Schnelle Fourier-Transformation

Die diskrete Fourier-Transformation,

$$f_j = \sum_{k=0}^{n-1} c_k w_n^{jk}\,, \quad j = 0, \ldots, n-1\,,$$

($w_n = \exp(2\pi \mathrm{i}/n)$), kann für $n = 2^\ell$ mit der sogenannten schnellen Fourier-Transformation (FFT, Fast Fourier Transform) mit $2n\ell$-Operationen berechnet werden.

In der rekursiven Version hat der Algorithmus die folgende Form:

$$
\begin{array}{l}
f = \mathrm{FFT}(c) \\
\quad n = \mathrm{length}(c) \\
\quad \textbf{if } n = 1,\ f = c,\ \textbf{return} \\
\quad \textbf{else} \\
\qquad g = \mathrm{FFT}(c_0, c_2, \ldots, c_{n-2})\,, \quad h = \mathrm{FFT}(c_1, c_3, \ldots, c_{n-1}) \\
\qquad p = \left(1, w_n, w_n^2, \ldots, w_n^{n/2-1}\right) \\
\qquad f = (g + p \mathbin{.*}\ h,\ g - p \mathbin{.*}\ h) \\
\quad \textbf{end}
\end{array}
$$

Dabei bezeichnet $.*$ die komponentenweise Multiplikation von Vektoren, d.h. $(a \mathbin{.*}\ b)_j = a_j b_j$.

Die inverse diskrete Fourier-Transformation

$$c_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j w_n^{-jk}$$

kann vollkommen analog berechnet werden. Man bezeichnet den entsprechenden Algorithmus mit $c = \mathrm{IFFT}(f)$.

Beweis:

(i) Induktive Herleitung des Algorithmus:

$$f_j = \sum_{k=0}^{n-1} c_k w^{kj} = \sum_{k=0}^{m-1} c_{2k} \tilde{w}^{kj} \; + \; w^j \sum_{k=0}^{m-1} c_{2k+1} \tilde{w}^{kj}, \quad j = 0, \ldots, n-1$$

mit $m = n/2$ und $\tilde{w} = \exp(2\pi \mathrm{i}/m) = w^2$
Summen entsprechen den im Algorithmus rekursiv berechneten Transformierten g und h der Länge m:

$$f_j = g_j + w^j h_j, \quad j = 0, \ldots, m-1$$

$\tilde{w}^m = 1$ und $w^{j+m} = w^j \exp((2\pi \mathrm{i}/n)(n/2)) = -w^j \quad \Longrightarrow$

$$f_{j+m} = g_j - w^j h_j, \quad j = 0, \ldots, m-1\,,$$

d.h. $(f_m, f_{m+1}, ..., f_{n-1})^{\mathrm{t}}$ ist ebenfalls mit Hilfe von g und h berechenbar

(ii) Anzahl $\operatorname{op}(n)$ der Operationen des FFT-Algorithmus:
Addition der zur Berechnung von g, h, p und f benötigten Operationen
$\rightsquigarrow$

$$\operatorname{op}(n) = \operatorname{op}(n/2) + \operatorname{op}(n/2) + (n/2) + 3(n/2) = 2\operatorname{op}(n/2) + 2n$$

Iteration der Identität $\quad\Longrightarrow$

$$\begin{aligned}
\operatorname{op}(n) &= 2\,(2\operatorname{op}(n/4) + 2(n/2)) + 2n = 4\operatorname{op}(n/4) + 2n + 2n \\
&= \cdots \\
&= 2^\ell \operatorname{op}(1) + \underbrace{2n + \cdots + 2n}_{\ell\text{-mal}}
\end{aligned}$$

$\operatorname{op}(1) = 0 \quad \rightsquigarrow \quad$ Gesamtoperationenzahl $2\ell n$

Beispiel:

diskrete Fourier-Transformation des Vektors

$$c = \begin{pmatrix} 3 \\ -2 \\ 0 \\ 1 \end{pmatrix}$$

$g = \mathrm{FFT}(3, 0)$:

$$\begin{aligned} \tilde{g} &= 3 \\ \tilde{h} &= 0 \\ w_2 &= \exp(2\pi \mathrm{i}/2) = -1\,,\ p = (1) \\ g &= (3 + 1 \cdot 0, 3 - 1 \cdot 0)^{\mathrm{t}} = (3, 3)^{\mathrm{t}} \end{aligned}$$

$h = \mathsf{FFT}(-2, 1)$:

$$
\begin{aligned}
\tilde{g} &= -2 \\
\tilde{h} &= 1 \\
w_2 &= \exp(2\pi\mathrm{i}/2) = -1\,,\ p = (1) \\
h &= (-2 + 1 \cdot 1, -2 - 1 \cdot 1)^{\mathrm{t}} = (-1, -3)^{\mathrm{t}}
\end{aligned}
$$

Addition der rekursiv berechneten diskreten Fourier-Transformierten

$$
g = (3, 3)^{\mathrm{t}}, \quad h = (-1, -3)^{\mathrm{t}}
$$

$\rightsquigarrow$

$$
\begin{aligned}
w_4 &= (\exp(2\pi\mathrm{i}/4) = \mathrm{i}\,,\ p = (1, \mathrm{i})^{\mathrm{t}} \\
f &= (g + p. * h, g - p. * h) \\
&= (3 + 1 \cdot (-1), 3 + \mathrm{i} \cdot (-3), 3 - 1 \cdot (-1), 3 - \mathrm{i} \cdot (-3))^{\mathrm{t}} \\
&= (2, 3 - 3\mathrm{i}, 4, 3 + 3\mathrm{i})^{\mathrm{t}}
\end{aligned}
$$