Kapitel 3

Schnelle Fourier-Transformation

3.1 Fourier-Transformation und Faltung

Die diskrete Fouriertransformation, oder Fourier-Transformation der Länge n, ist definiert durch

$$\hat{y}_k = \sum_{j=0}^{n-1} e^{-2\pi ijk/n} y_j \quad , \qquad k = 0, \ldots, n-1 \; .$$

Mit Hilfe der Orthogonalitätsrelationen

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{2\pi ijk/n} = \begin{cases} 1 & , \quad k = 0 \; , \; \pm n \; , \pm 2n, \ldots \\ 0 & , \quad \text{sonst} \end{cases} \quad ,$$

zeigt man leicht die Inversionsformel

$$y_j = \frac{1}{n} \sum_{k=0}^{n-1} e^{2\pi ijk/n} \hat{y}_k \; , \qquad j = 0, \ldots, n-1 \; .$$

Die Fourier-Transformation ist also eine lineare Abbildung in $\mathbb{C}^n$ mit der Matrix

$$W = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & & & & \\ 1 & \omega^{n-1} & \omega^{(n-1)2} & \ldots & \omega^{(n-1)(n-1)} \end{pmatrix} \quad , \quad \omega = e^{2\pi ij/n} \; .$$

Wegen $\omega^n = 1$ kann in dem Exponenten in W^n mod n gerechnet werden. Die Orthogonalitätsrelationen ergeben

$$W_n W_n^* = nI \quad .$$

Insbesondere ist also $\frac{1}{\sqrt{n}} W_n$ eine Isometrie.

Für zwei Vektoren x, $y \in \mathbb{C}^n$, deren Komponenten wir von 0 bis $n-1$ numerieren, nennen wir den Vektor z mit den Komponenten

$$z_k = \sum_{j=0}^{n-1} x_{k-j} y_j \; , \qquad k = 0, \ldots, n-1$$

die (zyklische) Faltung von x und y. Dabei rechnen wir im Index von x mod n. Wir schreiben $z = x * y$. In Matrix-Form lautet die Faltung $z = Xy$

$$X = \begin{pmatrix} x_0 & x_{n-1} & \cdots & x_1 \\ x_1 & x_0 & \cdots & x_2 \\ \vdots & & & \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{pmatrix} \quad .$$

Die $(i+1)$-te Zeile dieser Matrix entsteht also aus der i-ten durch einen Rechts-Ring-Schift.

Satz 3.1.1 *(Faltungssatz) Für x, $y \in \mathbb{Z}^n$ gilt*

$$(x * y)_k^\wedge = \hat{x}_k \hat{y}_k \; , \qquad k = 0, \ldots, n-1 \; .$$

Beweis: Auch dies ist eine unmittelbare Folgerung aus den Orthogonalitätsrelationen.

$\square$

3.2 Cooley-Tukey

Wir beschreiben nun verschiedene Versionen der schnellen Fourier-Transformation nach Cooley-Tukey. Wir nehmen an, die Länge n der Fourier-Transformation faktoriere gemäß $n = mp$. Den Fall $p = 2$ haben wir teilweise schon in der Einleitung behandelt.

1. Dezimierung in der Zeit

Wir haben auswerten

$$\hat{y}_k = \sum_{j=0}^{n-1} \omega_n^{jk} y_j \ , \quad k = 0, \dots, n-1 \ ,$$

wo $\omega_n = e^{-2\pi i/n}$ eine n-te Einheitswurzel ist. Wir fassen in der Summe diejenigen Indizes j zusammen, welche der gleichen Restklasse mod p angehören. Dies ergibt

$$\hat{y}_k = \sum_{r=0}^{p-1} \sum_{\ell=0}^{m-1} \omega_n^{(p\ell+r)k} y_{p\ell+r} \ .$$

Nun ist $\omega_n^p = \omega_m$. Also folgt

$$\hat{y}_k = \sum_{r=0}^{p-1} \omega_n^{rk} \sum_{\ell=0}^{m-1} \omega_m^{\ell k} y_{p\ell+r} \ .$$

Führen wir die Vektoren

$$y^r = \begin{pmatrix} y_r \\ y_{p+r} \\ \vdots \\ y_{(m-1)p+r} \end{pmatrix} \in \mathbb{C}^m \quad , \quad r = 0, \dots, p-1$$

ein, so lautet dies

$$\hat{y}_k = \sum_{r=0}^{p-1} \omega_n^{rk} (\hat{y}^r)_k \ ,$$

wo $\hat{y}^r$ die Fourier-Transformation der Länge m von y^r bedeutet. Diese hat die Periode m. Also gilt

$$\hat{y}_{k+\ell m} = \sum_{r=0}^{p-1} \omega_n^{(k+\ell m)r} (\hat{y}^r)_k \quad , \qquad k = 0, \ldots, m-1 \ , \quad \ell = 0, \ldots, p-1$$

und wegen $\omega_n^m = \omega_p$

$$\hat{y}_{k+\ell m} = \sum_{r=0}^{p-1} \omega_p^{\ell r} \omega_n^{k\ell} (\hat{y}^r)_k \ . \tag{3.2.1}$$

Bis auf die Multiplikation mit $\omega_n^{k\ell}$ sind dies m Fourier-Transformationen der Länge p. Die Fourier - Transformation der Länge n kann also ausgeführt werden durch p Fourier-Transformationen der Länge m und m Fourier-Transformationen der Länge p, zuzüglich einiger Multiplikationen mit Einheitswurzeln und Permutationen.

Wir wollen (3.2.1) in Matrixform schreiben. Mit W_n, W_m, W_p die Fourier-Transformation der Längen n, m, p und

$$D_n = \begin{pmatrix} 1 & & & \\ & \omega_n & & \\ & & \ddots & \\ & & & \omega_n^{m-1} \end{pmatrix}$$

lautet (3.2.1), wenn wir zunächst die m Gleichungen für $\ell = 0$, dann die m Gleichungen für $\ell = 1$ usw. aufschreiben,

$$\hat{y} = \begin{pmatrix} I_m & I_m & \cdots & I_m \\ I_m & \omega_p I_m & \cdots & \omega_p^{p-1} I_m \\ \vdots & \vdots & & \\ I_m & \omega_p^{p-1} I_m & \cdots & \omega^{(p-1)^2} I_m \end{pmatrix} \begin{pmatrix} D_m^0 & & & \mathbf{O} \\ & D_m^1 & & \\ & & \ddots & \\ \mathbf{O} & & & D_m^{p-1} \end{pmatrix} * \tag{3.2.2}$$

$$* \begin{pmatrix} W_m & & & \mathbf{O} \\ & W_m & & \\ & & \ddots & \\ \mathbf{O} & & & W_m \end{pmatrix} \begin{pmatrix} y^0 \\ y^1 \\ \vdots \\ y^{p-1} \end{pmatrix} .$$

Hier führen wir folgende Schreibweise ein. Ist A eine (m,n)-Matrix und B eine (p,q)-Matrix, so bezeichnet das Tensorprodukt $A \otimes B$ die (mp,nq)-Matrix

$$\begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ \vdots & & & \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} .$$

Damit lautet (3.2.2)

$$W_n = (W_p \otimes I_m) \begin{pmatrix} D_m^0 & & & \\ & D_m^1 & & \\ & & \ddots & \\ & & & D_m^{p-1} \end{pmatrix} (I_p \otimes W_m) P_{p,m} \quad (3.2.3)$$

mit einer Permutationsmatrix $P_{p,m}$. Ist $y = (y_0, \ldots, y_{n-1})^T$, so ist $P_{p,m}y$ der Vektor

$$(y_0, y_p, \ldots, y_{(n-1)p}, y_1, y_{p+1}, \ldots, y_{(m-1)p+1}, \ldots, y_{m-1}, y_{p+m-1}, \ldots, y_{(m-1)p+n-1}) .$$

Anders ausgedrückt: Ist $0 \leq i < n$ und $i = mq + r$ mit ganzen Zahlen $0 \leq q < p$, $0 \leq r < m$ (Division mit Rest), so ist

$$(Py)_i = y_{rp+q} .$$

Ist n ein Produkt kleiner Faktoren, so nennt man n FT-freundlich. Wiederholte Anwendungen von (3.2.4) führt dann zu einem effizienten Algorithmus.

Das einfachste Beispiel von FT-freundlichen Zahlen sind Potenzen von 2. Diesen Fall wollen wir nun etwas ausführlicher betrachten. Wir führen dazu den *butterfly* der Ordnung m ein als die (m,m)-Matrix

$$B_m = \begin{pmatrix} I_{m/2} & D_{m/2} \\ I_{m/2} & -D_{m/2} \end{pmatrix}$$

ein. Für $p = 2$ lautet (3.2.4)

$$\begin{aligned} W_n &= (W_2 \otimes I_{n/2}) \begin{pmatrix} I_{n/2} & \mathbf{O} \\ \mathbf{O} & D_{n/2} \end{pmatrix} (I_2 \otimes W_{n/2}) P_{2,n/2} \\ &= B_n (I_2 \otimes W_{n/2}) P_{2,n/2} . \end{aligned}$$

Rekursive Anwendung ergibt

$$\begin{aligned} W_n &= B_n(I_2 \otimes B_{n/2})(I_4 \otimes W_{n/4})(I_2 \otimes P_{2,n/4})P_{2,n/2} \\ &\vdots \\ &= (I_1 \otimes B_n)(I_2 \otimes B_{n/2})(I_4 \otimes B_{n/4}) \cdots (I_{n/2} \otimes B_2)P_n \quad (3.2.4) \\ P_n &= \quad (I_{n/4} \otimes P_{2/2}) \cdot \cdots \cdot (I_{2,n/4})(I_1 \otimes P_{2,n/2}) \, . \quad (3.2.5) \end{aligned}$$

Dies ist die Radix-2-Version der Cooley-Tukey FFT.

Die Permutation P_n kann durch *bitreversal* angewendet werden. Die *bitreversal*-Funktion r_n von $\{0, \ldots, n-1\}$ in sich ist für $n = 2^t$ wie folgt definiert. Sei $(b_{t-1}, \ldots, b_0)_2$ die Dualdarstellung von k, also

$$k = b_{t-1}2^{t-1} + \cdots + b_1 2 + b_0$$

mit $b_i \in \{0, 1\}$. Dann ist $r_n(k) = (b_0, \ldots, b_{t-1})_2$.

Satz 3.2.1 *Die Komponenten von x, $P_n x$ seien von 0 bis $n-1$ numeriert, und $n = 2^t$. Dann gilt*

$$(Px)_k = x_{r_n(k)} \quad , \qquad k = 0, \ldots, n-1 \, .$$

Beweis: Für $t = 1$ ist $P_2 = I_2$ und die Behauptung trivial. Sei die Behauptung richtig für $t-1$, und sei $m = 2^{t-1}$.

Nach (3.2.5) ist

$$P_n x = (I_2 \otimes P_m)P_{2,m}x = \begin{pmatrix} P_m \begin{pmatrix} x_0 \\ x_2 \\ \vdots \\ x_{m-2} \end{pmatrix} \\ P_m \begin{pmatrix} x_1 \\ x_3 \\ \vdots \\ x_{m-1} \end{pmatrix} \end{pmatrix} \, .$$

Nach Induktionsannahme können wir P_m durch *bitreversal* auswerten. Ist also $k = (b_{t-2}, \ldots, b_0)_2$ und $k' = (b_0, \ldots, b_{t-2})_2$, so ist

$$\left(P_m \begin{pmatrix} x_0 \\ x_2 \\ \vdots \\ x_{m-1} \end{pmatrix} \right)_k = x_{2k'} \quad , \quad \left(P_m \begin{pmatrix} x_1 \\ x_3 \\ \vdots \\ x_{m-1} \end{pmatrix} \right)_k = x_{2k'+1} \quad .$$

Nun ist aber für $k = 0, \ldots, m-1$

$$k = (0, b_{t-2}, \ldots, b_0)_2 \quad , \quad 2k' = (b_{t-2}, \ldots, b_0, 0)_2$$

und für $k = m, \ldots, n-1$

$$k = (1, b_{t-2}, \ldots, b_0)_2 \quad , \quad 2k' + 1 = (b_{t-2}, \ldots, b_0, 1)_2 \ .$$

In jedem Fall erhält man den Index von x also durch *bitreversal.*

□

2. Dezimierung in der Frequenz

Jetzt fassen wir die Indizes k, welche zur gleichen Restklasse mod m gehören, zusammen, schreiben also

$$\begin{aligned}
\hat{y}_{m\ell+r} &= \sum_{j=0}^{n-1} \omega_n^{j(m\ell+r)} y_j \\
&= \sum_{j=0}^{n-1} \omega_n^{jr} y_j \\
&= \sum_{k=0}^{m-1} \sum_{j=0}^{p-1} \omega_p^{(j+kp)\ell} \omega_n^{(j+kp)r} y_{j+kp} \\
&= \sum_{k=0}^{m-1} \omega_m^{kr} \sum_{j=0}^{p-1} \omega_p^{j\ell} \omega_n^{rj} y_{j+kp} \ ,
\end{aligned}$$

wobei wir wieder $\omega_n^p = \omega_m$ und $\omega_n^m = \omega_p$ benutzt haben. Dieses Mal setzen wir

$$y^k = \begin{pmatrix} y_{kp} \\ y_{kp+1} \\ \vdots \\ y_{kp+p-1} \end{pmatrix} \in \mathbb{C}^p \quad , \qquad k = 0, \dots, m-1 \; .$$

Dann können wir schreiben

$$\hat{y}_{m\ell+r} = \sum_{k=0}^{m-1} \omega_m^{kr} \left(D_p^r y^k \right)_\ell^\wedge \; ,$$

wobei rechts die Fourier-Transformation der Länge p steht und D_p entsprechend zu D_m erklärt ist. Schreiben wir diese Gleichungen erst für $r = 0$, dann für $r = 1$ usw. hin, dann entsteht

$$\begin{aligned} P_{m,p}\hat{y} &= \begin{pmatrix} W_p D_p^0 y^0 & + & W_p D_p^0 y^1 & + & \cdots & + & W_p D_p^0 y^{m-1} \\ W_p D_p^1 y^0 & + & \omega_m W_p D_p^1 y^1 & + & \cdots & + & \omega_m^{m-1} W_p D_p^1 y^{m-1} \\ \vdots & & & & & & \\ W_p D_p^{m-1} y^0 & + & W_p D_p^{m-1} y^1 & + & \cdots & + & \omega^{(m-1)^2} W_p D_p^{m-1} y^{m-1} \end{pmatrix} \\ &= \begin{pmatrix} W_p & & & \\ & W_p & & \\ & & \ddots & \\ & & & W_p \end{pmatrix} \begin{pmatrix} D_p^0 & & & \\ & D_p^1 & & \\ & & \ddots & \\ & & & D_p^{m-1} \end{pmatrix} * \\ &\quad * \begin{pmatrix} I_p & I_p & \cdots & I_p \\ I_p & \omega_n I_p & \cdots & \omega_m^{m-1} I_p \\ \vdots & & & \\ I_p & \omega_n^{m-1} I_p & \cdots & \omega_m^{(m-1)} I_p \end{pmatrix} \begin{pmatrix} y^0 \\ y^1 \\ \vdots \\ y^{m-1} \end{pmatrix} \end{aligned}$$

oder

$$P_{m,p} W_n = (I_m \otimes W_p) \begin{pmatrix} D_p^0 & & & \\ & D_p^1 & & \\ & & \ddots & \\ & & & D_p^{m-1} \end{pmatrix} (W_m \otimes I_p) \; . \quad (3.2.6)$$

Mit Hilfe von

$$P_{m,p}^{T} = P_{m,p}^{-1} = P_{p,m} \tag{3.2.7}$$

sieht man, daß (3.2.7) unmittelbar aus (3.2.4) folgt. Durch Transposition von (3.2.7) ergibt sich ja

$$W_n P_{m,p}^{T} = (W_m \otimes I_p) \begin{pmatrix} D_p^0 & & & \\ & D_p^1 & & \\ & & \ddots & \\ & & & D_p^{m-1} \end{pmatrix} (I_n \otimes W_p) \; .$$

Wegen (3.2.7) ist dies identisch zu (3.2.3) mit vertauschten m, p.

3.3 Primfaktoren

Die Good'sche oder Primfaktor-FFT benutzt wie Cooley-Tukey auch eine Faktorisierung $n = mp$ der Länge n. Die Faktoren m, p müssen jetzt aber teilerfremd sein. Wir beginnen mit dem Beispiel $n = G = 2\cdot 3$. Wir schreiben W_6 in der Form

$$W_6 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ 0 & 4 & 2 & 0 & 4 & 2 \\ 0 & 5 & 4 & 3 & 2 & 1 \end{pmatrix} \quad ,$$

d.h. wir notieren an der Stelle (k, ℓ) nicht $\omega_6^{k\ell} = e^{-2\pi i(k+\ell)/6}$, sondern nur $k + \ell$ mod 6. Ordnen wir die Zeilen von W_6 in der Reihenfolge 0, 3, 4, 1, 2, 5 und die Spalten in der Reihenfolge 0, 3, 2, 5, 4, 1 an, so entsteht

$$W_6' = \left(\begin{array}{cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 3 & 0 & 3 \\ \hline 0 & 0 & 2 & 2 & 4 & 4 \\ 0 & 3 & 2 & 5 & 4 & 1 \\ \hline 0 & 0 & 4 & 4 & 2 & 2 \\ 0 & 3 & 4 & 1 & 2 & 5 \end{array}\right) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_5 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad .$$

Bis auf Zeilen- und Spaltenvertauschungen ist W_6, also das Tensor-Produkt von W_3 und W_2. W_6 kann also ausgeführt werden durch 3 Fourier - Transformation der Länge 2 und 2 Fourier-Transformation der Länge 3.

Satz 3.3.1 *Sei $n = n_1 n_2$ und $ggt(n_1, n_2) = 1$. Dann gibt es Permutationsmatrizen P, Q, so daß*

$$W_n = Q^T (W_{n_1} \otimes W_{n_2}) P \ .$$

Beweis: Wegen $ggt(n_1, n_2) = 1$ können wir zu j mit $0 \leq j <$ Zahlen j_1, j_2 mit $0 \leq j_1 < n_1$, $0 \leq j_2 < n_2$ finden mit $j = n_1 j_2 + n_2 j_1 \mod n$, und ebenso $k = n_1 k_2 + n_2 k_1 \mod n$. Aus

$$\hat{y}_k = \sum_{j=0}^{n-1} \omega^{jk} y_j \ , \quad \omega = e^{-2\pi i/n} \ , \quad k = 0, \ldots, n-1$$

wird dann

$$\hat{y}_{n_1 k_2 + n_2 k_1} = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} \omega^{(n_1 j_2 + n_2 j_1)(n_1 k_2 + n_2 k_1)} y_{n_1 j_1 + n_2 j_1} \ .$$

Mit $\omega^{n_1 n_2} = 1$, $\omega^{n_1} = e^{-2\pi i/n_2} = \omega_2$, $\omega^{n_2} = e^{-2\pi i/n_1} = \omega_1$ folgt

$$\hat{y}_{n_1 k_2 + n_2 k_1} = \sum_{j_1=0}^{n_1-1} \omega_1^{n_2 j_1 k_1} \sum_{j_2=0}^{n_2-1} \omega_2^{n_1 j_2 k_2} y_{n_1 j_2 + n_2 j_1} \ .$$

Hier ist $0 \leq k_1 < n_1$, $0 \leq k_2 < n_2$. Zu jedem solchen Paar k_1, k_2 können wir genau ein Paar (k_1', k_2') finden, so daß

$$n_1 k_2' \equiv k_2 \mod n_2 \ , \quad n_2 k_1' \equiv k_1 \ \ mod\, n_1 \ , \tag{3.3.1}$$

denn $ggt(n_1, n_2) = 1$. Mit diesen Zahlen k_1', k_2' gilt dann

$$\hat{y}_{n_1 k_2' + n_2 k_1'} = \sum_{j_1=0}^{n_1-1} \omega_1^{n_2 j_1 k_1'} \sum_{j_2=0}^{n_2-1} \omega_2^{n_1 j_2 k_2'} y_{n_1 j_2 + n_2 j_1} \ .$$

Aufgrund der Wahl von k_1', k_2' und wegen $\omega_1^{n_1} = \omega_2^{n_2} = 1$ gilt

$$\omega_1^{n_2 k_1'} = \omega_1^{k_1} \quad , \quad \omega^{n_1 k_2'} = \omega_2^{k_2} \ ,$$

also

$$\hat{y}_{n_1 k_2' + n_2 k_1'} = \sum_{j_1=0}^{n_1-1} \omega_1^{j_1 k_1} \sum_{j_2=0}^{n_2-1} \omega_2^{j_2 k_2} y_{n_1 j_2 + n_2 j_1} \ .$$

Mit den n_2-Vektoren

$$\begin{aligned} y^{j_1} &= (y_{n_1 j_2 + n_2 j_1})_{0 \leq j_2 < n_2} \ , \\ \hat{y}^{k_1} &= (\hat{y}^{n_1 k_2' + n_2 k_1'})_{0 \leq k_2 < n_2} \end{aligned}$$

lautet dies mit $W_2 = W_{n_2}$

$$(\hat{y}^{k_1})_{k_2} = \sum_{j_1=0}^{n_1-1} \omega_1^{j_1 k_1} (W_2 y^{j_1})_{k_2}$$

oder

$$\begin{pmatrix} \hat{y}^0 \\ \vdots \\ \hat{y}^{n_1-1} \end{pmatrix} = \begin{pmatrix} W_2 & W_2 & \cdots & W_2 \\ W_2 & \omega_1 W_2 & \cdots & \omega_1^{n_1-1} W_2 \\ \vdots & & & \\ W_2 & \omega_1^{n_1-1} W_2 & \cdots & \omega_1^{(n_1-1)^2} W_2 \end{pmatrix} \begin{pmatrix} y^0 \\ \vdots \\ y^{n_1-1} \end{pmatrix} .$$

Also, mit $W_1 = W_{n_1}$,

$$Q\hat{y} = W_1 \otimes W_2 P y ,$$

und dies ist die Behauptung.

$\square$

Dem Beweis kann man die Permutationen P, Q entnehmen. Es ist mit $j = n_2 j_1 + j_2$, $0 \leq j_2 < n_2$ (Division mit Rest)

$$(Py)_j = y_{n_1 j_2 + n_2 j_1} .$$

Entsprechend ist mit $k = n_2 k_1 + k_2$, $0 \leq k_2 < n_2$,

$$(Q\hat{y})_k = \hat{y}_{n_1 k_2' + n_2 k_1'} .$$

Dabei sind k_1', k_2' zu k_1, k_2 gemäß (3.3.1) bestimmt. In jedem Fall ist in den Indizes mod n zu rechnen.

Beispiel: $n = 2 \cdot 3$, also $n_1 = 2$, $n_2 = 3$. Wir berechnen P

j	j_1	j_2	$n_1 j_2 + n_2 j_1$
0	0	0	0
1	0	1	2
2	0	2	4
3	1	0	3
4	1	1	5
5	1	2	1

, also $P \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 4 \\ 3 \\ 5 \\ 1 \end{pmatrix}$

und Q

k	k_1	k_2	k'_1	k'_2	$n_1k'_2+n_2k'_1$
0	0	0	0	0	0
1	0	1	0	2	4
2	0	2	0	1	2
3	1	0	1	0	3
4	1	1	1	2	1
5	1	2	1	1	5

$$, \quad \text{also} \quad Q\begin{pmatrix}0\\1\\2\\3\\4\\5\end{pmatrix}=\begin{pmatrix}0\\4\\2\\3\\1\\5\end{pmatrix}.$$

Wir bilden QW_6P^T, d.h. wir vertauschen die Zeilen gemäß Q und die Spalten gemäß P:

$$\begin{pmatrix}0&0&0&0&0&0\\0&1&2&3&4&5\\0&2&4&0&2&4\\0&3&0&3&0&3\\0&4&2&0&4&2\\0&5&4&3&2&1\end{pmatrix}\overset{Q}{\to}\begin{pmatrix}0&0&0&0&0&0\\0&4&2&0&4&2\\0&2&4&0&2&4\\0&3&0&3&0&3\\0&1&2&3&4&5\\0&5&4&3&2&1\end{pmatrix}\overset{P}{\to}\left(\begin{array}{ccc|ccc}0&0&0&0&0&0\\0&2&4&0&2&4\\0&4&2&0&4&2\\\hline 0&0&0&3&3&3\\0&2&4&3&5&1\\0&4&2&3&1&5\end{array}\right)=W_2\otimes W_3$$

Oben haben wir schon den Fall $n_1=3$, $n_2=2$ behandelt.

Folgerung: Seien $M(n)$, $A(n)$ die Anzahl der Multiplikationen bzw. Additionen zur Fourier-Transformation der Länge n. Sei $n=n_1n_2$ mit $ggt(n_1,n_2)=1$. Dann gilt

$$\begin{aligned}M(n)&=n_1M(n_2)+n_2M(n_1)\\A(n)&=n_1A(n_2)+n_2A(n_1)\end{aligned}$$

Wir wollen die entsprechenden Überlegungen nun für die Faltung der Länge n, also

$$z_k=\sum_{j=0}^{n-1}x_{k-j}y_j \quad , \quad k=0,\dots,n-1 \tag{3.3.2}$$

durchführen. Äquivalent hierzu ist die Anwendung der Matrix

$$X_n=\begin{pmatrix}x_0&x_{n-1}&\cdots&&x_1\\x_1&x_0&x_{n-1}&\cdots&x_2\\\vdots&&&&\\x_{n-1}&x_{n-2}&\cdots&&x_0\end{pmatrix}$$

auf den Vektor y.

Satz 3.3.2 *Sei $n = n_1, n_2$ und $ggt(n_1, n_2) = 1$. Dann gibt es eine Permutationsmatrix P und Faltungen $F_0, \ldots, F_{n_1-1}$ der Länge n_2, so daß*

$$X_n = P^T \begin{pmatrix} F_0 & F_{n_1-1} & \cdots & & F_1 \\ F_1 & F_0 & F_{n_1-1} & \cdots & F_2 \\ \vdots & & & & \\ F_{n_1-1} & F_{n_1-1} & \cdots & & F_0 \end{pmatrix} P \; .$$

Beweis: Für $0 \leq k,\, j < n$ können wir mod n

$$\begin{array}{ccc} k = k_2 n_1 + k_1 n_2 & , & j = j_2 n_1 + j_1 n_2 \; , \\ 0 \leq j_1, k_1 < n_1 & , & 0 \leq j_2, k_2 < n_2 \end{array}$$

schreiben, und zwar in eindeutiger Weise. Damit lautet (3.3.1)

$$z_{k_2 n_1 + k_1 n_2} = \sum_{j_1=0}^{n_1-1} \sum_{j_2=0}^{n_2-1} x_{(k_1-j_1)n_2+(k_2-j_2)n_2} y_{j_1 n_2 + j_2 n_1} \; .$$

Dabei wird in den Indizes mod n gerechnet. Die j_2-Summe ist eine Faltung der n_2-Vektoren

$$y^{j_1} = (y_{j_1 n_2 + j_2 n_1})_{0 \leq j_2 < n_2} \quad , \quad f^{k_1 - j_1} = \left(x_{(k_1-j_1)n_2 + j_2 n_1} \right)_{0 \leq k < n_2} \; .$$

Mit dem n_2-Vektor

$$z^{k_1} = (z_{k_2 n_1 + k_1 n_2})_{0 \leq k_2 < n_2}$$

können wir daher schreiben

$$\begin{pmatrix} z^0 \\ \vdots \\ z^{n_1-1} \end{pmatrix} = \begin{pmatrix} F_0 & F_{n_1-1} & \cdots & & F_1 \\ F_1 & F_0 & F_{n_1-1} & \cdots & F_2 \\ \cdots & & & & \\ F_{n_1-1} & F_{n_1-2} & \cdots & & F_0 \end{pmatrix} \begin{pmatrix} y^0 \\ \vdots \\ y^{n_1-1} \end{pmatrix} \; .$$

Dies ist die Behauptung.

$\square$

Der Beweis ergibt eine explizite Darstellung für P. Setzen wir für $0 \leq j < n$ $j = j_1 n_2 + j_2$, $0 \leq j_1 < n_1$, $0 \leq j_2 < n_2$, so ist

$$(Py)_j = y_{j_1 n_2 + j_2 n_1} \ .$$

Beispiel: $n = 2 \cdot 3$, also $n_1 = 2$, $n_2 = 3$. P bewirkt die Permutation $(0,1,2,3,4,5) \rightarrow (0,2,4,3,5,1)$. Entsprechende Vertauschungen von Zeilen und Spalten ergibt

$$\begin{pmatrix} 0 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 5 & 4 & 3 \\ 3 & 2 & 1 & 0 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 5 & 4 & 3 & 2 & 1 \\ 2 & 1 & 0 & 5 & 4 & 3 \\ 4 & 3 & 2 & 1 & 0 & 5 \\ 3 & 2 & 1 & 0 & 5 & 4 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 0 & 5 & 4 & 3 & 2 \end{pmatrix} \rightarrow \left(\begin{array}{ccc|ccc} 0 & 4 & 2 & 3 & 1 & 5 \\ 2 & 0 & 4 & 5 & 3 & 1 \\ 4 & 2 & 0 & 1 & 5 & 3 \\ \hline 3 & 1 & 5 & 0 & 4 & 2 \\ 5 & 3 & 1 & 2 & 0 & 4 \\ 1 & 5 & 3 & 4 & 2 & 0 \end{array}\right)$$

$$= \begin{pmatrix} F^0 & F^1 \\ F^1 & F^0 \end{pmatrix}$$

mit Faltungen F^ℓ der Länge 3.

Folgerung: Sei $n = n_1 n_2$ mit $ggt(n_1, n_2) = 1$. Seien $M(n)$, $A(n)$ die Anzahl der Multiplikationen bzw. $A(n)$ der Länge n. Dann gilt

$$\begin{aligned} M(n) &= M(n_1) M(n_2) \\ A(n) &= M(n_1) A(n_2) + n_2 A(n_1) \end{aligned}$$

(Algorithmus von Cooley und Agarwal).

3.4 Schnelle Faltung nach Winograd

Wir benutzen nun den Chinesischen Restsatz, um Faltungen schnell auszuführen. Wir wissen, daß wir Faltungen der Länge n als Multiplikation von Polynomen in $F[z]$ modulo $p = z^n - 1$ auffassen können. Es ist ja mod p

$$\sum_{k=0}^{n-1} x_k z^k \sum_{\ell=0}^{n-1} y_\ell z^\ell = \sum_{j=0}^{n-1} \left(\sum_{\ell=0}^{j} x_{j-\ell} y_\ell \right) z^j + \sum_{j=n}^{2n-2} \left(\sum_{\ell=j-n+1}^{n-1} x_{j-\ell} y_\ell \right) z^j$$

$$= \sum_{j=0}^{n-1} \left(\sum_{\ell=0}^{j} x_{j-\ell} y_\ell + \sum_{\ell=j+1}^{n-1} x_{j-\ell} y_\ell \right) z^j = \sum_{j=0}^{n-1} (x * y)_j z^j \ ,$$

wenn wir im Index mod n rechnen.

Ist nun $p = p_1 p_2$ mit teilerfremden p_1, p_2, so haben wir nach Satz 2.3.4

$$F[z]/p = F[z]/p_1 \oplus F[z]/p_2 \ .$$

Dieser Isomorphismus ist leicht hergestellt. Nach Satz 2.3.4 gibt es Polynome q_1, q_2 mit $q_2 p_1 + q_1 p_2 = 1$. Also gilt für $f \in F[z]$ die Beziehung $q_2 p_1 f + q_1 p_2 f = f$. Wir definieren nun die Abbildung $\varphi : F[z]/p \to F[z]/p_1 \oplus F[z]/p_2$ durch

$$\varphi(f) = (q_1 f, q_2 f) \ .$$

Dies ist offenbar der gesuchte Isomorphismus. Die Multiplikation zweier Polynome f, g modulo p wird nun so ausgeführt: Zunächst ist

$$fg = q_2 p_1 fg + q_1 p_2 fg$$

oder, modulo p gerechnet,

$$fg = q_2 p_1 (fg \bmod p_2) + q_1 p_2 (fg \bmod p_1) \ . \tag{3.4.1}$$

Es genügt also, die Multiplikation mod p_1 und mod p_2 auszuführen. Dann läßt sich fg mod p leicht berechnen.

Wir betrachten zunächst den Fall $n = 2$. Hier ist $p_1 = z-1$, $p_2 = z+1$, und wir haben

$$-\frac{1}{2}(z-1) + \frac{1}{2}(z+1) = 1 \; ,$$

also $q_1 = +\frac{1}{2}$, $q_2 = -\frac{1}{2}$. Sei $f = x_0 + x_1 z$, $g = y_0 + y_1 z$ und $fg = w_0 + w_1 z \mod (z^2-1)$, also $w_0 = x_0 y_0 + x_1 y_1$, $w_1 = x_0 y_1 + y_1 x_0$.

Wir führen die Multiplikation $fg \mod p_1$ und $\mod p_2$ durch. Es ist

$$\begin{array}{lllllll} fg & \bmod(z-1) & = & (fg)(1) & = & f(1)g(1) & = & (x_0+x_1)(y_0+y_1) \; , \\ fg & \bmod(z+1) & = & (fg)(-1) & = & f(-1)g(-1) & = & (x_0-x_1)(y_0-y_1) \; . \end{array}$$

Nach (3.4.1) haben wir also mod (z^2-1)

$$fg = -\frac{1}{2}(z-1)(x_0-x_1)(y_0-y_1) + \frac{1}{2}(z+1)(x_0+x_1)(y_0+y_1) \; .$$

Dies liefert den folgenden Algorithmus zur Faltung der Länge 2:

$$\begin{array}{lcl} s_1 = y_0 + y_1 & , & s_2 = y_0 - y_1 \\ m_1 = \dfrac{x_0+x_1}{2} s_1 & , & m_2 = \dfrac{x_0-x_1}{2} s_2 \\ w_0 = m_1 + m_2 & , & w_1 = m_1 - m_2 \end{array} \qquad (3.4.2)$$

Zur Feststellung des Rechenaufwandes legen wir nun fest: Es zählen nicht die Rechnungen, die man auf den Koeffizienten x_i ausführt. Man stellt sich dazu vor, daß die x_i ein für allemal gewählt sind, so daß diese Rechnungen ein für allemal durchgeführt werden können. Bei dieser Zählweise erfordert (3.4.2) nur 4 Additionen und zwei Multiplikationen, im Gegensatz zu den 4 Multiplikationen und zwei Additionen der Schulmethode für die Faltung der Länge 2.

Nun zum Fall $n = 3$. Hier haben wir $z^3 - 1 = (z^2+z+1)(z-1)$, also $p_1 = z-1$, $p_2 = z^2+z+1$. Die Polynome q_1, q_2 werden durch Division mit Rest berechnet. Es ist

$$\begin{array}{rcl} p_2 : p_1 & = & z^2+z+1 : z-1 = z+z \\ & & \underline{z^2 - z} \\ & & 2z+1 \\ & & \underline{2z-2} \\ & & \quad 3 \end{array}$$

also

$$z^2 + z + 1 = (z-1)(z+2) + 3$$

und daher

$$-\frac{z+2}{3}(z-1) + \frac{1}{3}(z^2+z+1) = 1 \; .$$

Dieser Beziehung entnehmen wir $q_2 = -(z+2)/3$, $q_1 = 1/3$. Seien wieder $f = x_0 + x_1 z + x_2 z^2$, $g = y_0 + y_1 z + y_2 z^2$ und $fg \bmod (z^3-1) = w_0 + w_1 z + w_2 z^2$. Die Berechnung von $fg \bmod (z-1)$ und $\bmod (z^2+z+1)$ ergibt nun

$$\begin{aligned}
fg \bmod (z-1) &= (fg)(1) = f(1)g(1) = (x_0+x_1+x_2)(y_0+y_1+y_2) \\
fg \bmod (z^2+z+1) &= f \bmod (z^2+z+1) \cdot g \bmod (z^2+z+1) \\
&= (x_0 + x_1 z - x_2(1+z))(y_0 + y_1 z - y_2(1+z)) \\
&= (x_0 - x_2 + (x_1-x_2)z)(y_0 - y_2 + (y_1-y_2)z) \\
&= (x_0-x_2)(y_0-y_2) + ((x_1-x_2)(y_0-y_2) \\
&\qquad +(x_0-x_2)(y_1-y_2))z - (x_1-x_2)(y_1-y_2)(1+z) \\
&= (x_0-x_2)(y_0-y_2) - (x_1-x_2)(y_1-y_2) \\
&\qquad +((x_1-x_2)(y_0-y_2) + (x_0-x_2)(y_1-y_2) \\
&\qquad -(x_1-x_2)(y_1-y_2))z \\
&= m_1' - m_2' + (m_1' - m_3')z \; ,
\end{aligned}$$

$$\begin{aligned}
m_1' &= (x_0-x_2)(y_0-y_2) \; , \\
m_2' &= (x_1-x_2)(y_1-y_2) \; , \\
m_3' &= (x_0-x_1)(y_0-y_1) \; .
\end{aligned}$$

Wir setzen auch noch

$$m_0' = (x_0+x_1+x_2)(y_0+y_1+y_2) \; .$$

Zusammensetzen dieser Resultate mittels (3.4.1) ergibt

$$\begin{aligned}
fg \bmod (z^3-1) &= -\tfrac{z+2}{3}(z-1)(fg \bmod (z^2+z+1)) \\
&\quad +\tfrac{1}{3}(z^2+z+1) fg \bmod z-1 \\
&= -\tfrac{z+2}{3}(z-1)(m_1' - m_2' + (m_1'-m_3')z) \\
&\quad +\tfrac{1}{3}(z^2+z+1)(x_0+x_1+x_2)(y_0+y_1+y_2) \\
&= \tfrac{2}{3}(m_1'-m_2') + (-\tfrac{1}{3}(m_1'-m_2') + \tfrac{2}{3}(m_1 - m_3') + \tfrac{1}{3}m_0')z \\
&\quad +(-\tfrac{1}{3}(m_1'-m_2') - \tfrac{1}{3}(m_1'-m_3') + \tfrac{1}{3}m_0)z^2 - \tfrac{1}{3}(m_1'-m_3')z^3 \; .
\end{aligned}$$

Dies ergibt den Algorithmus

$$\begin{aligned}
&s_0 = y_0 + y_1 + y_2\ ,\ \ s_1 = y_0 - y_2\ \ ,\ \ s_2 = y_1 - y_2\ \ ,\ \ s_3 = y_0 - y_1\ , \\
&m_0 = \frac{x_0 + x_1 x_2}{3} s_0\ ,\ \ m_1 = \frac{x_0 - x_2}{3} s_1\ \ ,\ \ m_2 = \frac{x_1 - x_2}{3} s_2\ \ ,\ \ m_3 = \frac{x_0 - x_1}{3} s_3 \\
&w_0 = m_0 + m_1 - 2m_2 + m_3\ \ ,\ \ w_1 = m_0 + m_1 + m_2 - 2m_3\ \ ,\ \ w_2 = m_0 - 2m_1 + m_2 + m_3\ .
\end{aligned} \tag{3.4.3}$$

Er benötigt 4 Multiplikationen und 17 Additionen (Multiplikationen mit 2 werden durch Additionen realisiert). Die Anzahl der Multiplikationen hat sich gegenüber der Schulmethode für die Faltung der Länge 3 also gewaltig reduziert (von 9 auf 4). Die Anzahl der Additionen ist noch nicht endgültig.

Zur Reduzierung der Anzahl der Additionen führen wir nun das Hilfsmittel der Transposition ein. Es seien t Bilinearformen

$$w_k = \sum_{i=1}^{s} \sum_{j=1}^{r} a_{ijk} x_i y_j \quad , \quad k = 1, \ldots, t \tag{3.4.4}$$

auszuwerten. Dafür stehe ein Algorithmus der Form

$$w_k = \sum_{\ell=1}^{m} \gamma_{\ell k} \left(\frac{1}{f_\ell} \sum_{i=1}^{s} \alpha_{\ell i} x_i \right) \left(\sum_{j=1}^{r} \beta_{\ell j} y_j \right) \tag{3.4.5}$$

zur Verfügung mit ganzen Zahlen $\gamma_{\ell k}$, f_ℓ, $\alpha_{\ell i}$ $\beta_{\ell j}$. Zur Zählung der Operationen in (3.4.3) machen wir folgende Konventionen:

(a) Operationen auf den x_i werden nicht gezählt. Man stellt sich vor, daß die x_i immer die gleichen sind, so daß

$$\frac{1}{f_\ell} \sum_{i=1}^{s} \alpha_{\ell i} x_i \quad , \quad \ell = 1, \ldots, m$$

vorberechnet werden kann.

(b) Die Multiplikationen mit $\gamma_{\ell k}$, $\beta_{\ell j}$ werden durch Additionen realisiert.

So gezählt benötigt (3.4.5) m Multiplikationen.

Faltungen mit festem x und Fourier-Transformationen sind Beispiele für (3.4.4). Die oben gefundenen Algorithmen für die Faltungen der Länge 2 und 3 sind Beispiele für (3.4.5).

Unter dem zu (3.4.4) transponierten System von Bilinearformen verstehen wir das System

$$\sum_{k=1}^{t} \sum_{j=1}^{r} a_{ijk} z_k y_j \quad , \quad i = 1, \ldots, s \; . \tag{3.4.6}$$

Wir erhalten das transponierte System aus (3.4.4), indem wir die k-te Gleichung (3.4.4) mit z_k multiplizieren, über k summieren, und die Koeffizienten der x_i betrachten.

Wir wollen zeigen, daß (3.4.5) auch einen Algorithmus zur Auswertung des transponierten Systems (3.4.6) liefert, der ebenfalls mit m Multiplikationen auskommt. Dazu multiplizieren wir (3.4.6) mit x_i und summieren über i. Mit (3.4.5) ergibt sich

$$\begin{aligned}\sum_{i=1}^{s} x_i \sum_{k=1}^{t} \sum_{j=1}^{r} a_{ijk} z_k y_j &= \sum_{k=1}^{t} z_k \sum_{i=1}^{s} \sum_{j=1}^{r} a_{ijk} x_i y_j \\ &= \sum_{k=1}^{t} z_k \sum_{\ell=1}^{m} \gamma_{\ell k} \left(\frac{1}{f_\ell} \sum_{i=1}^{s} \alpha_{\ell i} x_i \right) \left(\sum_{j=1}^{r} \beta_{\ell j} y_j \right) \\ &= \sum_{i=1}^{s} x_i \sum_{\ell=1}^{m} \alpha_{\ell i} \left(\frac{1}{f_\ell} \sum_{k=1}^{t} \gamma_{\ell k} z_k \right) \left(\sum_{j=1}^{r} \beta_{\ell j} y_j \right) \; .\end{aligned}$$

Vergleichen wir den Koeffizienten von x_i, so erhalten wir

$$\sum_{k=1}^{t} \sum_{j=1}^{r} a_{ijk} z_k y_j = \sum_{\ell=1}^{m} \alpha_{\ell i} \left(\frac{1}{f_\ell} \sum_{l=1}^{t} \gamma_{\ell k} z_k \right) \left(\sum_{j=1}^{r} \beta_{\ell j} y_j \right) \; . \tag{3.4.7}$$

Dies ist ein Algorithmus zur Auswertung des transponierten Systems (3.4.6) mit m Multiplikationen.

Wir wenden die Transposition an auf die Faltung

$$w_k = \sum_{i=0}^{n-1} x_{i-k} y_k \quad , \quad k = 0, \ldots, n-1 \; .$$

Das dazu transponierte System ist

$$w'_i = \sum_{k=0}^{n-1} y_{k-i} z_k \quad , \quad i = 0, \ldots, n-1 \; ,$$

und dies ist auch eine Faltung. (3.4.7) ist also wieder ein Algorithmus zur Faltung mit m Multiplikationen, aber einer anderen (und hoffentlich geringeren) Anzahl von Additionen.

Wir führen die Transposition durch für den Algorithmus (3.4.3) zur Faltung der Länge 3. Die Koeffizienten f_ℓ, $\gamma_{\ell k}$, $\alpha_{\ell i}$, $\beta_{\ell j}$ sind in folgendem Schema zusammengestellt:

f	γ			α			β		
3	1	1	1	1	1	1	1	1	1
3	1	1	-2	1	0	-1	1	0	-1
3	-2	1	1	0	1	-1	0	1	-1
3	1	-2	1	1	-1	0	1	-1	0
	w_0	w_1	w_2	x_0	x_1	x_2	y_0	y_1	y_2

Die Anwendung von α und β zeilenweise erfordert jeweils 5 Additionen, die Anwendung von γ spaltenweise aber 12 (Multiplikation mit 2 wird durch Addition realisiert). In dieser Form benötigt der Algorithmus also 17 Additionen. Bei der Transposition wird α mit γ vertauscht, während β unverändert bleibt.

Die Anwendung von β (zeilenweise) erfordert nach wie vor 5 Additionen, die Anwendung von α (spaltenweise) aber nur noch 6 Additionen. Der transponierte Algorithmus (3.4.7) kommt also mit 11 (statt 17) Additionen und 5 Multiplikationen aus. Er lautetet:

$$s_0 = y_0 + y_1 + y_2 \quad , \quad s_1 = y_0 - y_2 \;\; , \;\; s_2 = y_1 - y_2 \;\; , \;\; s_3 = y_0 - y_1 \; ,$$

$$\begin{aligned}
m_0 &= \frac{z_0+z_1+z_2}{3}s_0 \quad , \quad m_1 = \frac{z_0+z_1-2z_2}{3}s_1 \quad , \\
m_2 &= \frac{-2z_0+z_1+z_2}{3} \quad , \quad m_3 = \frac{z_0-2z_1+z_2}{3} \\
w_0' &= m_0+m_1+m_3 \quad , \quad w_1' = m_0+m_1-m_3 \quad , \quad w_2' = m_0-m_1-m_2 \; .
\end{aligned} \tag{3.4.8}$$

Die Winograd-Faltung versucht mit möglichst wenig Multiplikationen auszukommen. Wir wollen eine Aussage über die benötigten Multiplikationen für Faltungen beliebiger Länge herleiten. Sie gilt für einen beliebigen Körper F der Charakteristik 0, also z.B. $F = \mathbb{R}, \mathbb{C}$. Die Kreisteilungspolynome ϕ_d (siehe z.B. Fischer/Sacher: Einführung in die Algebra, S. 172 ff.) in einem solchen Körper sind Polynome mit der Eigenschaft

$$x^n - 1 = \prod_{d/n} \phi_d(x) \; .$$

Ihre Anzahl k entspricht also der Anzahl der Teiler von n (einschließlich 1, n). Eine kleine Liste von Kreisteilungspolynomen ist

$$\begin{aligned}
\phi_1 &= x-1 \\
\phi_2 &= x+1 \\
\phi_3 &= x^2+x+1 \\
\phi_4 &= x^2+1 \\
\phi_5 &= x^4+x^3+x^2+x+1
\end{aligned}$$

Sie haben folgende Eigenschaften:

(i) ϕ_d hat nur ganzzahlige Koeffizienten

(ii) ϕ_d ist irreduzibel

(iii) ϕ_d, ϕ_d' sind für $d \neq d'$ teilerfremd.

Satz 3.4.1 *Sei F ein Körper der Charakteristik* 0 *und k die Anzahl der Teiler von n, welche $\leq n$ sind (1 und n eingeschlossen). Dann kann man die Faltung der Länge n in F mit $2n-k$ Multiplikationen ausführen (Vorberechnungen auf einem der Faktoren werden nicht gezählt).*

Beweis: Seien $p_1, \ldots, p_k$ die Kreisteilungspolynome zu n, also

$$x^n - 1 = p_1 p_2 \cdots p_n \ .$$

Sei $P_2 = p_2 \cdots p_k$. Wir wenden den Chinesischen Restsatz an und bestimmen q_1, $q_2 \in F[z]$, so daß

$$1 = q_2 p_1 + q_1 P_2 \ .$$

Dann ist wie oben für f, $g \in F[z]$

$$\begin{aligned} fg \bmod (z^n - 1) &= q_2 p_1((fg) \bmod P_2) + q_2 P_2((fg) \bmod p_1) \\ &= q_2 p_1(f \bmod P_2)(g \bmod P_2) + q_1 P_2(f \bmod p_1)(g \bmod p_1) \ . \end{aligned}$$

Da p_1, P_2 nur ganzzahlige Koeffizienten haben, können f und g mod P_2 bzw. mod p_1 durch Additionen berechnet werden. Die Multiplikation zweier Polynome q, f vom Grade k, ℓ kann mit $k+\ell+1$ Multiplikationen ausgeführt werden. man braucht ja nur $k+\ell+1$ paarweise verschiedene Punkte $z_0, \ldots, z_{k+1}$ aus dem Primkörper von F zu wählen und für fg die Lagrange'sche Interpolationsformel hinzuschreiben, also

$$(fg)(z) = \sum_{j=0}^{\ell+k} \omega_j(z) f(z_j) g(z_j) \ , \quad \omega_j(z) = \prod_{\substack{i=0 \\ j \neq j}}^{k+\ell} \frac{z - z_j}{z_i - z_j} \ .$$

Da die ω_j Koeffizienten aus dem Primkörper von F haben, können die Koeffizienten von fg durch Addition berechnet werden, sobald die $k + \ell + 1$ Produkte $f(z_j)g(z_j)$ bekannt sind. Also kann fg tatsächlich mit $k + \ell + 1$ Multiplikationen berechnet werden.

Sei nun $M(p)$ die Anzahl der Multiplikationen für die Berechnung von fg mod p, wobei p irgendein Polynom über dem Primkörper von F ist. Nach dem Chinesischen Restsatz gilt dann

$$\begin{aligned} M(x^n - 1) &= M(p_1) + M(P_2) \\ &= M(p_1) + M(p_2) + \cdots + M(p_k) \ . \end{aligned}$$

Wir haben gerade gesehen, daß $M(p) = 2$ Grad $(p) - 1$. Also folgt

$$M(x^n - 1) = \sum_{i=1}^{k} (2 \operatorname{Grad}(p_i) - 1) = 2n - k \ .$$

□

3.5 Die schnelle Fourier–Transformation nach Winograd

Winograd führt die Fourier-Transformation auf Faltungen zurück. Dazu definieren wir zunächst die Faltung auf einer beliebigen Gruppe G. Sei $|G| = n$, und seien x, $y \in \mathbb{C}^n$. Wir interpretieren x, y als Funktionen auf G, schreiben also $x = (x_s)_{s\in G}$, $y = (y_s)_{s\in G}$. Die Faltung $z = x * y$ ist dann eine ebensolche Funktion auf G, und zwar ist

$$\begin{aligned} z_g &= \sum_{\substack{s,t\in G\\ ts=g}} x_s y_t = \sum_{t\in G} x_{t^{-1}g} y_t \ , \\ &= \sum_{t\in G} x_{tg} y_{t^{-1}} \ . \end{aligned}$$

In Matrix-Form schreibt sich dies

$$z = XPy \quad , \qquad X = (x_{tg})_{g,t\in G}$$

mit einer Permutationsmatrix P. Anders ausgedrückt: Eine Faltung auf G wird durch eine Matrix beschrieben, in der die n Komponenten des Vektors x entsprechend der Gruppentafel von G angeordnet sind.

Nun zur Fourier-Transformation W_n der Länge n, also

$$W_n = (\omega^{k\ell})_{k,\ell=0,\dots,n} \quad , \qquad \omega = e^{-2\pi i/n} \ .$$

Wegen $\omega_n^n = 1$ kann man die Produkte im Exponenten mod n berechnen. Streicht man aus W_n die Zeilen und Spalten k mit $k = 0$ oder $k|n$, so bleibt übrig eine $(\varphi(n), \varphi(n))$-Matrix $\tilde{W}_n$ (φ ist die Euler'sche φ-Funktion, vgl. 2.4), in welcher die $\varphi(n)$ Elemente ω^k, $0 \leq k < n-1$, $k \neq 0$ und $ggt(n,k) = 1$ entsprechend der Gruppentafel von M_n angeordnet sind.

Wir haben eben gesehen, daß eine solche Matrix nichts anderes als eine Faltung auf M_n darstellt. Ein großer Teil der zur Anwendung von W_n notwendigen Rechenoperationen kann also als Faltung auf M_n ausgeführt werden.

Aus Satz 2.4.3 wissen wir, daß M_n für $n = p^r$, $p > 2$, $r > 1$ zyklisch ist. Dann ist also $\tilde{W}_n$ sogar eine Faltung auf C_n, und diese können nach den Methoden aus §4 schnell ausgeführt werden. Diese Art der schnellen Fourier-Transformation wird auch Rader FFT genannt. Wir führen dies nun für einige Beispiele durch.

Beginnen wir mit $n = 3$. Nach Satz 2.4.3 ist $M_3 \cong C_2$. In der Tat sind in

$$W_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \quad , \quad \omega = e^{-2\pi i/3}$$

die beiden letzten Zeilen und Spalten eine Faltung der Länge 2. Diese können wir mit 2 Multiplikationen und 4 Additionen ausführen, dazu kommen dann nach 4 Additionen in Zeile und Spalte 0. Von den acht Additionen kann man noch zwei einsparen, wenn man

$$\begin{pmatrix} \hat{y}_1 - \hat{y}_0 \\ \hat{y}_2 - \hat{y}_0 \end{pmatrix} = \begin{pmatrix} \omega - 1 & \omega^2 - 1 \\ \omega^2 - 1 & \omega - 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

schreibt. Ausführung dieser Faltung mit schnellem Algorithmus ergibt

$$\begin{aligned} \hat{y}_1 - \hat{y}_0 &= \frac{\omega + \omega^2 - 2}{2}(y_1 + y_2) + \frac{\omega - \omega^2}{2}(y_2 - y_2) \\ \hat{y}_2 - \hat{y}_0 &= \frac{\omega + \omega^2 2}{2}(y_1 + y_2) - \frac{\omega - \omega^2}{2}(y_1 - y_2) \end{aligned}$$

Damit kommen wir zu folgendem Algorithmus mit 2 Multiplikationen und 6 Additionen:

$$\begin{aligned} s_1 &= y_1 + y_2 \quad &,\quad s_2 &= y_1 - y_2 \\ m_1 &= \frac{\omega + \omega^2 - 2}{2} s_1 \;, & m_2 &= \frac{\omega - \omega^2}{2} s_2 \\ \hat{y}_0 &= y_0 + s_1 \quad &,\quad s_3 &= \hat{y}_0 + m_1 \\ \hat{y}_1 &= s_3 - m_2 \quad &,\quad \hat{y}_2 &= s_3 + m_2 \;. \end{aligned} \tag{3.5.1}$$

Wir bemerken, daß

$$\begin{aligned} \tfrac{\omega+\omega^2-2}{2} &= \tfrac{\omega+\overline{\omega}-2}{2} &= \cos(2\pi/3) - 1 \\ \tfrac{\omega-\omega^2}{2} &= \tfrac{\omega-\overline{\omega}}{2} &= i\sin(2\pi/3) \end{aligned}$$

reell bzw. rein imaginär sind, so daß sich die beiden Multiplikationen von (3.5.1) in insgesamt 4 reellen Multiplikationen ausführen lassen.

Im Falle $n = 4$ haben wir nach Satz 2.4.3 $M_4 = \{1, 3\} = C_2$. Die entsprechende Umordnung von W_4 ist

$$\begin{array}{c|rrrr} & 0 & 2 & 1 & 3 \\ \hline 0 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & i & -i \\ 3 & 1 & -1 & -i & i \end{array}$$

Ein schneller Algorithmus ist

$$\begin{aligned} s_2 &= y_0 + y_2 \quad s_1 = y_1 + y_3 \quad , \quad s_2 = y_0 - y_2 \quad , \quad s_3 = y_1 - y_3 \\ \hat{y}_0 &= s_0 + s_1 \quad \hat{y}_1 = s_2 + i s_3 \quad , \quad \hat{y}_2 = s_0 - s_1 \quad , \quad \hat{y}_3 = s_2 - i s_3 \; . \end{aligned} \tag{3.5.2}$$

Dieser Algorithmus kommt also ganz ohne Multiplikation aus. Die Anzahl der Additionen ist 8.

Als nächstes nehmen wir $n = 7$. Dann ist $M_7 = C_6 = \{1, a, \ldots, a^5\}$ mit dem erzeugenden Element $a = 3$, vgl. Abschnitt 2.4. Also ist $M_7 = \{1, 3, 2, 6, 4, 5\}$. Wir ordnen W_7 so an, daß die Zeilen und Spalten $1, 2, 3, 4, 5, 6$ in der Reihenfolge 1, 3, 2, 6, 4, 5 erscheinen. Notieren wir nur die Exponenten, so bedeutet dies

$$\begin{array}{c|ccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 0 & 2 & 4 & 6 & 1 & 3 & 5 \\ 3 & 0 & 3 & 6 & 2 & 5 & 1 & 4 \\ 4 & 0 & 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 0 & 5 & 3 & 1 & 6 & 4 & 2 \\ 6 & 0 & 6 & 5 & 4 & 3 & 2 & 1 \end{array} \;\underset{\text{Zeilen}}{\longrightarrow}\; \begin{array}{c|ccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 3 & 3 & 4 & 5 & 6 \\ 3 & 0 & 3 & 6 & 2 & 5 & 1 & 4 \\ 2 & 0 & 2 & 4 & 6 & 1 & 3 & 5 \\ 6 & 0 & 6 & 5 & 4 & 3 & 2 & 1 \\ 4 & 0 & 4 & 1 & 5 & 2 & 6 & 3 \\ 5 & 0 & 5 & 3 & 1 & 6 & 4 & 2 \end{array} \;\underset{\text{Spalten}}{\longrightarrow}\; \begin{array}{c|ccccccc} & 0 & 1 & 3 & 2 & 6 & 4 & 5 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 3 & 2 & 6 & 4 & 5 \\ 3 & 0 & 3 & 2 & 6 & 4 & 5 & 1 \\ 2 & 0 & 2 & 6 & 4 & 5 & 1 & 3 \\ 6 & 0 & 6 & 4 & 5 & 1 & 3 & 2 \\ 4 & 0 & 4 & 5 & 1 & 3 & 2 & 6 \\ 5 & 0 & 5 & 1 & 3 & 2 & 6 & 4 \end{array}$$

Es entsteht die Gruppentafel von C_6, wie es auf Grund der Isomorphie $M_7 \cong C_6$ zu erwarten ist.

Dies wird in folgender Weise in einen Algorithmus umgesetzt. Die Faltung der Länge 6 realisieren wir durch eine Faltung der Länge 2 auf

Blöcken, die ihrerseits Faltungen der Länge 3 sind. Die Faltungen der Länge 3 führen wir durch Algorithmus (3.4.8) aus ($M(3) = 4$, $A(3) = 11$), für die Faltungen der Länge 2 ist $M(2) = 2$, $A(2) = 4$. Nach 3.4 ist dann $M(6) = M(2)M(3) = 8$, $A(6) = M(2)A(3) + 3A(2) = 34$. Die Faltung der Länge 6 läßt sich also mit 8 Multiplikationen und 34 Additionen ausführen. Bei den Faltungen der Länge 3 fallen auch die Summen $y_1 + y_2 + y_3$ und $y_4 + y_5 + y_6$ an. Die Addition $\hat{y}_6 = (y_1 + y_2 + y_3) + (y_4 + y_5 + y_6)$ fällt bei der Blockfaltung an. Jetzt sind nur noch die 6 Additionen mit $\hat{y}_0$ auszuführen. Nun ist aber jeder der beiden Faltungsalgorithmen für die Faltung der Länge 3 von der Form $w'_k = m_0 + \cdots$, $k = 0, 1, 2$ (vgl. 3.4.8). Also braucht man für jede dieser Faltungen nur $m_0 + \mathrm{y}_0$ zu bilden, kommt also mit 2 Additionen aus. Insgesamt reichen also 36 Additionen und 8 Multiplikationen.

Nehmen wir als weiteres Beispiel $n = 9 = 3^2$. Dann ist nach Satz 2.4.3 $M_9 = \{1, 2, 4, 5, 7, 8\} \cong C_6 = \{1, a, \ldots, a^5\}$ mit $a = 5$ (wir könnten ebensogut $a = 2$ nehmen, nicht aber $a = 4$). Ordnen wir demgemäß $M_9 = \{1, 5, 7, 8, 4, 2\}$ an und schreiben wir in W_9 erst die (nicht zu M_9 gehörigen) Indizes 0, 3, 6 an, so nimmt W_9 die Form

	0	3	6	1	5	7	8	4	2
0	0	0	0	0	0	0	0	0	0
3	0	0	0	3	6	3	6	3	6
6	0	0	0	6	3	6	3	6	3
1	0	3	6	1	5	7	8	4	2
5	0	6	3	5	7	8	4	2	1
7	0	3	6	7	8	4	2	1	5
8	0	6	3	8	4	2	1	5	7
4	0	3	6	4	2	1	5	7	8
3	0	6	3	2	1	5	7	8	4

Man erkennt die Faltung der Länge 6 in den letzten 6 Zeilen und Spalten.

Für die Winograd-FFT kann man folgende Tabelle verifizieren:

n	Multiplikationen	Additionen	Herleitung
2	0	2	$--$
3	2	6	$M_3 \cong C_2$
4	0	8	$M_4 \cong C_2$
5	5	17	$M_5 \cong C_4$
7	8	36	$M_7 \cong C_2 \otimes C_3$
8	2	26	$M_8 \cong C_2 \otimes C_2$
9	10	45	$M_9 \cong C_6$
16	10	74	$M_16 \cong C_2 \otimes C_4$

Aus dieser Tabelle kann man nun mittels Primfaktorzerlegung und Tensorprodukten nach §3 Algorithmen für andere Längen n herleiten.

3.6 Schnelle Poisson-Löser

Als Anwendung der schnellen Fourier-Transformation wollen wir nun einen schnellen Algorithmus zur Lösung der Randwertaufgabe

$$\Delta u = f \quad \text{in} \quad \Omega \,, \qquad u = 0 \quad \text{auf} \quad \partial\Omega \tag{3.6.1}$$

für ein Rechteck Ω herleiten. Durch Diskretisierung auf einem Gitter der Schrittweise h wird (3.6.1) angenähert durch

$$\begin{aligned} &-4u_{k,\ell} + u_{k+1,\ell} + u_{k-1,\ell} + u_{k,\ell-1} + u_{k,\ell+1} = h^2 f_{k,\ell} \,, \quad 0 < k < m,\ 0 < \ell < n \\ &u_{k\ell} = 0 \,, \quad k = 0 \quad \text{oder} \quad m \quad \text{oder} \quad \ell = 0 \ \text{ oder } n \,. \end{aligned} \tag{3.6.2}$$

Dabei ist $u_{k,\ell}$ eine Näherung für u in dem durch k, ℓ bezeichneten Gitterpunkt, und $f_{k,\ell}$ ist der dortige Funktionswert von f. Unser Ziel ist es, (3.6.2) in $\mathrm{O}(mn\log(mn))$ Rechenoperationen zu lösen.

Wir beginnen mit dem eindimensionalen Fall, also

$$u'' = f \quad \text{in} \quad 0 < x < a \,, \qquad u(0) = u(a) = 0 \,. \tag{3.6.3}$$

Durch Diskretisierung geht dies über in

$$\begin{aligned} &u_{k+1} - 2u_k + u_{k-1} = h^2 f_k \,, \quad 0 < k < m \\ &u_0 = u_m = 0 \,. \end{aligned} \tag{3.6.4}$$

Führen wir die Vektoren und Matrizen

$$U = \begin{pmatrix} u_1 \\ \vdots \\ u_{m-1} \end{pmatrix} \quad , \quad F = h^2 \begin{pmatrix} f_1 \\ \vdots \\ f_{m-1} \end{pmatrix} \quad T_{m-1} = \begin{pmatrix} -2 & 1 & & & \mathbf{O} \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & & \ddots & \\ \mathbf{O} & & & 1 & -2 \end{pmatrix}$$

ein, so schreibt sich (3.6.4) als

$$T_{m-1} U = F \,. \tag{3.6.5}$$

Da T_{m-1} tridiagonal ist, kann man dies durch Elimination in $\mathrm{O}(m)$ Rechenoperationen lösen. Wir gehen komplizierter vor und berechnen zunächst die Eigenwerte und Eigenvektoren von T_m. Dazu setzen wir

$$c_k = \cos(\theta)\ , \qquad s_k = \sin(k\theta)$$

und bekommen dann aus den Additionstheoremen

$$\begin{array}{rclcrcl} s_{k-1} & = & c_1 s_k - s_1 c_k & , & s_{k+1} & = & c_1 s_k + s_1 c_k\ , \\ c_{k-1} & = & c_1 c_k + s_1 s_k & , & c_{k+1} & = & c_1 s_k - s_1 s_k\ . \end{array}$$

Daraus folgt weiter

$$s_{k-1} - 2c_1 s_k + s_{k+1} = 0\ , \quad c_{k-1} - 2c_1 c_k + c_{k+1} = 0\ .$$

Die erste Beziehung schreiben wir in der Form

$$s_{k-1} - 2s_k + s_{k+1} = -2(1-c_1)s_k = -4\sin^2(\frac{\theta}{2})s_k\ ,$$

wobei wir $1 - \cos\theta = 2\sin^2(\theta/2)$ benutzt haben. Setzen wir hier $\theta = \theta_j = \pi j/m$ und bezeichnen wir mit s^j den zugehörigen Vektor $(s_1, \ldots, s_{m-1})$, so ist $s_m = 0$ und damit s^j Eigenvektor zu T_{m-1} zum Eigenwert $\lambda_j = -4\sin^2(\pi j/2m)$. Mit den Matrizen

$$S_{m-1} = (s^1, \ldots, s^{m-1})\ , \qquad \Lambda_{m-1} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{m-1} \end{pmatrix}$$

hat man als

$$S_{m-1}^{-1} T_{m-1} S_{m-1} = \Lambda_{m-1}\ . \tag{3.6.6}$$

Nun behandeln wir den zweidimensionalen Fall (3.6.2). Wir ordnen die $u_{k\ell}$ zeilenweise an, definieren also einen $(n-1)(m-1)$-Vektor U mit den Komponenten ($m = 4$, $n = 5$)

$$u_{11}, u_{12}, u_{13}, u_{14}, u_{21}, u_{22}, u_{23}, u_{24}, u_{31}, u_{32}, u_{33}, u_{34}\ .$$

Schreiben wir die Gleichungen (3.6.2) ebenfalls zeilenweise hin, so entsteht

$$
\left(
\left[\begin{array}{cccc|cccc|cccc}
-2 & 1 & & & & & & & & & & \\
1 & -2 & 1 & & & & & & & & & \\
 & 1 & -2 & 1 & & & & & & & & \\
 & & 1 & -2 & & & & & & & & \\
\hline
 & & & & -2 & 1 & & & & & & \\
 & & & & 1 & -2 & 1 & & & & & \\
 & & & & & 1 & -2 & 1 & & & & \\
 & & & & & & 1 & -2 & & & & \\
\hline
 & & & & & & & & -2 & 1 & & \\
 & & & & & & & & 1 & -2 & 1 & \\
 & & & & & & & & & 1 & -2 & 1 \\
 & & & & & & & & & & 1 & -2
\end{array}\right]
+
\left[\begin{array}{cccc|cccc|cccc}
-2 & & & & 1 & & & & & & & \\
 & -2 & & & & 1 & & & & & & \\
 & & -2 & & & & 1 & & & & & \\
 & & & -2 & & & & 1 & & & & \\
\hline
1 & & & & -2 & & & & 1 & & & \\
 & 1 & & & & -2 & & & & 1 & & \\
 & & 1 & & & & -2 & & & & 1 & \\
 & & & 1 & & & & -2 & & & & 1 \\
\hline
 & & & & 1 & & & & -2 & & & \\
 & & & & & 1 & & & & -2 & & \\
 & & & & & & 1 & & & & -2 & \\
 & & & & & & & 1 & & & & -2
\end{array}\right]
\right)
\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \underline{U_4} \\ U_5 \\ U_6 \\ U_7 \\ \underline{U_8} \\ U_9 \\ U_{10} \\ U_{11} \\ U_{12} \end{pmatrix}
=
\begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ \underline{F_4} \\ F_5 \\ F_6 \\ F_7 \\ \underline{U_8} \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \end{pmatrix} .
$$

Dies kann man in der Form

$$(I_3 \otimes T_4 + T_3 \otimes I_4)U = F$$

schreiben. Im allgemeinen erhält man

$$(I_{m-1} \otimes T_{n-1} + T_{m-1} \otimes I_{n-1})U = F \; . \tag{3.6.7}$$

Satz 3.6.1 *Seien A_n, B_m (n,n)- bzw. (m,m)-Matrizen, und sei $M = I_m \otimes A_n + B_m \otimes I_n$ invertierbar. Sei $U_n^{-1} A_n U_n = \Lambda_n$ und $V_m^{-1} B_m V_m =$*

M_m. Dann gilt

$$M^{-1} = (V_m \otimes U_n)(I_m \otimes \Lambda_n + M_m \otimes I_n)^{-1}(V_m^{-1} \otimes U_n^{-1}) \ .$$

Beweis: Dies folgt sofort aus den Rechenregeln

$$(A \otimes B)(C \otimes D) = AC \otimes BD \ , \quad (A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \ .$$

□

Sind Λ_n, M_m diagonal, so auch $I_n \otimes \Lambda_n + M_n \otimes I_n$ und damit leicht zu invertieren. In der Anwendung auf (3.6.7) sind U_n, V_m die Sinus-Transformation, die $n \log n$ bzw. $m \log m$ Operationen erfordern. $V_m \otimes U_n$ verlangt also m Sinus-Transformationen der Länge n und n Sinus-Transformationen der Länge m, also

$$mn \log n + nm \log m = mn \log(mn)$$

Operationen. Das gleiche gilt für $V_m^{-1} \otimes U_n^{-1}$, und die Anwendung von $(I_m \otimes \Lambda_n + M_m \otimes I_n)^{-1}$ verlangt mn Operationen. Also kann (3.6.2) in $O(mn \log(mn)$ Operationen gelöst werden.