

Diskrete und Schnelle Fourier-Transformation

Diese Aufzeichnungen habe ich als Begleittext zu dem entsprechenden Abschnitt der Vorlesung *Einführung in die Theoretische Informatik III* verfasst. Sie sollen von der Motivation für schnell auszuführende Faltungsoperationen zu den mathematischen Prinzipien der Diskreten und Schnellen Fourier-Transformation führen. Darüber hinaus werden Anregungen für das Weiterstudium in verschiedene Richtungen, von mathematischen Grundlagen bis zu Implementierung und Anwendung gegeben. Der Leser wird ausdrücklich aufgefordert, sich nicht auf ein "theoretisches" Studium zu beschränken, sondern diese Prinzipien und Techniken durch konkretes Rechnen und Anwenden zu erfahren. (Hinweis: MAPLE enthält FFT-Prozeduren.)

V. Strehl, 5. Juli 1994

Inhaltsverzeichnis

1	Beispiel: die Faltungsoperation	2
2	"Klassische" Fouriertransformation	3
3	Das Abtasttheorem	4
4	Darstellungen von Polynomen	5
5	Multiplikation von Polynomen	8
6	Polynome und ihre Wurzeln	9
7	Einheitswurzeln	11
8	Diskrete Fourier-Transformation (DFT)	14
9	DFT-Matrizen	15
10	Schnelle Fourier-Transformation (FFT)	17
11	Weiterführende Bemerkungen	22
11.1	Zur Implementierung	22
11.1.1	Iterative Implementierung	22
11.1.2	Parallele Implementierung	22
11.1.3	Rundungsfehler	22
11.2	Zur Komplexität	22
11.2.1	Lineare Komplexität	22
11.2.2	Faktorisierung	25
11.3	Algebraische Aspekte	30
11.3.1	Transformation über endlichen Körpern	30
11.3.2	Transformation über anderen Gruppen	33
11.4	Anwendungen	33
11.5	Historische Bemerkung	33
11.6	Literatur	34

1 Beispiel: die Faltungsoperation

Als motivierendes Beispiel betrachten wir ein lineares, (zeit-)diskretes Impuls-Antwort-System (*impulse-response-system*), das gegeben ist durch eine Folge (*Filter*)

$$\mathbf{f} = (f_0, f_1, f_2, \dots)$$

von (komplexen) Zahlen. Die Funktionsweise des Systems ist spezifiziert durch die Aussagen

- Ein Impuls p zur Zeit t verursacht eine Antwort *response* $p \cdot f_i$ zur Zeit $t+i$, ($i \geq 0$)
- Die Antworten auf zu verschiedenen Zeitpunkten gegebene Impulse überlagern sich additiv.

Daraus folgt: ist dem System eine Folge $\mathbf{p} = (p_0, p_1, p_2, \dots)$ als input (*pulse*) gegeben, so reagiert das System mit der output-Folge (*response*) $\mathbf{r} = (r_0, r_1, r_2, \dots)$ wobei

$$r_t = p_0 \cdot f_t + p_1 \cdot f_{t-1} + \dots + p_t \cdot f_0 \quad (t \geq 0)$$

Man bezeichnet $\mathbf{r} = \mathbf{p} * \mathbf{f}$ als die *Faltung* der Folgen \mathbf{p} und \mathbf{f} . Repräsentiert man die Folgen durch Potenzreihen

$$p(z) := \sum_{i \geq 0} p_i z^i, \quad f(z) := \sum_{j \geq 0} f_j z^j, \quad r(z) := \sum_{k \geq 0} r_k z^k$$

so ist die Faltungsoperation nichts anderes als die übliche (Cauchy-)Multiplikation von Potenzreihen:

$$r(z) = p(z) \cdot f(z) = \sum_{k \geq 0} \left(\sum_{i+j=k} p_i \cdot f_j \right) z^k$$

Speziell ist der Fall von Interesse, wo die beteiligten Potenzreihen nur endlich viele von 0 verschiedene Koeffizienten haben (das ist ja das, was man in endlicher Zeit beobachten kann). Dann sind $p(z), f(z), r(z)$ Polynome und die Faltungsoperation reduziert sich auf die Multiplikation von Polynomen.

Von erheblicher praktischer Bedeutung ist es, die Faltung, also das Produkt von Polynomen, möglichst effizient zu berechnen. Der direkte Weg, die *Schulmethode*, besteht darin, die Faltungsformel direkt auszuwerten: sei dazu $\deg p = m$, $\deg f = n$, also $\deg r = m + n$, dann ist

$$r_k = \sum_{i+j=k} p_i \cdot f_j = \sum_{j=\max\{0, k-m\}}^{\min\{k, n\}} p_{k-j} \cdot f_j$$

und dies erfordert $(m+1)(n+1)$ Multiplikationen und $m n$ Additionen im Koeffizientenbereich.

Wir wissen bereits, daß es besser geht: Methode von KARATSUBA. Geht es noch besser? JA! Das ist der Zweck der *schnellen Fourier-Transformation*.

2 “Klassische” Fouriertransformation

Die klassische Integraltransformation der *Fouriertransformation* ist definiert durch

$$\mathcal{F} : f(z) \mapsto f^\wedge(s) := \int_{-\infty}^{\infty} f(z) e^{-2\pi i s z} dz$$

Sie bildet also, sofern die Definition analytischen Sinn macht, Funktionen $f : \mathbf{R} \rightarrow \mathbf{C}$ in Funktionen gleichen Typs ab. Dies tut auch die *inverse Fourier-Transformation*

$$\mathcal{F}_{inv} : f(z) \mapsto f^\vee(s) := \int_{-\infty}^{\infty} f(z) e^{2\pi i s z} dz$$

Unter geeigneten Voraussetzungen sind beide Transformationen invers zueinander:

$$\mathcal{F}_{inv}(\mathcal{F}(f))(z) = (f^\wedge)^\vee(z) = f(z)$$

Das bedeutet

$$f(z) = \int_{-\infty}^{\infty} f^\wedge(s) e^{2\pi i s z} ds$$

d.h. die Funktion $f(z)$ lässt sich als “Überlagerung” von periodischen Funktionen darstellen, und $f^\wedge(s)$ gibt an, mit welcher “Leistung” die periodische Funktion $e^{2\pi i s z}$ (als Funktion von z , mit “Frequenz” $2\pi s$) in $f(z)$ “enthalten” ist. Man nennt $f^\wedge(s)$ auch das “Spektrum” von $f(z)$.

Es gibt also zwei gleichwertige Darstellungen einer Funktion: einmal die direkte Darstellung durch ihren Funktionsverlauf — das nennt man die Darstellung im *Zeitbereich* —, dann die Darstellung durch die in ihr enthaltenen periodischen Funktionen — das nennt man die Darstellung im *Frequenzbereich* — symbolisch:

$$\begin{array}{ccc} f(z) & \begin{array}{c} \xrightarrow{\mathcal{F}} \\ \xleftarrow{\mathcal{F}_{inv}} \end{array} & f^\wedge(s) \\ | & & | \\ \text{Zeitbereich} & & \text{Frequenzbereich} \end{array}$$

Die Transformationen \mathcal{F} und \mathcal{F}_{inv} sind lineare Transformationen. Eine ihrer bemerkenswertesten und nützlichsten Eigenschaften wird durch den sogenannten *Faltungssatz* ausgedrückt. Dazu bezeichne

$$(f * g)(z) := \int_{-\infty}^{\infty} f(t) g(z - t) dt$$

die *Faltung* der Funktionen $f(z)$ und $g(z)$ (die Analogie zum Beispiel im vorigen Abschnitt ist unübersehbar). Dann gilt

$$(f * g)^\wedge(s) = f^\wedge(s) \cdot g^\wedge(s)$$

oder, anders ausgedrückt,

$$(f * g)(z) = (f^\wedge \cdot g^\wedge)^\vee(z)$$

Dabei bezeichnet der Punkt “ \cdot ” die *punktweise* Multiplikation von Funktionen – also eine Operation, die man als sehr viel “einfacher” als das Faltungsprodukt ansprechen kann. Die letzte Gleichung drückt die fundamentale Tatsache aus, daß man das Faltungsprodukt auf dem “Umweg” über die Darstellung im Frequenzbereich mittels des punktweisen Produkts berechnen kann. Im Schema dargestellt:

$$\begin{array}{ccc}
 f, g & \xrightarrow{\mathcal{F}} & f^\wedge, g^\wedge \\
 * \downarrow & & \downarrow \bullet \\
 f * g & \xleftarrow{\mathcal{F}_{inv}} & f^\wedge \cdot g^\wedge \\
 \text{Zeitbereich} & & \text{Frequenzbereich}
 \end{array}$$

Dieser Umweg “lohnt” sich, falls es gelingt, die Transformationen \mathcal{F} und \mathcal{F}_{inv} zu Kosten zu berechnen, die deutlich unter denen für die Berechnung von der Faltungsoperation liegen.

Der Witz der Situation liegt darin, daß dieses Konzept auch im “diskreten” Bereich existiert und funktioniert (Diskrete Fouriertransformation — DFT), und daß man dort in der Tat mit einem Aufwand die Fourier-Transformation ausführen kann, der unter dem für die “normale” Faltung — sprich: Polynom-Multiplikation — liegt (“Schnelle” Fourier-Transformation — FFT).

3 Das Abtasttheorem

Zur Motivation des folgenden betrachten wir Funktionen $f : \mathbf{R} \rightarrow \mathbf{C}$ mit Periode 1, d.h. $\forall t : f(t+1) = f(t)$. Eine solche Funktion $f(t)$ heißt *W-bandbeschränkt*, wenn sie als Linearkombinationen der $2W+1$ einfachen periodischen Funktionen $e^{2\pi ikt}$ mit $-W \leq k \leq W$ dargestellt werden kann, wenn es also (komplexe) Konstante F_k , ($-W \leq k \leq W$) gibt mit

$$f(t) = \sum_{k=-W}^W F_k e^{2\pi ikt} \quad (t \in \mathbf{R})$$

Die Funktion $f(t)$ wird in diesem Fall mit dem Koeffizientenvektor $[F_{-W}, F_{-W+1}, \dots, F_W]$ identifiziert.

Folgendes praktische Problem stellt sich nun:

Kann man die Funktion $f(t)$ (also den Koeffizientenvektor $[F_{-W}, F_{-W+1}, \dots, F_W] \in \mathbf{C}^{2W+1}$) “rekonstruieren”, falls man von der Funktion $f(t)$ nur die T “Abtastwerte”

$$f(0), f\left(\frac{1}{T}\right), f\left(\frac{2}{T}\right), \dots, f\left(1 - \frac{1}{T}\right)$$

kennt ?

Sei nun $\omega_T := e^{2\pi i/T}$. Diese Zahl ist eine komplexe T -te *Einheitswurzel*, denn es gilt $(\omega_T)^T = 1$. Sie ist *primitiv* in dem Sinne, daß die T Potenzen $\omega_T^0 = 1, \omega_T^1, \omega_T^2, \dots, \omega_T^{T-1}$

gerade die T verschiedenen Lösungen der Gleichung $X^T = 1$ in \mathbf{C} sind. (Genauerer zu Einheitswurzeln im Abschnitt 7). Für $0 \leq \tau < T$ gilt nun

$$f\left(\frac{\tau}{T}\right) = \sum_{k=-W}^W F_k (\omega_T)^{\tau k} = \omega_T^{-W\tau} \sum_{k=0}^{2W} F_{k-W} \omega_T^{\tau k}$$

In Matrixform geschrieben bedeutet dies

$$\begin{pmatrix} f(0) \\ f(1/T) \\ \vdots \\ f(1-1/T) \end{pmatrix} = \text{Diag}(1, \omega_T^{-W}, \omega_T^{-2W}, \dots, \omega_T^{-(T-1)W}) \begin{pmatrix} \ddots & & & & \\ & \omega_T^{\tau k} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \omega_T^{\tau k} \end{pmatrix}_{\substack{0 \leq \tau < T \\ 0 \leq k \leq 2W}} \begin{pmatrix} F_{-W} \\ F_{-W+1} \\ \vdots \\ F_W \end{pmatrix}$$

wobei Diag eine Diagonalmatrix mit den entsprechenden Einträgen bezeichnet.

Rekonstruierbarkeit bedeutet, daß die lineare Transformation

$$\mathbf{F} = (F_{-W}, F_{-W+1}, \dots, F_W) \mapsto (f(0), f(1), \dots, f(T-1/T)) = \mathbf{f}$$

injektiv ist, also den Kern $\{0\}$ haben muß. Dafür muß aber notwendig $T > 2W$ sein:

die Abtastfrequenz T muß größer als das Doppelte der Grenzfrequenz W sein.

Ist $T > 2W$, so hat die Matrix $(\omega_T^{\tau k})_{0 \leq \tau < T, 0 \leq k \leq 2W}$ Maximalrang $2W + 1$, denn die ersten $2W + 1$ Spalten dieser Matrix bilden eine VANDERMONDE-Matrix (siehe folgenden Abschnitt)

$$\mathbf{V}(1, \omega_T, \omega_T^2, \dots, \omega_T^{2W}) = (\omega_T^{jk})_{0 \leq j, k \leq 2W} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_T & \omega_T^2 & \dots & \omega_T^{2W} \\ 1 & \omega_T^2 & \omega_T^4 & \dots & \omega_T^{4W} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_T^{2W} & \omega_T^{4W} & \dots & \omega_T^{4W^2} \end{pmatrix}$$

und da die Argumente paarweise verschieden sind, ist diese Matrix regulär. Im Fall $T = 2W + 1$ ist also Rekonstruierbarkeit gesichert (die injektive Transformation ist dann ja auch surjektiv). Im Fall $T > 2W + 1$ ist Rekonstruierbarkeit gegeben, falls der Abtastvektor \mathbf{f} im Bildbereich der Transformation liegt.

4 Darstellungen von Polynomen

Ist k ein Körper, so bezeichnet $k[X]$ der Ring der Polynome in einer Variablen X mit Koeffizienten in k . Ist $A(X) = \sum_{j=0}^{n-1} a_j X^j \in k[X]$, so bezeichnet man mit $\deg A = \max\{j; a_j \neq 0\}$ den Grad von $A(X)$ und n als Gradschranke. Der Grad des Nullpolynoms wird mit $-\infty$ festgesetzt. Mit $k[X]_n$ wird die Menge aller Polynome $A(X) \in k[X]$ bezeichnet, deren Grad $< n$ ist, für die also n eine Gradschranke ist.

$k[X]_n$ ist ein n -dimensionaler Vektorraum über dem Körper k . Tatsächlich ist

$$\underbrace{A(X) = \sum_{j=0}^{n-1} a_j X^j}_{\in k[X]_n} \mapsto \underbrace{[a_0, a_1, \dots, a_{n-1}] =: \mathbf{A}}_{\in k^n}$$

ein Isomorphismus von Vektorräumen. Man bezeichnet \mathbf{A} als die *Koeffizientendarstellung* von $A(X)$.

Die Addition von Polynomen entspricht der Addition von Vektoren. Für die Addition zweier Polynome aus $k[X]_n$ benötigt man üblicherweise n Additionen in k .

Ist nun $x_0 \in k$, so bezeichnet man die Abbildung

$$\mathcal{L}_{x_0} : k[X] \rightarrow k : A(X) \mapsto A(x_0)$$

als die *Auswertung an der Stelle x_0* . Dies ist ein Skalarprodukt von Vektoren, denn für $A(X) \in k[X]_n$ gilt:

$$\begin{aligned} A(x_0) &= \sum_{j=0}^{n-1} a_j x_0^j = (1, x_0, \dots, x_0^{n-1}) \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \\ &= a_0 + x_0 (a_1 + x_0 (\dots + x_0 (a_{n-2} + x_0 a_{n-1}) \dots)) \end{aligned}$$

Die Auswertungsvorschrift, die in der zweiten Zeile enthalten ist, nennt man das *Horner-Schema*. Es besagt, daß für $A(X) \in k[X]_n$ die Berechnung von $\mathcal{L}_{x_0}(A)$ mit $n - 1$ Additionen und $n - 1$ Multiplikationen in k möglich ist.

Die Abbildung \mathcal{L}_{x_0} ist ein Ring- und Vektorraumhomomorphismus, sie ist mit Addition, Skalarmultiplikation und Multiplikation verträglich:

$$\mathcal{L}_{x_0}(A+B) = \mathcal{L}_{x_0}(A) + \mathcal{L}_{x_0}(B) , \quad \mathcal{L}_{x_0}(a \cdot A) = a \cdot \mathcal{L}_{x_0}(A) , \quad \mathcal{L}_{x_0}(A \cdot B) = \mathcal{L}_{x_0}(A) \cdot \mathcal{L}_{x_0}(B)$$

Neben der Koeffizientendarstellung gibt es noch weitere Darstellungen von Polynomen — man denke nur daran, daß ein normiertes Polynom (höchster Koeffizient = 1) durch seine Nullstellen eindeutig bestimmt ist: der Übergang zur Koeffizientendarstellung geschieht mit Hilfe der *elementarsymmetrischen Funktionen*. Allgemeiner gilt:

Ein Polynom $A(X) \in k[X]_n$ ist durch seine Werte an n verschiedenen Stellen eindeutig bestimmt.

Etwas genauer formuliert: sind x_0, x_1, \dots, x_{n-1} Elemente von k , so betrachte man die Abbildung der *simultanen Auswertung*

$$\mathcal{L}_{x_0, \dots, x_{n-1}} : k[X] \rightarrow k^n : A(X) \mapsto [\mathcal{L}_{x_0}(A), \dots, \mathcal{L}_{x_{n-1}}(A)] = [A(x_0), \dots, A(x_{n-1})]$$

Es ist eine fundamentale Tatsache, daß gilt:

Für paarweise verschiedene $x_0, x_1, \dots, x_{n-1} \in k$ ist die simultane Auswertung

$$\mathcal{L}_{x_0, \dots, x_{n-1}} : k[X]_n \rightarrow k^n : A(X) \mapsto [A(x_0), \dots, A(x_{n-1})]$$

ein Isomorphismus.

Den Vektor $[[x_0, A(x_0)], \dots, [x_{n-1}, A(x_{n-1})]]$ bezeichnet man als *modulare Darstellung* von $A(X)$ (im Englischen auch *point-value-representation*).

Natürlich erbt die Abbildung $\mathcal{L}_{x_0, \dots, x_{n-1}}$ von ihren Komponentenabbildungen \mathcal{L}_{x_j} ($0 \leq j < n$) alle guten Eigenschaften. Beachtung verdient die Behauptung, daß $\mathcal{L}_{x_0, \dots, x_{n-1}}$ *bijektiv* ist, d.h. daß der Kern von $\mathcal{L}_{x_0, \dots, x_{n-1}}$ nur aus dem Nullvektor 0 besteht. In Matrixschreibweise gilt nämlich:

$$\mathcal{L}_{x_0, \dots, x_{n-1}} : \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} \mapsto \underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{n-1} \end{pmatrix}}_{\mathbf{V}(x_0, x_1, \dots, x_{n-1})} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} A(x_0) \\ A(x_1) \\ \vdots \\ A(x_{n-1}) \end{pmatrix}$$

Die Matrix $\mathbf{V}(x_0, x_1, \dots, x_{n-1})$ ist eine sog. VANDERMONDE-Matrix, und man sollte wissen, daß

$$\det \mathbf{V}(x_0, x_1, \dots, x_{n-1}) = \prod_{0 \leq i < j < n} (x_j - x_i)$$

gilt. Für paarweise verschiedene x_0, x_1, \dots, x_{n-1} ist diese Determinante $\neq 0$, d.h. die VANDERMONDE-Matrix $\mathbf{V}(x_0, x_1, \dots, x_{n-1})$ hat Maximalrang n und die zugehörige lineare Transformation $\mathcal{L}_{x_0, \dots, x_{n-1}}$ ist injektiv (und auch surjektiv).

In dieser Situation kann man die Umkehrabbildung zu $\mathcal{L}_{x_0, \dots, x_{n-1}}$ betrachten. Dies ist die *Interpolationsabbildung*

$$\mathcal{I}_{x_0, x_1, \dots, x_{n-1}} : k^n \rightarrow k[X]_n$$

die zu jedem Vektor $[y_0, y_1, \dots, y_{n-1}] \in k^n$ das eindeutig bestimmte (!) Polynom $A(X) \in k[X]_n$ konstruiert, für das

$$A(x_i) = y_i \quad (0 \leq i < n)$$

gilt.

Die *algorithmische* Aufgabe der Interpolation kann man auf verschiedene Weisen lösen:

- Man berechnet den Koeffizientenvektor \mathbf{A} zu dem gesuchten Polynom $A(X)$ durch

$$\mathbf{A}^T = V(x_0, x_1, \dots, x_{n-1})^{-1} \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Diese Lösung eines linearen Gleichungssystems erfordert mit traditionellen Mittel (GAUSS-Elimination) $O(n^3)$ Operationen im Körper k .

- Man benutzt die Interpolationsformel von LAGRANGE

$$A(X) = \sum_{j=0}^{n-1} y_j \frac{\prod_{i \neq j} (X - x_i)}{\prod_{i \neq j} (x_j - x_i)}$$

Zur Verifikation beachte man, daß die Funktionen

$$\delta_j(X) := \frac{\prod_{i \neq j} (X - x_i)}{\prod_{i \neq j} (x_j - x_i)} \quad (0 \leq j < n)$$

Polynome $(n - 1)$ -ten Grades sind mit der Eigenschaft von "Indikatorfunktionen":

$$\mathcal{L}_{x_i}(\delta_j) = \delta_j(x_i) = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j \end{cases}$$

Benutzung dieser Interpolationsformel führt zu einem Algorithmus, der $O(n^2)$ Operationen in k benötigt.

5 Multiplikation von Polynomen

Es geht nun um die Abbildung

$$\cdot : k[X] \times k[X] \rightarrow k[X] : (A(X), B(X)) \mapsto (A \cdot B)(X)$$

Wegen

$$\deg(A \cdot B) = \deg(A) + \deg(B)$$

gilt insbesondere

$$A(X), B(X) \in k[X]_n \Rightarrow (A \cdot B)(X) \in k[X]_{2n}$$

Im folgenden werden wir die Multiplikation von Polynomen immer in dieser beschränkten Situation

$$k[X]_n \times k[X]_n \rightarrow k[X]_{2n}$$

betrachten.

Das Schema *Auswertung-Interpolation* für Polynome legt es nahe, die Aufgabe der Multiplikation von Polynomen auf folgendem Wege anzugehen:

1. Sind $A(X), B(X) \in k[X]_n$ zu multiplizieren, so werte man beide Polynome an $2n$ Stellen $x_0, x_1, \dots, x_{2n-1}$ aus. Man erhält

$$\begin{aligned} \mathcal{L}_{x_0, \dots, x_{2n-1}}(A) &= [A(x_0), \dots, A(x_{2n-1})] \in k^{2n} \\ \mathcal{L}_{x_0, \dots, x_{2n-1}}(B) &= [B(x_0), \dots, B(x_{2n-1})] \in k^{2n} \end{aligned}$$

2. Man berechnet die $2n$ Produkte $y_i = A(x_i) \cdot B(x_i)$, $0 \leq i < 2n$.
3. Durch Interpolation an den Stellen $x_0, x_1, \dots, x_{2n-1}$ bestimmt man das eindeutig bestimmte Polynom $C(X) \in k[X]_{2n}$ mit

$$C(x_i) = y_i = A(x_i) \cdot B(x_i) \quad , \quad 0 \leq i < 2n.$$

Da $A(X) \cdot B(X)$ ein Polynom vom Grad $< 2n$ ist, muß also $C(X) = A(X) \cdot B(X)$ sein.

Schematisch:

$$\begin{array}{ccc}
 A(X), B(X) & \xrightarrow{\mathcal{L}_{x_0, \dots, x_{2n-1}}} & [A(x_0), \dots, A(x_{2n-1})], [B(x_0), \dots, B(x_{2n-1})] \\
 \downarrow & & \downarrow C(x_i) := A(x_i) \cdot B(x_i) \\
 C(X) = A(X) \cdot B(X) & \xleftarrow{\mathcal{I}_{x_0, \dots, x_{2n-1}}} & [C(x_0), \dots, C(x_{2n-1})]
 \end{array}$$

Die *Korrektheit* dieses Vorgehens sollte außer Frage stehen. Interessant ist die Frage nach dem *Aufwand* !

1. Die beiden simultanen Auswertungen $\mathcal{L}_{x_0, \dots, x_{2n-1}}(A)$ bzw $\mathcal{L}_{x_0, \dots, x_{2n-1}}(B)$ erfordern, wenn man konventionell vorgeht (d.h die jeweils $2n$ Auswertungen unabhängig voneinander durchführt), $2 \times 2n \times \Theta(n) \in \Theta(n^2)$ Operationen in k .
2. Die $2n$ Multiplikationen $C(x_i) := A(x_i) \cdot B(x_i)$ für $0 \leq i < 2n$ erfordern $2n \in \Theta(n)$ Operationen in k .
3. Die Interpolation $\mathcal{I}_{x_0, \dots, x_{2n-1}}$ erfordert, wenn an sie etwa nach dem Verfahren der LAGRANGE-Interpolation durchführt, $\Theta((2n)^2) \in \Theta(n^2)$ Operationen in k .

Insgesamt erfordert dieses Vorgehen also $\Theta(n^2)$ Operationen in k — und das ist gegenüber der üblichen Polynom-Multiplikation überhaupt kein Vorteil (im Gegenteil!).

ABER: Dieses Vorgehen gilt für *beliebige* Wahlen der Auswertungs/Interpolationspunkte x_0, \dots, x_{2n-1} . Wenn es gelingt, die Verfahren für *spezielle*, geschickte Wahl der x_0, \dots, x_{2n-1} mit geringeren Kosten durchzuführen, also etwa mit einem Aufwand $\in \Theta(n \log n)$, dann wird auch die ganze Operation der Polynom-Multiplikation auf diesem Wege zu einem $\Theta(n \log n)$ -Verfahren.

6 Polynome und ihre Wurzeln

In diesem Abschnitt werden einige einfache Tatsachen über Polynome und ihrer Wurzeln (Nullstellen) zusammengestellt. Man beachte, daß es sich stets um Polynome mit Koeffizienten aus einem *Körper* k handelt. Die Aussagen sind i.a. nicht mehr richtig, falls der Koeffizientenbereich Nullteiler enthält.

- *Divisionseigenschaft*: Zu $f, g \in k[X]$ mit $g(X) \neq 0$ gibt es eindeutig bestimmte $q, r \in k[X]$ mit

$$f(X) = g(X) \cdot q(X) + r(X)$$

wobei r entweder das Nullpolynom (in diesem Fall sagt man: g teilt f , in Zeichen: $f|g$) ist oder $0 \leq \deg r < \deg g$ gilt. Man benennt $q(X)$ als den *Quotienten* und $r(X)$ als den *Rest* der Division. Man sagt auch: “ $f(X)$ liegt in der Restklasse von $r(X)$ modulo $g(X)$ ” und schreibt $f \equiv r \pmod{g}$.

- *Division und Auswertung*: Ist $f \in k[X]$ und $x_0 \in k$, so gilt speziell

$$f(X) = (X - x_0) \cdot q(X) + f(x_0)$$

d.h. die *Auswertung* von $f(x)$ an der Stelle x_0 ist der *Divisionrest* bei Division durch $X - x_0$. Also ist $f(X) \equiv f(x_0) \pmod{X - x_0}$. (Dabei bezeichnet $f(x_0)$ das konstante Polynom mit eben diesem Funktionswert.) Daher auch die Bezeichnung "modulare Darstellung" eines Polynoms, wenn man ein Polynom durch seine Funktionswerte an bestimmten Stellen repräsentiert.

- *Division und Nullstellen* : Die Eigenschaft eines $x_0 \in k$ Nullstelle von $f(X)$ zu sein, drückt sich so aus:

$$f(x_0) = 0 \Leftrightarrow (X - x_0) | f(X) \Leftrightarrow f(X) \equiv 0 \pmod{X - x_0}$$

- *Vielfachheiten von Nullstellen* : Ist $f \in k[X]$ und $x_0 \in k$, so heißt die eindeutig bestimmte natürliche Zahl n mit

$$(X - x_0)^n | f(X) \quad , \quad (X - x_0)^{n+1} \nmid f(X)$$

die *Vielfachheit* von x_0 als Nullstelle von $f(X)$, geschrieben $n = \nu(f, x_0)$.

- *Lemma* : Sind $f, g, h \in k[X]$ mit $f = g \cdot h$ und ist $x_0 \in k$, so gilt $\nu(f, x_0) = \nu(g, x_0) + \nu(h, x_0)$.

Beweis: Sei $s = \nu(g, x_0)$ und $t = \nu(h, x_0)$. Dann ist $s + t \leq \nu(f, x_0)$ klar! Also existiert ein $\tilde{f} \in k[X]$ mit

$$g(X) \cdot h(X) = (X - x_0)^{s+t} \cdot \tilde{f}(X)$$

Sind außerdem $\tilde{g}, \tilde{h} \in k[X]$ mit

$$g(X) = (X - x_0)^s \cdot \tilde{g}(X) \quad , \quad h(X) = (X - x_0)^t \cdot \tilde{h}(X)$$

so muß $g(x_0) \neq 0 \neq h(x_0)$ sein. Da k ein Körper ist, muß dann wegen $\tilde{f}(X) = \tilde{g}(X) \cdot \tilde{h}(X)$ aber auch $f(x_0) \neq 0$ sein. Also $s + t = \nu(f, x_0)$.

- *Satz* : Ist $f \in k[X]$, $f \neq 0$, so gilt $\sum_{x \in k} \nu(f, x) \leq \deg f$.
[Ein Polynom hat in einem Körper höchstens so viele Nullstellen (mit Berücksichtigung der Vielfachheiten), wie sein Grad angibt].

Beweis:

– Falls $f(X)$ konstant ($\neq 0$) ist, also $\deg f = 0$, ist die Behauptung offensichtlich korrekt.

– Induktion: sei nun $\deg f > 0$.

* Falls f in k überhaupt keine Nullstelle hat, ist auch nichts zu zeigen.

* Sei andernfalls x_0 eine Nullstelle von f in k , also $\nu(f, x_0) = n > 0$. Dann ist also $f(X) = (X - x_0)^n \cdot g(X)$, wobei $g(X) \in k[X]$ mit $g(x_0) \neq 0$ und $\deg g = \deg f - n < \deg f$. Aus dem Lemma folgt, daß für alle $y \in k \setminus \{x_0\}$ die Gleichheit $\nu(f, y) = \nu(g, y)$ gilt. Damit und mit der Induktionsannahme für $g(X)$ folgt

$$\begin{aligned} \sum_{x \in k} \nu(f, x) &= \nu(f, x_0) + \sum_{y \in k \setminus \{x_0\}} \nu(f, y) \\ &= \nu(f, x_0) + \sum_{y \in k \setminus \{x_0\}} \nu(g, y) \\ &\leq \nu(f, x_0) + \deg g(X) = \deg f(X) \end{aligned}$$

- *Fundamentalsatz der Algebra* : Jedes komplexe Polynom $f(X) \in \mathbf{C}[X]$ hat mindestens eine Nullstelle in \mathbf{C} .
[Aus diesem Grund nennt man den Körper \mathbf{C} der komplexen Zahlen auch “algebraisch abgeschlossen”. Die Körper \mathbf{Q} und \mathbf{R} sind offensichtlich *nicht* algebraisch abgeschlossen.]
- *Folgerung* : Ist $f \in \mathbf{C}[X]$, so gilt $\sum_{x \in \mathbf{C}} \nu(f, x) = \deg f$.

7 Einheitswurzeln

Ist k ein Körper (oder allgemeiner ein Ring) und $n \geq 1$ eine ganze Zahl, so bezeichnet man jedes Element $\omega \in k$ mit der Eigenschaft $\omega^n = 1$, also jede Lösung (“Wurzel”) der Gleichung $X^n - 1 = 0$, als eine n -te Einheitswurzel “root of unity”).

Aus der Definition folgt fast unmittelbar, daß die Beziehungen der n -ten Einheitswurzeln für verschiedene n durch die *Teilbarkeitsstruktur* der Zahlen n geregelt werden. Es gilt für $m, n \geq 1$:

$$\omega^m = 1 \wedge \omega^n = 1 \Leftrightarrow \omega^{\gcd(m,n)} = 1$$

Dies folgt einerseits aus der offensichtlichen Tatsache, daß

$$\omega^d = 1 \wedge d | n \Rightarrow \omega^n = 1$$

Ist andererseits $d = \gcd(m, n)$, so gibt es $\alpha, \beta \in \mathbf{Z}$ mit $\alpha \cdot m + \beta \cdot n = d$. Folglich ist

$$\omega^d = \omega^{\alpha \cdot m + \beta \cdot n} = (\omega^m)^\alpha \cdot (\omega^n)^\beta = 1^\alpha \cdot 1^\beta = 1$$

Aus den Darlegungen des vorigen Abschnitts folgt, daß jeder Körper höchstens n n -te Einheitswurzeln enthält. Der Körper \mathbf{C} der komplexen Zahlen enthält in der Tat genau n n -te Einheitswurzeln; dies sind die Elemente von

$$\mathcal{R}_n := \{ e^{2\pi it/n}; 0 \leq t < n \} = \{ \omega_n^t; 0 \leq t < n \} \quad \text{wobei } \omega_n := e^{2\pi i/n}$$

(Diese n Zahlen sind n -te Einheitswurzeln und sie sind paarweise verschieden; weitere kann es also nicht geben). Die Körper \mathbf{Q} und \mathbf{R} sind dagegen sehr arm an Einheitswurzeln: sie enthalten, wie jeder Körper, die beiden zweiten Einheitswurzeln $+1$ und -1 , die ihrerseits natürlich auch n -te Einheitswurzeln für jedes n (bzw. jedes gerade n) sind, aber das ist auch schon alles.

Aus den generellen (vom jeweiligen Körper unabhängigen) Eigenschaften von Einheitswurzeln folgt

$$\begin{aligned} n | m &\Rightarrow \mathcal{R}_n \subseteq \mathcal{R}_m \\ \mathcal{R}_m \cap \mathcal{R}_n &= \mathcal{R}_{\gcd(m,n)} \end{aligned}$$

Hier eine Aufstellung der komplexen n -ten Einheitswurzeln für $1 \leq n \leq 12$. Wegen der gerade erwähnten Inklusionseigenschaft der Mengen \mathcal{R}_n genügt es, für jedes

n diejenigen n -ten Einheitswurzeln anzugeben, die nicht schon d -te Einheitswurzeln für einen echten Teiler d von n sind. Bezeichnen wir diese Menge mit \mathcal{P}_n , so gilt also

$$\mathcal{P}_n = \mathcal{R}_n \setminus \bigcup_{\substack{d|n \\ d \neq n}} \mathcal{R}_d$$

Umgekehrt ergibt sich daraus

$$\mathcal{R}_n = \bigcup_{d|n} \mathcal{P}_d$$

Die Mengen \mathcal{P}_n , ($n \geq 0$), sind paarweise disjunkt. Es folgt eine Liste der ersten 12 Mengen:

$$\mathcal{P}_1 = \{\omega_1\} = \{1\}$$

$$\mathcal{P}_2 = \{\omega_2\} = \{-1\}$$

$$\mathcal{P}_3 = \{\omega_3, \omega_3^2\} = \left\{ -\frac{1 \pm i\sqrt{3}}{2} \right\}$$

$$\mathcal{P}_4 = \{\omega_4, \omega_4^3\} = \{\pm i\}$$

$$\mathcal{P}_5 = \{\omega_5, \omega_5^2, \omega_5^3, \omega_5^4\} = \left\{ \frac{\sqrt{5} - 1 \pm i\sqrt{2}\sqrt{5 + \sqrt{5}}}{4}, \frac{-\sqrt{5} - 1 \pm i\sqrt{2}\sqrt{5 - \sqrt{5}}}{4} \right\}$$

$$\mathcal{P}_6 = \{\omega_6, \omega_6^5\} = \left\{ \frac{1 \pm i\sqrt{3}}{2} \right\}$$

$$\mathcal{P}_7 = \{\omega_7, \omega_7^2, \omega_7^3, \omega_7^4, \omega_7^5, \omega_7^6\} = \left\{ \cos\left(\frac{k\pi}{7}\right) \pm i \sin\left(\frac{k\pi}{7}\right) ; 1 \leq k \leq 3 \right\}$$

$$\mathcal{P}_8 = \{\omega_8, \omega_8^3, \omega_8^5, \omega_8^7\} = \left\{ \pm \frac{1 \pm i}{\sqrt{2}} \right\}$$

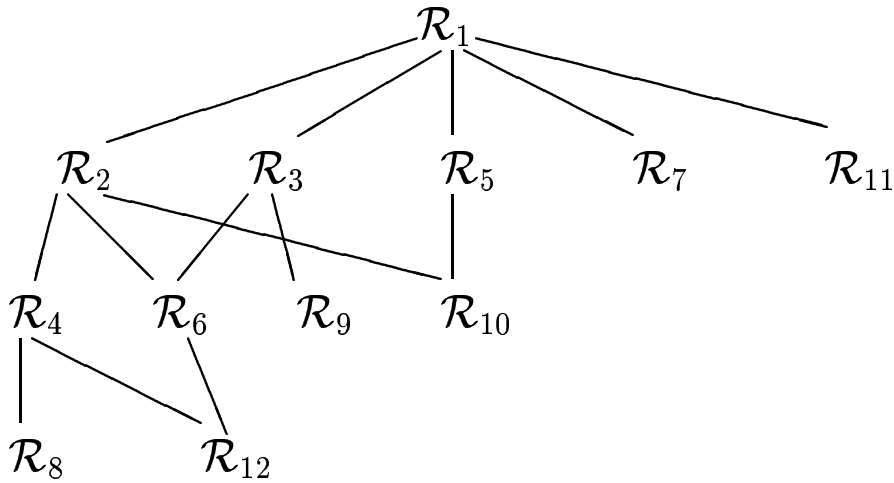
$$\mathcal{P}_9 = \{\omega_9, \omega_9^2, \omega_9^4, \omega_9^5, \omega_9^7, \omega_9^8\} = \left\{ \cos\left(\frac{k\pi}{9}\right) \pm i \sin\left(\frac{k\pi}{9}\right) ; k \in \{1, 2, 4\} \right\}$$

$$\mathcal{P}_{10} = \{\omega_{10}, \omega_{10}^3, \omega_{10}^7, \omega_{10}^9\} = \left\{ \cos\left(\frac{k\pi}{10}\right) \pm i \sin\left(\frac{k\pi}{10}\right) ; k \in \{1, 3\} \right\}$$

$$\begin{aligned} \mathcal{P}_{11} &= \{\omega_{11}, \omega_{11}^2, \omega_{11}^3, \omega_{11}^4, \omega_{11}^5, \omega_{11}^6, \omega_{11}^7, \omega_{11}^8, \omega_{11}^9, \omega_{11}^{10}\} \\ &= \left\{ \cos\left(\frac{k\pi}{11}\right) \pm i \sin\left(\frac{k\pi}{11}\right) ; 1 \leq k \leq 5 \right\} \end{aligned}$$

$$\mathcal{P}_{12} = \{\omega_{12}, \omega_{12}^5, \omega_{12}^7, \omega_{12}^{11}\} = \left\{ \pm \frac{\sqrt{3} \pm i}{2} \right\}$$

Aufgrund der Teilbarkeitsbeziehungen hat man



wobei die Verbindungslinien die Inklusionsbeziehungen darstellen.

Die Elemente $\xi \in \mathcal{P}_n$ bezeichnet man als die *primitiven* (komplexen) n -ten Einheitswurzeln. Das sind also die Elemente “mit Periode n ”, die keine echt kleinere Periode haben. Folgende Aussagen, die sich an den Beispielen leicht überprüfen lassen, sind äquivalent für eine komplexe Zahl ξ :

- $\xi \in \mathcal{P}_n$
- $\xi = \omega_n^t$ für ein $t \in \{1, 2, \dots, n-1\}$ mit $\gcd(t, n) = 1$
- $\mathcal{P}_n = \{ \xi^t ; 1 \leq t < n, \gcd(n, t) = 1 \}$

Die spezielle n -te Einheitswurzel $\omega_n := e^{2\pi i/n}$ bezeichnet man als n -te *Haupt-Einheitswurzel* (“principal root of unity”). Die Anzahl der primitiven n -ten Einheitswurzeln ist demnach (für $n > 1$)

$$\phi(n) := \#\{ t ; 1 \leq t < n, \gcd(t, n) = 1 \} = \#\mathcal{P}_n$$

Dies ist die berühmte EULERSche Phi-Funktion, deren erste Werte sich aus obigen Beispielen ablesen lassen:

n	:	1	2	3	4	5	6	7	8	9	10	11	12	...
$\phi(n)$:	1	1	2	2	4	2	6	4	6	4	10	4	...

Die folgenden Eigenschaften von ergeben sich unmittelbar aus der Definition:

- Ist p Primzahl, so gilt für $n \geq 1$: $\phi(p^n) = p^n - p^{n-1} = p^n \left(1 - \frac{1}{p}\right)$
- Sind m, n teilerfremd, also $\gcd(m, n) = 1$, so gilt $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$

Das erlaubt eine kompakte Darstellung der ϕ -Funktion, die — algorithmisch betrachtet — allerdings voraussetzt, daß man die Primfaktorisierung des Arguments n kennt:

$$\begin{aligned}
 n = p_1^{\alpha_1} p_2^{\alpha_2} \dots &\Rightarrow \phi(n) = p_1^{\alpha_1} \left(1 - \frac{1}{p_1}\right) p_2^{\alpha_2} \left(1 - \frac{1}{p_2}\right) \dots \\
 &= n \cdot \prod_{\substack{p|n \\ p \text{ Primzahl}}} \left(1 - \frac{1}{p}\right)
 \end{aligned}$$

Hierbei bezeichnen p_1, p_2, \dots die *verschiedenen* Primteiler von n , und das Produkt erstreckt sich über eben diese Menge.

Schließlich drückt sich die Beziehung $\mathcal{R}_n = \bigcup_{d|n} \mathcal{P}_d$ numerisch so aus

$$\forall n : n = \sum_{d|n} \phi(d)$$

was man an den Beispielen schnell überprüft.

Es ist leicht einzusehen, daß die n -ten Einheitswurzeln eines Körpers eine Gruppe bilden und zwar eine zyklische Gruppe (mit der Multiplikation als Operation). Am speziellen Fall der komplexen Zahlen verdeutlicht: \mathcal{R}_n ist eine zyklische Gruppe der Ordnung n , und die "Erzeuger" dieser Gruppe sind genau die primitiven n -ten Einheitswurzeln, als die $\xi \in \mathcal{P}_n$. In dieser Situation ist

$$(\mathbf{Z}/n\mathbf{Z}) \rightarrow \mathcal{R}_n : t \mapsto \xi^t$$

ein Isomorphismus zwischen den beiden zyklischen Gruppen $(\mathbf{Z}/n\mathbf{Z})$ und \mathcal{R}_n .

- Einige Rechenregeln für Einheitswurzeln

Die nachfolgenden Regeln werden für die Haupt-Einheitswurzeln $\omega_n = e^{2\pi i/n}$ in \mathbf{C} formuliert, gelten aber entsprechend auch für beliebige primitive Einheitswurzeln.

- Für $n, d > 0, t \in \mathbf{Z} : \omega_{dn}^t = \omega_n^t$
- Für $n > 0 : \omega_{2n}^n = -1$
- Durch Quadrieren $x \mapsto x^2$ wird \mathcal{R}_{2n} auf \mathcal{R}_n abgebildet. Dabei hat jedes ω_n^t , ($0 \leq t < n$), die beiden Urbilder ω_{2n}^t und ω_{2n}^{t+n} .
- Für $n \geq 1$ und $t \in \mathbf{Z}$ gilt

$$\sum_{x \in \mathcal{R}_n} x^t = \sum_{j=0}^{n-1} (\omega_n^t)^j = \begin{cases} n & \text{falls } n | t \\ 0 & \text{falls } n \nmid t \end{cases}$$

Der *Beweis* ist einfach: Im Fall $n | t$ ist $\omega_n^t = 1$. Im Fall $n \nmid t$ ist $\omega_n^t \neq 1$, also

$$\sum_{j=0}^{n-1} (\omega_n^t)^j = \frac{(\omega_n^t)^n - 1}{\omega_n^t - 1} = \frac{(\omega_n^n)^t - 1}{\omega_n^t - 1} = 0$$

8 Diskrete Fourier-Transformation (DFT)

Ist $\mathbf{a} = (a_0, a_1, \dots, a_{n-1}) \in \mathbf{C}^n$ ein Vektor von komplexen Zahlen, so bezeichnet man den Vektor

$$\mathbf{y} = (y_0, y_1, \dots, y_{n-1}) \text{ mit } y_j = \sum_{t=0}^{n-1} a_t \omega_n^{jt} \quad (0 \leq j < n)$$

als *diskrete Fourier-Transformierte* von \mathbf{a} . Identifiziert man, im Sinne der Koeffizientendarstellung von Polynomen, den Vektor \mathbf{a} mit dem Polynom $A(X) = \sum_{j=0}^{n-1} a_j X^j \in \mathbf{C}[X]_n$, so erkennt man, daß die *diskrete Fourier-Transformation*

$$DFT_n : \mathbf{C}^n \rightarrow \mathbf{C}^n : \mathbf{y} \mapsto (A(\omega_n^0), A(\omega_n^1), \dots, A(\omega_n^{n-1}))$$

in Grunde nichts anderes ist als die *simultane Auswertung* $\mathcal{L}_{\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}}$ von Polynomen an den n -ten Einheitswurzeln $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$.

DFT_n ist also eine lineare Transformation, gegeben durch die VANDERMONDE-Matrix

$$\mathbf{V}_n := V(\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}) = (\omega_n^{jk})_{0 \leq j, k < n}$$

DFT_n ist invertierbar, da die $\omega_n^j \in \mathcal{R}_n$ paarweise verschieden sind. Die Matrix der inversen Transformation, der "Interpolation" also, ist

$$\mathbf{V}_n^{-1} = \left(\frac{1}{n} \omega_n^{-jk} \right)_{0 \leq j, k < n}$$

Der *Beweis* hierfür ist einfach. Er ergibt sich aus der letzten Rechenregel des vorigen Abschnitts:

$$(\omega_n^{jk})_{0 \leq j, k < n} \left(\frac{1}{n} \omega_n^{-kl} \right)_{0 \leq k, l < n} = \left(\frac{1}{n} \sum_{k=0}^{n-1} \omega_n^{(j-l)k} \right)_{0 \leq j, l < n} = \frac{1}{n} (\delta_{j,l})_{0 \leq j, l < n}$$

Es ist bemerkenswert, daß die inverse Transformation DFT_n^{-1} , sieht man einmal von der Skalarmultiplikation mit $1/n$ ab, wieder eine Transformation vom Typ der DFT_n ist: mit ω_n^{-1} an Stelle von ω_n . (Natürlich ist auch ω_n^{-1} eine *primitive* n -te Einheitswurzel).

9 DFT-Matrizen

In diesem Abschnitt werden für (sehr) kleine Werte von n die Matrizen der Diskreten Fourier-Transformation DFT_n explizit angegeben. Man beachte: alle Einträge ω_n^{jk} , $0 \leq j, k < n$, der Matrix \mathbf{V}_n sind n -te Einheitswurzeln — wegen $\omega_n^n = 1$ gilt ja

$$\omega_n^{jk} = \omega_n^t$$

wobei t mit $0 \leq t < n$ der Rest der Division von $j \cdot k$ durch n ist, $jk \equiv t \pmod{n}$.

- $n = 2$

Es ist $\omega_2 = -1$ und daher

$$\mathbf{V}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Die inverse Matrix ist

$$\mathbf{V}_2^{-1} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{bmatrix}$$

- $n = 3$

Es ist $\omega_3 = \frac{-1+i\sqrt{3}}{2}$ und daher

$$\begin{aligned} \mathbf{V}_3 &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1/2 + \frac{\sqrt{-1}\sqrt{3}}{2} & -1/2 - \frac{\sqrt{-1}\sqrt{3}}{2} \\ 1 & -1/2 - \frac{\sqrt{-1}\sqrt{3}}{2} & -1/2 + \frac{\sqrt{-1}\sqrt{3}}{2} \end{bmatrix} \end{aligned}$$

Die inverse Transformation ist gegeben durch

$$\mathbf{V}_3^{-1} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & -\frac{\sqrt{-1}\sqrt{3}}{6} - 1/6 & \frac{\sqrt{-1}\sqrt{3}}{6} - 1/6 \\ 1/3 & \frac{\sqrt{-1}\sqrt{3}}{6} - 1/6 & -\frac{\sqrt{-1}\sqrt{3}}{6} - 1/6 \end{bmatrix}$$

- $n = 4$

Es ist $\omega_4 = i = \sqrt{-1}$ und daher

$$\begin{aligned} \mathbf{V}_4 &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & 1 & i^2 \\ 1 & i^3 & i^2 & i \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \sqrt{-1} & -1 & -\sqrt{-1} \\ 1 & -1 & 1 & -1 \\ 1 & -\sqrt{-1} & -1 & \sqrt{-1} \end{bmatrix} \end{aligned}$$

Die inverse Transformation ist gegeben durch

$$\mathbf{V}_4^{-1} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & -\frac{\sqrt{-1}}{4} & -1/4 & \frac{\sqrt{-1}}{4} \\ 1/4 & -1/4 & 1/4 & -1/4 \\ 1/4 & \frac{\sqrt{-1}}{4} & -1/4 & -\frac{\sqrt{-1}}{4} \end{bmatrix}$$

- $n = 5$

Die Einträge der Matrix sind die Elemente von \mathcal{R}_5 , wie oben aufgeführt.

$$\mathbf{V}_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_5 & \omega_5^2 & \omega_5^3 & \omega_5^4 \\ 1 & \omega_5^2 & \omega_5^4 & \omega_5^1 & \omega_5^3 \\ 1 & \omega_5^3 & \omega_5^1 & \omega_5^4 & \omega_5^2 \\ 1 & \omega_5^4 & \omega_5^3 & \omega_5^2 & \omega_5^1 \end{bmatrix}$$

Für die inverse Transformation beachte man, daß die Elemente ω_5 und ω_5^4 , wie auch die Elemente ω_5^2 und ω_5^3 , invers zueinander sind. Also

$$\mathbf{V}_5^{-1} = \frac{1}{5} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_5^4 & \omega_5^3 & \omega_5^2 & \omega_5^1 \\ 1 & \omega_5^3 & \omega_5^1 & \omega_5^4 & \omega_5^2 \\ 1 & \omega_5^2 & \omega_5^4 & \omega_5^1 & \omega_5^3 \\ 1 & \omega_5^1 & \omega_5^2 & \omega_5^3 & \omega_5^4 \end{bmatrix}$$

- $n = 6$

Es ist $\omega_6 = \frac{1+i\sqrt{3}}{2}$. Bei der Darstellung von \mathbf{V}_6 kann man sich folgende Tatsachen zunutze machen

$$\omega_6 = \omega_2 \cdot \omega_3^2 = -\omega_3^2, \quad \omega_6^2 = \omega_3, \quad \omega_6^3 = -1, \quad \omega_6^4 = \omega_3^2, \quad \omega_6^5 = -\omega_3$$

Daher kann man \mathbf{V}_6 mit Hilfe von ω_3 ausdrücken

$$\begin{aligned} \mathbf{V}_6 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega_6 & \omega_6^2 & \omega_6^3 & \omega_6^4 & \omega_6^5 \\ 1 & \omega_6^2 & \omega_6^4 & 1 & \omega_6^2 & \omega_6^4 \\ 1 & \omega_6^3 & 1 & \omega_6^3 & 1 & \omega_6^3 \\ 1 & \omega_6^4 & \omega_6^2 & 1 & \omega_6^4 & \omega_6^2 \\ 1 & \omega_6^5 & \omega_6^4 & \omega_6^3 & \omega_6^2 & \omega_6^1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -\omega_3^2 & \omega_3 & -1 & \omega_3^2 & -\omega_3 \\ 1 & \omega_3 & \omega_3^2 & 1 & \omega_3 & \omega_3^2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \omega_3^2 & \omega_3 & 1 & \omega_3^2 & \omega_3 \\ 1 & -\omega_3 & \omega_3^2 & -1 & \omega_3 & -\omega_3^2 \end{bmatrix} \end{aligned}$$

Die Matrix der inversen Transformation läßt sich entsprechend mit Hilfe der dritten Einheitswurzeln ausdrücken.

Bereits diese kleinen Beispiele lassen erkennen, daß die Matrizen der diskreten Fourier-Transformation sehr spezielle Matrizen sind. Man ahnt zumindest, daß in der Transformation DFT_n die Transformationen DFT_d für Teiler d von n "enthalten" sind. Genau diese Beobachtung ist der Ansatzpunkt für eine "schnelle" Transformationstechnik.

10 Schnelle Fourier-Transformation (FFT)

Der "Trick" bei der "schnellen" Berechnung von DFT_n besteht darin, die spezielle Struktur der Menge $\mathcal{R}_n = \{\omega_n^t; 0 \leq t < n\}$ der Auswertungs/Interpolationspunkte auszunutzen. In der Situation eines geraden n , also $n = 2m$, kann man jedes Polynom $A(X) = \sum_{0 \leq j < 2m} a_j X^j \in k[X]_{2m}$ zerlegen:

$$A(X) = A^{[0]}(X^2) + X \cdot A^{[1]}(X^2)$$

mit

$$\begin{aligned} A^{[0]}(X) &= a_0 + a_2 X + a_4 X^2 + \cdots + A_{2m-2} X^{m-1} \\ A^{[1]}(X) &= a_1 + a_3 X + a_5 X^2 + \cdots + A_{2m-1} X^{m-1} \end{aligned}$$

Die Berechnung der $n = 2m$ Auswertungen (für $0 \leq t < n = 2m$)

$$\begin{aligned} A(\omega_n^t) &= A^{[0]}(\omega_n^{2t}) + \omega_n^t \cdot A^{[1]}(\omega_n^{2t}) \\ &= A^{[0]}(\omega_m^t) + \omega_n^t \cdot A^{[1]}(\omega_m^t) \end{aligned}$$

kann nun dadurch geschehen, daß man die beiden Polynome $A^{[0]}(X)$ und $A^{[1]}(X)$ mit Gradschranke m an den m Stellen

$$\mathcal{R}_m = \{ \omega_{2n}^{2t} (= \omega_m^t); 0 \leq t < m \}$$

auswertet. Dies erfordert also die Berechnung von

$$\begin{aligned} DFT_m(\mathbf{A}^{[0]}) &= (y_0^{[0]}, y_1^{[0]}, \dots, y_{m-1}^{[0]}) = \mathbf{y}^{[0]} \quad \text{mit } y_t^{[0]} = A^{[0]}(\omega_m^t) \\ DFT_m(\mathbf{A}^{[1]}) &= (y_0^{[1]}, y_1^{[1]}, \dots, y_{m-1}^{[1]}) = \mathbf{y}^{[1]} \quad \text{mit } y_t^{[1]} = A^{[1]}(\omega_m^t) \end{aligned}$$

Man erhält dann

$$DFT_n(\mathbf{A}) = (y_0, y_1, \dots, y_{n-1}) = \mathbf{y}$$

durch

$$y_t = A(\omega_n^t) = A^{[0]}(\omega_m^t) + \omega_n^t \cdot A^{[1]}(\omega_m^t) \quad (0 \leq t < n)$$

mit jeweils nur m (!) verschiedenen Auswertungen von $A^{[0]}(X)$ bzw. $A^{[1]}(X)$, denn es ist ja

$$\omega_m^{m+t} = \omega_m^m \cdot \omega_m^t = \omega_m^t$$

für $0 \leq t < m$. Dabei ist aber auch

$$\omega_n^{m+t} = \omega_{2m}^m \cdot \omega_n^t = (-1) \cdot \omega_n^t$$

sodaß man für $0 \leq t < m$ schreiben kann

$$\begin{aligned} y_t &= y_t^{[0]} + \omega_n^t \cdot y_t^{[1]} \\ y_{m+t} &= y_t^{[0]} - \omega_n^t \cdot y_t^{[1]} \end{aligned}$$

Noch suggestiver geschrieben:

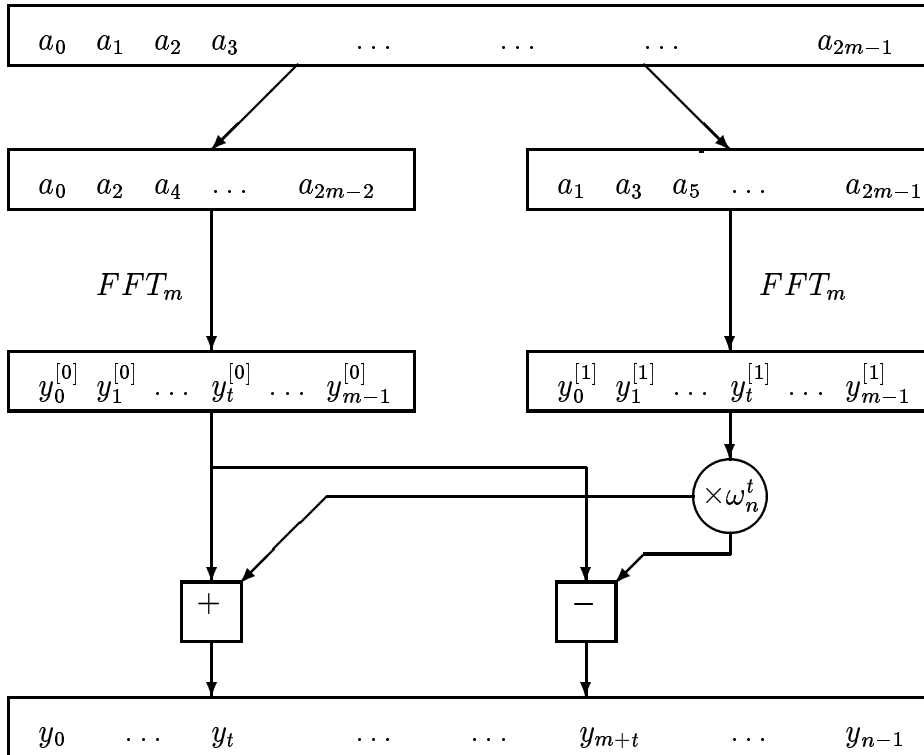
$$\begin{aligned} \begin{pmatrix} y_t \\ y_{m+t} \end{pmatrix} &= \begin{pmatrix} 1 & \omega_m^t \\ 1 & -\omega_m^t \end{pmatrix} \begin{pmatrix} y_t^{[0]} \\ y_t^{[1]} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \omega_m^t \end{pmatrix} \begin{pmatrix} y_t^{[0]} \\ y_t^{[1]} \end{pmatrix} \\ &= \mathbf{V}_2 \cdot \begin{pmatrix} 1 & 0 \\ 0 & \omega_m^t \end{pmatrix} \begin{pmatrix} y_t^{[0]} \\ y_t^{[1]} \end{pmatrix} \end{aligned}$$

für $0 \leq t < m$.

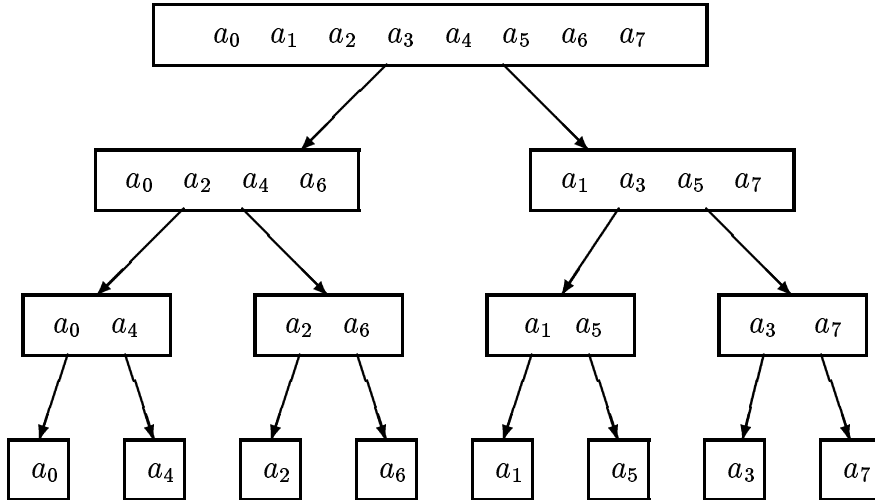
Diese simultane Berechnung von (y_t, y_{m+t}) aus $(y_t^{[0]}, y_t^{[1]})$ für jedes t mit $0 \leq t < m$ bezeichnet man als *butterfly*-Operation, was von entsprechenden graphischen Darstellungen suggeriert wird.

Das Wesen und die Effizienz der schnellen Fourier-Transformation liegen in der *rekursiven* Anwendung dieser Idee: Zurückführung eines DFT-Problems der Größe $n = 2m$ auf zwei DFT-Probleme der Größe m , und dazu noch Datenumordnungen und m *butterfly*-Operationen. Ist n eine Potenz von 2, also $n = 2^k$, so kann man folgendes Schema rekursiv anwenden:

Schema von FFT_{2m}



Schema der input-Vektoren für die rekursiven Aufrufe bei Berechnung von FFT_8 :



Man beachte, daß die Blätter dieses Baumes mit den Elementen des ursprünglichen input-Vektors

$$\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$$

beschriftet sind, allerdings in permutierter Reihenfolge

$$\mathbf{a}^{rev} = (a_0, a_4, a_2, a_6, a_1, a_5, a_3, a_7)$$

Der Übergang von \mathbf{a} zu \mathbf{a}^{rev} läßt sich am elegantesten so beschreiben: man denke sich die Indices der Komponenten von \mathbf{a} in Basis-2-Darstellung geschrieben; die an entsprechender Stelle von \mathbf{a}^{rev} stehende Komponente erhält man dadurch, daß man die Basis-2-Darstellung *rückwärts* liest: an der Stelle von a_3 in \mathbf{a} steht also a_6 in \mathbf{a}^{rev} , denn $3 = (011)_2$ und $6 = (110)_2$. Diese Regel gilt allgemein und diese Transformation wird als *bit reversal* bezeichnet. Sie spielt insbesondere bei iterativer und paralleler Implementierung der FFT eine wichtige Rolle.

Das nachfolgende Programm `FFT` (in Maple-pseudocode) realisiert die Idee der schnellen Fourier-Transformation:

```

FFT := proc (A,k)
n := 2^k;
if k=0 then RETURN(A) fi;
omega_n := exp(2*Pi*I/n);
omega := 1;
a_0 := [A[0],A[2],...,A[n-2]];
a_1 := [A[1],A[3],...,A[n-1]];
y_0 := FFT(a_0,k-1);
y_1 := FFT(a_1,k-1);
for t from 0 to (n/2)-1 do
y[t] := y_0[t] + omega*y_1[t];
y[t+(n/2)] := y_0[t] - omega*y_1[t];
end do;
end proc;
  
```

```

    omega := omega*omega_n
  od;
RETURN(y);
end;

```

Anhand dieses Programms, bei dem der Parameter k die jeweilige *Ordnung* angibt (entsprechend Listenlänge $n = 2^k$), kann man nun leicht die Komplexität, gemessen in Operationen im Körper der komplexen Zahlen, abschätzen. Man hat nur zu beachten, daß die `for t ... do` Schleife einen Aufwand erfordert, der linear in der Länge n wächst. Bezeichnet als $T(n)$ den Aufwand zur Berechnung von $DFT(A)$ für Listen der Länge $n = 2^k$ mittels **FFT**, so ergibt sich die *divide-and-conquer*-Gleichung

$$T(2n) = 2 \cdot T(n) + \mathcal{O}(n)$$

und damit

$$T(n) \in \Theta(n \log n)$$

Was nun die inverse Transformation, die “Interpolationstransformation” DFT_n^{-1} angeht, so wurde bereits festgestellt, daß diese — bis auf eine Skalarmultiplikation mit einem Faktor $1/n$ — durch die **VANDERMONDE**-Matrix $V(1, \omega_n^{-1}, \omega_n^{-2}, \dots, \omega_n)$ dargestellt wird. Das rekursive Schema für **FFT** läßt sich also ganz genauso auf die Rücktransformation anwenden. Ein Programm **iFFT** unterscheidet sich von **FFT** nur dadurch, daß $\exp(2\pi i/n)$ durch $\exp(-2\pi i/n)$ ersetzt wird und noch die Skalarmultiplikation berücksichtigt wird. Daher gilt die $\Theta(n \log n)$ -Aussage für die Komplexität ganz genauso für die inverse Transformation.

Zusammenfassend ist festzuhalten

- Der Algorithmus der schnellen Fourier-Transformation (**FFT**) erlaubt es, die diskrete Fourier-Transformation (**DFT**) und ihre Rücktransformation für Vektoren der Länge n mit einem Aufwand von $\Theta(n \log n)$ Operationen im Grundkörper (der komplexen Zahlen) zu berechnen.
- In der Sprache der Polynomarithmetik gesprochen: simultane Auswertung und Interpolation komplexer Polynome mit Gradschranke n sind in $\Theta(n \log n)$ Operationen möglich, wenn als Auswertungs/Interpolationspunkte die komplexen n -ten Einheitswurzeln gewählt werden. Daher ist auch die Berechnung des Produktes (Faltung) zweier Polynome mit Gradschranke n mit $\Theta(n \log n)$ Operationen möglich.

11 Weiterführende Bemerkungen

11.1 Zum Implementierung

Diese Bemerkungen können nur als Andeutungen verstanden werden. Für Einzelheiten muss auf die reichhaltige Literatur verwiesen werden.

11.1.1 Iterative Implementierung

Aus Gründen der Effizienz wird man für Anwendungen meist an einer iterativen Implementierung des FFT-Algorithmus interessiert sein. Man beachte, daß FFT aus zwei Dingen besteht:

- der Organisation der Daten (entsprechen dem *bit-reversal*)
- der *butterfly*-Operation

Eine iterative Implementierung hat also eine Kaskade von *butterfly*-Operationen auf den korrekt organisierten Daten abzarbeiten. Siehe z.B. das Buch von CORMEN/LEISERSON/RIVEST für einen Einstieg in solche Realisierungen.

11.1.2 Parallele Implementierung

Die dem FFT-Algorithmus innewohnende Parallelität ist offensichtlich. Der Datenfluss und die *butterfly*-Operationen eines iterativen Berechnungsschemas können unmittelbar in eine parallele Implementierung transformiert werden. Mit $(n/2) \log n$ "*butterfly*-Schaltkreisen" kann FFT der Länge n in $\log n$ "Takten" berechnet werden, wenn man einen "Takt" pro *butterfly*-Operation ansetzt. Siehe auch hierzu CORMEN/LEISERSON/RIVEST.

11.1.3 Rundungsfehler

Ein nicht zu übersehender Aspekt der Diskreten Fourier-Transformation besteht darin, daß man mit Einheitswurzeln rechnet und daß diese nun einmal komplexe Zahlen sind. Bis auf wenige kleine Fälle wird man diese Zahlen, auch unter Zuhilfenahme von Quadratwurzeln etwa, nicht exakt behandeln können oder wollen. Für viele Anwendungen ist komplexe floating-point-Arithmetik durchaus kein Problem, nur hat man dann das Problem der Rundungsfehler und ihrer Fortpflanzung zu studieren (siehe z.B. das Buch von KRONJÖ). Für manche Anwendungen (etwa in der Polynom- und Integer-Arithmetik, Codierungstheorie etc.) sind Rundungsfehler aber intolerabel. Glücklicherweise ist das Prinzip der Diskreten Fourier-Transformation nicht an den Körper der komplexen Zahlen gebunden, sondern funktioniert auch in *endlichen Körpern*, siehe unten.

11.2 Zur Komplexität

11.2.1 Lineare Komplexität

Es drängt sich naturgemäß die Frage auf, ob mit der Schnellen Fourier-Transformation die "Schallgrenze" für den Rechenaufwand der Diskreten Fourier-Transformation

erreicht ist. Angesichts der notorischen Probleme, überhaupt *nichtlineare* untere Schranken für Komplexitätsaussagen zu gewinnen, ist es schon erwähnenswert, daß man in einem bestimmten Kostenmaß tatsächlich eine $\mathcal{O}(n \log n)$ -Aussage als *untere* Schranke für die DFT herleiten kann. Es geht dabei um die sogenannte *lineare Komplexität* vom Linearformen und Matrizen. An dieser Stelle soll nur eine kurze Skizze gegeben werden.

- Problemstellung :

Gegeben sei eine $(p \times n)$ -Matrix $\mathbf{A} = (a_{ij})$ mit Einträgen aus einem Körper k . Für Vektoren $\mathbf{x} = (x_1, \dots, x_n) \in k^n$ soll das Produkt $\mathbf{A} \cdot \mathbf{x}^T$ berechnet werden. Über die x_i sei *a priori* nichts bekannt — sie werden wie *Unbestimmte* d.h. wie *Variable* behandelt.

- Begriffsbildung :

Eine $((n+r) \times n)$ -Matrix

$$\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n, \mathbf{g}_{n+1}, \mathbf{g}_{n+2}, \dots, \mathbf{g}_{n+r})^T$$

heißt *Berechnungsmatrix* für \mathbf{A} , wenn für die Zeilenvektoren \mathbf{g}_i dieser Matrix gilt

- $\mathbf{g}_i = \mathbf{e}_i$ ist i -ter Einheitsvektor für alle $1 \leq i \leq n$
- für alle $n < i \leq n+r$ ist \mathbf{g}_i von einer der beiden Formen

$$\begin{aligned} \mathbf{g}_i &= \mathbf{g}_j \pm \mathbf{g}_m \quad \text{mit } j, m < i \\ \mathbf{g}_i &= \lambda \cdot \mathbf{g}_j \quad \text{mit } \lambda \in k \text{ und } j < i \end{aligned}$$

und wenn alle Zeilenvektoren von \mathbf{A} in dieser Matrix \mathbf{G} als Zeilenvektoren vorkommen.

- Beispiel :

Sei

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & -6 & 2 \end{pmatrix}$$

so ist folgendes eine Berechnungsmatrix für \mathbf{A} :

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \\ 1 & -3 & 1 \\ 2 & -6 & 2 \end{pmatrix} \quad \text{denn es ist} \quad \begin{pmatrix} \mathbf{g}_1 = \mathbf{e}_1 \\ \mathbf{g}_2 = \mathbf{e}_2 \\ \mathbf{g}_3 = \mathbf{e}_3 \\ \mathbf{g}_4 = 3 \cdot \mathbf{g}_2 \\ \mathbf{g}_5 = \mathbf{g}_1 + \mathbf{g}_3 \\ \mathbf{g}_6 = \mathbf{g}_5 - \mathbf{g}_4 \\ \mathbf{g}_7 = 2 \cdot \mathbf{g}_6 \end{pmatrix}$$

- Definition :

Die *lineare Komplexität* einer $(p \times n)$ -Matrix \mathbf{A} ist die kleinste Zahl r , zu der es eine $((n+r) \times n)$ -Berechnungsmatrix \mathbf{G} für \mathbf{A} gibt. Diese Zahl r wird mit $L_s(\mathbf{A})$ bezeichnet.

- Kommentar :

Die lineare Komplexität einer Matrix kommt von einer “ökonomischsten” Berechnungsfolge für Matrix-Vektor-Produkte $\mathbf{A} \cdot \mathbf{x}^T$, die von eventuellen speziellen Eigenschaften von \mathbf{x} keinen Gebrauch macht. Bei der Berechnung der Komplexität sind die ersten n Zeilen der Matrix \mathbf{G} “kostenlos” — es handelt sich um die Bereitstellung der input-Werte. An dem obigen Beispiel illustriert:

- ist $\mathbf{x} = (x_1, x_2, x_3) \in \mathbf{C}^3$ ein input-Vektor, so kann das Produkt

$$\mathbf{A} \cdot \mathbf{x}^T = \begin{pmatrix} 1 & 0 & 1 \\ 2 & -6 & 2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 + x_3 \\ 2x_1 - 6x_2 + 2x_3 \end{pmatrix}$$

mit vier komplexen Operationen der beschriebenen Art berechnet werden:

$$g_4 = 3 \cdot x_2, \quad g_5 = x_1 + x_3, \quad g_6 = g_5 - g_4, \quad g_7 = 2 \cdot g_6$$

- Einfache Eigenschaften:

- Ist \mathbf{P} eine Permutationsmatrix, so ist $L_s(\mathbf{P}) = 0$.
- Ist \mathbf{B} eine Teilmatrix von \mathbf{A} , so ist $L_s(\mathbf{B}) \leq L_s(\mathbf{A})$,

- Definition :

Sei nun $k = \mathbf{C}$ und $c \in \mathbf{R}$ mit $c > 0$. Eine Berechnungsmatrix \mathbf{G} für eine Matrix \mathbf{A} heißt *c-beschränkte Berechnungsmatrix*, wenn bei den Skalarmultiplikationen (s.o.)

$$\mathbf{g}_i = \lambda \cdot \mathbf{g}_j \quad \text{mit } \lambda \in k, j < i$$

stets $|\lambda| \leq c$ gilt.

Der Begriff der *c-beschränkten linearen Komplexität* einer Matrix \mathbf{A} wird entsprechend definiert und mit $L_c(\mathbf{A})$ bezeichnet.

- Eigenschaften :

- $L_s(\mathbf{A}) \leq L_c(\mathbf{A})$
- Ist \mathbf{B} eine Teilmatrix von \mathbf{A} , so ist $L_c(\mathbf{B}) \leq L_c(\mathbf{A})$
- $L_s(\mathbf{A}) = \lim_{c \rightarrow \infty} L_c(\mathbf{A})$

- **Satz von MORGENSTERN** :

Sei $\mathbf{A} \neq \mathbf{0}$ eine komplexe Matrix und $c \geq 2$, so gilt für *jede* quadratische Teilmatrix \mathbf{B} von \mathbf{A}

$$L_c(\mathbf{A}) \geq \log_c(|\det \mathbf{B}|)$$

- Zum Beweis : Wegen der erwähnten Eigenschaften von Teilmatrizen genügt es, die Aussage

$$L_c(\mathbf{A}) \geq \log_c(|\det \mathbf{A}|)$$

zu beweisen.

Sei also \mathbf{A} eine komplexe $(n \times n)$ -Matrix und \mathbf{G} eine $((n+r) \times n)$ -Berechnungsmatrix für \mathbf{A} , deren Zeilen mit \mathbf{g}_j , $(1 \leq j \leq n+r)$, bezeichnet werden. Für $1 \leq j_1 < j_2 < \dots < j_n \leq n+r$ bezeichne $\mathbf{G}_{j_1, j_2, \dots, j_n}$ die $(n \times n)$ -Teilmatrix von \mathbf{G} , die aus den Zeilen $\mathbf{g}_{j_1}, \mathbf{g}_{j_2}, \dots, \mathbf{g}_{j_n}$ besteht. Per Induktion über den Index j_n zeigt man

$$|\det \mathbf{G}_{j_1, j_2, \dots, j_n}| \leq c^{i_n - n}$$

und daher ist (mit $i_n \leq n+r$)

$$|\det A| \leq c^r$$

- Anwendung :

Sei nun $c = 2$; für die VANDERMONDE-Matrix \mathbf{V}_n der Diskreten Fourier-Transformation gilt (siehe Aufgabe 19b)

$$|\det \mathbf{V}_n| = n^{n/2}$$

und damit ist nach dem Satz von MORGENSTERN

$$L_2(\mathbf{V}_n) \geq \log_2 |\det \mathbf{V}_n| = \log_2 n^{n/2} = \frac{n}{2} \log_2 n$$

d.h. FFT_n ist "optimal" bezüglich der L_2 -Komplexität.

11.2.2 Faktorisierung

Das weiter oben dargestellte Verfahren der Schnellen Fourier-Transformation ist für Vektoren definiert worden, deren Länge eine Potenz von 2 ist. Diskrete Fourier-Transformation ist aber für beliebige $n \in \mathbf{N}$ definiert. Was macht man also, wenn man Vektoren zu transformieren hat, deren Länge keine Zweierpotenz ist? Das kann durchaus von der Situation der jeweiligen Anwendung abhängen.

- "Padding"

Die einfachste Lösung besteht darin, Vektoren der Länge n bis zur nächst größeren Zweierpotenz 2^k zu verlängern und mit Nullen aufzufüllen. Dann kann man Schnelle Fourier-Transformation zu dieser Länge wie üblich anwenden.

- "Faktorisieren"

Wenn ein solche künstliche Verlängerung zu aufwendig erscheint oder sich aus anderen Gründen verbietet, erinnere man sich daran, daß das Funktionieren der Schnellen Fourier-Transformation ja nur an ganz speziellen Eigenschaften der Einheitswurzeln gerader Ordnung hängt und daß solche Teilbarkeitseigenschaften viel allgemeiner gelten. Das kann man immer dann ausnützen, falls die zu bearbeitende Vektorlänge nicht Primzahl ist. (Zu letzterem Fall wird hier nichts weiter gesagt. Hat man für eine prime Ordnung p zu transformieren, so werte man die Multiplikation mit der VANDERMONDE-Matrix \mathbf{V}_p direkt aus, was (etwa) $p(p-1)$ komplexe Multiplikationen und insgesamt $\Theta(p^2)$ komplexe Operationen kostet).

Betrachten wir die Aufgabe einer DFT zur Länge n , die keine Primzahl ist, also in echte Teiler zerlegt werden kann. Zur Illustration soll zunächst das Beispiel $n = 15 = 3 \times 5$ behandelt werden.

Schreiben wir ein Polynom vom Grad < 15 so, daß die Koeffizienten mit denjenigen Indices, die zu derselben Restklasse modulo 3 gehören, zusammengefaßt werden:

$$\begin{aligned} A(X) &= \sum_{j=0}^{14} a_j X^j \\ &= A^{[0]}(X^3) + X \cdot A^{[1]}(X^3) + X^2 \cdot A^{[2]}(X^3) \end{aligned}$$

mit

$$\begin{aligned} A^{[0]}(X) &= a_0 + a_3 X + a_6 X^2 + a_9 X^3 + a_{12} X^4 \\ A^{[1]}(X) &= a_1 + a_4 X + a_7 X^2 + a_{10} X^3 + a_{13} X^4 \\ A^{[2]}(X) &= a_2 + a_5 X + a_8 X^2 + a_{11} X^3 + a_{14} X^4 \end{aligned}$$

Dann ist

$$DFT_{15}(\mathbf{A}) = \hat{\mathbf{A}} = (\hat{a}_0, \dots, \hat{a}_{15})$$

mit

$$\begin{aligned} \hat{a}_i &= A(\omega_{15}^i) \\ &= A^{[0]}(\omega_{15}^{3i}) + \omega_{15}^i A^{[1]}(\omega_{15}^{3i}) + \omega_{15}^{2i} A^{[2]}(\omega_{15}^{3i}) \\ &= A^{[0]}(\omega_5^i) + \omega_{15}^5 A^{[1]}(\omega_5^i) + \omega_{15}^{10} A^{[2]}(\omega_5^i) \\ &= b_{i \bmod 5} + \omega_{15}^i b_{5+(i \bmod 5)} + \omega_{15}^{2i} b_{10+(i \bmod 5)} \end{aligned}$$

für $0 \leq i < 15$.

Beachte nun: fasst man die Koeffizienten b_i in Blöcken der Länge 5 zusammen, so erweisen sie sich diese als Fourier-Transformierte der Länge 5:

$$\begin{aligned} (b_0, \dots, b_4) &= DFT_5(\mathbf{A}^{[0]}) \\ (b_5, \dots, b_9) &= DFT_5(\mathbf{A}^{[1]}) \\ (b_{10}, \dots, b_{14}) &= DFT_5(\mathbf{A}^{[2]}) \end{aligned}$$

wobei die $\mathbf{A}^{[j]}$ die den gleichnamigen Polynomen entsprechenden Vektoren der Länge 5 sind.

Fasst man nun die \hat{a}_k nach ihren Restklassen von k modulo 5 zu Vektoren der Länge 3 zusammen, so erhält man

$$\begin{aligned} \hat{\mathbf{A}}_{[k]} &= \begin{pmatrix} \hat{a}_k \\ \hat{a}_{5+k} \\ \hat{a}_{10+k} \end{pmatrix} = \begin{pmatrix} 1 & \omega_{15}^k & \omega_{15}^{2k} \\ 1 & \omega_{15}^{5+k} & \omega_{15}^{10+2k} \\ 1 & \omega_{15}^{10+k} & \omega_{15}^{20+2k} \end{pmatrix} \begin{pmatrix} b_{k \bmod 5} \\ b_{5+(k \bmod 5)} \\ b_{10+(k \bmod 5)} \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega_{15}^5 & \omega_{15}^{10} \\ 1 & \omega_{15}^{10} & \omega_{15}^{20} \end{pmatrix}}_{DFT_3} \begin{pmatrix} 1 & & \\ & \omega_{15}^k & \\ & & \omega_{15}^{2k} \end{pmatrix} \underbrace{\begin{pmatrix} b_{k \bmod 5} \\ b_{5+(k \bmod 5)} \\ b_{10+(k \bmod 5)} \end{pmatrix}}_{\mathbf{b}_{[k]}} \end{aligned}$$

Man sieht: die gesuchten Koeffizienten \hat{a}_k erhält man aus den b_j durch geeignete Umgruppierung, Skalarmultiplikation mit passenden Einheitswurzeln und fünf DFTs der Länge 3.

Zusammengefasst: man kann eine DFT-Operation der Länge 15 zerlegen in :

1. drei DFT-Operationen der Länge 5
2. eine Reihe von (insgesamt 8) Skalarmultiplikationen mit passenden fünfzehnten Einheitswurzeln
3. fünf DFT-Operation der Länge 3

Dies soll nun noch einmal schematisch dargestellt werden:

1. Input ist ein Vektor

$$\mathbf{A} = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14})$$

Dieser wird spaltenweise in eine (3×5) -Matrix eingelesen und deren Zeilen werden einzeln mit einer DFT_5 behandelt:

$$\begin{pmatrix} a_0 & a_3 & a_6 & a_9 & a_{12} \\ a_1 & a_4 & a_7 & a_{10} & a_{13} \\ a_2 & a_5 & a_8 & a_{11} & a_{14} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{[0]} \\ \mathbf{A}^{[1]} \\ \mathbf{A}^{[2]} \end{pmatrix} \\ \Downarrow \\ DFT_5 \\ \Downarrow \\ \begin{pmatrix} b_0 & b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 & b_9 \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} \end{pmatrix} = \begin{pmatrix} \mathbf{b}^{[0]} \\ \mathbf{b}^{[1]} \\ \mathbf{b}^{[2]} \end{pmatrix} \\ \parallel \\ (\mathbf{b}_{[0]}, \mathbf{b}_{[1]}, \mathbf{b}_{[2]}, \mathbf{b}_{[3]}, \mathbf{b}_{[4]})$$

2. Die Einträge der so erhaltenen Matrix werden mit passende Potenzen der fünfzehnten Einheitswurzel ω_{15} multipliziert (beachte: das Schema entspricht genau dem der ersten drei Zeilen und ersten fünf Spalten der Matrix \mathbf{V}_{15}) :

$$\begin{pmatrix} b_0 & b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 & b_9 \\ b_{10} & b_{11} & b_{12} & b_{13} & b_{14} \end{pmatrix} \Rightarrow \begin{pmatrix} b_0 & b_1 & b_2 & b_3 & b_4 \\ b_5 & \omega b_6 & \omega^2 b_7 & \omega^3 b_8 & \omega^4 b_9 \\ b_{10} & \omega^2 b_{11} & \omega^4 b_{12} & \omega^6 b_{13} & \omega^8 b_{14} \end{pmatrix} \\ \parallel \qquad \qquad \qquad \parallel \\ (\mathbf{b}_{[0]}, \mathbf{b}_{[1]}, \mathbf{b}_{[2]}, \mathbf{b}_{[3]}, \mathbf{b}_{[4]}) \qquad (\mathbf{b}'_{[0]}, \mathbf{b}'_{[1]}, \mathbf{b}'_{[2]}, \mathbf{b}'_{[3]}, \mathbf{b}'_{[4]})$$

3. Schließlich werden die Spaltenvektoren dieser Matrix jeweils mit DFT_3 transformiert:

$$\begin{array}{c}
 (\mathbf{b}'_{[0]}, \mathbf{b}'_{[1]}, \mathbf{b}'_{[2]}, \mathbf{b}'_{[3]}, \mathbf{b}'_{[4]}) \\
 \downarrow \\
 DFT_3 \\
 \downarrow \\
 (\hat{\mathbf{A}}_{[0]}, \hat{\mathbf{A}}_{[1]}, \hat{\mathbf{A}}_{[2]}, \hat{\mathbf{A}}_{[3]}, \hat{\mathbf{A}}_{[4]}) \\
 \parallel \\
 \begin{pmatrix} \hat{a}_0 & \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \hat{a}_4 \\ \hat{a}_5 & \hat{a}_6 & \hat{a}_7 & \hat{a}_8 & \hat{a}_9 \\ \hat{a}_{10} & \hat{a}_{11} & \hat{a}_{12} & \hat{a}_{13} & \hat{a}_{14} \end{pmatrix}
 \end{array}$$

und das gewünschte Resultat erhält man durch zeilenweises Auslesen:

$$\hat{\mathbf{A}} = (\hat{a}_0, \hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4, \hat{a}_5, \hat{a}_6, \hat{a}_7, \hat{a}_8, \hat{a}_9, \hat{a}_{10}, \hat{a}_{11}, \hat{a}_{12}, \hat{a}_{13}, \hat{a}_{14})$$

- Die allgemeine Situation

Sei nun $n = p \cdot q$ eine echte Faktorisierung der Länge n . Dann gilt $\omega_n^q = \omega_p$. Man kann zerlegen:

$$\begin{aligned}
 A(X) &= \sum_{i=0}^{n-1} a_i X^i \\
 &= \sum_{j=0}^{p-1} \sum_{k=0}^{q-1} a_{jq+k} X^{jq+k} \\
 &= \sum_{k=0}^{q-1} X^k \underbrace{\sum_{j=0}^{p-1} a_{j+k} X^{jq}}_{A^{[k]}(X^q)}
 \end{aligned}$$

wobei für $0 \leq k < q$:

$$A^{[k]}(Z) = \sum_{j=0}^{p-1} a_{jq+k} Z^j$$

Die Fourier-Transformierte

$$DFT_n(\mathbf{A}) = [A(\omega_n^0), A(\omega_n^1), A(\omega_n^2), \dots, A(\omega_n^{n-1})]$$

stellt sich dann so dar:

$$A(\omega_n^s) = \sum_{k=0}^{q-1} \omega_n^{sk} A^{[k]}(\omega_n^{sq}) = \sum_{k=0}^{q-1} \omega_n^{sk} A^{[k]}(\omega_p^s)$$

für $0 \leq s < n$, wobei zu beachten ist, daß $A^{[k]}(\omega_p^s)$ die Komponente mit Index $s \bmod p$ von $DFT_p(\mathbf{A}^{[k]})$ ist.

- Komplexität

Für die Anzahl der komplexen Multiplikationen gilt bei der eben beschriebenen Zerlegung $n = p \cdot q$:

$$q \cdot T(p) + n \cdot (q - 1)$$

Hat man nun eine Zahl n vorliegen die nicht wie das obige Beispiel $n = 15 = 3 \cdot 5$ ein Produkt von zwei Primzahlen ist, so sind mehrere Zerlegungen von n möglich und man kann optimieren. Dies führt auf folgende Rekursion für die Anzahl $T(n)$ der komplexen Multiplikationen bei optimaler Zerlegung — wobei diese Idee natürlich rekursiv, wie bei der FFT — angewendet werden soll:

$$T(n) = \begin{cases} n(n-1) & \text{falls } n \text{ Primzahl} \\ \min_{d|n} \{d \cdot T(\frac{n}{d}) + n \cdot (d-1)\} & \text{falls } n \text{ zusammengesetzt} \end{cases}$$

wobei die Minimierung über alle echten Teiler d von n läuft.

- An einem Beispiel illustriert:

$$\begin{aligned} T(28) &= \min \{ 2 \cdot T(14) + 28 \cdot 1, \\ &\quad 4 \cdot T(7) + 28 \cdot 3, \\ &\quad 7 \cdot T(4) + 28 \cdot 6, \\ &\quad 14 \cdot T(2) + 28 \cdot 13 \} \\ T(14) &= \min \{ 2 \cdot T(7) + 14, 7 \cdot T(2) + 14 \cdot 6 \} \\ &= \min \{ 98, 98 \} = 98 \\ T(4) &= 2 \cdot T(2) + 4 = 8 \\ \implies T(28) &= \min \{ 224, 252, 224, 392 \} = 224 \end{aligned}$$

Man vergleiche das mit den Kosten von $n(n-1) = 28 \times 27 = 756$, die bei *direkter* Auswertung der Multiplikation mit \mathbf{V}_{28} entstehen würden.

Man kann leicht nachvollziehen, *welche* Zerlegungen zu diesem optimalen Resultat führen:

$$\begin{array}{ccc} DFT_{28} & DFT_{28} & DFT_{28} \\ \times 2 | & \times 2 | & \times 7 | \\ DFT_{14} & DFT_{14} & DFT_4 \\ \times 2 | & \times 7 | & \times 2 | \\ DFT_7 & DFT_2 & DFT_2 \end{array}$$

Man erkennt an diesem speziellen Fall:

- Bei den optimalen Zerlegungen tritt in allen Zerlegungsschritten eine *prime* Anzahl von Teilproblemen auf.
- Alle Zerlegungen, die diese Regel befolgen, sind auch optimal!

Diese Aussage gilt auch generell, also für beliebige Längen n , was noch festgehalten werden soll:

- Explizite Lösung der Rekursion für $T(n)$

Satz: Die Lösung $T(n)$ der obigen Rekursion für die Anzahl der komplexen Multiplikationen bei Schneller Fourier-Transformation nach optimalem Zerlegungsschema ist gegeben durch

$$T(n) = n \cdot \sum_{p|n} \alpha_p \cdot (p-1) \quad \text{wobei} \quad n = \prod_{p \text{ prim}} p^{\alpha_p}$$

und dies entspricht dem iterativen Zerlegen jedes DFT-Problems der Länge k in eine prime Anzahl p von Teilproblemen der Länge k/p . Die Reihenfolge des Auftretens der Primfaktoren spielt dabei keine Rolle.

Zum *Beweis* siehe Aufgabe 20.

11.3 Algebraische Aspekte

11.3.1 Transformation über endlichen Körpern

Endliche Körper sind algebraische Strukturen, bestehend aus endlich-vielen Elementen, in denen Operationen "Addition" und "Subtraktion" definiert sind, wobei für diese Operationen die üblichen Rechenregeln gelten, wie man sie aus den vertrauten Körpern der rationalen, reellen oder komplexen Zahlen kennt (abelsche Gruppe bezüglich der Addition, abelsche Gruppe bezüglich der Multiplikation für die von 0 verschiedenen Elemente, Distributivgesetz) Einfachste Beispiele für solche endlichen Körper liefern die Restklassenringe $\mathbf{Z}_n = \mathbf{Z}/n\mathbf{Z}$, bei denen auf der Menge der Zahlen $\{0, 1, 2, \dots, n-1\}$ Addition und Multiplikation per Division mit Rest modulo n definiert werden. \mathbf{Z}_n ist genau dann ein Körper, wenn n eine Primzahl ist.

Endliche Körper sind aus der Sicht der Fourier-Transformation deshalb so attraktiv, weil sie (sieht man von dem jeweiligen Null-Element ab) aus lauter Einheitswurzeln bestehen. Um das am Beispiel von \mathbf{Z}_{17} und \mathbf{Z}_{19} zu erläutern:

- Alle Elemente $x \in \mathbf{Z}_{17}^* = \mathbf{Z}_{17} \setminus \{0\}$ genügen der Gleichung $x^{16} = 1$, sind also sechzehnte Einheitswurzeln. Davon sind *primitive* sechzehnte Einheitswurzeln die Elemente von

$$\mathcal{P}_{16} = \{3, 5, 6, 7, 10, 11, 12, 14\}$$

Alle diese Element genügen keiner Gleichung $x^t = 1$ für ein t mit $1 \leq t < 16$. Die übrigen Elemente von \mathbf{Z}_{17}^* gruppieren sich folgendermaßen in Mengen von primitiven Einheitswurzeln für die Teiler $d = 1, 2, 4, 8$ von $n = 16$:

$$\mathcal{P}_1 = \{1\}, \quad \mathcal{P}_2 = \{16\}, \quad \mathcal{P}_4 = \{4, 13\}, \quad \mathcal{P}_8 = \{2, 8, 9, 15\}$$

Über diesem Körper \mathbf{Z}_{17} sind also alle Einheitswurzeln vorhanden, die man für eine FFT der Länge $n = 16$ benötigt.

- Alle Elemente $x \in \mathbf{Z}_{19}^* = \mathbf{Z}_{19} \setminus \{0\}$ genügen der Gleichung $x^{18} = 1$, sind also achtzehnte Einheitswurzeln. Davon sind *primitive* achtzehnte Einheitswurzeln die Elemente von

$$\mathcal{P}_{18} = \{2, 3, 10, 13, 14, 15\}$$

Alle diese Element genügen keiner Gleichung $x^t = 1$ für ein t mit $1 \leq t < 18$. Die übrigen Elemente von \mathbf{Z}_{19}^* gruppieren sich folgendermaßen in Mengen von primitiven Einheitswurzeln für die Teiler $d = 1, 2, 3, 6, 9$ von $n = 18$:

$$\mathcal{P}_1 = \{1\}, \mathcal{P}_2 = \{18\}, \mathcal{P}_3 = \{7, 11\}, \mathcal{P}_6 = \{8, 12\}, \mathcal{P}_9 = \{4, 5, 6, 9, 16, 17\},$$

Über diesem Körper \mathbf{Z}_{19} sind also alle Einheitswurzeln vorhanden, die man für eine FFT der Länge $n = 18$ benötigt.

- Ein Körper mit 8 Elementen

α sei ein Element, das sich beim Rechnen “modulo 2” wie eine Wurzel des Polynoms $X^3 + X + 1$ verhält, also der Gleichung

$$\alpha^3 = \alpha + 1$$

genügt. Elemente der Körpers \mathbf{F}_8 sind die 8 polynomialen Terme in α vom Grad < 3 mit Koeffizienten in \mathbf{F}_2 :

$$0 \quad 1 \quad \alpha \quad 1 + \alpha \quad \alpha^2 \quad \alpha^2 + 1 \quad \alpha + \alpha^2 \quad 1 + \alpha + \alpha^2$$

Addition und Multiplikation in \mathbf{F}_8 werden wie Polynomoperationen ausgeführt, wobei eventuell auftretende höhere Potenzen von α als α^2 gemäss der Vorschrift

$$\alpha^3 \leftarrow \alpha + 1$$

ersetzt werden.

Addition in \mathbf{F}_8 :

0	1	α	$1 + \alpha$	α^2	$\alpha^2 + 1$	$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$
1	0	$1 + \alpha$	α	$\alpha^2 + 1$	α^2	$1 + \alpha + \alpha^2$	$\alpha + \alpha^2$
α	$1 + \alpha$	0	1	$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$	α^2	$\alpha^2 + 1$
$1 + \alpha$	α	1	0	$1 + \alpha + \alpha^2$	$\alpha + \alpha^2$	$\alpha^2 + 1$	α^2
α^2	$\alpha^2 + 1$	$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$	0	1	α	$1 + \alpha$
$\alpha^2 + 1$	α^2	$1 + \alpha + \alpha^2$	$\alpha + \alpha^2$	1	0	$1 + \alpha$	α
$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$	α^2	$\alpha^2 + 1$	α	$1 + \alpha$	0	1
$1 + \alpha + \alpha^2$	$\alpha + \alpha^2$	$\alpha^2 + 1$	α^2	$1 + \alpha$	α	1	0

Multiplikation in \mathbf{F}_8 :

0	0	0	0	0	0	0	0
0	1	α	$1 + \alpha$	α^2	$\alpha^2 + 1$	$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$
0	α	α^2	$\alpha + \alpha^2$	$1 + \alpha$	1	$1 + \alpha + \alpha^2$	$\alpha^2 + 1$
0	$1 + \alpha$	$\alpha + \alpha^2$	$\alpha^2 + 1$	$1 + \alpha + \alpha^2$	α^2	1	α
0	α^2	$1 + \alpha$	$1 + \alpha + \alpha^2$	$\alpha + \alpha^2$	α	$\alpha^2 + 1$	1
0	$\alpha^2 + 1$	1	α^2	α	$1 + \alpha + \alpha^2$	$1 + \alpha$	$\alpha + \alpha^2$
0	$\alpha + \alpha^2$	$1 + \alpha + \alpha^2$	1	$\alpha^2 + 1$	$1 + \alpha$	α	α^2
0	$1 + \alpha + \alpha^2$	$\alpha^2 + 1$	α	1	$\alpha + \alpha^2$	α^2	$1 + \alpha$

Einheitswurzeln in \mathbf{F}_8 : die folgende Tabelle enthält (zeilenweise gelesen) die Potenzen der von 0 verschiedenen Elemente von \mathbf{F}_8 . Daran zeigt sich, daß alle diese Elemente siebte Einheitswurzeln sind; die 1 ist erste Einheitswurzel, alle übrigen Elemente sind primitive siebte Einheitswurzeln.

x^0	x^1	x^2	x^3	x^4	x^5	x^6	x^7
1	1	1	1	1	1	1	1
1	α	α^2	$1 + \alpha$	$\alpha^2 + \alpha$	$1 + \alpha + \alpha^2$	$1 + \alpha^2$	1
1	$1 + \alpha$	$1 + \alpha^2$	α^2	$1 + \alpha + \alpha^2$	α	$\alpha^2 + \alpha$	1
1	α^2	$\alpha^2 + \alpha$	$1 + \alpha^2$	α	$1 + \alpha$	$1 + \alpha + \alpha^2$	1
1	$1 + \alpha^2$	$1 + \alpha + \alpha^2$	$\alpha^2 + \alpha$	$1 + \alpha$	α^2	α	1
1	$\alpha + \alpha^2$	α	$1 + \alpha + \alpha^2$	α^2	$1 + \alpha^2$	$1 + \alpha$	1
1	$1 + \alpha + \alpha^2$	$1 + \alpha$	α	$1 + \alpha^2$	$\alpha^2 + \alpha$	α^2	1

Generell gilt:

- Zu jeder Primzahl p und jeder natürlichen Zahl $n \geq 1$ gibt es einen Körper mit p^n Elementen, bezeichnet als \mathbf{F}_{p^n} oder $GF(p^n)$ ¹.
- Alle Körper mit p^n Elementen sind *isomorph*, als mathematische Strukturen als nicht zu unterscheiden. Die konkreten Realisierungen können allerdings sehr unterschiedlich sein — was sich auf die Algorithmen natürlich stark auswirkt.
- Die Menge der (multiplikativ) invertierbaren Elemente $\mathbf{F}_{p^n}^* = \mathbf{F}_{p^n} \setminus \{0\}$ eines endlichen Körpers bilden eine *Gruppe*, die *zyklisch* ist: es handelt sich um alle $p^n - 1$ Lösungen der Gleichung

$$X^{p^n - 1} = 1$$

die “über dem Grundkörper” \mathbf{F}_p möglich sind.

- Ein endlicher Körper besteht (von der 0 abgesehen) also aus lauter Einheitswurzeln! Und zwar für jeden Teiler d von $p^n - 1$ aus genau $\phi(d)$ primitiven d -ten Einheitswurzeln, wenn d ein Teiler von $p^n - 1$ ist.
- Inklusion wird durch Teilerstruktur der Exponenten induziert:

$$\mathbf{F}_{p^d} \subseteq \mathbf{F}_{p^n} \Leftrightarrow d \mid n$$

- Wenn man Einheitswurzeln braucht, findet man sie (meist) auch:

- Zu jeder Primzahl p und jedem n mit $p \nmid n$ gibt es ein t mit $n \mid p^t - 1$, d.h. \mathbf{F}_{p^t} enthält die n -ten Einheitswurzeln.

Diskrete und Schnelle Fourier-Transformation über endlichen Körpern (und endlichen Ringen — aber Vorsicht! Die Frage nach Existenz und Anzahl von Einheitswurzeln kann delikater werden) ist deshalb interessant, weil man diesen Bereichen *rundungsfehlerfrei* rechnet und sich Algorithmen über diesen Körpern (zumindest im Fall $p = 2$) für eine direkte hardware-Realisierung eignen. Natürlich müssen die zu transformierenden Daten von dieser diskreten Natur sein oder dorthin übersetzt werden können. Diese Voraussetzung ist aber im Bereich von digitalen Signalen etwa, wie auch bei Problemen der integer-Arithmetik, erfüllt.

¹*GF* steht für *Galois field*, zu Ehren des französischen Mathematikers Evariste GALOIS (1811-1832).

11.3.2 Transformation über anderen Gruppen

Es sollte bei der obigen Darstellung klar geworden sein, daß das Prinzip der FFT von den Eigenschaften sehr einfacher *Gruppen*, eben der zyklischen Gruppen der n -ten Einheitswurzeln, Gebrauch macht. Man kann fragen, ob sich solche Zerlegungs-Prinzipien auch in anderen Gruppen realisieren lassen — und wozu man solche Erfindungen eventuell gebrauchen kann. Tatsächlich zeichnen sich hierzu in den letzten Jahren interessante Entwicklungen ab. Ein Schritt in diese Richtung stellt eine Erlanger Habilitationsschrift (TH. BETH, *Verfahren der Schnellen Fourier-Transformation*, Teubner-Verlag, 1984) dar. Das kürzlich erschienene Buch von M. CLAUSEN, *Fast Fourier Transforms*, BI-Wissenschaftsverlag, 1993, präsentiert diese Ansätze und Ergebnisse systematisch aus der Sicht der Darstellungstheorie endlicher Gruppen.

11.4 Anwendungen

Auch diese Punkt kann wegen seines Umfangs nur mit wenigen Zitaten gestreift werden.

- Numerische Approximation der klassischen FT
Diskretisierungsschemata für partielle Dgln
- Signalverarbeitung, Bildverarbeitung, digitales Filtern, bildgebende Verfahren der Medizin (CT, NMR), Geologie,...
- Fehlerkorrigierende Codes (RS-Codes)
- Zeitreihenanalyse, Spektraltechniken
- exakte Polynom- und Integer-Arithmetik (Schönhage/Strassen-Multiplikation), Potenzreihenarithmetik [→ KNUTH, AHU, LIPSON, NAUDIN/QUITTÉ , ...]

11.5 Historische Bemerkung

Als Auslöser für die *moderne* Periode der Schnellen Fourier-Transformation mit ihren Anwendungen und Varianten wird die folgende Veröffentlichung angesehen:

- J. W. COOLEY and J. W. TUKEY : An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, **19**, 1965, p. 297-301.

in der die Prinzipien der Methode zum Ausdruck kommen. Was die Breitenwirkung angeht, ist diese Zuordnung zweifellos richtig, entspricht aber nicht den historischen Prioritäten. Man findet die Rechentechnik der FFT bereits in dem Buch

- C. RUNGE und R. KÖNIG : *Vorlesungen über Numerisches Rechnen*, Grund-
lehren der Mathematischen Wissenschaften, Band 11, Springer, Berlin, 1924.

wobei C. RUNGE schon 1903 darüber in der *Zeitschrift für Mathematische Physik* geschrieben hat. Weitere Veröffentlichungen der Methode, die allerdings weniger Beachtung fanden, stammen von DANIELSON und LANZOS (1942), sowie GOOD (1958/1960). Die Autoren HEIDEMAN, JOHNSON und BURRUS behaupten, daß die Grundprinzipien schon C.F. GAUSS bekannt waren (*IEEE Magazine*, 1984).

11.6 Literatur

Die Schnelle Fourier-Transformation wird in Lehrbüchern heutzutage gebührend berücksichtigt, und es gibt eine ganze Reihe von guten bis vorzüglichen Darstellungen. Als besonders lesenswert sollen die Darstellungen in den Büchern von AHO/HOPCROFT/ULLMAN, LIPSON, GEDDES/ CZAPOR/LABAHN und von NAUDIN/QUITTÉ hervorgehoben werden (siehe Literaturliste).

Entsprechend der Bedeutung der Fourier-Transformation in theoretischer und praktischer Hinsicht, und angesichts der großen Vielfalt an Anwendungen, ist es nicht verwunderlich, daß zu diesem Thema eine immense Literatur zur Verfügung steht. Das ELIS-System gibt beispielsweise folgende Auskunft:

Autor, Titel, Schlagwort

Fourier	289
Transformation	781
Fourier UND Transformation	85

Aus der Fülle seien hier einige Titel herausgegriffen. Die erste Gruppe der hier aufgeführten Bücher betont die mathematischen Aspekte der Diskreten und Schnellen Fourier-Transformation (ohne deswegen Anwendungsaspekte auszuklammern):

- Beth : Verfahren der schnellen Fourier-Transformation
- Bracewell : The Fourier transform and its applications
- Brigham : FFT
- Burrus/Potts : DFT/FFT and convolution algorithms
- Cizek : Discrete Fourier transforms and their applications
- Clausen/Baum : Fast Fourier transforms
- Nussbaumer : Fast Fourier transform and convolution algorithms
- Tolimieri et al. : Algorithms for discrete Fourier transform and convolution
- VanLoan : Computational frameworks for the fast Fourier transform

Das breite Spektrum der Anwendungen sei durch eine Auswahl von Titeln repräsentativ beschrieben:

- Achilles : Die Fourier-Transformation in der Signalverarbeitung
- Achilles : Ueber die diskrete Fourier-Transformation und ihre Anwendungen auf lineare zeitinvariante Systeme
- Bestmann : Beitrage zur Mikrowellen-Fouriertransform-Spektroskopie
- Gaskill : Linear systems, Fourier transforms, and optics

- Geiger : Nullstellenbestimmung bei Polynomen und allgemeinen analytischen Funktionen als Anwendung der schnellen Fouriertransformation
- Lahmeyer : Anwendungen der schnellen Fouriertransformation und der quadratischen Programmierung bei der Interpretation von Schwerefeldern
- Kettler : Spektralanalyse regelloser Achskrümmungen in Hohlkabelrohren mittels schneller diskreter Fourier-Transformation zur Bestimmung der Zusatzdämpfung der HO1-Welle im dielektrisch beschichteten Rundhohlleiter
- Nagel : Schnelle Fourier- und Walsh-Transformation (Anwendungsmöglichkeiten in der Nuklearmedizin)
- Niederdrenk : Die endliche Fourier- und Walsh-Transformation mit einer Einführung in die Bildverarbeitung
- Pickering : An introduction to fast Fourier transform methods for partial differential equations, with applications
- Shaw : Fourier transform NMR spectroscopy
- Stark : Applications of optical Fourier transforms
- Treinies : Die Ermittlung aerodynamischer Derivativa mit Hilfe der diskreten Fourier-Transformation freier Schwingungen
- Weaver : Applications of discrete and continuous Fourier analysis