Fast Multipoint Evaluation

MCA Ch. 10.

September 9, 2021 5:24 PM

Let $f \in F[x]$, $\alpha_1, \alpha_2, \dots, \alpha_n \in F$ with $\deg f < n$ and $n = 2^k$.
How fast can we compute $f(\alpha_1), f(\alpha_2), \dots, f(\alpha_n)$?

Horner's rule: n^2 mults and n^2 adds in F.

$$f = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n = a_0 + x\cdot(a + x(a_2 + \dots + x\cdot a_n)\dots)).$$

$f(\alpha_1)$ $\qquad$ $n\times + n+$.

The division tree.

$\le \deg \frac{n}{2}-1$ $\qquad$ $\deg \le n-1$ $\qquad$ $\deg n/2$

$$r_1 = f \bmod \prod_{i=1}^{n/2}(x-\alpha_i) \qquad r_2 = f \bmod \prod_{i=\frac{n}{2}+1}^{n}(x-\alpha_i) \qquad 2D(\tfrac{n}{2})$$

$$r_{11} = r_1 \bmod \prod_{i=1}^{n/4}(x-\alpha_i) \qquad r_{21} = r_2 \bmod \prod_{i=n/2+1}^{3/4 n}(x-\alpha_i) \qquad 4D(\tfrac{n}{4})$$

$$r_{12} = r_1 \bmod \prod_{i=\frac{n}{4}+1}^{n/2}(x-\alpha_i) \qquad r_{22} = r_2 \bmod \prod_{i=3n/4}^{n}(x-\alpha_i)$$

$$8D(\tfrac{n}{8})$$

$\vdots$

$\deg 0$ $\qquad$ $\deg \le 1.$ $\qquad$ $\deg 1$

$$r_{\square 1} = r_{\square} \bmod (x-\alpha_1) = f(\alpha_1)?$$
$$r_{\square 2} = r_{\square} \bmod (x-\alpha_2) = f(\alpha_2)? \qquad n\,D(1)$$

Let $T(n)$ be the cost of the divisions. Suppose we use fast $\div$
So $D(n) \le 4M(n) + O(n)$. where $M(n)$ is the cost of mult. two polynomials of degree $\le n$.

$$\begin{aligned}
T(n) &= 2D(\tfrac{n}{2}) + 4D(\tfrac{n}{4}) + \dots + nD(1) \quad +O(n) \\
&\le 4\left[2M(\tfrac{n}{2}) + 4M(\tfrac{n}{4}) + \dots + nM(1)\right] \\
&< 4\left[M(n) + 1\cdot M(n) + M(n) + \dots + M(n)\right] \\
&< 4\log_2 n\, M(n) \in O(M(n)\log_2 n)
\end{aligned}$$

Assume $2M(\tfrac{n}{2}) < 1\cdot M(n)$.

What about computing the products?
What if $\deg f \ge n$? First compute $r_0 \leftarrow f \bmod \prod_{i=1}^{n}(x-\alpha_i)$ $\qquad$ $\deg < n.$

Claim. $\displaystyle \Big(\underbrace{f \bmod \prod_{i=1}^{4}(x-\alpha_i)}_{h}\Big) \bmod \underbrace{(x-\alpha_1)(x-\alpha_2)}_{g} = \underbrace{f \bmod (x-\alpha_1)(x-\alpha_2)}_{\hat{g}}$

Lemma Let $f, g, h \in F[x]$ where $g \mid h$.

Lemma Let $f, g, h \in F[x]$ where $g \mid h$.
Then $f \bmod g = (f \bmod h) \bmod g$.

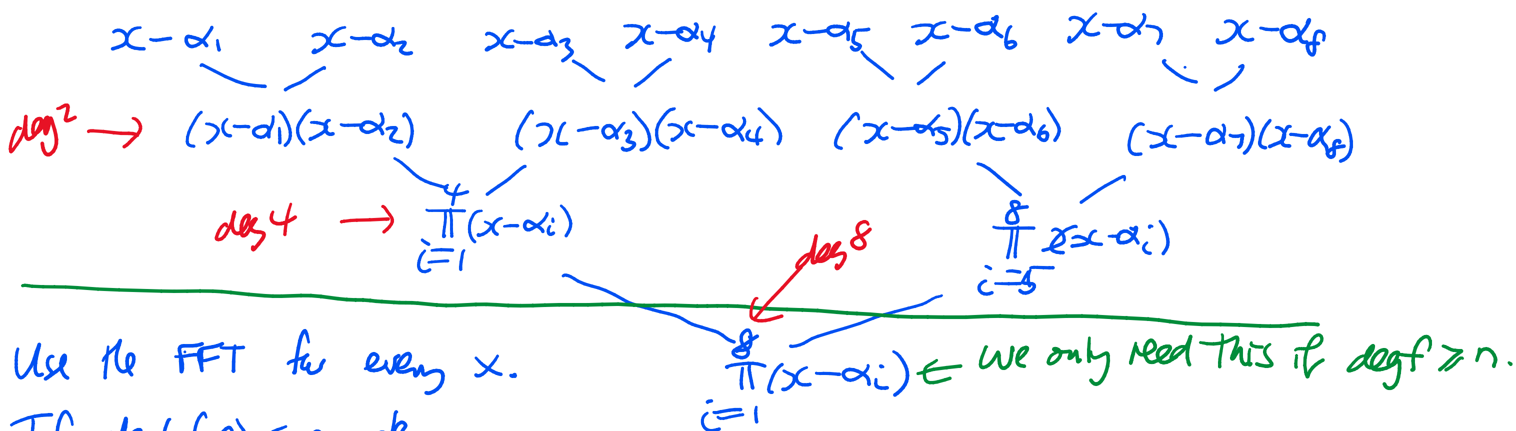
Proof. $f \div h$: Let $f = hq + r$ with $r = 0$ or $\deg r < \deg h$.

$g \mid h \Rightarrow h = g \cdot p$ for some $p \in F[x]$.

$$f \bmod g = hq + r \bmod g = gpq + r \bmod g = r \bmod g.$$

$$\begin{aligned}(f \bmod h) \bmod g &= (hq + r \bmod h) \bmod g \\ &= (r \bmod h) \bmod g \\ &= r \bmod g\end{aligned}$$

The product tree Π.



Use the FFT for every x.

If $\deg(f \cdot g) < n = 2^k$ we can use 3 FFTs of size n. But $\deg(f \cdot g) = 2^k$.
Do we need 3 FFTs of size $2n$?

Let $T(n)$ be the cost of the mults in the Π tree.

$$T(n) = \frac{n}{2}M(1) + \frac{n}{4}M(2) + \cdots + 4M\left(\frac{n}{8}\right) + 2M\left(\frac{n}{4}\right).. \quad \text{Assuming } \boxed{M(n) > 2M\left(\frac{n}{2}\right)}$$

$$\Rightarrow T(n) < M\left(\frac{n}{2}\right) + \ldots + M\left(\frac{n}{2}\right) + M\left(\frac{n}{2}\right)$$

$$= (\log_2 n - 1) M\left(\frac{n}{2}\right) \in O(M(n) \log n).$$

So the total cost to compute $f(\alpha_i)$ for $1 \leq i \leq n$ is $O(M(n) \log n)$.

If we need to evaluate polynomials $f_1, f_2, \ldots$ at $x = \alpha_1, \alpha_2, \ldots, \alpha_n$ we only need to compute the Π once.

If we need to evaluate polynomials $f_1, f_2, \dots$ at $x = \alpha_1, \alpha_2, \dots, \alpha_n$ we only need to compute the T once.

Let $a, b \in F[x]$, $n = 2^k$. Using the FFT we can compute $c = a \times b$ using 3 FFTs of size n provided $\deg c < n$.

Consider $(2x^2 + 3x + 4)(3x^2 + x + 5) = (6x^4 + \boxed{\underbrace{11x^3 + 25x^2 + 19x + 50}_{\bar{C}(x)}})$

$c_n = 6$.

Need 3 FFTs of size $n = 8$.

Consider using $n = 4$.

CASE $\deg a < n$, $\deg b < n$ and $\deg c = n$.

Let $c(x) = c_0 + c_1 x + \dots + c_{n-1} x^{n-1} + c_n x^n = \bar{C}(x) + c_n x^n$

Note $C(\omega^i) = \bar{C}(\omega^i) + c_n \omega^{in} = \bar{C}(\omega^i) + c_n$.

$\deg a < n \Rightarrow FFT(a) = [a(1), a(\omega), \dots, a(\omega^{n-1})] = A$

$\deg b < n \Rightarrow FFT(b) = [b(1), b(\omega), \dots, b(\omega^{n-1})] = B$.

$$C_i = A_i \cdot B_i = a(\omega^i) \cdot b(\omega^i) = c(\omega^i).$$

$$C = [\bar{C}(1) + c_n, \bar{C}(\omega) + c_n, \bar{C}(\omega^2) + c_n, \dots, \bar{C}(\omega^{n-1}) + c_n]$$

$FFT(A) = V(\omega) \cdot A$

$FFT^{-1}(C) = \frac{1}{n} \cdot V(\omega^{-1}) \cdot C$

$\downarrow FFT^{-1}$

$A(u+v) = Au + Av$.

$$FFT^{-1}[\bar{C}(1), \bar{C}(\omega), \dots, \underbrace{\bar{C}(\omega^{n-1})}_{\deg n-1}] + FFT^{-1}([c_n, c_n, \dots, c_n]).$$

$$\downarrow FFT^{-1} \qquad \downarrow FFT^{-1}$$

$$= [c_0 \; c_1 \; \dots \; c_{n-1}] + [c_n, 0, 0, \dots, 0]$$

$FFT([c_n, 0, 0, \dots, 0]) = [f(1), f(\omega), f(\omega^2), \dots, f(\omega^{n-1})] = [c_n, c_n, c_n, \dots, c_n]$

$f(x) = c_n$

$$= [c_0 + c_n, c_1, c_2, \dots, c_{n-1}].$$

We want $c(x) = c_0 + c_1 x + \dots + c_n x^n$.

We can recover $c(x)$ from the result of mult. $a \times b$ using an FFT of size n with 1 mult. and 1 subtraction.

$\deg n/2$ $\quad \deg \frac{n}{2}$