Algorithm Theory
3 Fast Fourier Transformation

Christian Schindelhauer
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Wintersemester 2007/08
Chapter 3

Fast Fourier Transformation
Polynomials

Polynomials $p$ over real numbers with a variable $x$

$$p(x) = a_n x^n + \ldots + a_1 x^1 + a_0$$

$a_0, \ldots, a_n \in R$, $a_n \neq 0$: coefficients of $p$

$n = \text{degree of } p$: highest power of $x$ in $p$

Example:

$$p(x) = 3x^3 - 15x^2 + 18x$$

Set of all polynomials over real polynomials: $R[x]$
2. Operations on Polynomials

\[ p, q \in \mathbb{R}[x] \]

\[ p(x) = a_n x^n + \ldots + a_1 x^1 + a_0 \]

\[ q(x) = b_n x^n + \ldots + b_1 x^1 + b_0 \]

1. Addition

\[ p(x) + q(x) = (a_n x^n + \ldots + a_0) + (b_n x^n + \ldots + b_0) \]

\[ = (a_n + b_n) x^n + \ldots + (a_1 + b_1) x^1 + (a_0 + b_0) \]
2. Multiplication

\[ p(x)q(x) = (a_n x^n + \ldots + a_0)(b_n x^n + \ldots + b_0) = c_{2n} x^{2n} + \ldots + c_1 x^i + c_0 \]

\[ c_i : \text{Which monomial products have degree } i ? \]

\[ \Rightarrow c_i = \sum_{j=0}^{i} a_j b_{i-j} \quad i = 0, \ldots, 2n. \]

\[ a_{n+1} = \ldots = a_{2n} = 0, b_{n+1} = \ldots = b_{2n} = 0 \]

Polynomial ring \( R[x] \)
Operation on Polynomials

3. Evaluation at $x_0$: Horner-Schema

$$p(x_0) = \left( \ldots (a_n x_0 + a_{n-1}) x_0 + \ldots + a_1 \right) x_0 + a_0$$

Runtime: $O(n)$
Representations of Polynomials

\( p(x) \in R[x] \)

Possible representations of \( p(x) \):

1. **Coefficient representation**

\[
p(x) = a_n x^n + \ldots + a_1 x^1 + a_0
\]

**Example:**

\[
p(x) = 3x^3 - 15x^2 + 18x
\]
Representations of Polynomials

2. Root representation

\[ p(x) \in R[x] \]

\[ p(x) = a_n (x - x_1) \ldots (x - x_n) \]

Beispiel:

\[ p(x) = 3x(x - 2)(x - 3) \]
Representations of Polynomials

3. Point-value representation

Interpolation lemma

Any polynomial $p(x)$ over $\mathbb{R}[x]$ of degree $n$ is uniquely defined by $n+1$ pairs $(x_i, p(x_i))$, where $i = 0, \ldots, n$ and $x_i \neq x_j$ für $i \neq j$

Beispiel:
The polynomial

$$p(x) = 3x(x - 2)(x - 3)$$

is determined by the point-values $(0,0), (1,6), (2,0), (3,0)$. 
Operations on Polynomials

\( p, q \in R[x], \ \text{Grad}(p) = \text{Grad}(q) = n \)

- **Coefficient representation**
  - Addition: \( O(n) \)
  - Product: \( O(n^2) \)
  - Evaluation at \( x_0 \): \( O(n) \)

- **Point-value representation**
  \[
  p = (x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)
  \]
  \[
  q = (x_0, z_0), (x_1, z_1), \ldots, (x_n, z_n)
  \]
Operations on Polynomials

Addition:
\[ p + q = (x_0, y_0 + z_0), (x_1, y_1 + z_1), \ldots, (x_n, y_n + z_n) \]
Runtime: \( O(n) \)

Multiplication:
\[ p \cdot q = (x_0, y_0 \cdot z_0), (x_1, y_1 \cdot z_1), \ldots, (x_n, y_n \cdot z_n) \]
(Condition: \( n \geq \text{degree}(pq) \))
Runtime: \( O(n) \)

Evaluation at \( x' \):
Convert polynomial to coefficient representation
(Interpolation)
Polynomial Multiplication

Multiplication of two polynomials $p, q$ of degree $< n$
$p,q$ of degree $n-1$, $n$ coefficients

\[
\text{Evaluation: } x_0, x_1, \ldots, x_{2n-1}
\]
2n point-value pairs $(x_i, p(x_i))$ and $(x_i, q(x_i))$

\[
\text{Pointwise multiplication}
\]
2n point-value pairs $(x_i, pq(x_i))$

\[
\text{Interpolation}
\]
$pq$ of degree $2n-2$, $2n-1$ coefficients
Divide and Conquer Approach

Idea: (for even $n$)

\[
p(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1} = a_0 + a_2x^2 + \ldots + a_{n-2}x^{n-2} + a_1x + a_3x^3 + \ldots + a_{n-1}x^{n-1}
\]

\[
= a_0 + a_2x^2 + \ldots + a_{n-2}(x^2)^{(n-2)/2} + x\left(a_1 + a_3x^2 + \ldots + a_{n-1}(x^2)^{(n-2)/2}\right)
\]

\[
= p_0(x^2) + xp_1(x^2)
\]

$p_0(x) = a_0 + a_2x + \ldots + a_{n-2}x^{(n-2)/2}$

$p_1(x) = a_1 + a_3x + \ldots + a_{n-1}x^{(n-2)/2}$

Select $x_0, \ldots, x_{2n-1}$ such that the computations of $p(x_k)$ and $p(x_{k+n})$ are almost identical.
Representations of $p(x)$

Assume: $\deg(p) < n$

3a. Values of the $n$ powers of the principal $n$th root of unity

$$\omega_n = e^{2\pi i/n}$$

$$i = \sqrt{-1} \quad e^{2\pi i} = 1$$

Power of $\omega_n$ (roots of unity):

$$1 = \omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$$
Discrete Fourier Transform

The values $p(\omega_n^i)$ uniquely define $p$ if $\deg(p)<n$.

Discrete Fourier Transformation (DFT)

\[ DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})) \]

Example: $n=4$

\[ e^{ix} = \cos x + i \sin x \]
\[ \omega_4^0 = e^{0i} = \cos(0) + i \sin(0) = 1 \]
\[ \omega_4^1 = e^{2\pi i/4} = \cos(\pi/2) + i \sin(\pi/2) = i \]
\[ \omega_4^2 = (e^{2\pi i/4})^2 = \cos\pi + i \sin\pi = -1 \]
\[ \omega_4^3 = (e^{2\pi i/4})^3 = \cos(3\pi/2) + i \sin(3\pi/2) = -i \]
Evaluation at the Unity Roots

\[ \omega_4^1 = i \]
\[ \omega_4^2 = -1 \]
\[ \omega_4^3 = -i \]
\[ \omega_4^0 = 1 \]
Evaluation at the Unity Roots

\[ p(x) = 3x^3 - 15x^2 + 18x \]

\[
\begin{align*}
(\omega_4^0, p(\omega_4^0)) &= (1, p(1)) = (1, 6) \\
(\omega_4^1, p(\omega_4^1)) &= (i, p(i)) = (i, 15 + 15i) \\
(\omega_4^2, p(\omega_4^2)) &= (-1, p(-1)) = (-1, -36) \\
(\omega_4^3, p(\omega_4^3)) &= (-i, p(-i)) = (-i, 15 - 15i)
\end{align*}
\]

\[ DFT_4(p) = (6, 15 + 15i, -36, 15 - 15i) \]
Multiplication of Polynomials

Computation of the product of $p$ and $q$ of degree $< n$

$p, q$ of degree $n-1$, $n$ coefficients

Evaluation: $\omega_0^{2n}, \omega_1^{2n}, \ldots, \omega_0^{2n-1}$

2n point-value pairs $\left(\omega_i^{2n}, p\left(\omega_i^{2n}\right)\right)$ and $\left(\omega_i^{2n}, q\left(\omega_i^{2n}\right)\right)$

Pointwise multiplication

2n point-value pairs $\left(\omega_i^{2n}, pq\left(\omega_i^{2n}\right)\right)$

Interpolation

$pq$ of degree $2n-2$, $2n-1$ coefficients
Properties of the Unity Roots

\( \omega^0_{2n}, \omega^1_{2n}, \ldots, \omega^{2n-1}_{2n} \) form a multiplicative group

Cancellation lemma
For all \( n > 0, 0 \leq k \leq n, \) and \( d > 0 \) we have

\[ \omega^{dk}_{dn} = \omega^k_n \]

Proof:

\[ \omega^{dk}_{dn} = e^{2\pi i dk/(dn)} = e^{2\pi ik/n} = \omega^k_n \]

Therefore:

\[ \omega^n_{2n} = \omega^1_2 = -1 \]
Discrete Fourier Transform

\[ DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1})) \]

Fast Fourier Transform:
Computation of \( DFT_n(p) \) using Divide-and-Conquer
Discrete Fourier Transform

Idee: (for even $n$)

$$p(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1}$$

$$= a_0 + a_2x^2 + \ldots + a_{n-2}x^{n-2} +$$

$$a_1x + a_3x^3 + \ldots + a_{n-1}x^{n-1}$$

$$= a_0 + a_2x^2 + \ldots + a_{n-2}\left(x^2\right)^{(n-2)/2} +$$

$$x\left(a_1 + a_3x^2 + \ldots + a_{n-1}\left(x^2\right)^{(n-2)/2}\right)$$

$$= p_0(x^2) + xp_1(x^2)$$

$$p_0(x) = a_0 + a_2x + \ldots + a_{n-2}x^{(n-2)/2}$$

$$p_1(x) = a_1 + a_3x + \ldots + a_{n-1}x^{(n-2)/2}$$
Discrete Fourier Transform

Evaluation for \( k = 0, \ldots, n - 1 \)

\[
p(\omega_n^k) = p_0\left((\omega_n^k)^2\right) + \omega_n^k p_1\left((\omega_n^k)^2\right) = \begin{cases} 
p_0(\omega_n^k) + \omega_n^k p_1(\omega_n^k), & \text{if } k < n/2 \\
p_0(\omega_n^{k-n/2}) + \omega_n^k p_1(\omega_n^{k-n/2}), & \text{if } k \geq n/2 \end{cases}
\]

\[
DFT_n(p) = (p_0(\omega_{n/2}^0), \ldots, p_0(\omega_{n/2}^{n/2-1}), \ p_0(\omega_{n/2}^0), \ldots, p_0(\omega_{n/2}^{n/2-1}))
\]

\[
+ \ (\omega_n^0 p_1(\omega_{n/2}^0), \ldots, \omega_n^{n/2-1} p_1(\omega_{n/2}^{n/2-1}), \ \omega_n^{n/2} p_1(\omega_{n/2}^0), \ldots, \omega_n^{n-1} p_1(\omega_{n/2}^{n/2-1})))
\]
Discrete Fourier Transform

Example:

\[ p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0) \]
\[ p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1) \]
\[ p(\omega_4^2) = p_0(\omega_2^0) + \omega_4^2 p_1(\omega_2^0) \]
\[ p(\omega_4^3) = p_0(\omega_2^1) + \omega_4^3 p_1(\omega_2^1) \]
Computation of $DFT_n$

$$DFT_n(p) = \left( p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1}) \right)$$

**Simple case:** $n = 1$ ($\text{degree}(p) = n - 1 = 0$)

$$DFT_1(p) = a_0$$

**General case:**

- **Divide:**
  Divide $p$ in $p_0$ and $p_1$

- **Conquer:**
  Compute $DFT_{n/2}(p_0)$ and $DFT_{n/2}(p_1)$ recursively

- **Merge:**
  Compute for $k = 0, \ldots, n-1$:

  $$DFT_n(p)_k = (DFT_{n/2}(p_0), DFT_{n/2}(p_0))_k + \omega_n^k \times (DFT_{n/2}(p_1), DFT_{n/2}(p_1))_k$$
Further Improvement

\[ p(\omega_n^k) = \begin{cases} 
  p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k) & \text{if } k < n/2 \\
  p_0(\omega_{n/2}^{k-n/2}) + \omega_n^k p_1(\omega_{n/2}^{k-n/2}) & \text{if } k \geq n/2
\end{cases} \]

Hence, if \( k < n/2 \):

\[
\begin{align*}
  p_0(\omega_{n/2}^k) + \omega_n^k p_1(\omega_{n/2}^k) &= p(\omega_n^k) \\
  p_0(\omega_{n/2}^k) - \omega_n^k p_1(\omega_{n/2}^k) &= p(\omega_n^{k+n/2})
\end{align*}
\]
Further Improvement

Example:

\[ p(\omega_4^0) = p_0(\omega_2^0) + \omega_4^0 p_1(\omega_2^0) \]
\[ p(\omega_4^1) = p_0(\omega_2^1) + \omega_4^1 p_1(\omega_2^1) \]
\[ p(\omega_4^2) = p_0(\omega_2^0) - \omega_4^0 p_1(\omega_2^0) \]
\[ p(\omega_4^3) = p_0(\omega_2^1) - \omega_4^1 p_1(\omega_2^1) \]
Fast Fourier Transform

Algorithm $FFT$

**Input:** An Array $a$ with $n$ coefficients of a polynomial $p$
and $n = 2^k$

**Output:** $DFT_n(p)$

1. if $n = 1$ then /* $p$ ist constant */
2. return $a$
3. $d[0] = FFT([a_0, a_2, \ldots, a_{n-2}], n/2)$
4. $d[1] = FFT([a_1, a_3, \ldots, a_{n-1}], n/2)$
5. $\omega_n = e^{2\pi i/n}$
6. $\omega = 1$
7. for $k = 0$ to $n/2 - 1$ do /* $\omega = \omega_n^k$ */
8. $d_k = d_k[0] + \omega \times d_k[1]$
9. $d_{k+n/2} = d_k[0] - \omega \times d_k[1]$
10. $\omega = \omega_n \times \omega$
11. return $d$
FFT : Example

\[ P(x) = 3x^3 - 15x^2 + 18x + 0 \]

\[ a = [0, 18, -15, 3] \]
\[ a^0 = [0, -15] \quad a^1 = [18, 3] \]
\[ FFT([0, -15], 2) = (FFT([0], 1) + FFT([-15], 1), FFT([0], 1) - FFT([-15], 1)) \]
\[ = (-15, 15) \]
\[ FFT([18, 3], 2) = (FFT([18], 1) + FFT([3], 1), FFT([18], 1) - FFT([3], 1)) \]
\[ = (21, 15) \]

\[ k = 0; \quad \omega = 1 \]
\[ d_0 = -15 + 1 \times 21 = 6 \quad d_2 = -15 - 1 \times 21 = -36 \]

\[ k = 1; \quad \omega = i \]
\[ d_1 = 15 + i \times 15 \quad d_3 = 15 - i \times 15 \]

\[ FFT(a, 4) = (6, 15 + 15i, -36, 15 - 15i) \]
Analysis

\[ T(n) = \text{running time for evaluating a polynomial of degree} < n \text{ at } n \text{ positions} \]
\[ \omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{2n-1} \]

\[ T(1) = O(1) \]

\[ T(n) = 2 \, T(n/2) + O(n) \]
\[ = O(n \log n) \]
Polynomial Multiplication

Compute the product of two polynomials $p$, $q$ of degree $< n$:

$p, q$ of degree $n-1$, $n$ coefficients

\[ \omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{2n-1} \]

Evaluation by FFT:

2$n$ point-value pairs \( (\omega_{2n}^i, p(\omega_{2n}^i)) \) und \( (\omega_{2n}^i, q(\omega_{2n}^i)) \)

Pointwise multiplication

2$n$ point-value pairs \( (\omega_{2n}^i, pq(\omega_{2n}^i)) \)

Interpolation via FFT

$pq$ of degree $2n-2$, $2n-1$ coefficients
Interpolation

Convert the point-value representation into coefficient representation.

Input: \((x_0, y_0), \ldots, (x_{n-1}, y_{n-1})\) mit \(x_i \neq x_j\), für alle \(i \neq j\)

Output: Polynomial \(p\) with coefficients \(a_0, \ldots, a_{n-1}\), such that

\[
p(x_0) = a_0 + a_1 x_0 + \ldots + a_{n-1} x_0^{n-1} = y_0
\]
\[
p(x_1) = a_0 + a_1 x_1 + \ldots + a_{n-1} x_1^{n-1} = y_1
\]
\[
p(x_2) = a_0 + a_1 x_2 + \ldots + a_{n-1} x_2^{n-1} = y_2
\]
\[
\vdots
\]
\[
p(x_{n-1}) = a_0 + a_1 x_{n-1} + \ldots + a_{n-1} x_{n-1}^{n-1} = y_{n-1}
\]
Interpolation

Matrix notation:

\[
\begin{pmatrix}
1 & x_0 & \cdots & x_0^{n-1} \\
1 & x_1 & \cdots & x_1^{n-1} \\
\vdots & & & \vdots \\
1 & x_{n-1} & \cdots & x_{n-1}^{n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{n-1}
\end{pmatrix}
=
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{pmatrix}
\]
Interpolation

System of equations

\[
\begin{pmatrix}
1 & x_0 & \cdots & x_0^{n-1} \\
1 & x_1 & \cdots & x_1^{n-1} \\
\vdots & & & \vdots \\
1 & x_{n-1} & \cdots & x_{n-1}^{n-1}
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
\vdots \\
a_{n-1}
\end{pmatrix}
= 
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{pmatrix}
\]

solvable if \( x_i \neq x_j \) for all \( i \neq j \).

Special Case (here): \( x_i = \omega_i^n \)

Definition: \( V_n = (\omega_i^n)_{i,j} \), \( a = (a_i) \), \( y = (y_i) \)

\[
V_n a = y \quad \Rightarrow \quad a = V_n^{-1} y
\]
Interpolation

Theorem
For all $0 \leq i, j \leq n - 1$ we have

$$\left(V_n^{-1}\right)_{ij} = \frac{\omega_n^{-ij}}{n}$$

Proof

$$V_n^{-1} = \left(\frac{\omega_n^{-ij}}{n}\right)_{i,j}$$

We have to show:

$$V_n^{-1}V_n = I_n$$
Interpolation

Consider the entry of $V_n^{-1}V_n$ in line $i$ and column $j$:

$$(V_n^{-1}V_n)_{ij} =$$

\[
\begin{pmatrix}
\frac{1}{n} & \frac{\omega_n^{-i}}{n} & \cdots & \frac{\omega_n^{-i(n-1)}}{n} \\
\vdots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \frac{\omega_n^{(n-1)i}}{n}
\end{pmatrix}
\begin{pmatrix}
\cdots & 1 & \cdots \\
\cdots & \omega_n & \cdots \\
\cdots & \omega_n^{j} & \cdots \\
\cdots & \omega_n^{(n-1)j} & \cdots
\end{pmatrix}_{ij}
\]
Interpolation

\[(V_n^{-1}V_n)_{ij} = \sum_{k=0}^{n-1} \frac{\omega_n^{-ik}}{n} \omega_n^{jk} = \frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k}\]

Case 1: \(i = j\)

\[\frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = \frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{0k} = 1\]

Case 2: \(i \neq j\), i.e. \(- (n-1) \leq -i + j \leq n-1\)

Thus \(n \nmid -i + j\):

\[\frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = 0\]
Interpolation

Summation lemma:
For any integer \( n > 0, \, l \geq 0 \) with \( n \not| \, l \):

\[
\sum_{k=0}^{n-1} \omega_n^{lk} = 0
\]

Proof:

\[
\sum_{k=0}^{n-1} \left( \frac{\omega_n^l}{\omega_n} \right)^k = \frac{\left( \frac{\omega_n^l}{\omega_n} \right)^n - 1}{\omega_n^l - 1} = \frac{\left( \omega_n^n \right)^l - 1}{\omega_n^l - 1} = 0
\]
Interpolation

\[ a_i = \left( V_n^{-1} y \right)_i \]

\[ = \left( \frac{1}{n}, \frac{\omega_n^{-i}}{n}, \ldots, \frac{\omega_n^{-(i-1)}}{n} \right) \left( \begin{array}{c} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{array} \right) \]

\[ = \sum_{k=0}^{n-1} y_k \frac{\omega_n^{-ik}}{n} \]

\[ = \frac{1}{n} \sum_{k=0}^{n-1} y_k \left( \omega_n^{-i} \right)^k \]
Interpolation

\[ a = \frac{1}{n} \left( \sum_{k=0}^{n-1} y_k \left( \omega_n^{-0} \right)^k, \sum_{k=0}^{n-1} y_k \left( \omega_n^{-1} \right)^k, \ldots, \sum_{k=0}^{n-1} y_k \left( \omega_n^{-(n-1)} \right)^k \right) \]

\[ r(x) = y_0 + y_1 x + y_2 x^2 + \cdots + y_{n-1} x^{n-1} \]

\[ a = \frac{1}{n} \left( r(\omega_n^{-0}), r(\omega_n^{-1}), \ldots, r(\omega_n^{-(n-1)}) \right) \]
Interpolation and DFT

\[ a = \frac{1}{n} (r(\omega_n^0), r(\omega_n^{-1}), \ldots, r(\omega_n^{-(n-1)})) \]

\[ a = \frac{1}{n} (r(\omega_n^n), r(\omega_n^{n-1}), \ldots, r(\omega_n^1)) \quad \text{since} \quad \omega_n^n = 1 \]

\[ a_i = \frac{1}{n} (DFT_n(r))_{n-i} \quad (i \neq 0) \]

\[ a_0 = \frac{1}{n} (DFT_n(r))_0 \]
Polynomial Multiplication

Compute the product of two polynomials \( p, q \) of degree \( < n \):

- \( p, q \) of degree \( n-1 \), \( n \) coefficients
- Evaluation by FFT: \( \omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{2n-1} \)
- 2\( n \) point-value pairs \( (\omega_{2n}^i, p(\omega_{2n}^i)) \) and \( (\omega_{2n}^i, q(\omega_{2n}^i)) \)
- Pointwise multiplication
- 2\( n \) point-value pairs \( (\omega_{2n}^i, pq(\omega_{2n}^i)) \)
- Interpolation via FFT
- \( pq \) of degree \( 2n-2, 2n-1 \) coefficients
Algorithm Theory
3 Fast Fourier Transform

Christian Schindelhauer

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Wintersemester 2007/08