Numerische Mathematik II
Universität Hamburg
WS06/07

K. Taubert

Komplexe und trigonometrische Polynome

# § 9. DIE SCHNELLE FOURIERTRANSFORMATION (FFT) und Anwendungen

Gegenstand dieses Kapitels sind zunächst komplexe Polynome

$$t_N(z) = \sum_{k=0}^{N-1} c_k z^k .$$

und drei damit zusammenhängende Aufgaben:

1. Für vorgegebene $z_j \in \mathbf{C}$, $j = 0,1,2, \dots ,N-1$, und bei bekannten $c_k \in \mathbf{C}$ sollen die Werte $t_N(z_j)$ bestimmt werden.
2. Für vorgegebene $\tau_j \in \mathbf{C}$, $j = 0,1,2, \dots ,N-1$, sollen Koeffizienten $c_k \in \mathbf{C}$ gefunden werden mit $\tau_j = \sum_{k=0}^{N-1} c_k z_j^k$.
3. Anschließend wird noch auf die Approximation im Mittel von periodischen Funktionen durch reelle trigonometrische Polynome eingegangen.

Im Allgemeinen erfordert die Lösung dieser Aufgaben einen Aufwand der Größenordnung von $N^2$ Multiplikationen bzw. Divisionen. Werden jedoch die $z_j$ als j-te Potenzen der N-ten komplexen Einheitswurzel gewählt und ist $N = 2^n$, dann kann dieser Aufwand auf die Größenordnung $N \ln_2 N$ reduziert werden. Dieses leistet die Schnelle Diskrete Fourier-Transformation (FFT) bzw. die Inverse Schnelle Fourier-Transformation (IFFT).
Geeignete Umschreibungen führen zu ähnlichen Aussagen über reelle trigonometrische Polynome

$$t_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos(kx) + b_k \sin(kx) .$$

Die numerische Behandlung der drei Aufgaben kann unter geeigneten Voraussetzungen durch die FFT und ihre Varianten durchgeführt werden. Dies geschieht sehr einheitlich.

## § 9.1 Die Schnelle Diskrete Fourier-Transformation (FFT)

Gegeben sei das komplexe Polynom

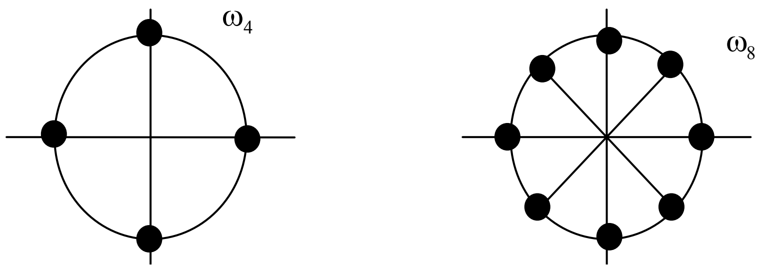
$$t_N(z) = \sum_{k=0}^{N-1} c_k z^k , \qquad c_k \in \mathbf{C}.$$

Die Wertebestimmung eines solchen Polynoms an einer Stelle $\omega_N \in \mathbf{C}$

$$t_N(\omega_N) = \sum_{k=0}^{N-1} c_k \omega_N^k$$

erfordert bekanntlich N komplexe Multiplikationen. Soll der Wert des Polynoms an den N Stellen

$$\omega_N^j = \exp(i2\pi j/N) = \cos(2\pi j/N) + i \sin(2\pi j/N), \quad j = 0,1,2, \dots ,N\text{-}1$$




N=4 N=8

(Potenzen der N-ten Einheitswurzel) berechnet werden, dann würde dieses – auf den ersten Blick – einen Aufwand der Größenordnung von $N^2$ Multiplikationen erfordern. Ist $N = 2^n$, dann kann mit Hilfe des folgenden Satzes der Aufwand auf die Größenordnung $N\ln_2 N$ reduziert werden.

**Satz**

Es sei N gerade, $x_j = 2\pi j / N$, $j = 0,1,2, \dots , N\text{-}1$ und

$$t_N(x_j) = \sum_{k=0}^{N-1} c_k (\omega_N^j)^k , \qquad j = 0,1,2, \dots ,N\text{-}1.$$

Dann gilt mit $M = N/2$

$$t_N(x_{2\ell}) = \sum_{k=0}^{M-1} d_k (\omega_N^\ell)^k \qquad \begin{array}{c} \ell = 0,1, \dots , M-1 \\ d_k = c_k + c_{M+k} \end{array}$$

$$t_N(x_{2\ell+1}) = \sum_{k=0}^{M-1} d_k (\omega_N^\ell)^k \qquad \begin{array}{c} \ell = 0,1, \dots , M-1 \\ d_k = (c_k - c_{M+k})\omega_N^k \end{array} .$$

**Der Satz zeigt, dass die Bestimmung der Werte von $t_N(.)$ an den Stellen $z_j$ ersetzt werden kann durch die Auswertung von zwei Ausdrücken gleicher Bauart mit halber Länge. Die Schnelle Diskrete Fourier-Transformation (FFT) besteht darin, diesen Reduktionsschritt für jede der neu entstehenden Formeln so lange zu wiederholen bis M = 1 ist. Am Ende stehen dann die gesuchten Werte.**

**Beweis des Satzes:**
Es sei N eine gerade Zahl, M=N/2 und

$$\omega_N^j = \exp(i2\pi j/N) \quad \text{für} \quad j = 0,1,2, \ldots, N-1.$$

Für $k = 2\ell$, $\ell = 0,1,2, \ldots, M-1$, sind

$$\omega_N^{jk} = \exp(i2\pi jk/N) = \exp(i2\pi j2l/2M) = \omega_M^{j\ell}$$

und

$$\omega_M^{(j+M)\ell} = \omega_M^{j\ell}\omega_M^{M\ell} = \omega_M^{j\ell}\exp((i2\pi/M)M\ell) = \omega_M^{j\ell}.$$

Damit ist

$$t_N(x_{2\ell}) = \sum_{j=0}^{N-1} c_j\omega_N^{jk} = \sum_{j=0}^{N-1} c_j\omega_M^{j\ell} = \sum_{j=0}^{M-1} c_j\omega_M^{j\ell} + \sum_{j=M}^{N-1} c_j\omega_M^{j\ell}$$

$$= \sum_{j=0}^{M-1} c_j\omega_M^{j\ell} + \sum_{j=0}^{M-1} c_{j+M}\omega_M^{(j+M)\ell} = \sum_{j=0}^{M-1}(c_j + c_{j+M})\omega_M^{j\ell}$$

und analog ergibt sich

$$t_N(x_{2\ell+1}) = \sum_{j=0}^{M-1}(c_j - c_{M+j})\omega_N^j.$$

Zum besseren Verständnis betrachten wir die folgende Tabelle, in der exemplarisch eine FFT für den Fall $N = 8 = 2^3$ durchgeführt wird.

FFT für $N = 8 = 2^3$

| Binäre Adressen | k | $c_k$ | $c_k^{(1)}$ | $c_k^{(2)}$ | $c_k^{(3)}$ | j | Binäre Adressen |
|---|---|---|---|---|---|---|---|
| 000 | 0 | $c_0$ | $c_0+c_4$ | $c_0^{(1)} + c_2^{(1)}$ | $c_0^{(2)} + c_1^{(2)}$ | 0 | 000 |
| 001 | 1 | $c_1$ | $c_1+c_5$ | $c_1^{(1)} + c_3^{(1)}$ | $(c_0^{(2)} - c_1^{(2)})\omega_2^0$ | 4 | 100 |
| 010 | 2 | $c_2$ | $c_2+c_6$ | $(c_0^{(1)} - c_2^{(1)})\omega_4^0$ | $c_2^{(2)} + c_3^{(2)}$ | 2 | 010 |
| 011 | 3 | $c_3$ | $c_3+c_7$ | $(c_1^{(1)} - c_3^{(1)})\omega_4^1$ | $(c_2^{(2)} - c_3^{(2)})\omega_2^0$ | 6 | 110 |
| 100 | 4 | $c_4$ | $(c_0-c_4)\omega_8^0$ | $c_4^{(1)} + c_6^{(1)}$ | $c_4^{(2)} + c_5^{(2)}$ | 1 | 001 |
| 101 | 5 | $c_5$ | $(c_1-c_5)\omega_8^1$ | $c_5^{(1)} + c_7^{(1)}$ | $(c_4^{(2)} - c_5^{(2)})\omega_2^0$ | 5 | 101 |
| 110 | 6 | $c_6$ | $(c_2-c_6)\omega_8^2$ | $(c_4^{(1)} - c_6^{(1)})\omega_4^0$ | $c_6^{(2)} + c_7^{(2)}$ | 3 | 011 |
| 111 | 7 | $c_7$ | $(c_3-c_7)\omega_8^3$ | $(c_5^{(1)} - c_7^{(1)})\omega_4^1$ | $(c_6^{(2)} - c_7^{(2)})\omega_2^0$ | 7 | 111 |

Die gesuchten Werte $t_N(x_j)$ stehen in der sechsten Spalte. Die zugehörigen Stellen $x_j$ (bzw. deren Indizes j) können aus der siebten Spalte entnommen werden und ergeben sich durch Umkehrung der binären Adressen der zugehörigen Zeilen in der ersten Spalte.

**Beispiel**
Gegeben sei das Polynom

$$t_4(z) = \sum_{k=0}^{3} c_k z^k$$

mit den Koeffizienten

| k | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| $c_k$ | $4\pi^2/3+1/4$ | $20/9+i4\pi/3$ | 1 | $20/9-i4\pi/3$ |

Für die Bestimmung der Werte dieses Polynoms an den Stellen

$$z_j = \omega_4^j = \exp(i2\pi j/4) \quad \text{für} \quad j = 0,1,2,3$$

kann die FFT eingesetzt werden und liefert

| k | $c_k$ | $c_k^{(1)}$ | $c_k^{(2)}$ | j |
|---|---|---|---|---|
| $0 = 00_2$ | $4\pi^2/3+1/4$ | $c_0+c_2$<br>$4\pi^2/3+5/4$ | $c_0^{(1)} + c_1^{(1)}$<br>$4\pi^2/3+205/36$ | $0 = 00_2$ |
| $1 = 01_2$ | $20/9+i4\pi/3$ | $c_1+c_3$<br>$40/9$ | $c_0^{(1)} - c_1^{(1)}$<br>$4\pi^2/3-115/36$ | $2 = 10_2$ |
| $2 = 10_2$ | 1 | $c_0-c_2$<br>$4\pi^2/3-3/4$ | $c_2^{(1)} + c_3^{(1)}$<br>$4\pi^2/3-8\pi/3-3/4$ | $1 = 01_2$ |
| $3 = 11_2$ | $20/9-i4\pi/3$ | $(c_1-c_3)i$<br>$-8\pi/3$ | $c_2^{(1)} - c_3^{(1)}$<br>$4\pi^2/3+8\pi/3-3/4$ | $3 = 11_2$ |

Das Polynom hat also z.B. an der Stelle $z_2$ den Wert $t_3(z_2) = 4\pi^2/3-115/36$.

Ein elegantes (rekursives) MATLAB Programm für die FFT mit einem Zeilenvektor c ist:

```
function z = sft(c)
N = length(c);
m=N/2;
wn = exp(i*pi/m);
if N==1
z =c;
else
for k=1:m
d(k) = c(k) +c(k+m);
d(k+m)= (c(k)-c(k+m))*wn^(k-1);
end
z = [sft(d(1:m)),sft(d(m+1:N))];
end
```

Mit dem Befehl “bitrevorder” kann schließlich noch die kanonische Reihenfolge der Elemente aus z hergestellt werden.

## § 9.2 Komplexe Polynominterpolation und die IFFT

Gegeben seien N Punkte $(z_j, \tau_j)$, $j = 0,1,2, \dots ,N-1$, mit $z_j = \exp(i2\pi j/N)$ und $\tau_j \in \mathbf{C}$.
Gesucht ist dann ein komplexes (Interpolations-) Polynom vom (Höchst-) Grad N-1

$$t_N(z) = \sum_{k=0}^{N-1} c_k z^k \qquad c_k \in \mathbf{C}.$$

mit der Eigenschaft

$$\tau_j = \sum_{k=0}^{N-1} c_k z_j^k \ , \quad j = 0,1,2, \dots ,N-1.$$

Wird $x_j = 2\pi j/N$, $j = 0,1,2, \dots , N-1$ gesetzt, dann liegt die Aufgabe vor, komplexe Zahlen $c_0, c_1, c_2, \dots , c_{N-1}$ zu finden mit der Eigenschaft

$$\tau_j = \sum_{k=0}^{N-1} c_k \exp(ikx_j) \ , \qquad j = 0,1,2, \dots , N-1.$$

Wegen der besonderen Wahl der Stellen $z_j$ wird die Aufgabe auch als trigonometrische Interpolationsaufgabe bezeichnet.
Die Aufgabe besitzt eine eindeutig bestimmte Lösung. Das Interpolationspolynom könnte mit dem Schema der dividierten Differenzen ermittelt werden und würde einen Aufwand der Größenordnung von $N^2$ Divisionen erfordern. Mit Hilfe des folgenden Satzes kann gezeigt werden, dass der Aufwand auf die Größenordnung von $N \ln_2 N$ Multiplikationen reduziert werden kann.

Erfreulicherweise kann die Lösung der Interpolationsaufgabe explizit angegeben werden, denn es gilt der

**Satz**
Die Interpolationsaufgabe besitzt eine eindeutig bestimmte Lösung.
Die Koeffizienten $c_j$ sind gegeben durch

$$c_j = \frac{1}{N}\sum_{k=0}^{N-1} \tau_k \exp(-ikx_j) , \qquad j = 0,1,2, \dots , N-1.$$

Zum Beweis des Satzes benötigen wir noch ein

**Lemma**
Für festes $N \in \mathbf{N}$ sei

$$\omega_N = \exp(i2\pi/N)$$

Dann gilt

$$S = \sum_{k=0}^{N-1} \omega_N^{k\ell} = \begin{cases} N & \text{falls } \ell = \nu N \text{ mit einem } \nu \in \mathbf{Z} \\ 0 & \text{falls } \ell \neq \nu N \text{ mit einem } \nu \in \mathbf{Z} \end{cases}.$$

**Beweis des Lemmas**
Es ist

$$\omega_N^{\ell} = \exp(i2\pi\ell/N).$$

Ist $\ell$ ein ganzzahliges Vielfaches von N, dann ist jeder Summand von S gleich 1 und dieser Teil des Lemmas ist somit bewiesen.
Ist $\ell$ kein ganzzahliges Vielfaches von N, dann kann S dargestellt werden in der Form

$$S = \sum_{k=0}^{N-1} (\omega_N^{\ell})^k = \frac{1-\omega_N^{\ell N}}{1-\omega_N^{\ell}} = 0 .$$

**Beweis des Satzes**
Der Ausdruck

$$\tau_j = \sum_{k=0}^{N-1} c_k (z_j)^k \quad , \quad z_j = \exp(ix_j) \ , \quad x_j = 2\pi j/N \ , \quad j = 0,1,2, \dots ,N\text{-}1$$

kann dargestellt werden in der Form

$$\tau_j = \sum_{k=0}^{N-1} c_k \omega_N^{jk} \ , \qquad \omega_N = \exp(i2\pi/N) \ , \qquad j = 0,1,2, \dots ,N\text{-}1.$$

Multiplikation mit $\omega_N^{-j\ell}$ und Addition über j liefert

$$\sum_{j=0}^{N-1} \omega_N^{-j\ell} \tau_j = \sum_{j=0}^{N-1} \omega_N^{-j\ell} \sum_{k=0}^{N-1} c_k \omega_N^{jk} = \sum_{k=0}^{N-1} c_k \sum_{j=0}^{N-1} \omega_N^{-j\ell} \omega_N^{kj} = \sum_{k=0}^{N-1} c_k \sum_{j=0}^{N-1} \omega_N^{j(k-\ell)} = N c_\ell$$

und mithin die Behauptung.

**Bemerkung**
Der Satz zeigt, dass die gesuchten Koeffizienten $c_j$ der Interpolationsaufgabe sich offenbar aus einem Austausch der $\tau_j$ mit den $c_k$, einem Vorzeichenwechsel in der Exponentialfunktion (!) und einer Multiplikation mit 1/N ergeben.

Der Satz zeigt auch, dass die gesuchten Koeffizienten $c_k$ mit einem Aufwand der Größenordnung von $N^2$ Multiplikationen berechnet werden können. Eine günstigere Berechnung der gesuchten Koeffizienten $c_k$ kann mit der Inversen Schnellen Fourier-Transformation (IFFT) durchgeführt werden:

Es ist

$$Nc_j = \sum_{k=0}^{N-1} \tau_k \exp(-ikx_j) = \sum_{k=0}^{N-1} \tau_k (\overline{\omega}_N^{j})^k \quad , \qquad j = 0,1,2, \dots , N\text{-}1,$$

d.h. es liegt eine Darstellung vor, die der Aufgabe des ersten Abschnittes sehr ähnlich ist.

Nicht überraschend ist, dass auch diese Aufgabe für $N = 2^n$ mit einem Aufwand der Größenordnung von $N\ln_2 N$ Multiplikationen gelöst werden kann. Dieses leistet die Inverse Schnelle Fourier-Transformation (IFFT) . Wir gehen hier nicht weiter darauf ein und verwenden dafür das MATLAB-Programm

ifft(t),

wobei t der Zeilenvektor mit den Einträgen $\tau_k$ ist.

## § 9.3 Wertebestimmung von reellen trigonometrischen Polynomen

Gegenstand dieses Abschnittes sind trigonometrische Polynome N-ten Grades

$$t_N(x) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(kx) + b_k \sin(kx)$$

mit reellen Koeffizienten $a_0, a_1, b_1, \ldots, a_N, b_N$.
Der Wert des trigonometrischen Polynoms soll an den Stellen

$$x_j = 2\pi j/N, \qquad j = 0,1,2, \ldots, N\text{-}1$$

bestimmt werden.
Der Ausdruck

$$t_N(x_j) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(kx_j) + b_k \sin(kx_j), \qquad j = 0,1,2, \ldots\ldots, N\text{-}1$$

heißt diskretes Fourierpolynom.
Der folgende Satz zeigt, dass das diskrete Fourierpolynom in der Form

$$t_N(x_j) = \sum_{k=0}^{N-1} c_k \omega_N^{jk}, \qquad j = 0,1,2, \ldots\ldots, N\text{-}1$$

mit $\omega_N = \exp(i2\pi/N)$ und geeigneten Koeffizienten $c_k$ dargestellt werden kann.

**Satz**
Das diskrete Fourierpolynom

$$t_N(x_j) = \frac{a_0}{2} + \sum_{k=1}^{N} a_k \cos(kx_j) + b_k \sin(kx_j), \qquad j = 0,1,2, \ldots\ldots, N\text{-}1$$

mit $x_j = 2\pi j/N$, $j = 0,1,2, \ldots, N\text{-}1$, kann dargestellt werden in der Form

$$t_N(x_j) = \sum_{k=0}^{N-1} c_k \omega_N^{jk}, \qquad \omega_N = \exp(i2\pi/N), \qquad j = 0,1,2, \ldots\ldots, N\text{-}1.$$

Dabei ist

$$c_0 = (1/2)a_0 + a_N, \qquad c_k = (1/2)(\gamma_k + \overline{\gamma}_{N-k}), \qquad \gamma_k = (a_k - ib_k), \quad k = 1,2, \ldots, N\text{-}1.$$

**Beweis**
Es ist

$$\begin{aligned}
\gamma_k \exp(ikx_j) &= (a_k - ib_k)\exp(ikx_j) \\
&= (a_k - ib_k)(\cos(kx_j) + i\sin(kx_j)) \\
&= a_k \cos(kx_j) + b_k \sin(kx_j) + i(a_k \sin(kx_j) - b_k \cos(kx_j))
\end{aligned}$$

Damit kann $t_N(x_j)$ dargestellt werden in der Form

$$t_N(x_j) = \frac{a_0}{2} + \sum_{k=1}^{N} Re\,(\gamma_k \exp(ikx_j)) = \frac{a_0}{2} + Re\sum_{k=1}^{N}\gamma_k \exp(ikx_j)$$

$$= \frac{a_0}{2} + \frac{1}{2}\Big(\sum_{k=1}^{N}\gamma_k \exp(ikx_j) + \sum_{k=1}^{N}\overline{\gamma}_k \exp(-ikx_j)\Big)$$

und wegen $\exp(iNx_j) = 1$

$$= \frac{a_0}{2} + \frac{1}{2}\Big(\sum_{k=1}^{N}\gamma_k \exp(ikx_j) + \exp(iNx_j)\sum_{k=1}^{N}\overline{\gamma}_k \exp(-ikx_j)\Big)$$

$$= \frac{a_0}{2} + \frac{1}{2}\Big(\sum_{k=1}^{N}\gamma_k \exp(ikx_j) + \sum_{k=1}^{N}\overline{\gamma}_k \exp(i(N-k)x_j)\Big)$$

$$= \frac{a_0}{2} + \sum_{k=1}^{N-1}\frac{1}{2}(\gamma_k + \overline{\gamma}_{N-k})\exp(ikx_j) + \frac{1}{2}(\gamma_N + \overline{\gamma}_N)$$

$$= \frac{a_0}{2} + \sum_{k=1}^{N-1}\frac{1}{2}(\gamma_k + \overline{\gamma}_{N-k})\exp(ikx_j) + a_N$$

$$= c_0 + \sum_{k=1}^{N-1} c_k\,\omega_N^{jk}$$

**Bemerkung**
**Ist $N = 2^n$, dann zeigt dieser Satz, dass die Werte eines reellen trigonometrischen Polynoms an den Stellen $x_j$ durch eine FFT bestimmt werden können.**

**Beispiel**
Gegeben sei die „Sägezahnfunktion“

$$f(x) = \begin{cases} x/\pi & \text{für} \quad 0 \le x < \pi \\ 0 & \text{für} \quad x = \pi \\ -2 + x/\pi & \text{für} \quad \pi < x \le 2\pi \end{cases}$$

Die Fourierreihe von f hat bekanntlich die Gestalt

$$f(x) = \frac{2}{\pi}\Big(\sin x - \frac{1}{2}\sin 2x + \frac{1}{3}\sin 3x - \frac{1}{4}\sin 4x + \cdots\Big) = \frac{2}{\pi}\sum_{k=1}^{\infty}(-1)^{k+1}\frac{1}{k}\sin kx\,.$$

Bestimmt werden soll für $N = 2^n$ ( n=4,5,8)

$$t_N(x_j) = \frac{2}{\pi}\sum_{k=1}^{N}(-1)^{k+1}\frac{1}{k}\sin kx_j\,, \quad x_j = 2\pi j/N\,, \quad j = 0,1,2, \ldots ,N-1.$$

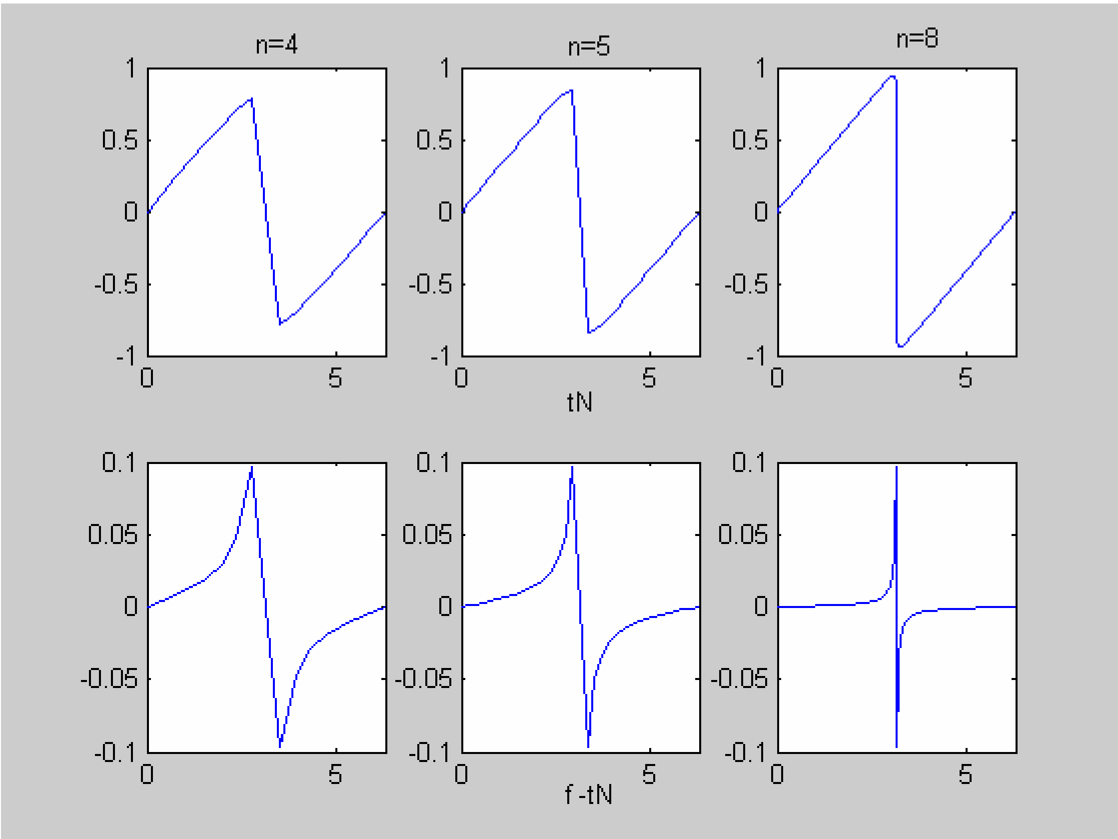
Die Koeffizienten des zugehörigen komplexen Ausdruckes

$$t_N(x_j) \;=\; \sum_{k=0}^{N-1} c_k \omega_N^{jk}\,, \qquad\qquad j = 0,1,2, \ldots\ldots , N\text{-}1,$$

sind $\left(\gamma_k = \dfrac{2i}{\pi}(-1)^k(1/k)\right)$

$$c_0 = 0 \quad \text{und} \quad c_k = \frac{i}{\pi}\{(-1)^k(1/k) - (-1)^{N-k}(1/(N-k))\}, \qquad k = 1,2,3, \ldots , N\text{-}1.$$

Die FFT liefert




**Bemerkungen**
Ein zugehöriges Programm finden Sie im Anhang dieses Textes.
Zu beachten ist die hohe Anzahl ( $N = 2^8$) der benötigten Stützstellen für eine „gute“ Approximation von f.
Beim Fehler f-tN kann das Gibb’sche Phänomen oder das grundsätzliche Fehlverhalten in der Umgebung der Unstetigkeit von f beobachtet werden.

## § 9.4 Interpolation mit reellen trigonometrischen Polynomen

Leider muss bei der Interpolationsaufgabe mit reellen trigonometrischen Polynomen eine Fallunterscheidung vorgenommen werden. Die Fälle richten sich danach, ob die Anzahl der Interpolationspunkte gerade oder ungerade ist.

Gegeben seien erneut die N Stützstellen

$$x_j = 2\pi j / N , \qquad j = 0,1,2, \dots , N-1,$$

und zugehörige reelle Werte $\tau_j \in \mathbf{R}$.

Ist N gerade mit N =2M, dann hat die Interpolationsaufgabe

$$\tau_j = \frac{a_0}{2} + \sum_{k=1}^{M-1} (a_k \cos(kx_j) + b_k \sin(kx_j)) + \frac{a_M}{2} \cos(Mx_j), \qquad j = 0,1,2, \dots\dots , N-1,$$

eine eindeutig bestimmte Lösung.
Ist N ungerade mit N= 2M+1, dann hat die Interpolationsaufgabe oder die Aufgabe, reelle Koeffizienten $a_0, a_1, \dots , a_M, b_1, b_2, \dots , b_M$ zu finden mit

$$\tau_j = \frac{a_0}{2} + \sum_{k=1}^{M} a_k \cos(kx_j) + b_k \sin(kx_j), \qquad j = 0,1,2, \dots\dots , N-1,$$

eine eindeutig bestimmte Lösung.

**Bemerkung**
Die - vielleicht nahe liegende - Auffassung, dass für gerade N=2M auch die Interpolations-aufgabe

$$\tau_j = \frac{a_0}{2} + \sum_{k=1}^{M-1} (a_k \cos(kx_j) + b_k \sin(kx_j)) + \frac{b_M}{2} \sin(Mx_j), \qquad j = 0,1,2, \dots\dots , N-1,$$

zum Erfolg führen muss, ist nicht richtig.
Dieses zeigt das einfache Beispiel mit N=4. Die zugehörige Interpolationsaufgabe führt auf das Gleichungssystem

$$\begin{pmatrix} 0.5 & \cos(0) & \sin(0) & \sin(0) \\ 0.5 & \cos(2\pi/4) & \sin(2\pi/4) & \sin(\pi) \\ 0.5 & \cos(\pi) & \sin(\pi) & \sin(2\pi) \\ 0.5 & \cos(6\pi/4) & \sin(6\pi/4) & \sin(3\pi) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} \qquad \Leftrightarrow \qquad Ax = b$$

mit der singulären Matrix

$$A = \begin{pmatrix} 0.5 & 1 & 0 & 0 \\ 0.5 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0.5 & -1 & 0 & 0 \\ 0.5 & \sqrt{2}/2 & \sqrt{2}/2 & 0 \end{pmatrix}.$$

**Satz**
Gegeben sei die gerade Zahl N=2M und Punkte $(x_j, \tau_j)$, $j = 0,1,2, \dots ,N-1$, mit $x_j = 2\pi j/N$ und $\tau_j \in \mathbf{R}$.

Außerdem seien

$$a_j = \frac{2}{N}\sum_{k=0}^{N-1}\tau_k \cos(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots , M$$

und

$$b_j = \frac{2}{N}\sum_{k=0}^{N-1}\tau_k \sin(kx_j) \qquad \text{für} \quad j = 1,2, \ldots , M\text{-}1,$$

dann ist

$$\tau_j = \frac{a_0}{2} + \sum_{k=1}^{M-1} a_k \cos(kx_j) + b_k \sin(kx_j) + \frac{a_M}{2}\cos(Mx_j) \quad \text{für} \quad j = 0,1,2,3, \ldots ,N\text{-}1.$$

**Beweis**
Der Satz aus Abschnitt 2 zeigte, dass die komplexe Interpolationsaufgabe

$$\tau_j = \sum_{k=0}^{N-1} c_k z_j^k \quad , \qquad j = 0,1,2, \ldots , N\text{-}1,$$

für die Punkte $(z_j, \tau_j)$, $j = 0,1,2, \ldots ,N\text{-}1$, $z_j = \exp(i2\pi j/N)$ und $\tau_j \in \mathbf{R}$, durch die Koeffizienten

$$c_j = \frac{1}{N}\sum_{k=0}^{N-1}\tau_k \exp(-ikx_j), \qquad j = 0,1,2, \ldots , N\text{-}1$$

gelöst wird.
Die im Satz angegebenen Koeffizienten $a_j$, $b_j$ werden jetzt ergänzt zu

$$a_j = \frac{2}{N}\sum_{k=0}^{N-1}\tau_k \cos(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots , N\text{-}1$$

und

$$b_j = \frac{2}{N}\sum_{k=0}^{N-1}\tau_k \sin(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots , N\text{-}1.$$

Es sei

$$c_j = \frac{a_j - ib_j}{2} \qquad \text{für} \quad j = 0,1,2, \ldots , N\text{-}1.$$

Damit ist

$$c_{N-j} = \frac{1}{N}\sum_{k=0}^{N-1}\tau_k \exp(-ik\frac{2\pi}{N}(N-j)) = \frac{1}{N}\sum_{k=0}^{N-1}\tau_k \exp(ik\frac{2\pi}{N}j)) \quad \text{für} \quad j = 1,2, \ldots , N\text{-}1$$

und mithin

$$c_{N-j} = \frac{a_j + ib_j}{2} = \overline{c}_j \qquad \text{für} \quad j = 1,2,3, \ldots , N\text{-}1.$$

Die Ausnutzung der Symmetrien der Funktionen Cosinus und Sinus liefert

$$c_j \exp(\frac{i2\pi}{N}kj) + c_{N-j} \exp(\frac{i2\pi}{N}k(N-j)) \; = \; a_j \cos jx_k + b_j \sin jx_k \quad \text{für} \quad j = 1,2,3, \ldots, N\text{-}1.$$

Mit Hilfe der letzten Gleichungen kann jedes $\tau_j$ dargestellt werden in der Form

$$\tau_j = \sum_{k=0}^{N-1} c_k \exp(ikx_j) \; = \; c_0 + \sum_{k=1}^{N-1} c_k \exp(ikx_j)$$

$$= c_0 + \sum_{k=1}^{M-1} (c_k \exp(ikx_j) + c_{N-k} \exp(i(N-k)x_j)) + c_M \exp(iMx_j)$$

$$= c_0 + \sum_{k=1}^{M-1} (a_k \cos(kx_j) + b_k \sin(kx_j)) \; + \; c_M \exp(iMx_j).$$

Wegen

$$\exp(iMx_j) = 1,$$

$$c_0 = \frac{a_0 - ib_0}{2} = \frac{a_0}{2} \quad \text{und}$$

$$c_M = \frac{a_M - ib_M}{2} = \frac{a_M}{2}$$

folgt die Behauptung des Satzes.

Ein entsprechender Satz kann für den Fall $N = 2M+1$ formuliert und bewiesen werden. Dieses wird dem Leser überlassen.

**Wir kommen nun zur eigentlichen Aufgabe, nämlich die Berechnung der Koeffizienten eines reellen trigonometrischen Interpolationspolynoms:**

Aus dem Satz folgt, dass die Koeffizienten eines reellen trigonometrischen Interpolationspolynoms für die $N = 2^n$ Punkte $(x_j, \tau_j)$ gegeben sind durch

$$a_j = \frac{2}{N} \sum_{k=0}^{N-1} \tau_k \cos(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots, N/2$$

und

$$b_j = \frac{2}{N} \sum_{k=0}^{N-1} \tau_k \sin(kx_j) \qquad \text{für} \quad j = 1,2, \ldots, N/2\text{-}1.$$

Für eine numerisch effiziente Ermittlung dieser Koeffizienten werden hier obige Gleichungen ergänzt zu

$$a_j = \frac{2}{N} \sum_{k=0}^{N-1} \tau_k \cos(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots, N\text{-}1$$

und

$$b_j = \frac{2}{N} \sum_{k=0}^{N-1} \tau_k \sin(kx_j) \qquad \text{für} \quad j = 0,1,2, \ldots, N\text{-}1.$$

Mit dem Ausdruck

$$c_j = \frac{a_j - ib_j}{2} \qquad \text{für} \quad j = 0,1,2, \ldots, N-1$$

ergibt sich die Gleichung

$$c_j = \frac{1}{N}\sum_{k=0}^{N-1} \tau_k \exp(-ikx_j), \qquad j = 0,1,2, \ldots, N-1.$$

Die Koeffizienten $c_j$ werden mit der IFFT ermittelt und aus diesen die benötigten Koeffizienten $a_j$ und $b_j$ entnommen.

**Bemerkung**
Die angegebene Methode zur Ermittlung der gesuchten Koeffizienten erfordert hier eine Ergänzung von Gleichungen. Dieses kann vermieden werden (*Siehe: Approximation SS94 C. Geiger, K. Glashoff, Universität Hamburg)*.

**Beispiel**
Gegeben sei die „Dachfunktion“

$$f(x) = \begin{cases} x & \text{für} \quad 0 \le x < \pi/2 \\ \pi - x & \text{für} \quad \pi/2 \le x < 3\pi/2 \\ x - 2\pi & \text{für} \quad 3\pi/2 \le x \le 2\pi \end{cases} .$$

Für $N = 8 = 2^3$ und $N = 16 = 2^4$ werden die zugehörigen reellen trigonometrischen Interpolationspolynome $T_N(x)$ mit den Interpolationspunkten

$$(x_j, f(x_j)), \qquad x_j = 2\pi j / N, \qquad j = 0,1,2, \ldots, 15,$$

gesucht.
Die entstehenden reellen trigonometrischen Interpolationspolynome sollen nun an den ( $p = 32$ bzw. $p = 64$ ) Punkten $(y_j, T_N(y_j))$, $y_j = 2\pi j / p$, $j = 0,1,2, \ldots, p$, bestimmt werden und ein Vergleich mit f(x) angestellt werden.

Die Fourierreihe von f hat bekanntlich die Gestalt

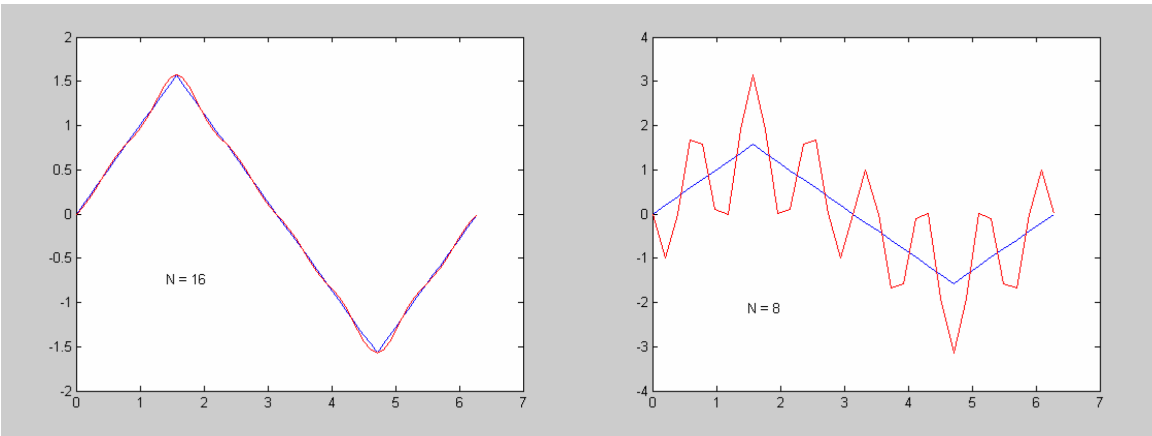
$$f(x) = \frac{4}{\pi}\left(\sin x - \frac{1}{3^2}\sin 3x + \frac{1}{5^2}\sin 5x - \cdots\right) = \frac{4}{\pi}\sum_{k=0}^{\infty}(-1)^k \frac{1}{(2k+1)^2}\sin(2k+1)$$

Das Programm aus dem Anhang liefert das folgende Ergebnis, wobei nur die grau unterlegten Werte von Bedeutung sind:

| | N = 16 | N=8 |
|---|---|---|
| $c_0$ | 0 | 0 |
| $c_1$ | -0.0000 - 0.6449i | 0 - 0.6704i |
| $c_2$ | -0.0000 - 0.0000i | 0 |
| $c_3$ | 0 +0.0795i | 0 + 0.1150i |
| $c_4$ | 0 - 0.0000i | 0 |
| $c_5$ | 0 - 0.0355i | 0 - 0.1150i |
| $c_6$ | 0.0000 - 0.0000i | 0 |
| $c_7$ | 0.0000 + 0.0255i | 0 +0.6704i |
| $c_8$ | -0.0000 | |
| | 0.0000 - 0.0255i | |
| | 0.0000 + 0.0000i | |
| | 0 + 0.0355i | |
| | 0 + 0.0000i | |
| | 0 - 0.0795i | |
| | -0.0000 + 0.0000i | |
| | -0.0000 + 0.6449i | |
| | -0.0000 + 0.0000i | |

Die zugehörigen (nicht trivialen) Koeffizienten können aus der folgenden Tabelle entnommen werden (man beachte den Faktor -2):

| | $b_1$ | $b_3$ | $b_5$ | $b_7$ |
|---|---|---|---|---|
| f(x) | 1.2732 | -0.1415 | 0.0509 | 0.0260 |
| N=8 | 1.3408 | -0.2300 | | |
| N=16 | 1.2897 | -0.1590 | 0.0710 | -0.0510 |




## § 9.5 Approximationsaufgaben mit trigonometrischen Polynomen

Einige Fragen bei den Approximationsaufgaben mit trigonometrischen Polynomen können leicht mit dem sehr allgemeinen Projektionssatz beantwortet werden:

**Projektionssatz**
Es sei $(X,(.,.))$ ein unitärer Raum über **R**, $V \subset X$ ein linearer Teilraum, $b \in X$ und $b \notin V$.
Dann gilt

1. Ist die Dimension von V endlich, dann gibt es eine beste Approximation $v^*$ für b in V, d.h. ein Element $v^* \in V$ mit der Eigenschaft

$$\| b-v^* \| \leq \| b-v \| \quad \text{für alle } v \in V.$$

2. Wenn es eine beste Approximation gibt, dann ist diese eindeutig bestimmt.
3. $v^*$ ist die beste Approximation aus V für b genau dann, wenn

$$(b-v^*,v) = 0 \quad \text{für alle } v \in V.$$

**Beweis**
Ist $v^*$ eine beste Approximation aus V für b, dann ist

$$\| v^*-b \| \leq \| 0-b \| \leq \| b \|.$$

Damit ist $\| v^* \| \leq 2\| b \|$ und eine beste Approximation muss in der Kugel

$$K = \{ v \mid v \in V , \ \| v \| \leq 2\| b \| \}$$

liegen.
Da V endlich dimensional ist, ist K eine kompakte Menge und die stetige Funktion

$$g(v) = \| v-b \|$$

nimmt auf K ihr Minimum an.

Es soll nun der Punkt 3 bewiesen werden und damit auch der Punkt 2:

$\Rightarrow$ Es sei $v^*$ eine beste Approximation. Angenommen, es gäbe ein $\tilde{v} \in V$

$$\text{mit } (b-v^*, \tilde{v}) = \delta \neq 0 .$$

Dann ist $\tilde{v} \neq 0$ und für $g = v^* + \delta \frac{\tilde{v}}{\| \tilde{v} \|^2} \in V$ gilt

$$( b-g, b-g) = ( b - v^* - \delta \frac{\tilde{v}}{\| \tilde{v} \|^2} , b - v^* - \delta \frac{\tilde{v}}{\| \tilde{v} \|^2} )$$

$$= ( b-v^*, b-v^*) - 2 \frac{\delta}{\| \tilde{v} \|^2} (b - v^*, \tilde{v}) + \frac{\delta^2}{\| \tilde{v} \|^4} (\tilde{v}, \tilde{v})$$

$$= ( b-v^*, b-v^*) - \frac{\delta^2}{\| \tilde{v} \|^2} (b - v^*, \tilde{v}) < \| b - v^* \|^2$$

und folglich kann $v^*$ keine beste Approximation sein.

$\Leftarrow$ Es sei $(b-v^*,v) = 0$ für alle $v \in V$. Für beliebiges $v \in V$ gilt dann

$$
\begin{aligned}
\| \, b-v \, \|^2 &= ( b-v^*+v^*-v \, , \, b-v^*+v^*-v ) \\
&= ( b-v^* \, , \, b-v^* ) + 2( b-v^* \, , \, v^*-v ) + ( v^*-v \, , \, v^*-v ) \\
&= \| \, b-v^* \, \|^2 + \| \, v-v^* \, \|^2 \; \geq \; \| \, b-v^* \, \|^2,
\end{aligned}
$$

folglich ist $v^*$ die beste Approximation und auch die einzige (!).

Ist nun V von endlicher Dimension und $\{v_1, v_2, v_3, \dots, v_n\}$ eine Basis von V, dann kann die beste Approximation

$$v^* = \sum_{i=1}^{n} a_i v_i$$

für ein $b \in X$ und $b \notin V$ direkt berechnet werden:
Die dritte Bedingung aus dem Projektionssatz ist äquivalent zu

$$( b-\sum_{i=1}^{n} a_i v_i \, , \, v_j) = 0 \qquad \text{für alle} \quad j = 1,2,3, \dots ,n$$

und die Koeffizienten $a_i$ ergeben sich dann aus dem linearen Gleichungssystem

$$\sum_{i=1}^{n} a_i (v_i, v_j) = (b, v_j) \,, \qquad j = 1,2,3, \dots , n.$$

Das Gleichungssystem vereinfacht sich wesentlich, wenn die angegebene Basis von V ein orthogonales System ist, d.h.

$$(v_i,v_j) = 0 \qquad \text{für alle } i,j \text{ mit } i \neq j.$$

Ist die Basis $\{v_1, v_2, v_3, \dots, v_n\}$ von V ein orthogonales System, dann ist

$$a_i = \frac{(b, v_i)}{(v_i, v_i)} \qquad \text{für} \quad i = 1,2,3, \dots ,n.$$

Ist die Basis sogar orthonormiert, d.h. es gilt

$$(v_i,v_j) = \begin{cases} 1 & \text{für} \quad i = j \\ 0 & \text{für} \quad i \neq j \end{cases},$$

dann kann die beste Approximation geschrieben werden in der Form

$$v^* = \sum_{i=1}^{n} (b, v_i) v_i \, .$$

Es sei $(X,(.,.))$ der Raum $C_{2\pi}$ der $2\pi-$periodischen, reellwertigen und quadratisch integrierbaren Funktionen mit

$$(f,g) = \frac{1}{\pi} \int_0^{2\pi} f(x) g(x) dx$$

und V die Menge der trigonometrischen Summen mit der Basis (dim V = N=2M)

$$\{1, \cos x, \sin x, \cos 2x, \sin 2x, \dots, \cos(M-1)x, \sin(M-1)x, \cos Mx\}.$$

Diese Basis ist eine Orthogonalbasis (siehe z.B. Schwarz-Köckler, Numerische Mathematik, 2004)

Die beste Approximation

$$\| f - v^* \|_2 \le \| f - v \|_2 \qquad \text{für alle } v \in V$$

mit $$v^*(x) = S_n(f)(x) = \frac{a_0}{2} + \sum_{k=1}^{M-1} (a_k \cos kx + b_k \sin kx) + \frac{a_M}{2} \cos(Mx)$$

hat die „Fourier-Koeffizienten“

$$a_0 = \frac{1}{\pi} \int_0^{2\pi} f(x)dx,$$

$$a_j = \frac{1}{\pi} \int_0^{2\pi} f(x)\cos(jx)dx, \quad j = 1,2, \dots, M,$$

$$b_j = \frac{1}{\pi} \int_0^{2\pi} f(x)\sin(jx)dx, \quad j = 1,2, \dots, M-1.$$

Die Bestimmung der Fourier-Koeffizienten erfordert die Berechnung von Integralen.

Obwohl es vielleicht auf den ersten Blick erstaunlich ist, kann es i.A. zweckmäßig sein, die Fourier-Koeffizienten näherungsweise mit der N-mal wiederholten Trapezregel zu berechnen:

$$\tilde{a}_0 = \frac{2}{N} \sum_{k=0}^{N-1} f(\frac{2\pi}{N}k),$$
$$\tilde{a}_j = \frac{2}{N} \sum_{k=1}^{N-1} f(\frac{2\pi}{N}k)\cos(\frac{2\pi}{N}jk), \quad j = 1,2, \dots, M,$$
$$\tilde{b}_j = \frac{2}{N} \sum_{k=1}^{N-1} f(\frac{2\pi}{N}k)\sin(\frac{2\pi}{N}jk), \quad j = 1,2, \dots, M-1.$$

Vorteile dieser Wahl sind:

1. Die Näherungen $\tilde{a}_j, \tilde{b}_j$ für die Fourier-Koeffizienten $a_j$, $b_j$ können mit Hilfe der FFT berechnet werden.
2. Mit den $\tilde{a}_j, \tilde{b}_j$ wird die Interpolationsaufgabe

$$f(x_j) = \frac{\tilde{a}_0}{2} + \sum_{k=1}^{M-1} (\tilde{a}_k \cos(kx_j) + \tilde{b}_k \sin(kx_j)) + \frac{\tilde{a}_M}{2} \cos(Mx_j),$$

$$x_j = \frac{2\pi}{N} j, \qquad j = 0,1,2, ... ,N-1,$$

gelöst.

3. Die Abschnitte

$$\frac{\tilde{a}_0}{2} + \sum_{k=1}^{m-1} (\tilde{a}_k \cos(kx_j) + \tilde{b}_k \sin(kx_j)), \qquad 1 \le m \le M,$$

lösen eine analoge „Diskrete Approximationsaufgabe“.

4. Das beim Romberg-Verfahren genutzte günstige „Fehlergesetz“ der Trapezregel.

Einige der genannten Vorteile sollen nun näher erläutert werden:

1. (Anwendung der FFT)
Werden die Näherungen $\tilde{a}_j, \tilde{b}_j$ ergänzt zu

$$\tilde{a}_j = \frac{2}{N} \sum_{k=1}^{N-1} f(\frac{2\pi}{N} k) \cos(\frac{2\pi}{N} jk), \qquad j = 0,1, 2, ... , N-1,$$
$$\tilde{b}_j = \frac{2}{N} \sum_{k=1}^{N-1} f(\frac{2\pi}{N} k) \sin(\frac{2\pi}{N} jk), \qquad j = 0, 1,2, ... , N-1,$$

und wird

$$c_j := \frac{\tilde{a}_j + \tilde{b}_j}{2}, \quad j = 0\ 1,2, .... , N-1$$

gesetzt, dann liefert die FFT die gesuchten Werte.

<u>Beispiel zu 1</u>
Gegeben sei die stückweise konstante Funktion

$$f(x) = \operatorname{sign}(\sin(x))$$

Die Fourierreihe von f hat die Gestalt

$$f(x) = \frac{4}{\pi}(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x + \cdots) = \frac{4}{\pi}\sum_{k=0}^{\infty} \frac{1}{2k+1}\sin((2k+1)x).$$

Es sollen die Näherungen $\tilde{a}_j, \tilde{b}_j$ für die Fourier-Koeffizienten $a_j$, $b_j$, $0 \le j \le 32$, mit Hilfe der FFT berechnet werden.
Das Programm aus dem Anhang liefert die folgende Tabelle von Werten: In der zweiten
Spalte stehen die nichttrivialen Koeffizienten $b_{2k+1}$ und in der ersten Spalte die Näherungen $\tilde{b}_{2k+1}$, k = 0,1,2, … ,15. Alle sonstigen Koeffizienten sind gleich Null.

| | |
|---|---|
| 1.2722 | 1.2732 |
| 0.4213 | 0.4244 |
| 0.2495 | 0.2546 |
| 0.1747 | 0.1819 |
| 0.1321 | 0.1415 |
| 0.1043 | 0.1157 |
| 0.0843 | 0.0979 |
| 0.0690 | 0.0849 |
| 0.0566 | 0.0749 |
| 0.0464 | 0.0670 |
| 0.0375 | 0.0606 |
| 0.0296 | 0.0554 |
| 0.0224 | 0.0509 |
| 0.0157 | 0.0472 |
| 0.0093 | 0.0439 |
| 0.0031 | 0.0411 |

2. (Interpolationsaufgabe)
   Dieser Punkt wurde in Abschnitt 4 ausführlich behandelt.

3. (Analoge „Diskrete Approximationsaufgabe“)
   Der Projektionssatz und die anschließenden Bemerkungen zeigten, dass für ein $f \in X$, $f \notin V$, die beste Approximation für f in V mit Hilfe der Fourier-Koeffizienten von f angegeben werden konnte. Die folgenden Überlegungen zeigen, dass durch die Zahlen $\tilde{a}_j, \tilde{b}_j$ eine beste Approximation für f in einem zu X verwandten unitären Raum angegeben werden kann.
   Es sei $(X_B, (.,.))$ der Raum der reellwertigen Funktionen $C_{2\pi}(B)$ mit

$$B = \{ \frac{2\pi}{N}k \ / \ k = 0,1,2, \dots, N\text{-}1 \}$$

und für $f,g \in C_{2\pi}(B)$

$$(f,g) = \frac{2}{N}\sum_{k=0}^{N-1} f(\frac{2\pi}{N}k)g(\frac{2\pi}{N}k).$$

Es sei $V_B$ der lineare Raum der Funktionen (Vektoren) mit Basis

$$\{ 1, \cos(x), \sin(x), \cos(2x), \sin(2x) \dots, \cos(mx), \sin(mx) \}, \quad 1 \le m \le M = N/2, \quad x \in B.$$

Die genannten Basisvektoren von $V_B$ bilden ein Orthogonalsystem (beachten Sie dazu das Lemma aus Abschnitt 2).

Der Projektionssatz und die anschließenden Bemerkungen zeigen, dass für ein $f \in X_B$, $f \notin V_B$, die beste Approximation $v^*$ für f in $V_B^m$ durch die Koeffizienten

$$\widetilde{a}_j = ( f, \cos(jx))$$

und

$$\widetilde{b}_j = ( f, \sin(jx))$$

gegeben ist.
Es gilt also

$$v_m^*(x) = \frac{\widetilde{a}_0}{2} + \sum_{k=1}^{m} (\widetilde{a}_k \cos(kx) + \widetilde{b}_k \sin(kx)), \quad 1 \le m \le M$$

und $x \in B$.

<u>Beispiel zu 3</u>
Wir betrachten erneut die stückweise konstante Funktion

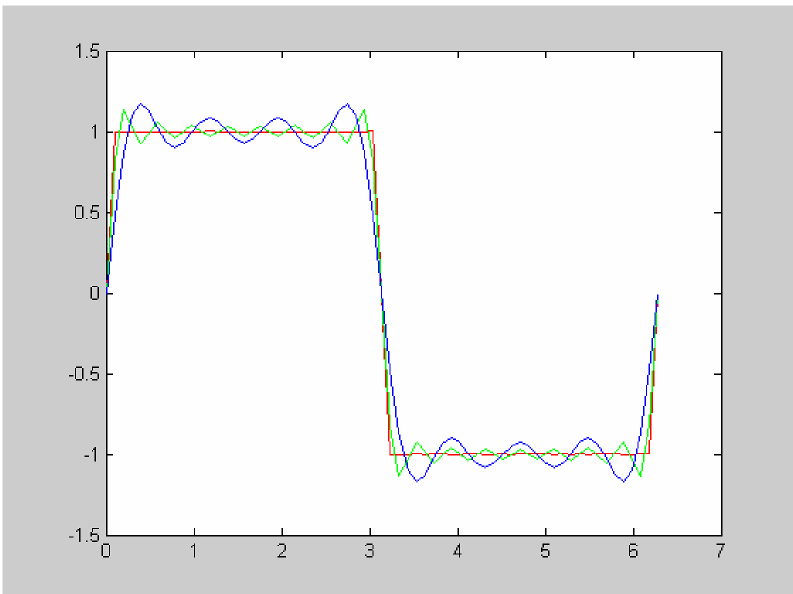
$$f(x) = \operatorname{sign}(\sin(x))$$

mit der Fourierreihe

$$f(x) = \frac{4}{\pi}(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x + \cdots ) = \frac{4}{\pi}\sum_{k=0}^{\infty}\frac{1}{2k+1}\sin((2k+1)x) .$$

Gesucht sind die besten Approximationen von $f \in C_{2\pi}(B)$, für $m = 7,15, 31$, in $V_B^m$.
Die benötigten Koeffizienten $\widetilde{b}_j$ können aus der obigen Tabelle entnommen werden.
Die zugehörigen Näherungen sind im folgenden Bild wiedergegeben:

$m = 7$ (blau)
$m = 15$ (grün)
$m = 31$ (rot)



# Anhang

## Programm zum Abschnitt 3

```
clear all
j=0;
for n = [4,5,8];
j=j+1;
N = 2^n;
x = (2*pi/N)*(1:N)-2*pi/N;
f = x/pi;                                   % Funktion
f = f+(x>pi)*(-2);
f(N+1)=0;
x(N+1)=2*pi;
f(N/2+1)=0;
c(1) = 0;                                   % Koeffizienten
for k = 2:N
   c(k) = (i/pi)*((1/(k-1))*(-1)^(k-1)-(1/(N-(k-1)))*(-1)^(N-k+1));
end
z=sft(c);                                   % FFT
tN = bitrevorder(z)                         % Werte
tN(N+1) = tN(1)
subplot(2,3,j)                              % Zeichnungen
plot(x,tN),axis([0 2*pi -1 1])
subplot(2,3,j+3)
plot(x,f-tN),axis([0 2*pi -.1 .1])
end
```

## Programm zum Abschnitt 4

```
clear all
N = 16;
x = 0:(2*pi)/N:2*pi
f = x;                                                        % Funktion
f = f-(x>pi/2).*(2*x-pi);
f = f-(x>3*pi/2).*(-2*x+3*pi);
t = f(1:N);
z = ifft(t)'                                                  % IFFT
b = -2*imag([z(2) z(4) z(6) z(8)])                            % Koeffizienten
x = 0:(2*pi)/(4*N):2*pi;
f = x;
f = f-(x>pi/2).*(2*x-pi);
f = f-(x>3*pi/2).*(-2*x+3*pi);
TN = b(1)*sin(x)+b(2)*sin(3*x)+b(3)*sin(5*x)+b(4)*sin(7*x);    %  Druck
%plot(x,f)
plot(x, TN,'r')
```

**Programme zum Abschnitt 5**

```
clear all
m=64
x = (0:1/m:1-1/m)*2*pi
f = sign(sin(x))                % Funktion
f(1) = 0
f (m/2+1)=0
t =(2/m)*f
z = sft(t)'                     % FFT
c = bitrevorder(z)              % Komplexen Resultate
for i=2:2:m/2
   fk(i)= (4/pi)*(1/(i-1))      % Fourierkoeffizienten
end
b = -imag(c(2:2:m/2))'          % Nichttrivialen Koeffizienten der Approximation
v = fk(2:2:m/2)
d = [b;v]'
```

```
clear all
m=64
x = (0:1/m:1-1/m)*2*pi
f =sign(sin(x))
f(1) = 0
f(m/2+1) = 0
t = (2/m)*f
c = fft(t)'
```

```
b = imag(c)
f(m+1)= 0
x(m+1) =2*pi
y(1:m+1)=0
for i = 1:4
   y = y +b(2*i)*sin((2*i-1)*x)          %So sollte man es nicht machen
end                                      %Machen Sie es sachgerechter
plot(x,y,'b')
hold on
```