Notes on the FFT

October 11, 2023 4:55 PM

How can we solve a first order recurrence.

$$M(n) = 2\,M(n/2) + n/2 \quad \text{with } M(1)=0,\ n=2^k.$$

$$\begin{aligned}
2^0 \cdot M(n) &= 2\,\cancel{M(n/2)} + \frac{n}{2} \\
2^1\,\cancel{M(\tfrac{n}{2})} &= 2^2\,\cancel{M(n/4)} + 2(n/4) = n/2 \\
2^2\,\cancel{M(\tfrac{n}{4})} &= 2^3\,\cancel{M(n/8)} + (n/8)\cdot 2^2 = n/2 \\
&\ \vdots \\
2^{k-1}\,\cancel{M(2)} &= 2^k\,\cancel{M(1)} + 1\cdot 2^{k-1} = n/2 \\
2^k\,\cancel{M(1)} &= 0 \\
\hline
+ \quad M(n) &= \frac{n}{2}\cdot k = \frac{n}{2}\cdot \log_2 n.
\end{aligned}$$

Recap.

$$\begin{array}{ccccc}
 & c(x) & = & a(x)\cdot b(x) & \leftarrow \text{coefficient representation} \\
\text{interp.} & \uparrow F_w^{-1} & & \text{eval}\downarrow F_w \quad \downarrow\text{eval } F_w & \\
 & c(w^i) & = & a(w^i)\cdot b(w^i) & \leftarrow \text{points representation.}
\end{array}$$

Let $F_w : F^n \to F^n$ denote the Fourier transform.

Let $F_w^{-1}(a) = \frac{1}{2} F_{w^{-1}}(a)$ denote the inverse transform.

Since the F_w is a Linear Transformation, for constants $\alpha, \beta \in F$ and $A, B \in F^n$ representing $a, b \in F[x]$,

$\frac{1}{2} n \log_2 n$ mults.

$$F_w(\alpha \cdot A + \beta \cdot B) = \alpha \cdot F_w(A) + \beta \cdot F_w(B).$$

scalar $\times$ F^n

n mults n adds

Application.

Let $A \in F[x]^{2\times 2}$ and $R \in F[x]^2$.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}.$$

$$= \begin{bmatrix} A_{11}\cdot R_1 + A_{21}\cdot R_2 \\ A_{21}\cdot R_1 + A_{22}\cdot R_2 \end{bmatrix} = \begin{bmatrix} F_\omega^{-1}(F_\omega(A_{11})\cdot F_\omega(R_1)) + F_\omega^{-1}(F_\omega(A_{21})\cdot F_\omega(R_2)) \\ \end{bmatrix}$$

If we do each of the 4 polynomial multiplications using the FFT we will do $4 \cdot 3 = 12$ FFTs.

Consider

$$A \cdot R = \begin{bmatrix} F_\omega^{-1}\left(F_\omega(A_{11}) \cdot F_\omega(R_1) + F_\omega(A_{21}) \cdot F_\omega(R_2)\right) \\ F_\omega^{-1}\left(F_\omega(A_{21}) \cdot F_\omega(R_1) + F_\omega(A_{22}) \cdot F_\omega(R_2)\right) \end{bmatrix}$$

↑ n mults ↑ n adds

We can compute $A \cdot R$ using $6+2=8$ FFTs.

This saves $1/3$ of the FFTs.

How can we compute $\alpha \cdot ab + \beta \cdot a \cdot c + \gamma \cdot b \cdot c$ for $\alpha, \beta, \gamma \in F$ and polynomials a, b, c?