Kapitel 5

Interpolation

5.1 Interpolation durch Polynome

Sei $\mathcal{P}_n$die Menge der Polynome mit komplexen Koeffizienten vom Grade $\leq n$. $\mathcal{P}_n$ besteht also aus den Ausdrücken der Form

$$\sum_{k=0}^{n} a_n x^k \quad , \quad a_n \in \mathbb{C} \; .$$

Als Polynominterpolation bezeichnet man folgende Aufgabe: Gegeben seien $n+1$ "Stützstellen" $x_0, \ldots, x_n \in \mathbb{C}$ und ebensoviel "Stützwerte" $y_0, \ldots, y_n \in \mathbb{C}$. Gesucht ist $p \in \mathcal{P}_n$ mit $p(x_j) = y_j$, $j = 0, \ldots, n$.

Satz 5.1.1 *p ist eindeutig bestimmt, falls die x_j paarweise verschieden sind.*

Beweis: Das Interpolationsproblems ist äquivalent dem linearen Gleichungssystem

$$\sum_{k=0}^{n} a_k x_j^k = y_j \quad , \quad j = 0, \ldots, n \tag{1.1}$$

für $a_0, \ldots, a_n$. Wir zeigen, daß dieses eindeutig lösbar ist. Dazu ist hinreichend, daß das zugehörige homogene System, also das System mit $y_0 = y_1 = \cdots = y_n = 0$, nur die Lösung $a_0 = a_1 = \cdots a_n = 0$ hat. Dies ist aber der Fall, weil ein Polynom vom Grade $\leq n$ nicht mehr als $n+1$ paarweise verschiedene Nullstellen haben kann.

$\square$

Zur Berechnung des interpolierenden Polynoms könnte man das lineare Gleichungssystem (1.1) lösen, etwa durch die Cramer'sche Regel. Die Determinante von (1.1) ist die Vandermond'sche Determinante

$$V = \begin{vmatrix} 1 & \cdots & 1 \\ x_0 & & x_n \\ \vdots & & \vdots \\ x_0^n & & x_n^n \end{vmatrix} = \prod_{i>k}(x_i - x_k) .$$

Diese ist also $\neq 0$, falls die x_j paarweise verschieden sind, was einen neuen Beweis von Satz 1 liefert. Wir ersetzen in V die k-te Spalte durch die Zahlen $y_0, \ldots, y_n$ und bezeichnen das Resultat mit V_k. Die Lösung des Interpolationsproblems ist dann

$$p(x) = \sum_{k=0}^{n} a_k x^k \quad , \quad a_k = \frac{V_k}{V} , \quad k = 0, \ldots, n .$$

Für das praktische Rechnen ist diese Lösung völlig ungeeignet. Wir werden drei sehr viel praktischere Darstellungen von p finden.

1. Die Form von Lagrange

Man setzt

$$\omega_j(x) = \prod_{i=0,\, i\neq j}^{n} \frac{x - x_i}{x_j - x_i} \quad , \quad j = 0, \ldots, n .$$

Es ist $\omega_j \in \mathcal{P}_n$ Lösung des speziellen Interpolationsproblems

$$\omega_j(x_k) = \begin{cases} 1 & , \quad k = j , \\ 0 & , \quad \text{sonst} . \end{cases}$$

Für das gesuchte Polynom gilt dann

$$p(x) = \sum_{j=0}^{n} y_j \omega_j(x) .$$

Beispiel: $n = 2$. Stützstellen und Stützwerte seien

j	x_j	y_j
0	0	1
1	1	3
2	3	2

Wir berechnen

$$\begin{aligned}
\omega_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} &= \tfrac{1}{3}(x-1)(x-3)\\
\omega_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} &= -\tfrac{1}{2}x(x-3)\\
\omega_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} &= \tfrac{1}{6}x(x-1)
\end{aligned}$$

und erhalten

$$p(x) = \frac{1}{3}(x-1)(x-3) - \frac{3}{2}x(x-3) + \frac{1}{3}x(x-1) = -\frac{5}{6}x^2 + \frac{17}{6}x + 1 \ .$$

2. Die Rekursionsformel von Neville

Wir bezeichnen mit $p_{i,\ldots,k}$ das nach Satz 1 eindeutig bestimmte Polynom $\in \mathcal{P}_{k-i}$, welches $y_i, \ldots, y_k$ an den Stellen $x_i, \ldots, x_k$ interpoliert. Das gesuchte Polynom ist dann $p = p_{0,\ldots,n}$.

Satz 5.1.2 *Es ist*

$$p_{i,\ldots,k}(x) = \frac{1}{x_k - x_i}\ \left((x-x_i)p_{i+1,\ldots,k}(x) + (x_k - x)p_{i,\ldots,k-1}(x)\right) \ .$$

Beweis: Auf der rechten Seite steht ein Polynom vom Grade $k-i$, das an den Stellen $x_i, \ldots, x_k$ die Werte $y_i, \ldots, y_k$ annimmt. Nach Satz 1 ist dieses eindeutig bestimmt und muß daher mit $p_{i,\ldots,k}$ übereinstimmen.

□

Die Neville'sche Formel erlaubt die Berechnung des Interpolationspolynoms p nach folgendem Schema ($n = 3$):

$$\begin{array}{ccccccc}
x_0 & y_0 = p_0 & & & & & \\
 & & p_{0,1} & & & & \\
x_1 & y_1 = p_1 & & p_{0,1,2} & & & \\
 & & p_{1,2} & & p_{0,1,2,3} = p & & \\
x_2 & y_2 = p_2 & & p_{1,2,3} & & & \\
 & & p_{2,3} & & & & \\
x_3 & y_3 = p_3 & & & & &
\end{array}$$

Bei Hinzunahme einer weiteren Stützstelle (und Erhöhung des Polynomgrads) wird das Schema einfach um eine Zeile erweitert, ohne daß die bereits berechneten Teile des Schemas neu berechnet werden müßten.

3. Die Newton'sche Form

Für $p = p_{0,\ldots,n}$ machen wir den Ansatz

$$\begin{aligned}
p_{0,\ldots,n}(x) \;=\; & A_0 + A_1(x - x_0) + A_2(x - x_0)(x - x_1) + \cdots \\
& + A_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})
\end{aligned}$$

mit noch zu bestimmenden Koeffizienten $A_0, \ldots, A_n$. Die Bedingungen $p(x_j) = y_j,\ \ j = 0, \ldots, n$ ergeben für die A_k das lineare Gleichungssystem

$$\begin{aligned}
A_0 &= y_0 \, , \\
A_0 + A_1(x_1 - x_0) &= y_1 \, , \\
A_0 + A_1(x_2 - x_0) + A_2(x_2 - x_0)(x_2 - x_1) &= y_2 \, , \\
\cdots & \\
A_0 + A_1(x_n - x_0) + \cdots + A_n(x_n - x_0)\cdots(x_n - x_{n-1}) &= y_n \, .
\end{aligned}$$

Die A_k können also durch Vorwärtseinsetzen berechnet werden. Für das Resultat gibt es eine elegante Darstellung durch "dividierte Differenzen" $[y_i, \ldots, y_k]$. Diese sind rekursiv definiert durch

$$\begin{aligned}
[y_i] &= y_i \, , \\
[y_i, \ldots, y_k] &= \frac{1}{x_k - x_i}([y_{i+1}, \ldots, y_k] - [y_i, \ldots, y_{k-1}]) \, .
\end{aligned}$$

Z.B. ist

$$\begin{aligned}
[y_0, y_1] &= \frac{y_1 - y_0}{x_1 - x_0} \, , \\
[y_0, y_1, y_2] &= \frac{1}{x_2 - x_0} \left(\frac{y_1 - y_2}{x_1 - x_2} - \frac{y_0 - y_1}{x_0 - x_1} \right) \, .
\end{aligned}$$

Man ordnet die dividierten Differenzen im "Differenzenschema" an:

$$\begin{array}{ccccc}
x_0 & [y_0] & & & \\
 & & [y_0, y_1] & & \\
x_1 & [y_1] & & [y_0, y_1, y_2] & \\
 & & [y_1, y_2] & & [y_0, y_1, y_2, y_3] \\
x_2 & [y_2] & & [y_1, y_2, y_3] & \\
 & & [y_2, y_3] & & \\
x_3 & [y_3] & & &
\end{array}$$

Satz 5.1.3 *Für die Koeffizienten A_i gilt*

$$A_i = [y_0, \ldots, y_i] \, , \qquad i = 0, \ldots, n \, .$$

Beweis: Wir zeigen

$$\begin{aligned}
p_{i,\ldots,k}(x) &= [y_i] + [y_i, y_{i+1}](x - x_i) + [y_i, y_{i+1}, y_{i+2}](x - x_i)(x - x_{i+1}) \\
&\quad + \cdots + [y_i, \ldots, y_k](x - x_i) \cdots (x - x_{k-1})
\end{aligned} \tag{1.2}$$

durch Induktion nach $m = k - i$. Für $m = 0$ ist (1.2) richtig. Sei (1.2) richtig für ein $m \geq 0$. Dann ist

$$\begin{aligned}
p_{i,\ldots,k}(x) &= [y_i, \ldots, y_k](x - x_i) \cdots (x - x_{k-1}) + q_1(x) \\
&= [y_i, \ldots, y_k] x^{k-i} + q_2(x)
\end{aligned}$$

mit $q_\ell \in \mathcal{P}_{k-i-1}$, und ebenso gilt

$$p_{i+1,\ldots,k+1}(x) = [y_{i+1}, \ldots, y_{k+1}] x^{k-i} + q_3(x) \, .$$

Nach Satz 2 ist mit $r_\ell \in \mathcal{P}_{k-i}$

$$\begin{aligned}
p_{i,\ldots,k+1}(x) &= \frac{1}{x_{k+1}-x_i}((x-x_i)p_{i+1,\ldots,k+1}(x) + (x_{k+1}-x)p_{i,\ldots,k}(x)) \\
&= \frac{1}{x_{k+1}-x_i}([y_{i+1},\ldots,y_{k+1}] - [y_i,\ldots,y_k])x^{k-i+1} + r_1(x) \\
&= [y_i,\ldots,y_{k+1}]x^{k-i+1} + r_1(x) \\
&= [y_i,\ldots,y_{k+1}](x-x_i)\cdots(x-x_k) + r_2(x)\ .
\end{aligned}$$

Offenbar muß $r_2(x_j) = y_j,\ \ j = i,\ldots,k$ sein. Nach Satz 1 ist also $r_2 = p_{i,\ldots,k}$. Da für $p_{i,\ldots,k}$ (1.2) bereits gilt, gilt (1.2) auch für $p_{i,\ldots,k+1}$.

□

Beispiel: Wir haben oben das Interpolationsproblem

j	x_j	y_j
0	0	1
1	1	3
2	3	2

durch die Lagrange'sche Form des Interpolationspolynoms gelöst. Zur Lösung des gleichen Problems mit der Newton'schen Form stellen wir zunächst einmal das Differenzenschema auf:

$$\begin{array}{cccc}
0 & 1 & & \\
 & & 2 & \\
1 & 3 & & -5/6 \\
 & & -1/2 & \\
3 & 2 & &
\end{array}$$

Die Koeffizienten des Newton'schen Interpolationspolynoms stehen in der obersten Zeile:

$$p(x) = 1 + 2x - \frac{5}{6}x(x-1) = 1 + \frac{17}{x}x - \frac{5}{6}x^2\ .$$

Fügen wir noch die Stützstelle $x_3 = 3$ mit dem Stützwert $y_3 = 4$ hinzu, so lautet das erweiterte Differenzenschema

$$\begin{array}{ccccc}
0 & 1 & & & \\
 & & 2 & & \\
1 & 3 & & -5/6 & \\
 & & -1/2 & & -1/3 \\
3 & 2 & & -3/2 & \\
 & & -2 & & \\
2 & 4 & & &
\end{array}$$

und das zugehörige Interpolationsproblem ist

$$p(x) = 1 + 2x - \frac{5}{6}x(x-1) - \frac{1}{3}x(x-1)(x-3)\ .$$

5.2 Der Interpolationsfehler

Seien in einem Intervall $[a, b]$ $n+1$ paarweise verschiedene Stützstellen $x_0, \ldots, x_n$ und eine Funktion $f \in C_{n+1}[a, b]$ gegeben. Wir wollen $f(x)$ für $x \neq x_i$ approximieren.

Betrachten wir das eindeutig bestimmte Polynom $p \in \mathcal{P}_n$ mit $p(x_j) = f(x_j)$, $j = 0, \ldots, n$. Der folgende Satz macht eine Aussage über den Fehler, der bei einer Approximation von f durch p auftritt:

Satz 5.2.1 *Zu jedem $x \in [a, b]$ existiert ein $\tilde{x} \in [a, b]$, so daß gilt:*

$$f(x) - p(x) = w(x) \frac{f^{(n+1)}(\tilde{x})}{(n+1)!}$$

mit

$$w(x) = \prod_{j=0}^{n} (x - x_j)$$

Beweis: Sei $\overline{x} \in [a, b]$ beliebig gewählt mit $\overline{x} \neq x_j$, $j = 0, \ldots, n$. Wir setzen

$$F(x) = f(x) - p(x) - Kw(x)$$

mit einer Konstanten K. Dann ist

$$F(x_j) = 0 \quad , \quad j = 0, \ldots, n \; .$$

Wir wählen K nun so, daß auch $F(\overline{x})$ verschwindet, also

$$K = \left(\frac{f - p}{w} \right) (\overline{x}) \; .$$

Damit hat F in $[a, b]$ mindestens die $n+2$ Nullstellen $\overline{x}$, $x_0, \ldots, x_n$. Durch wiederholte Anwendung des Satzes von Rolle folgt, daß $F^{(n+1)}$ mindestens eine Nullstelle $\tilde{x}$ in $[a, b]$ besitzt.

Aus der Beziehung

$$F^{(n+1)}(x) = f^{(n+1)}(x) - K(n+1)!$$

folgt

$$K = \left(\frac{f-p}{w}\right)(\overline{x}) = \frac{f^{(n+1)}(\tilde{x})}{(n+1)!} \; .$$

Also

$$f(\overline{x}) - p(\overline{x}) = w(\overline{x}) \, \frac{f^{(n+1)}(\tilde{x})}{(n+1)!} \; .$$

Diese Beziehung ist offenbar auch für $\overline{x} = x_j, \;\; j = 0, \ldots, n$ erfüllt.

$\square$

Für den Interpolationsfehler erhalten wir die Abschätzung

$$|f(x) - p(x)| \leq |w(x)| \, \max_{x \in [a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} \; .$$

1) Wir betrachten zuerst den Fall gleichmäßig verteilter Stützstellen $x_j = a + jh$ mit der "Schrittweite" $h = \frac{b-a}{n}$. Für $x = a + \theta h$, $0 \leq \theta \leq n$ gilt

$$w(x) = h^{n+1} \prod_{j=0}^{n} (\theta - j) \; ,$$

also

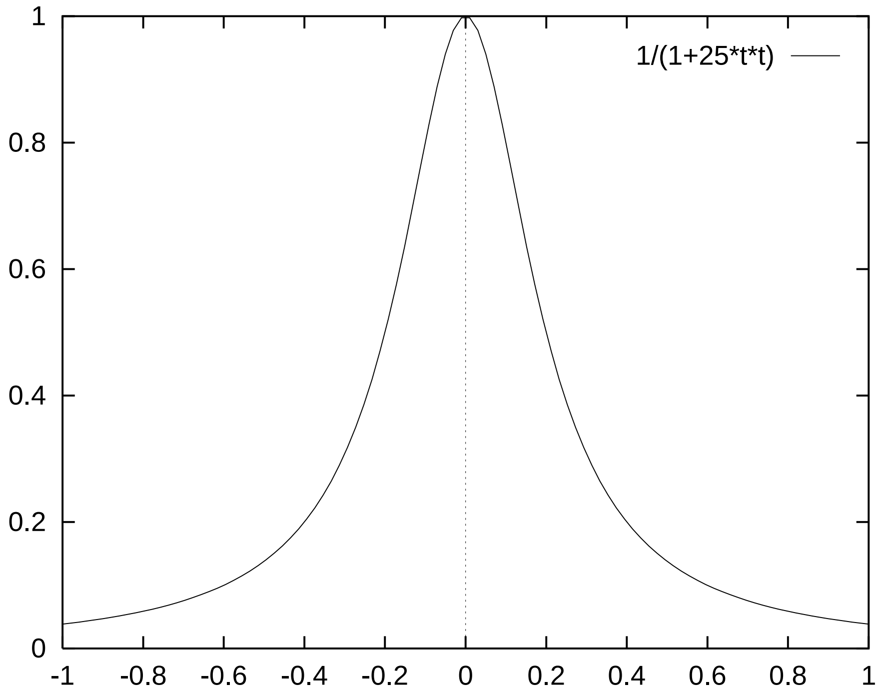
$$|f(x) - p(x)| \leq \frac{h^{n+1}}{(n+1)!} \left(\prod_{j=0}^{n} |\theta - j| \right) \max_{x \in [a,b]} |f^{(n+1)}(x)| \; .$$

(a) Für $b - a \to 0$ bei festem n erhalten wir

$$f(x) - p(x) = \mathrm{O}(h^{n+1}) \; .$$

(b) Der Fall $n \to \infty$ bei fester Intervallänge $b - a$ führt i.a. zu keiner Konvergenz. Wir betrachten hierzu das Beispiel von Runge:

$$f(x) = (1 + 25x^2)^{-1} \quad \text{in } [-1, 1]$$



f wird an den Stellen $x_j = -1 + \frac{2j}{n}$, $j = 0, \ldots, n$ durch ein Polynom vom Grad n interpoliert. Die folgende Tabelle zeigt, daß der Interpolationsfehler für große n stark anwächst.

n	$\max_{x\in[-1,1]} \lvert f(x) - p(x) \rvert$
1	$0,96$
5	$0,43$
13	$1,07$
19	$8,57$

2) Wir wählen die Stützstellen $x_0, \ldots, x_n$ nun so, daß $\max_{[a,b]} |w(x)|$ möglichst klein ist.

Für $[a, b] = [-1, 1]$ erhalten wir

$$w(x) = 2^{-n} T_{n+1}(x) ,$$

wobei für $x \in [-1, 1]$ die Tschebyscheff-Polynome T_n wie folgt definiert sind:

$$T_n(x) = \cos nt \quad , \quad x = \cos t \quad , \quad 0 \leq t \leq \pi .$$

In IV.3 haben wir gesehen, daß

$$T_{n+1} = 2xT_n - T_{n-1} \ .$$

Die Rekursion zeigt, daß T_n für $n \geq 1$ die Form

$$T_n(x) = 2^{n-1}x^n + \text{Polynom} \quad \in \mathcal{P}_{n-1}$$

haben muß.
Daher hat $w(x) = 2^{-n}T_{n+1}(x)$ den Höchstkoeffizienten 1 und es gilt

$$|w(x)| \leq 2^{-n} \quad \text{in} \ [-1,1] \ .$$

Die Nullstellen $x_j = \cos\frac{(j+1/2)\pi}{n}$, $j = 0,\dots,n$ von w sind unsere Stützstellen.
Bei dieser Wahl ergibt sich die Fehlerabschätzung

$$|f(x) - p(x)| \leq \frac{\text{Max}|f^{(n+1)}(x)|}{2^n(n+1)!} \ .$$

Für das obige Beispiel ergeben sich folgende Werte:

n	$\max_{x\in[-1,1]} \lvert f(x) - p(x)\rvert$
1	0,93
5	0,56
13	0,12
19	0,04

Die Verbesserung ist erheblich, die Approximation aber immer noch unbefriedigend.

5.3 Trigonometrische Interpolation

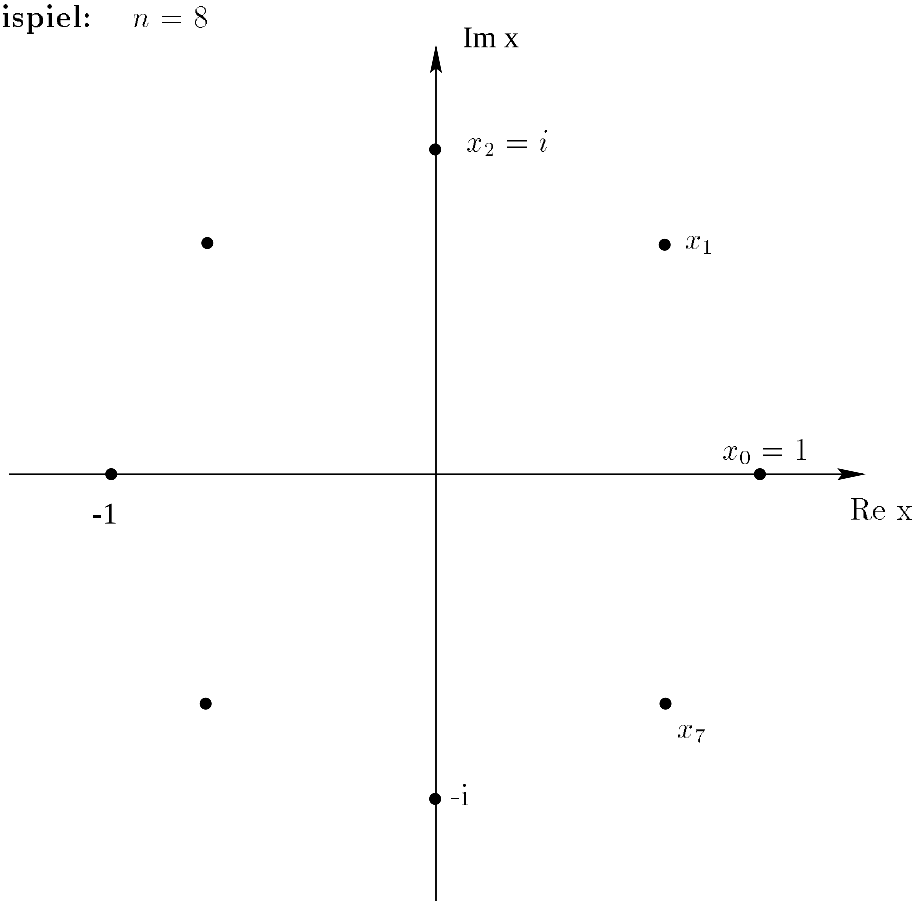
Wir betrachten in diesem Abschnitt einen wichtigen Spezialfall der Polynominterpolation, bei dem die Stützstellen in regelmäßigen Abständen auf dem komplexen Einheitskreis liegen.

Diese spezielle Problemstellung wird zu einem der wichtigsten Hilfsmittel der angewandten Mathematik führen: Zur diskreten Fouriertransformation.

Die Stützstellen seien gegeben durch

$$x_j = e^{it_j} = \cos t_j + i \sin t_j \ ; \quad t_j = 2\pi j/n \ ; \quad j = 0, \ldots, n-1 \ .$$

Beispiel: $n = 8$



Nach Satz 5.1.1 gibt es ein eindeutig bestimmtes Polynom $p \in \mathcal{P}_{n-1}$ mit $p(x_j) = y_j,\;\; j = 0, \ldots, n-1$.

Die Koeffizienten von p bezeichnen wir mit $\hat{y}_k$:

$$p(x) = \sum_{k=0}^{n-1} \hat{y}_k x^k$$

Seien $y = (y_0, \ldots, y_{n-1})^T \in \mathbb{C}^n$ und $\hat{y} = (\hat{y}_0, \ldots, \hat{y}_{n-1})^T \in \mathbb{C}^n$. Dann können wir die Interpolationsaufgabe

$$y_j = \sum_{k=0}^{n-1} \hat{y}_k x_j^k \;; \quad j = 0, \ldots, n-1$$

in die Form

$$y = \begin{pmatrix} 1 & x_0 & \ldots & x_0^{n-1} \\ 1 & x_1 & \ldots & x_1^{n-1} \\ \vdots & & \ddots & \\ 1 & x_{n-1} & \ldots & x_{n-1}^{n-1} \end{pmatrix} \hat{y} = W\hat{y} \;.$$

umschreiben. Die Inversion der Matrix W erweist sich als sehr einfach:

Satz 5.3.1 *Mit obigen Definitionen für x_j, $j = 0, \ldots, n-1$ gilt für die Matrix W:*

$$WW^* = nI \;.$$

Beweis: Für k, $\ell = 0, \ldots, n-1$ gilt:

$$\begin{aligned}
(WW^*)_{k\ell} &= \sum_{j=0}^{n-1} x_k^j \overline{x}_\ell^j \\
&= \sum_{j=0}^{n-1} e^{i(t_k - t_\ell)j} \\
&= \sum_{j=0}^{n-1} e^{2\pi i(k-\ell)j/n} \\
&= \sum_{j=0}^{n-1} q^j \qquad \text{mit } \; q = e^{2\pi i(k-\ell)/n}
\end{aligned}$$

$$= \begin{cases} \frac{q^n-1}{q-1} & , \text{ falls } q \neq 1 \\ n & , \text{ falls } q = 1 \end{cases}$$

$$= \begin{cases} 0 & , \text{ falls } k \neq \ell \\ n & , \text{ falls } k = \ell \end{cases}$$

□

Damit haben wir gleichzeitig eine Orthogonalitätseigenschaft der trigonometrischen Funktionen gezeigt:

$$\frac{1}{n} \sum_{j=0}^{n-1} e^{2\pi ijk/n} = \begin{cases} 1 & , \text{ falls } k \in n\mathbb{Z} \\ 0 & , \text{ sonst} \end{cases}$$

Aus Satz 5.3.1 können wir folgern:

$$W^{-1} = \frac{1}{n} W^*$$

und damit

$$\hat{y} = \frac{1}{n} W^* y \ .$$

In Komponenten:

$$\boxed{\begin{array}{rcll} \hat{y}_k & = & \frac{1}{n} \sum\limits_{j=0}^{n-1} e^{-2\pi ijk/n} y_j \ , & \qquad (1) \\ y_j & = & \sum\limits_{k=0}^{n-1} e^{2\pi ijk/n} \hat{y}_k \quad , & \qquad (2) \end{array}}$$

Gleichung (1) heißt diskrete Fouriertransformation der Länge n, (2) heißt

entsprechend inverse diskrete Fouriertransformation der Länge n. Beide werden in der Angewandten Mathematik sehr häufig benutzt. Man programmiert jedoch nicht nach den Formeln (1) und (2), was jeweils n^2 komplexe Rechenoperationen beanspruchen würde, sondern benutzt erheblich schnellere Algorithmen. Einen davon werden wir in 5.4 kennenlernen.

Zunächst stellen wir jedoch die Beziehung zum Titel dieses Paragraphen her und drücken $p(x)$ durch trigonometrische Funktionen aus:

Wir setzen

$$\begin{aligned} a_k &= \frac{2}{n} \sum_{j=0}^{n-1} y_j \cos(2\pi kj/n) , \\ b_k &= \frac{2}{n} \sum_{j=0}^{n-1} y_j \sin(2\pi kj/n) . \end{aligned}$$

Dann ist

$$\begin{aligned} \hat{y}_k &= \frac{1}{n} \sum_{j=0}^{n-1} y_j e^{-2\pi ijk/n} \\ &= \frac{1}{n} \sum_{j=0}^{n-1} y_j (\cos(2\pi jk/n) - i \sin(2\pi jk/n)) \\ &= \frac{1}{2}(a_k - ib_k) . \end{aligned}$$

Man beachte, daß dies keine Zerlegung in Real- und Imaginärteil von $\hat{y}_k$ darstellt, da die a_k und b_k nicht notwendig reell sind. Es gilt:

$$a_{n-k} = a_k \quad \text{und} \quad b_{n-k} = -b_k$$

und damit

$$\hat{y}_{n-k} = \frac{1}{2}(a_k + ib_k) .$$

Sei n nun ungerade, also $n = 2m+1$. Dann erhalten wir

$$\begin{aligned} p(x_j) &= \sum_{k=0}^{n-1} \hat{y}_k e^{2\pi ijk/n} \\ &= \hat{y}_0 + \sum_{k=1}^{m} \hat{y}_k e^{2\pi ijk/n} + \sum_{k=m+1}^{n-1} \hat{y}_k e^{2\pi ijk/n} \\ &= \hat{y}_0 + \sum_{k=1}^{m} \hat{y}_k e^{2\pi ijk/n} + \sum_{k'=1}^{m} \hat{y}_{n-k'} e^{2\pi ij(n-k')/n} \quad , \quad k' = n-k \\ &= \hat{y}_0 + \frac{1}{2} \sum_{k=1}^{m} \{(a_k - ib_k)(\cos(2\pi jk/n) + i \sin(2\pi jk/n)) \\ &\qquad + (a_k + ib_k)(\cos(2\pi jk/n) - i \sin(2\pi jk/n))\} \ , \end{aligned}$$

also
$$p(x_j) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)) \ .$$
Für gerades n, $n = 2m$, erhält man analog
$$\begin{aligned} p(x_j) = \frac{a_0}{2} \quad &+ \quad \sum_{k=1}^{m-1} (a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)) \\ &+ \quad \frac{a_m}{2} \cos(2\pi jm/n) \ , \end{aligned}$$
wenn man beachtet, daß b_m verschwindet.

Wir bezeichnen die Menge
$$\mathcal{T}_m = \left\{ \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos kt + b_k \sin kt) \ ; \ a_k, b_k \in \mathbb{C} \right\}$$
als die Menge der trigonometrischen Polynome vom Grad m.

Damit haben wir die folgende Aussage über die trigonometrische Interpolation gewonnen:

Satz 5.3.2 *Gesucht sei $T \in \mathcal{T}_m$ mit*
$$\begin{aligned} T(t_j) &= y_j \quad , \quad j = 0, \ldots, n-1 \\ t_j &= 2\pi j/n \ . \end{aligned}$$
Für $n = 2m + 1$ ist die Aufgabe eindeutig lösbar und die Lösung ist durch
$$T(t) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos kt + b_k \sin kt)$$
gegeben.

Für $n = 2m$ gibt es genau eine Lösung mit $b_m = 0$, die durch
$$T(t) = \frac{a_0}{2} + \sum_{k=1}^{m-1} (a_k \cos kt + b_k \sin kt) + \frac{a_m}{2} \cos mt$$
gegeben ist. Dabei sind a_k, b_k die oben angegebenen Summen.

5.4 Schnelle Fouriertransformation

Wir betrachten nun einen effizienten Algorithmus zur Berechnung von $\hat{y}$, die schnelle Fouriertransformation (FFT) von Cooley-Tukey.

Sei n gerade, $n = 2m$. Dann gilt

$$\hat{y}_k = \sum_{j=0}^{n-1} y_j q^{jk} \quad \text{mit} \quad q = e^{-2\pi i/n} \; .$$

Es sind $q^n = 1$ und $q^m = q^{n/2} = -1$.

Die Idee ist nun, die Summe nach geraden und ungeraden Indizes zu zerlegen:

$$\begin{aligned} \hat{y}_k &= \sum_{\ell=0}^{m-1} q^{(2\ell)k} y_{2\ell} + \sum_{\ell=0}^{m-1} q^{(2\ell+1)k} y_{2\ell+1} \\ &= \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell} + q^k \sum_{\ell=0}^{m-1} (q^2)^{\ell k} y_{2\ell+1} \\ &=: g_k + q^k u_k \end{aligned}$$

Wir sehen nun, daß sich g_k und u_k als Fouriertransformationen der Länge $m = n/2$ berechnen, da gilt:

$$q^2 = e^{-2\pi i/(n/2)}$$

Weiter haben wir

$$\begin{aligned} g_{k+m} &= g_k \\ u_{k+m} &= u_k \end{aligned}$$

und damit für $k = 0, \ldots, m-1$

$$\begin{aligned} \hat{y}_k &= g_k + q^k u_k \; , \\ \hat{y}_{k+m} &= g_k + q^{k+m} u_k = g_k - q^k u_k \; . \end{aligned}$$

Sei nun M_p die Anzahl der komplexen Multiplikationen und A_p die Anzahl der komplexen Additionen, die für eine schnelle Fouriertransformation der Länge

$n = 2^p$ benötigt werden. Wenn wir die Berechnung von q^k vernachlässigen, erhalten wir $A_0 = 0$, $M_0 = 0$ und

$$\begin{aligned} M_{p+1} &= 2M_p + 2^p \ , \\ A_{p+1} &= 2A_p + 2^{p+1} \ , \end{aligned}$$

also

$$\begin{aligned} M_p &= \tfrac{1}{2}p2^p &= \tfrac{1}{2}(\log_2 n) \cdot n \\ A_p &= p2^p &= (\log_2 n) \cdot n \ . \end{aligned}$$

Damit gilt der

Satz 5.4.1 *Die Fouriertransformation der Länge $n = 2^p$ kann durch $\frac{1}{2} n \log_2 n$ komplexe Multiplikationen und $n \log_2 n$ komplexe Additionen berechnet werden.*

Die Implementierung der FFT durch ein rekursives Programm ist sehr einfach:

```
fft (y, n)
/⋆ Führt die FFT der Länge n = 2^p durch ⋆/
{ m = n/2;
    for ℓ = 0, ..., m − 1 {
        g[ℓ] = y[2ℓ];
        u[ℓ] = y[2ℓ + 1];
    }
    if (m > 1) {
        fft (g, m);
        fft (u, m);
    }
    for k = 0, ..., m − 1 {
        u[k] = q^k ∗ u[k];
        y[k] = g[k] + u[k];
        y[k + m] = g[k] − u[k];
    }
}
```

Die FFT nach Cooley-Tukey ist ein typisches Beispiel für das "divide and conquer"-Prinzip der Informatik:

1. Zerlege das Problem in Teilprobleme.
2. Löse die Teilprobleme.
3. Setze die Lösung der Teilprobleme zur Lösung des ganzen Problems zusammen.

Beispiele:

$$\begin{array}{rclcll}
n=1 & : & \hat{y}_0 & = & y_0 & \\
n=2 & : & \hat{y}_0 & = & y_0+y_1 & \\
 & & \hat{y}_1 & = & y_0-y_1 & \\
n=4 & : & q & = & p^{-2\pi i/4}=-i & \\
 & & g_0 & = & y_0+y_2 & u_0=y_1+y_3 \\
 & & g_1 & = & y_0-y_2 & u_1=y_1-y_3 \\
 & & \hat{y}_0 & = & g_0+u_0 & \\
 & & \hat{y}_1 & = & g_1-iu_1 & \\
 & & \hat{y}_2 & = & g_0-u_0 & \\
 & & \hat{y}_3 & = & g_1+iu_1 &
\end{array}$$

Wir können die FFT auch als Faktorisierung der Matrix W_n für die Fourier-Transformation der Länge n ansehen, also

$$W_n = \left(e^{-2\pi i k\ell/n}\right)_{k,\ell=0,\ldots,n-1} .$$

Sei P_n die Permutationsmatrix

$$P_n = \begin{pmatrix} e_0 \\ e_2 \\ \vdots \\ e_{n-2} \\ e_1 \\ e_3 \\ \vdots \\ e_{n-1} \end{pmatrix}$$

mit dem Einheitsvektor e_j^* des C^n, in dem wir die Komponenten von 0 bis $n-1$ durchnummerieren. Sei $m = n/2$ und I_m die (m,m) Einheitsmatrix, D_m die Diagonalmatrix

$$D_m = \begin{pmatrix} 1 & & & \\ & q & & \\ & & \ddots & \\ & & & q^{m-1} \end{pmatrix} , \qquad q = e^{-2\pi i/n} .$$

Dann ist

$$W_n = \begin{pmatrix} I_m & I_m \\ I_m & -I_m \end{pmatrix} \begin{pmatrix} I_m & 0 \\ 0 & D_m \end{pmatrix} \begin{pmatrix} W_m & 0 \\ 0 & W_m \end{pmatrix} P_n .$$

Die FFT ergibt sich durch sukzessives Anwenden dieser Formel. Die Durchführung der FFT verlangt also nur drei Operationen:

1) Permutationen

2) Multiplikation mit q^ℓ (Twiddle factor)

3) Die Anwendung von $\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ (Butterfly)

5.5 Auswertung von Polynomen

Sei $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + A_0$. Der Standard-Algorithmus zur Auswertung von p an der Stelle x ist

$$\begin{aligned} s_n &= a_n , \\ s_j &= x s_{j+1} + a_j \quad , \quad j = n-1, \ldots, 0 \\ p &= s_0 . \end{aligned}$$

Man kommt also mit n flops aus.

Ein entsprechender Algorithmus für trigonometrische Polynome ist der Algorithmus von Goertzel (1958). Er wertet die Ausdrücke

$$c = \sum_{k=0}^{n-1} y_k \cos k\varphi \ , \qquad s = \sum_{k=0}^{n-1} y_k \sin k\varphi$$

aus. Für $\varphi = 2\pi j/n$, $j = 0, \ldots, n-1$, ist das ein Fall für FFT.

Satz 5.5.1 *Für* $\varphi \notin \pi\mathbb{Z}$, $n \in \mathbb{N}$ *sei*

$$\begin{aligned} u_j &= \frac{1}{\sin\varphi} \sum_{k=j}^{n-1} y_k \sin(k-j+1)\varphi \quad , \quad j = 1, \ldots, n-1 \\ u_n &= u_{n+1} = 0 \ . \end{aligned}$$

Dann gilt:

(a) $u_j = y_j + 2\cos\varphi \, u_{j+1} - u_{j+2} \quad , \quad j = n-1, n-2, \ldots, 1$

(b) $\sum_{k=0}^{n-1} y_k \sin k\varphi = u_1 \sin\varphi$

(c) $\sum_{k=0}^{n-1} y_k \cos k\varphi = y_0 + \cos\varphi \, u_1 - u_2$

Beweis:

Zu **(a)**: Das Additionstheorem für den Sinus

$$\sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta$$

liefert uns die Formel

$$\sin(k-j\pm 1)\varphi = \sin(k-j)\varphi\cos\varphi \pm \cos(k-j)\varphi\sin\varphi \ .$$

Hiermit können wir (a) durch einfaches Nachrechnen zeigen:

$$\begin{aligned} & y_j + 2\cos\varphi \, u_{j+1} - u_{j+2} \\ & \quad = y_j + \frac{1}{\sin\varphi}\left\{ 2\cos\varphi \sum_{k=j+1}^{n-1} y_k \sin(k-j)\varphi - \sum_{k=j+2}^{n-1} y_k \sin(k-j-1)\varphi \right\} \end{aligned}$$

$$
\begin{aligned}
&= y_j + \frac{1}{\sin\varphi} \sum_{k=j+1}^{n-1} y_k \{2\cos\varphi \sin(k-j)\varphi - \sin(k-j-1)\varphi\} \\
&= y_j + \frac{1}{\sin\varphi} \sum_{k=j+1}^{n-1} y_k \{2\cos\varphi \sin(k-j)\varphi - (\sin(k-j)\varphi\cos\varphi - \cos(k-j)\varphi\sin\varphi)\} \\
&= y_j + \frac{1}{\sin\varphi} \sum_{k=j+1}^{n-1} y_k \{\cos\varphi \sin(k-j)\varphi + \sin\varphi\cos(k-j)\varphi\} \\
&= \frac{1}{\sin\varphi} \left\{ y_j \sin\varphi + \sum_{k=j+1}^{n-1} y_k \sin(k-j+1)\varphi \right\} \\
&= \frac{1}{\sin\varphi} \sum_{k=j}^{n-1} y_k \sin(k-j+1)\varphi \\
&= u_j
\end{aligned}
$$

(b) Dies folgt direkt aus der Definition von u_1.

(c) Es ist

$$
\begin{aligned}
y_0 + \cos\varphi u_1 - u_2 &= y_0 + \frac{\cos\varphi}{\sin\varphi} \sum_{k=1}^{n-1} y_k \sin k\varphi - \frac{1}{\sin\varphi} \sum_{k=2}^{n-1} y_k \sin(k-1)\varphi \\
&= y_0 + y_1 \cos\varphi + \sum_{k=2}^{n-1} y_k \frac{\cos\varphi \sin k\varphi - \sin(k-1)\varphi}{\sin\varphi} \\
&= y_0 + y_1 \cos\varphi + \sum_{k=2}^{n-1} y_k \cos k\varphi\ ,
\end{aligned}
$$

wobei wir zuletzt wieder das Additionstheorem benutzt haben.

□

Dies führt zu folgendem Programm:

goertzel (φ, y, n, s, c)
/$\star$ Wertet für gegebene $y_0, \ldots, y_{n-1}$ die Zahlen c, s aus $\star$/
$\{u_n = u_{n+1} = 0;$
    $\gamma = \cos\varphi;$
    for $j = n-1,\ n-2, \ldots, 1$

$$u_j = y_j + 2\gamma u_{j+1} - u_{j+2};$$
$$c = y_0 + u_1\gamma - u_2;$$
$$s = u_1 \sin\varphi;$$
}

Zahlenbeispiel: $n = 1001$, $\varphi = \frac{\pi}{2(n-1)}$, $y_j = \begin{cases} 1 & , \quad j = n-1 \\ 0 & , \quad \text{sonst} . \end{cases}$

Korrekt: $c = \cos\frac{\pi}{2} = 0$, $s = \sin\frac{\pi}{2} = 1$

Sun 4 mit float-Zahlen: $\tilde{c} = -0.012144$, $\tilde{s} = 0.992777$

Dies ist nicht befriedigend. Eine Erklärung findet sich im nächsten Paragraphen.

5.6 Differenzengleichungen

Eine lineare Differenzengleichung der Ordnung m mit konstanten Koeffizienten lautet

$$\sum_{\nu=0}^{m} u_{k+\nu}\alpha_\nu = f_k \ , \qquad k = 0, 1, \ldots \tag{6.1}$$

Wir setzen $\alpha_m \neq 0$ voraus.

Beispiele:

1) Die Formel von Goertzel (Satz 5.5.1), also

$$u_k = y_k + 2u_{k+1}\cos\xi - u_{k+2} \ .$$

2) Für die Tschebyscheff-Polynome $T_k(x)$ gilt

$$T_{k+1}(x) + T_{k-1}(x) = 2xT_k(x) \ , \quad T_0 = 1 \ , \quad T_1 = x \ .$$

3) Ebenfalls linear, aber nicht mit konstanten Koeffizienten ist die Differenzengleichung

$$J_{k+1}(x) + J_{k-1}(x) = \frac{2k}{x}J_k(x)$$

für die Bessel-Funktion $J_k(x)$ 1. Art der Ordnung k:

$$J_k(x) = \frac{1}{\pi}\int\limits_0^\pi \cos(x\sin t - kt)dt$$

Die Differenzengleichung (6.1) heißt homogen, falls $f_k = 0$ ist für $k = 0, 1, \ldots,$ andernfalls inhomogen. Der homogenen Gleichung ordnen wir das charakteristische Polynom

$$p(\lambda) = \sum_{\nu=0}^{m} \alpha_\nu \lambda^\nu$$

zu.

Satz 5.6.1 *p habe die Nullstellen* $\lambda_1, \ldots, \lambda_r$ *mit den Vielfachheiten* $\sigma_1, \ldots, \sigma_r$, $\sigma_1 + \cdots + \sigma_r = m$. *Dann ist*

$$u_k = k^\sigma \lambda_\ell^k \ , \qquad 0 \leq \sigma < \sigma_\ell \ , \quad \ell = 1, \ldots, r \tag{6.2}$$

Lösung der homogenen Gleichung (6.1), und jede Lösung der homogenen Gleichung (6.1) ist eine Linearkombination solcher Lösungen.

Beweis: Für $\sigma = 0$ ist

$$\begin{aligned} \sum_{\nu=0}^{m} u_{k+\nu}\alpha_\nu &= \sum_{\nu=0}^{m} \lambda_\ell^{k+\nu}\alpha_\nu = \lambda_\ell^k \sum_{\nu=0}^{n} \lambda_\ell^\nu \alpha_\nu \\ &= \lambda_\ell^k p(\lambda_\ell) = 0 \ . \end{aligned}$$

Ist λ_ℓ eine σ_ℓ-fache Wurzel von p, so ist für $0 \leq \sigma < \sigma_\ell$

$$\left(\frac{d}{d\lambda}\right)^\sigma p(\lambda)\Big|_{\lambda=\lambda_\ell} = \sum_{\nu=0}^{m} \nu(\nu-1)\cdots(\nu-\sigma+1)\lambda_\ell^{\nu-\sigma}\alpha_\nu = 0 \ .$$

Es folgt

$$\sum_{\nu=0}^{m} \nu^\sigma \lambda_\ell^\nu \alpha_\nu = 0 \ , \quad \sigma = 0, \ldots, \sigma_\ell - 1$$

und damit

$$\sum_{\nu=0}^{m} u_{k+\nu}\alpha_\nu = \sum_{\nu=0}^{m} (k+\nu)^\sigma \lambda_\ell^{k+\nu}\alpha_\nu = 0 \ , \quad \sigma = 0, \ldots, \sigma_\ell - 1 \ .$$

Also sind (6.2) tatsächlich Lösungen der homogenen Gleichung (6.1).

Die Lösungen der homogenen Gleichung (6.1) bilden offenbar einen Vektorraum über $\mathbb{C}$. Jede Lösung u ist durch m Anfangswerte $u_0, \ldots, u_{m-1}$ eindeutig bestimmt. Es folgt, daß der Lösungsraum die Dimension m hat. Die m Lösungen (6.2) sind linear unabhängig und bilden daher eine Basis des Lösungsraumes.

□

Man nennt eine Differenzengleichung stabil, wenn es eine Konstante C gibt, so daß für jede Lösung der homogenen Gleichung

$$|u_k| \leq C \max_{0\leq \ell < m} |u_\ell|$$

für alle $k \geq 0$ gilt. Sind dann für eine Lösung u_k der inhomogenen Gleichung $u_0, \ldots, u_{m-1}$ mit dem Fehler ε bekannt, so kann u_k für $k \geq m$ mit dem Fehler $\leq C\varepsilon$ berechnet worden.

Satz 5.6.2 *Die Differenzengleichung (6.1) ist genau dann stabil, wenn für* $\ell = 1, \ldots, r$ *gilt:*

$$|\lambda_\ell| < 1 \quad \textit{oder} \quad |\lambda_\ell| = 1 \quad \textit{und} \quad \sigma_\ell = 1 \ .$$

Beweis: Folgt sofort aus Satz 6.1.

□

Beispiele:

1) Bei der Formel von Goertzel ist $p(\lambda) = 1 - 2\lambda\cos\xi + \lambda^2$, also $\lambda_{1,2} = \cos\xi \pm i\sin\xi$. Die Rekursion ist stabil für $\xi \neq 0$, instabil für $\xi = 0$. Man hat Probleme für $\xi \sim 0$.
2) Es gibt Probleme bei $|x| \geq 1$.
3) Hier ist die Theorie nicht streng anwendbar, da die Differenzengleichung nicht konstante Koeffizienten hat. Aus dem Vergleich mit 2) vermutet man aber, daß man Probleme hat für $|k/x| \geq 1$.

Zahlenbeispiel: Wir wollen $J_6(1)$ berechnen, ausgehend von $J_0(1)$, $J_1(1)$:

$J_0(1)$	0.765198	exakt
$J_1(1)$	0.440051	exakt
$J_2(1)$	0.114904	1
$J_3(1)$	0.019565	2
$J_4(1)$	0.002486	9
$J_5(1)$	0.000323	73
$J_6(1)$	0.000744	723

Der Fehler in der letzten Spalte ist in Einheiten der letzten Stelle angegeben. Deutlich sieht man die Auswirkung der Instabilität.

5.7 Harmonische Analyse

Sei f 2π-periodisch. Solche Funktionen kann man in eine Fourier-Reihe entwickeln:

Satz 5.7.1 *Sei f stetig. Dann gilt*

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{k=0}^{\infty} (a_k \cos kx + b_k \sin kx) \;, \\ a_k &= \frac{1}{\pi} \int\limits_0^{2\pi} f(x) \cos kx dx \\ b_k &= \frac{1}{\pi} \int\limits_0^{2\pi} f(x) \sin kx dx \end{aligned}$$

Beweis: Siehe etwa **Heuser** :*Lehrbuch der Analysis II.*

□

Bemerkungen:

1) Gilt sogar $f \in C^m$, so ist $a_k, b_k = \mathrm{O}(k^{-m})$.

2) Ist f stetig mit Ausnahme von x_0 und existieren dort die linken und rechten Grenzwerte $f(x_0 \pm 0)$, so konvergiert die Reihe in x_0 gegen $\frac{1}{2}\left(f(x_0 + 0) + f(x_0 - 0)\right)$.

3) Die Berechnung der a_k, b_k aus f heißt Fourier-Analyse, die Berechnung von f aus a_k, b_k heißt Fourier-Synthese. a_k, b_k heißen Fourier-Koeffizienten von f, die Reihe Fourier-Reihe von f.

Beispiel: $f(x) = \begin{cases} 1 & , \quad 0 \le x < \pi \\ -1 & , \quad \pi \le x < 2\pi \end{cases}$. Offenbar ist $a_k = 0$ für alle k und $b_k = 0$ für k gerade, $b_k = 4/\pi k$ für k ungerade. Also lautet die Fourier-Reihe

$$f(x) = \frac{4}{\pi}\left(\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \cdots\right) .$$

Zur numerischen Fourier-Analyse nehmen wir an, daß f an den Stützstellen $x_j = 2\pi j/n$, $j = 0, \ldots, n-1$ gegeben ist. Dann hat man die Approximationen

$$a_k \sim a_{k,n} = \frac{2}{n}\sum_{j=0}^{n-1} f(x_j)\cos kx_j \ , \quad b_k \sim b_{k,n} = \frac{2}{n}\sum_{j=0}^{n-1} f(x_j)\sin kx_j \ .$$

Im Abschnitt über numerische Integration werden wir sehen, daß diese Näherungen (für 2π-periodisches f) kaum noch verbesserbar sind. Trotzdem verhalten sich die genäherten Fourier-Koeffizienten $a_{k,n}$, $b_{k,n}$ ganz anders als die exakten: $a_{k,n}$, $b_{k,n}$ haben in k die Periode n, während a_k, b_k für $k \to \infty$ gegen 0 streben. Man kann daher nur für $k \ll n$ erwarten, daß $a_{k,n}$, $b_{k,n}$ gute Näherungen für a_k, b_k darstellen.

Wie in §3 gehen wir zur komplexen Schreibweise und das Intervall $[-\pi, \pi]$ über und schreiben dann für Satz 1

$$\begin{aligned} f(x) &= \sum_{k=-\infty}^{+\infty} c_k \, e^{ikx} \ , \\ c_k &= \frac{1}{2\pi}\int_{-\pi}^{+\pi} f(x)\, e^{-ikx} dx \ . \end{aligned}$$

Die Approximation $c_{n,k}$ für c_k ist mit $x_j = \pi j/n, \ j = -n, \ldots, n-1$

$$c_{n,k} = \frac{1}{n}\sum_{j=-n}^{n-1} f_j \, e^{-ikx_j} = \frac{1}{n}\sum_{j=-n}^{n-1} f_j \, e^{-\pi ikj/n}$$

mit $f_j = f(x_j)$. Dies ist genau eine Fourier-Transformation der Länge $2n$. Die Fourier-Analyse kann also näherungsweise durch die schnelle Fourier-Transformation durchgeführt werden. Zur Fourier-Synthese verwenden wir aus den oben angedeuteten Gründen nur $c_{n,k}$ für $k = -n, \ldots, n-1$, d.h.

$$\begin{aligned} f(x_j) &\sim \sum_{k=-n}^{n-1} c_{n,k}\, e^{ikx_j} \\ &= \sum_{k=-n}^{n-1} c_{n,k}\, e^{\pi ikj/n} \ , \end{aligned}$$

weil $c_{n,k}$ in k die Periode n hat. Dies ist genau eine inverse diskrete Fourier-Transformation der Länge $2n$. Auch die Fourier-Synthese kann daher mit der schnellen Fourier-Transformation durchgeführt werden.

Um das asymptotische Verhalten der $c_{n,k}$ an das der c_k anzupassen, multipliziert man die $c_{n,k}$ noch mit "Abminderungsfaktoren" oder "Filterfaktoren" $d_{n,k}$ mit der Eigenschaft

$$\begin{aligned} d_{n,k} &\sim 1 \quad , \quad |k| \ll n \ , \\ d_{n,k} &\sim 0 \quad , \quad |k| \sim n \ . \end{aligned}$$

Eine häufige Wahl ist

$$d_{n,k} = \left(\frac{\sin k\pi/n}{k\pi/n} \right)^p$$

mit einem geeigneten p, etwa $p = 1, 2$ oder 3.

5.8 Splines

Ein Spline (spline (engl.) = Straklatte) ist ein dünner elastischer Stab, der zwischen Gewichte ("Knoten") gespannt Kurven darstellt. Sie wurden im Schiffsbau verwendet. Wir werden unten sehen, daß die Straklatte zwischen den Knoten durch Polynome beschrieben wird, die an den Knoten unter gewissen Differenzierbarkeitsbedingungen zusammengefügt werden. Unter Splines oder Splinefunktionen versteht man daher in der Numerik Funktionen, welche stückweise Polynome sind und die an den Nahtstellen gewissen Differenzierbarkeitsbedingungen genügen.

Definition 5.8.1 *Seien $x_0 < x_1 < \cdots < x_n$ reelle Zahlen. Eine Funktion s heißt Spline der Ordnung k zu $x_0, \ldots, x_n$, falls*

1) $s \in C^{k-2}(\mathbb{R}^1)$.

2) In jedem Intervall $[x_i, x_{i+1}]$ $i = -1, \ldots, n$ ist s ein Polynom vom Grade $\leq k-1$ $(x_{-1} = -\infty,\ x_{n+1} = +\infty)$.

(Für $k = 1$ ist 1) leer.)

Beispiel: $n = 0$, $x_0 = t$,

$$f(x) = (x-t)_+^{k-1} = \begin{cases} (x-t)^{k-1} & , \quad x > t\ , \\ 0 & , \quad \text{sonst}\ . \end{cases}$$

f ist ein Spline der Ordnung k zu x_0.

Mit Hilfe von f können wir bereits sehr nützliche Splines erzeugen. Sei $x_0 < x_1 < \cdots < x_n$. Wir setzen $f_i = f(x_i)$, $i = 0, \ldots, n$. Die f_i sind Funktionen von t, was wir aber in der Bezeichnung nicht zum Ausdruck bringen. Dann sind auch die dividierten Differenzen $[f_i, \ldots, f_{i+k}]$ Funktionen von t. Wir setzen

$$B_{i,k}(t) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}]\ . \tag{8.1}$$

Beispiel: $k = 2$, $f(x) = (x-t)_+$. Es ist

$$\begin{aligned} B_{i,2}(t) &= (x_{i+2} - x_i)[f_i, f_{i+1}, f_{i+2}] \\ &= [f_{i+1}, f_{i+2}] - [f_i, f_{i+1}] \\ &= \frac{f_{i+1} - f_{i+2}}{x_{i+1} - x_{i+2}} - \frac{f_i - f_{i+1}}{x_i - x_{i+1}}\ . \end{aligned}$$

Wir betrachten 4 Fälle.

1) $t < x_i$. Dann ist $f_i = x_i - t$, $f_{i+1} = x_{i+1} - t$, $f_{i+2} = x_{i+2} - t$, und es wird $B_{i,2}(t) = 0$.

2) $x_i \leq t < x_{i+1}$. Jetzt ist $f_i = 0$, $f_{i+1} = x_{i+1} - t$, $f_{i+2} = x_{i+2} - t$, und es wird $B_{i,2}(t) = 1 - (x_{i+1} - t)/(x_{i+1} - x_i)$.

3) $x_{i+1} \leq t < x_{i+2}$. Jetzt ist $f_i = f_{i+1} = 0$, $f_{i+2} = x_{i+2} - t$, also $B_{i,2}(t) = (x_{i+2} - t)/(x_{i+2} - x_{i+1})$.

4) $x_{i+2} \leq t$. Es ist $f_i = f_{i+1} = f_{i+2} = 0$ und damit $B_{i,2}(t) = 0$.

Wir erhalten also die stückweise lineare stetige Funktion, die außerhalb $[x_i, x_{i+2}]$ verschwindet und die bei x_{i+1} den Wert 1 annimmt, also einen Spline der Ordnung 2.

Satz 5.8.1 *$B_{i,k}$ ist ein Spline der Ordnung k, der außerhalb $[x_i, x_{i+k}]$ verschwindet.*

Beweis: f_j ist für $t \neq x_j$ ein Polynom vom Grade $\leq k-1$ in t und für alle t $k-2$ mal stetig differenzierbar (für $k = 1$ ist die letzte Aussage leer). Als Linearkombination solcher Ausdrücke ist $[f_i, \ldots, f_{i+k}]$ in jedem Intervall $[x_j, x_{j+1}]$ $(j = i-1, \ldots, i+k)$ ein Polynom vom Grade $\leq k-1$ und $k-2$ mal stetig differenzierbar, also ein Spline der Ordnung k zu $x_i, \ldots, x_{i+k}$. Für $t < x_i$ ist $f_j = (x_j - t)^{k-1}$ ein Polynom vom Grade $k-1$ in x_j für $j = i, \ldots, i+k$. Die dividierte Differenz $[f_i, \ldots, f_{i+k}]$ ist daher 0. Für $t > x_{i+k}$ ist $f_j = 0$ für $j = i, \ldots, i+k$. In beiden Fällen ist $B_{i,k}(t) = 0$.

□

Für das Weitere benötigen wir ein Hilfsmittel über dividierte Differenzen, das mit Splines eigentlich gar nichts zu tun hat.

Lemma 5.8.1 *(Leibniz'sche Formel)* *Sei $f_j = g_j h_j$, $j = i, \ldots, i+k$. Dann gilt*

$$[f_i, \ldots, f_{i+k}] = \sum_{r=i}^{i+k} [g_i, \ldots, g_r][h_r, \ldots, h_{i+k}] \, .$$

Beweis: Seien f, g, h die Interpolationspolynome vom Grade k zu den Stützstellen $x_i, \ldots, x_{i+k}$ und den Stützwerten f_j, g_j, h_j, $j = i, \ldots, i+k$. Das Newton'sche Interpolationspolynom für g, h lautet

$$\begin{aligned}
g(x) &= [g_i] + [g_i, g_{i+1}](x - x_i) + \cdots [g_i, \ldots, g_{i+k}](x - x_i) \cdots (x - x_{i+k-1}) \\
&= \sum_{r=i}^{i+k} [g_i, \ldots, g_r](x - x_i) \cdots (x - x_{r-1}) , \\
h(x) &= [h_{i+k}] + [h_{i+k}, h_{i+k-1}](x - x_{i+k}) + \cdots \\
&\qquad + [h_{i+k}, \ldots, h_i](x - x_{i+k}) \cdots (x - x_{i+1}) \\
&= \sum_{s=i}^{i+k} [h_s, \ldots, h_{i+k}](x - x_{i+k}) \cdots (x - x_{s+1}) .
\end{aligned}$$

Multiplikation ergibt

$$(gh)(x) = \sum_{r,s=i}^{i+k} [g_i, \ldots, g_r][h_s, \ldots, h_{i+k}](x-x_i) \cdots (x-x_{r-1})(x-x_{s+1}) \cdots (x-x_{i+k}) . \tag{8.2}$$

Wir teilen die Summe auf in die Summe für $r > s$ und den Rest. Die erste Summe verschwindet für $x = x_i, \ldots, x_{i+k}$. Die restliche Summe ist ein Polynom vom Grade $\leq k$, welches an den Stellen $x_i, \ldots, x_{i+k}$ die Werte $f_i, \ldots, f_{i+k}$ annimmt und daher mit f übereinstimmen muß.

Der Höchstkoeffizient des restlichen Polynoms ergibt sich als Summe über $r = s$, also

$$\sum_{s=i}^{i+k} [g_i, \ldots, g_s][h_s, \ldots, h_{i+k}] ,$$

und dies muß mit dem Höchstkoeffizienten des Newton'schen Interpolationspolynoms, also $[f_i, \ldots, f_{i+k}]$ übereinstimmen. Dies beweist die Leibniz'sche Formel.

$\square$

Satz 5.8.2 *Für $k \geq 2$ und $i = 0, \ldots, n-k$ gilt*

$$B_{i,k}(x) = \frac{x - x_i}{x_{i+k-1} - x_i} B_{i,k-1}(x) + \frac{x_{i+k} - x}{x_{i+k} - x_{i+1}} B_{i+1,k-1}(x) .$$

Beweis: Nach Definition (5.8.1) ist

$$B_{i,k}(x) = (x_{i+k} - x_i)[f_i, \ldots, f_{i+k}] \ , \quad f_j = (x_j - x)_+^{k-1} \ .$$

Wir setzen $g_j = (x_j - x)_+^{k-2}$ und $h_j = x_j - x$. Dann ist $f_j = g_j h_j, \quad j = i, \ldots, i+k$ und damit nach Leibniz

$$B_{i,k}(x) = (x_{i+k} - x_i) \sum_{r=i}^{i+k} [g_i, \ldots, g_r][h_r, \ldots, h_{i+k}] \ .$$

Da h_j ein Polynom vom Grade 1 in x_j ist, verschwindet $[h_r, \ldots, h_{i+k}]$ für $r < i+k-1$, und wir erhalten

$$\begin{aligned}
B_{i,k}(x) &= (x_{i+k} - x_i)\{[g_i, \ldots, g_{i+k}][h_{i+k}] + [g_i, \ldots, g_{i+k-1}][h_{i+k-1}, h_{i+k}]\} \\
&= (x_{i+k} - x_i)\{[g_i, \ldots, g_{i+k}](x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}]\} \\
&= (x_{i+k} - x_i)\left\{\frac{[g_{i+1}, \ldots, g_{i+k}] - [g_i, \ldots, g_{i+k-1}]}{x_{i+k} - x_i}(x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}]\right\} \\
&= [g_{i+1}, \ldots, g_{i+k}](x_{i+k} - x) + [g_i, \ldots, g_{i+k-1}](x - x_i) \\
&= \frac{B_{i+1,k-1}}{x_{i+k} - x_{i+1}}(x_{i+k} - x) + \frac{B_{i,k-1}(x)}{x_{i+k-1} - x_i}(x - x_i) \ .
\end{aligned}$$

□

Bemerkungen:

1) Zusammen mit

$$B_{i,1}(x) = \begin{cases} 1 & , \quad x_i \leq x < x_{i+1} \ , \\ 0 & , \quad \text{sonst} \end{cases}$$

ermöglicht Satz 5.8.2 die rekursive Berechnung der $B_{i,k}$.

2) Diese Rekursion ist numerisch stabil, da nur Linearkombinationen positiver Zahlen gebildet werden.

Satz 5.8.3 *Für $k \geq 3$ gilt*

$$B'_{i,k}(x) = (k-1)\left\{\frac{B_{i,k-1}(x)}{x_{i+k-1}-x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k}-x_{i+1}}\right\} .$$

Beweis: Nach Definition ist

$$B_{i,k}(x) = (x_{i+k}-x_i)[f_i, \ldots, f_{i+k}] , \quad f_j = (x_j - x)_+^{k-1} .$$

Differentiation nach x ergibt

$$B'_{i,k}(x) = -(k-1)(x_{i+k}-x_i)[g_i, \ldots, g_{i+k}] ,$$

$$g_j = (x_j - x)_+^{k-2} .$$

Nach der rekursiven Definition der dividierten Differenzen ist

$$\begin{aligned} B'_{i,k}(x) &= -(k-1)\{[g_{i+1}, \ldots, g_{i+k}] - [g_i, \ldots, g_{i+k-1}]\} \\ &= (k-1)\left\{\frac{B_{i,k-1}(x)}{x_{i+k-1}-x_i} - \frac{B_{i+1,k-1}(x)}{x_{i+k}-x_{i+1}}\right\} . \end{aligned}$$

□

5.9 Interpolation mit Splines

Sei $x_0 < x_1 < \cdots x_n$ und

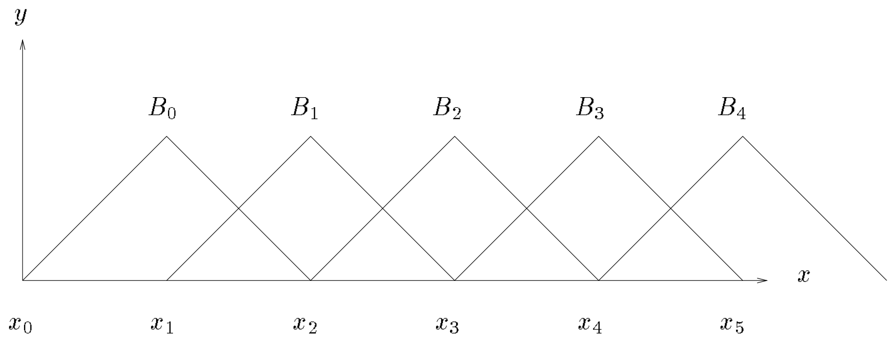
$$s(t) = \sum_{i=0}^{n-k} a_i B_{i,k}(t) \ .$$

Seien $n-k+1$ Stützstellen $t_0 < t_1 < \cdots < t_{n-k}$ und ebenso viele Stützwerte $y_0, \ldots, y_{n-k}$ gegeben. Wir wollen die a_i so bestimmen, daß $s(t_j) = y_j$, $j = 0, \ldots, n-k$ ist.

Beispiel: $k = 1$. Dann ist $s(t) = a_i$ für $x_i \leq t < x_{i+1}$. Das Interpolationsproblem ist genau dann eindeutig lösbar, wenn $x_i \leq t_i < x_{i+1}$, und zwar ist $a_i = y_i, \ i = 0, \ldots, n-k$.

Satz 5.9.1 *Das Interpolationsproblem ist eindeutig lösbar, wenn* $x_i < t_i < x_{i+k}, \ i = 0, \ldots, n-k$.

Beweis: Wir beginnen mit dem Fall $k = 2$ und schreiben B_i für $B_{i,k}$.



Wir haben zu zeigen, daß die $(n-1, n-1)$-Matrix

$$B = \begin{pmatrix} B_0(t_0) & B_1(t_0) & 0 & 0 \\ B_0(t_1) & B_1(t_1) & B_2(t_1) & 0 \\ 0 & B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} \qquad (n = 5)$$

invertierbar ist. Für $n = 2, 3$ ist dies elementar einzusehen. Den Fall $n > 3$ kann man auf $n = 2, 3$ zurückführen. Dazu zeigen wir, daß für $n > 3$ nicht alle Außerdiagonalelemente $\neq 0$ sein können. Ist nämlich etwa $B_0(t_1) \neq 0$, so ist $t_1 < x_2$ und damit $B_2(t_1) = 0$. Entsprechend argumentiert man in den anderen Fällen. Ist aber ein Außerdiagonalelement $= 0$, so zerfällt die Matrix in Teilmatrizen. Im Falle $B_2(t_1) = 0$ wären dies

$$\begin{pmatrix} B_0(t_0) & B_1(t_0) \\ B_0(t_1) & B_1(t_1) \end{pmatrix} \quad , \quad \begin{pmatrix} B_2(t_2) & B_3(t_2) \\ B_2(t_3) & B_3(t_3) \end{pmatrix} ,$$

im Falle $B_1(t_0) = 0$

$$B_1(t_0) \quad , \quad \begin{pmatrix} B_1(t_1) & B_2(t_1) & 0 \\ B_1(t_2) & B_2(t_2) & B_3(t_2) \\ 0 & B_2(t_3) & B_3(t_3) \end{pmatrix} .$$

Die letzte Matrix zerfällt wieder in eine $(1,1)$- und eine $(2,2)$-Matrix. Insgesamt genügt es also, die Invertierbarkeit der $(1,1)$- und der $(2,2)$-Matrizen nachzuweisen, also die Fälle $n = 2,3$ zu betrachten. Damit ist der Satz für $k = 2$ bewiesen.

Sei nun die Behauptung richtig bis zur Ordnung $< k$ für ein $k \geq 2$. Der Fall $n = k$ ist trivial. Eine typische Stelle der Matrix für die Ordnung k sieht dann so aus:

$$\begin{array}{ccc} & B_j(t_{j-1}) & B_{j+1}(t_{j-1}) \\ B_{j-1}(t_j) & B_j(t_j) & B_{j+1}(t_j) \end{array}$$

Wäre hier $B_{j+1}(t_j) = 0$, d.h. $t_j \leq x_{j+1}$, so wären erst recht alle Elemente rechts und oberhalb dieses Elementes 0. Entsprechend würde aus $B_{j-1}(t_j) = 0$, d.h. $t_j \leq x_{j+k-1}$, folgen, daß alle Elemente links und unterhalb dieses Elementes 0 wären. In beiden Fällen zerfiele die Matrix in kleinere, und wir könnten Induktion nach n treiben. Wir können also annehmen, daß $x_{j+1} < t_j < x_{j+k-1}, \; j = 0, \ldots, n-k$.

Sei nun a ein Vektor mit den Komponenten $a_0, \ldots, a_{n-k}$ und $Ba = 0$. Der Spline

$$s = \sum_{j=0}^{n-k} a_j B_{j,k}$$

wäre eine C^{n-k}-Funktion mit den $n-k+3$ Nullstellen $t_0, \ldots, t_{n-k}, \; x_0, \, x_n$. Nach dem Satz von Rolle hätte s' $n-k+2$ Nullstellen

$$\tau_0 \in (x_0, t_0) \, , \quad \tau_{n-k+1} \in (t_{n-k}, x_n) \, ,$$
$$\tau_j \in (t_{j-1}, t_j) \, , \quad j = 1, \ldots, n-k \, .$$

Wegen $x_{j+1} < t_j < x_{j+k-1}$ muß dann

$$\tau_0 \in (x_0, x_{k-1})\ , \quad \tau_{n-k+1} \in (x_{n-k+1}, x_n)\ ,$$
$$\tau_j \in (x_j, x_{j+k-1})\ , \quad j = 1, \ldots, n-k$$

gelten, wie man sich an Hand einer Figur klarmacht.

Mit $\tau_0, \ldots, \tau_{n-k+1}$ haben wir $n-k+2$ Nullstellen von s' gefunden, die in den Trägern von $B_{0,k-1}, \ldots, B_{n-k+1,k-1}$ liegen. s' löst also das Interpolationsproblem für Splines der Ordnung $k-1$ mit den Interpolationsstellen $\tau_0, \ldots, \tau_{n-k+1}$ und den Werten 0. Nach Induktionsannahmen ist dieses Problem eindeutig lösbar. Eine Lösung ist offenbar $s' = 0$. Also ist $s' = 0$ und damit $s = 0$. Damit ist auch $a = 0$.

□

Der Beweis des Satzes gibt auch eine Methode, den interpolierenden Spline zu berechnen. Man löse das Gleichungssystem

$$Ba = y\ , \quad a = \begin{pmatrix} a_0 \\ \vdots \\ a_{n-k} \end{pmatrix}\ , \quad y = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-k} \end{pmatrix}$$

mit der Matrix B des Beweises. Diese Matrix ist eine Bandmatrix mit der Bandbreite k, der Rechenaufwand also entsprechend gering.

Bei speziellen Interpolationsproblemen kann man eine explizite Form des Gleichungssystems angeben. Unter dem natürlichen kubischen Spline verstehen wir einen Spline der Ordnung 4, der außerhalb $x_0, \ldots, x_n$ linear ist. Der interpolierende natürliche kubische Spline s erüllt

$$\begin{aligned} s(x_j) &= y_j\ , \quad j = 0, \ldots, n \\ s''(x_0) &= s''(x_n) = 0\ . \end{aligned}$$

Wir werden zeigen, daß s eindeutig bestimmt und leicht zu berechnen ist. Wir machen in $[x_j, x_{j+1}]$ den Ansatz

$$\begin{aligned} s(x) &= y_j + \left(\frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6} h_j \right)(x - x_j) \\ &\quad + \frac{M_j}{2}(x - x_j)^2 + \frac{M_{j+1} - M_j}{6h_j}(x - x_j)^3 \end{aligned}$$

mit $h_j = x_{j+1} - x_j$ und gewissen Zahlen M_j. Man bestätigt leicht

$$\begin{aligned}
s(x_j \pm 0) &= y_j \ , \quad s''(x_j \pm 0) = M_j \ , \\
s'(x_j + 0) &= \frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6} h_j \ , \\
s'(x_j - 0) &= \frac{y_j - y_{j-1}}{h_{j-1}} + \frac{2M_j + M_{j-1}}{6} h_{j-1} \ .
\end{aligned}$$

s ist also genau dann Lösung unseres Interpolationsproblems, wenn

$$\frac{y_{j+1} - y_j}{h_j} - \frac{2M_j + M_{j+1}}{6} h_j = \frac{y_j - y_{j-1}}{h_{j-1}} + \frac{2M_j + M_{j-1}}{6} h_{j-1} \ , \quad j = 1, \ldots, n-1$$

$$M_0 = M_n = 0 \ .$$

Dies ist ein lineares Gleichungssystem von $n-1$ Gleichungen für die $n-1$ Unbekannten $M_1, \ldots, M_{n-1}$. Es lautet in Matrix-Form

$$AM = b \ , \quad M = \begin{pmatrix} M_1 \\ \vdots \\ M_{n-1} \end{pmatrix} \ , \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_{n-1} \end{pmatrix} \ , \quad b_j = \frac{y_{j+1} - y_j}{h_j} - \frac{y_j - y_{j-1}}{h_{j-1}}$$

$$A = \frac{1}{3} \begin{pmatrix} h_0 + h_1 & \frac{h_1}{2} & & \\ \frac{h_1}{2} & h_1 + h_2 & \frac{h_2}{2} & \\ & & \cdots & \\ & & \frac{h_{n-1}}{2} & h_{n-2} + h_{n-1} \end{pmatrix} \ .$$

A ist offenbar invertierbar, und es gilt sogar

$$k(A) = \|A\|_\infty \|A^{-1}\|_\infty \leq \frac{\max\limits_j (h_j + h_{j+1})}{\min\limits_j (h_j + h_{j+1})} \ .$$