# The Fast Fourier Transform Andreas Klappenecker

#### Fourier Transform in Matrix Form

For a polynomial  $A(x) = a_0 + a_1x + ... + a_{n-1}x^{n-1}$  of degree n-1, a conversion from coefficient representation to p.v. representation at n distinct points 1,  $\omega$ , ...,  $\omega^{n-1}$  can be done as follows:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

#### Fast Fourier Transform

Evaluate a degree n-1 polynomial  $A(x) = a_0 + ... + a_{n-1} x^{n-1}$  at its n<sup>th</sup> roots of unity:  $\omega^0$ ,  $\omega^1$  1, ...,  $\omega^{n-1}$ .

Divide. Divide the polynomial into even and odd powers.

$$A(x) = A_{even}(x^2) + x A_{odd}(x^2).$$

Conquer. Evaluate  $A_{\text{even}}(x)$  and  $A_{\text{odd}}(x)$  at the ½n<sup>th</sup> roots of unity:  $v^0$ ,  $v^1$ , ...,  $v^{n/2-1}$ . Combine.

$$A(\omega^{k}) = A_{even}(v^{k}) + \omega^{k} A_{odd}(v^{k}), \quad 0 \le k < n/2$$

$$A(\omega^{k+n/2}) = A_{even}(v^{k}) - \omega^{k} A_{odd}(v^{k}), \quad 0 \le k < n/2$$

## Fourier Transform in Matrix Form

Split into even and odd powers (green and gray)

Each gray column vector is two copies of the same vector

$$\begin{pmatrix} A(1) \\ A(\omega) \\ \vdots \\ A(\omega^{n-1}) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{bmatrix} 1 \\ \omega \\ \vdots \\ \omega^{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ \omega^2 \\ \vdots \\ \omega^{2(n-1)} \end{bmatrix} \cdots \begin{bmatrix} 1 \\ \omega^{n-1} \\ \vdots \\ \omega^{(n-1)^2} \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

Realize that 1,  $\omega$ , ...,  $\omega^{n/2-1}$ , -1,  $-\omega$ , ...,  $-\omega^{n/2-1}$  (upper/lower part)

## Reorganization

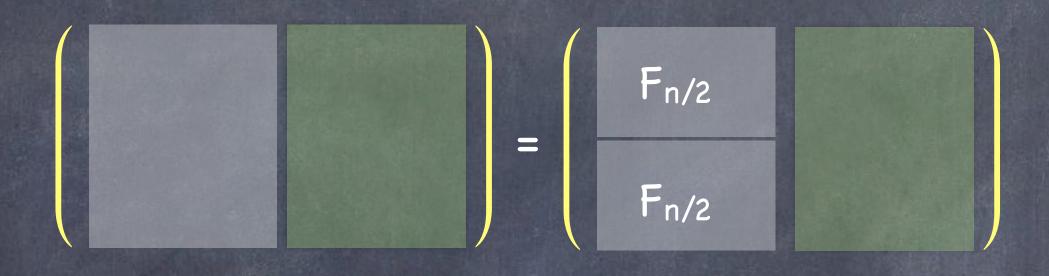
$$\begin{pmatrix} A(1) \\ A(\omega) \\ \vdots \\ A(\omega^{n-1}) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ \omega \\ \vdots \\ \omega^{n-1} \end{pmatrix} \begin{pmatrix} 1 \\ \omega^2 \\ \vdots \\ \omega^{2(n-1)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

$$\vdots \\ a_{n-1} \end{pmatrix}$$

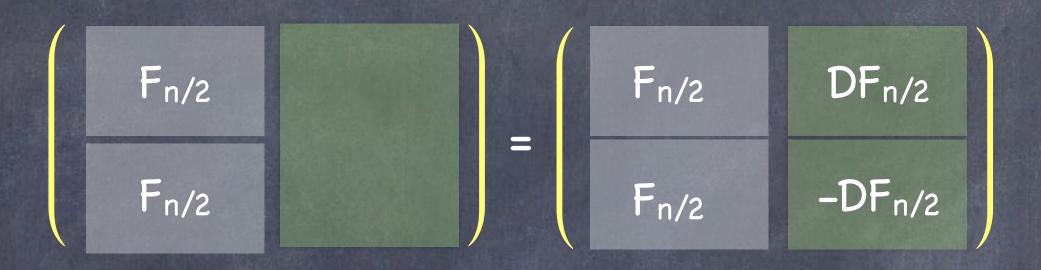
$$\vdots \\ a_{n-1} \end{pmatrix}$$
Put gray columns

Idea: Put grey columns together, and green columns together

## Reorganization



## Reorganization



$$D = \operatorname{diag}(1, \omega, \omega^2, ..., \omega^{n/2-1})$$

### Fast Fourier Transform

$$F_n = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_{n/2} & 0 \\ 0 & F_{n/2} \end{pmatrix} P$$

$$D = \operatorname{diag}(1, \omega, \omega^2, ..., \omega^{n/2-1})$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

#### Fast Fourier Transform

Evaluate a degree n-1 polynomial  $A(x) = a_0 + ... + a_{n-1} x^{n-1}$  at its n<sup>th</sup> roots of unity:  $\omega^0$ ,  $\omega^1$  1, ...,  $\omega^{n-1}$ .

Divide. Divide the polynomial into even and odd powers.

$$A(x) = A_{even}(x^2) + x A_{odd}(x^2).$$

Conquer. Evaluate  $A_{\text{even}}(x)$  and  $A_{\text{odd}}(x)$  at the ½n<sup>th</sup> roots of unity:  $v^0$ ,  $v^1$ , ...,  $v^{n/2-1}$ . Combine.

$$A(\omega^{k}) = A_{even}(v^{k}) + \omega^{k} A_{odd}(v^{k}), \quad 0 \le k < n/2$$

$$A(\omega^{k+n/2}) = A_{even}(v^{k}) - \omega^{k} A_{odd}(v^{k}), \quad 0 \le k < n/2$$

## FFT Algorithm

```
FFT (n, a_0, a_1, ..., a_{n-1}) {
     if (n == 1) return a_0
     (e_0, e_1, ..., e_{n/2-1}) \leftarrow FFT(n/2, a_0, a_2, a_4, ..., a_{n-2})
     (d_0, d_1, ..., d_{n/2-1}) \leftarrow FFT(n/2, a_1, a_3, a_5, ..., a_{n-1})
     for k = 0 to n/2 - 1 {
          \omega^{k} \leftarrow e^{2\pi i k/n}
          y_k \leftarrow e_k + \omega^k d_k
         y_{k+n/2} \leftarrow e_k - \omega^k d_k
     return (y_0, y_1, ..., y_{n-1})
```

## Summary

- The FFT evaluates a polynomial of degree n-1 at n-th roots of unity in O(n log n) steps.
- Inverse of FFT just as fast.
- Find more details in CLRS or Kleinberg/Tardos.