on a tower $\mathbb{F}_p, \mathbb{F}_{p^p}, \mathbb{F}_{p^{p^2}}, \ldots$ of Artin-Schreier extensions of $\mathbb{F}_p$
- Gao and Mateer gave an $O(n \log n \log\log n)$ Cooley-Tukey-style algorithm, a.k.a. *really* fast Fourier transform, when $p = 2$ and $Z = \mathbb{F}_{p^q}$
- We showed that *van der Hoeven and Larrieu's idea of using Frobenius map to accelerate polynomial multiplication beautifully generalizes to Cantor-Gao-Mateer-$\cdots$ FFT*

Cantor's construction

- Let $\wp(t) = t^p - t$ be the Artin-Schreier polynomial and

$$s_m(t) := \wp^{\circ m}(t) = \underbrace{(\wp \circ \wp \circ \cdots \circ \wp)}_{m}(t) = \sum_{i=0}^{m} (-1)^{m-i} \binom{m}{i} t^{p^i}$$

- Let W_i be the zero set of $s_i(t) = \prod_{\omega \in W_i}(t - \omega)$; then $s_j(W_i) = W_{i-j}$, and

$$\mathbb{F}_p = W_1 \subset W_2 \subset \cdots \subset \tilde{\mathbb{F}}_p = \bigcup_{i=0}^{\infty} \mathbb{F}_{p^{p^i}}$$

- Since s_i's are linear, W_i's are vector (sub)spaces over $\mathbb{F}_p$
- $\dim_{\mathbb{F}_p} W_i = i$, and W_i is a field iff $i = p^d$ for some integer d

Cantor's basis

- Choose $u_0, u_1, \ldots$ from $\tilde{\mathbb{F}}_p$ such that

$$\phi(u_j) = (u_0 u_1 \cdots u_{j-1})^{p-1} + \text{ [a sum of monomials of lower degree]}$$

- Let $m_k m_{k-1} \cdots m_0$ be the base-p expansion of m and define $y_m = u_0^{m_0} u_1^{m_1} \cdots u_k^{m_k}$
- Cantor showed that $(y_0, y_1, \ldots, y_m)$ is a basis for W_{m+1}, and $y_m \in W_{m+1} - W_m$
- Can Gaussian-eliminate and get a (Cantor) basis $(v_0, v_1, \ldots, v_m)$ such that $\forall i, s(v_{i+1}) = v_i$

A closer look at cosets of W_j in W_i, $0 < j < i$

- Can put the cosets of W_j in W_i into a one-to-one correspondence with the elements in $s_j(W_i) = W_{i-j}$
  - If ω and ω' are representatives from the same coset, then $0 = s_j(\omega - \omega') = s_j(\omega) - s_j(\omega')$, or $s_j(\omega) = s_j(\omega')$
  - Conversely, if ω and ω' are from different cosets, then $s_j(\omega) \neq s_j(\omega')$
- Can label the coset containing ω as $W_j + s_j^{-1}(\alpha)$ for $\alpha = s_j(\omega) \in W_{i-j}$:

$$
\begin{aligned}
W_i &= \bigcup_{\omega_1 \in W_1} W_{i-1} + s_{i-1}^{-1}(\omega_1) & \quad s_i(t) &= \prod_{\omega_1 \in W_1} \Big(s_{i-1}(t) - \omega_1\Big) \\
&= \bigcup_{\omega_2 \in W_2} W_{i-2} + s_{i-2}^{-1}(\omega_2) & &= \prod_{\omega_2 \in W_2} \Big(s_{i-2}(t) - \omega_2\Big) \\
&\;\;\vdots & &\;\;\vdots \\
&= \bigcup_{\omega_{i-1} \in W_{i-1}} W_1 + s^{-1}(\omega_{i-1}) & &= \prod_{\omega_{i-1} \in W_{i-1}} \Big(s(t) - \omega_{i-1}\Big)
\end{aligned}
$$

Cantor's algorithm

- To compute $f(W_m)$, let

$$\mathcal{A}_m = \left\{ f_\omega^{(i)}(t) := f(t) \bmod \left(s_i(t) - \omega\right) : 0 \le i \le m, \omega \in W_{m-i} \right\}$$

- Start from $f_0^{(m)}(t) = f(t)$ and compute $f_\omega^{(m-1)}, \ldots$
  - $s_i(x) - \omega$ divides $s(s_i(x) - \omega) = s_{i+1}(x) - s(\omega)$, so

$$\begin{aligned} f_\omega^{(i)}(t) =& f(t) \bmod \left(s_i(t) - \omega\right) \\ =& \left(f(t) \bmod \left(s_{i+1}(t) - s(\omega)\right)\right) \bmod \left(s_i(t) - \omega\right) \\ =& f_{s(\omega)}^{(i+1)}(t) \bmod \left(s_i(t) - \omega\right) \end{aligned}$$

- Then $f(W_m) = (f_{\omega_1}^{(0)}, f_{\omega_2}^{(0)}, \ldots,)$, the constant polynomials

Gao-Mateer's (a-ha) algorithm

- To evaluate $f_\omega^{(i)}(W_j + s_j^{-1}(\omega))$ for all $\omega \in W_{i-j}$, can set $T = s_j(t)$ and "Taylor-expand" f around it: $f(t) = f_0(t) + f_1(t)T + f_2(t)T^2 + \cdots$
- Again think $p = 2$ and $i = 2j$:

$$
\begin{aligned}
f(t) =&(f_{0,0} + f_{0,1}t + \cdots + f_{0,2^j-1}t^{2^j-1}) \\
&+(f_{1,0} + f_{1,1}t + \cdots + f_{1,2^j-1}t^{2^j-1})s_j(t) \\
&+(f_{2,0} + f_{2,1}t + \cdots + f_{2,2^j-1}t^{2^j-1})s_j^2(t) \\
&\vdots \\
&+(f_{2^{j-1}-1,0} + f_{2^j-1,1}t + \cdots + f_{2^j-1,2^j-1}t^{2^j-1})s_j^{2^j-1}(t)
\end{aligned}
$$

- Observe that $s_j(t) = \omega$ on $W_j + s_j^{-1}(\omega)$, so can recursively break down as Cooley and Tukey did if $i = p^q$

Orbits under the action of $\mathrm{Gal}(\mathbb{F}_{p^q}/\mathbb{F}_p)$

- Let $\mathbb{F}_{p^q}$ be the smallest field containing $\alpha \in \tilde{\mathbb{F}}_p$
- This means that $\alpha^{p^q} = \alpha$ but $\alpha^{p^i} \neq \alpha \, \forall i < q$, so

$$\left|\mathrm{Orb}_\alpha\right| = \left|\left\{\alpha, \phi(\alpha), \phi^{\circ 2}(\alpha), \ldots, \phi^{\circ(q-1)}(\alpha)\right\}\right| = q$$

- Now $\phi = 1 + s$, so

$$\phi^{\circ m} = (1+s)^{\circ m} = \sum_{i=0}^{m} \binom{m}{i} s^{\circ i}$$

- Lemma: $\binom{nq}{q} = n \bmod p$ for $n = 1, 2, \ldots, p-1$

m	1	s	$s^{\circ 2}$	$s^{\circ 3}$	$s^{\circ 4}$	$\dots$	$s^{\circ p}$	$\dots$	$s^{\circ p^2}$
1	1	1							
2	1	2	1						
3	1	3	3	1					
4	1	4	6	4	1				
			$\vdots$						
i	$\binom{i}{0}$	$\binom{i}{1}$	$\binom{i}{2}$	$\binom{i}{3}$	$\binom{i}{4}$	$\dots$			
			$\vdots$						
p	1	0	0	0	0	$\dots$	1		
			$\vdots$						
$2p$	1	0	0	0	0	$\dots$	2	$\dots$	
			$\vdots$						
$3p$	1	0	0	0	0	$\dots$	3	$\dots$	
			$\vdots$						
p^2	1	0	0	0	0	$\dots$	0	$\dots$	1

Our main result

- Let $\mathbb{F}_{p^q}$ be the smallest field containing $\alpha \in \tilde{\mathbb{F}}_p$
- Write $\alpha = \sum_{i=0}^{m} a_i v_i = (a_m a_{m-1} \dots a_0)$, where $q/p \le m < q$, $a_i \in \mathbb{F}_p$, and $a_m \neq 0$
- Theorem:

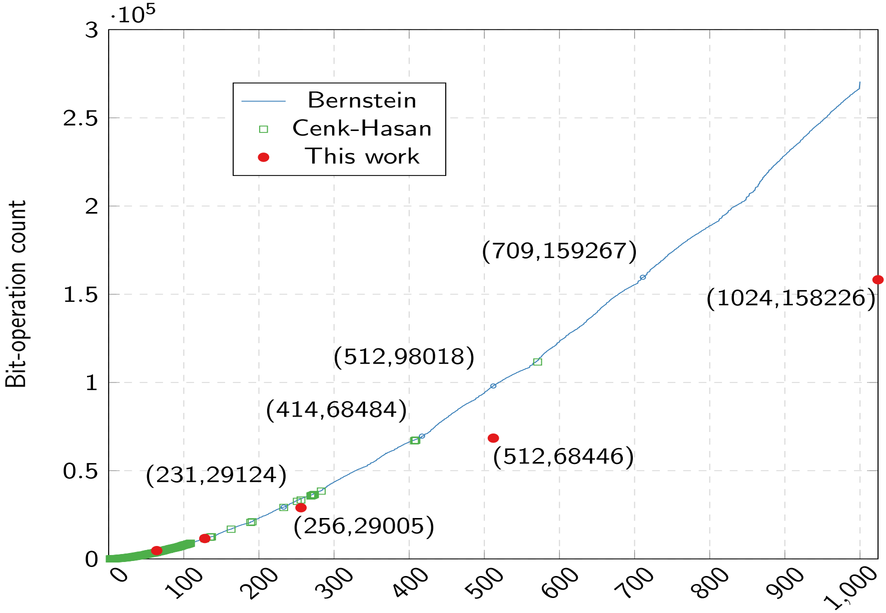
$$\mathrm{Orb}_\alpha = \Big\{ (a_m \underbrace{\underbrace{\underbrace{b_{m-1} X \cdots X b_{m-p}}_{p} X \cdots X b_{m-p^2}}_{p^2} X \cdots X b_{m-q/p}}_{q/p} X \cdots) \Big\},$$

where $b_i \in \mathbb{F}_p$ and X "don't care's"

- Corollary: Fixing b_i's and varying a_m and X's, we get a cross section for $W_{m+1} - W_m$

Partial cross sections of W_i

- For multiplying $f, g \in \mathbb{F}_p[t]$ with $\deg fg < n$, we just need to evaluate on a set Z of size n
- Idea: since Frobenius map gives us a factor of q gain, let's choose Z as the union of some cosets of W_j in W_i such that $Z \subset \mathbb{F}_{p^q} - \mathbb{F}_{p^{q/p}}$ and $|Z| = n/q$
- Furthermore, if we choose j a power of p and $i \geq j + q/p$, then the action of $\mathrm{Gal}(\mathbb{F}_{p^q}/\mathbb{F}_p)$ will leave W_j "intact," which greatly simplifies software implementation
- For hardware implementation, can use full-fledged cross sections


$\cdot 10^5$
3
2.5
2
1.5
1
0.5
0
Bit-operation count
0
100
200
300
400
500
600
700
800
900
1,000
Bernstein
Cenk-Hasan
This work
(709,159267)
(1024,158226)
(512,98018)
(414,68484)
(512,68446)
(231,29124)
(256,29005)

Concluding remarks

- Can avoid Kronecker segmentation in polynomial multiplication in $\mathbb{F}_p[t]$ (for small-ish p) by working in some extension field $\mathbb{F}_q$ of $\mathbb{F}_p$, with help from Frobenius map
- Sorry didn't talk about all the detail due to time constraints
- If your favorite PQC scheme involves such polynomial multiplication, then please come talk to us!

Thanks!

- Questions or comments?