Mathematik IV für Elektrotechnik
Mathematik III für Informatik

Vorlesungsskriptum

Stefan Ulbrich

Fachbereich Mathematik
Technische Universität Darmstadt

Sommersemester 2011

Inhaltsverzeichnis

Numerische Mathematik

Kapitel 1

Einführung

Viele Problemstellungen aus den Ingenieur- und Naturwissenschaften lassen sich durch mathematische Modelle beschreiben, in denen häufig lineare oder nichtlineare Gleichungssysteme, Integrale, Eigenwertprobleme, gewöhnliche oder partielle Differentialgleichungen auftreten. In nahezu allen praxisrelevanten Fällen läßt das mathematische Modell keine analytische Lösung zu. Vielmehr muss die Lösung durch geeignete Verfahren auf einem Rechner näherungsweise bestimmt werden. Hierbei ist es wichtig, dass das verwendete Verfahren robust, genau und möglichst schnell ist. Die Entwicklung derartiger Verfahren ist Gegenstand der Numerischen Mathematik, einem inzwischen sehr bedeutenden Gebiet der Angewandten Mathematik. Die Numerische Mathematik entwickelt effiziente rechnergestützte Verfahren zur Lösung mathematischer Problemstellungen, unter anderem der oben genannten. Die Vorlesung gibt eine Einführung in die numerische Behandlung der folgenden Problemstellungen

- Interpolation
- Lineare Gleichungssysteme
- Nichtlineare Gleichungssysteme
- Eigenwertprobleme
- Numerische Integration
- Anfangswertprobleme für gewöhnliche Differentialgleichungen
- Partielle Differentialgleichungen (gegebenenfalls ganz kurz)

Kapitel 2

Interpolation

Häufig liegen von einem funktionalen Zusammenhang $y = f(x)$, $f : [a, b] \to \mathbb{R}$ nur eine begrenzte Zahl von Werten $y_i = f(x_i)$, $i = 0, \ldots, n$, vor, man möchte jedoch $f(x)$ für beliebiges $x \in [a, b]$ näherungsweise berechnen, plotten, etc.. Dies führt auf das

Interpolationsproblem: Suche eine einfache Ersatzfunktion $\Phi(x)$ mit

$$\Phi(x_i) = y_i, \quad i = 0, \ldots, n.$$

Wunsch: Der Fehler $f(x) - \Phi(x)$ sollte auf $[a, b]$ klein sein.

Beispiele:

1. Die Funktion $f(x)$ ist aufwändig zu berechnen (z.B. $\sin(x)$, $\exp(x)$, $\ln(x)$, $\Gamma(x)$, etc.) und es sind nur die Werte $y_i = f(x_i)$, $i = 0, \ldots, n$, bekannt.
   Gesucht: Genaue Approximation $\Phi(x)$ für $f(x)$, oder $\Phi'(x)$ für $f'(x)$.

2. Ein Experiment (oder eine numerische Berechnung) beschreibt einen unbekannten funktionalen Zusammenhang $y = f(x)$ und liefert zu Eingangsparametern x_i die Werte y_i.
   Gesucht: Gutes Modell $\Phi(x)$ für das unbekannte $f(x)$.

3. Ein digitales Audiosignal (CD, MP3-Player, DVD, ...) liefert zum Zeitpunkt t_i, $i = 0, \ldots, n$, die Amplitude y_i.
   Gesucht: Wie sieht das zugehörige analoge Audiosignal $y(t)$ aus?

4. Ein digitales Audiosignal (t_i, y_i), $i = 0, \ldots, n$, zur Abtastrate 44,1 kHz (CD) soll umgesampelt werden auf die Abtastrate 48 kHz (DAT, DVD-Video).
   Gesucht: $(\tilde{t}_j, y(\tilde{t}_j))$ für die 48 kHz-Abtastzeiten $\tilde{t}_j$.

5. 2D-Beispiel: Durch Datenpunkte (x_i, y_i, z_i) soll eine glatte Fläche $(x, y, z(x, y))$ gelegt werden (CAD, Computergrafik, Laserscanner, etc.).

Formale Aufgabenstellung

Gegeben sei eine Ansatzfunktion $\Phi(x; a_0, \dots, a_n)$, $x \in \mathbb{R}$, die von Parametern $a_0, \dots, a_n \in \mathbb{R}$ abhängt. In diesem Kapitel beschäftigen wir uns mit der folgenden

Interpolationsaufgabe: Zu gegebenen Paaren

$$(x_i, y_i), \quad i = 0, \dots, n \ \text{ mit } x_i, y_i \in \mathbb{R}, \quad x_i \neq x_j \text{ für } i \neq j$$

sollen die Parameter $a_0, \dots, a_n$ so bestimmt werden, dass die Interpolationsbedingungen gelten

$$\Phi(x_i; a_0, \dots, a_n) = y_i, \quad i = 0, \dots, n.$$

Die Paare (x_i, y_i) werden als *Stützpunkte* bezeichnet.

2.1 Polynominterpolation

Sehr verbreitet ist die Polynominterpolation. Hier verwendet man als Ansatzfunktion Polynome vom Grad $\leq n$, also

$$p_n(x) = \Phi(x; a_0, \dots, a_n) = a_0 + a_1 x + \dots + a_n x^n.$$

Die Interpolationsaufgabe lautet dann: Finde ein Polynom $p_n(x)$ vom Grad $\leq n$, das die Interpolationsbedingungen erfüllt

$$p_n(x_i) = y_i, \quad i = 0, \dots, n. \tag{2.1}$$

Naiver Lösungsansatz: Ein naheliegender, aber in der Praxis untauglicher Ansatz ist folgender: (2.1) liefert die $n+1$ linearen Gleichungen

$$a_0 + x_i a_1 + x_i^2 a_2 + \dots + x_i^n a_n = y_i, \quad i = 0, \dots, n,$$

für die $n+1$ Koeffizienten $a_0, \dots, a_n$. In Matrixform lautet das Gleichungssystem

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdot & x_0^n \\ 1 & x_1 & x_1^2 & \cdot & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdot & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}. \tag{2.2}$$

Gründe für Unbrauchbarkeit des Verfahrens:

- Das Auflösen des Gleichungssystems (2.2) ist mit $O(n^3)$ elementaren Rechenoperationen im Vergleich zu den nachfolgenden $O(n^2)$-Verfahren sehr teuer.

- Die Koeffizientenmatrix in (2.2) (Vandermonde-Matrix) ist zwar invertierbar, aber für größere n *extrem schlecht konditioniert*. Daher kann das Gleichungssystem (2.2) auf einem Computer nicht genau gelöst werden, da Rundungsfehler wegen der schlechten Kondition dramatisch verstärkt werden (siehe Kapitel 3).

2.1.1 Interpolationsformel von Lagrange

Als numerisch stabile und effiziente Lösung der Interpolationsaufgabe bietet sich folgendes Vorgehen an: Wir betrachten das

Lagrangesche Interpolationspolynom

$$p_n(x) = \sum_{k=0}^{n} y_k L_{k,n}(x) \quad \text{mit} \quad L_{k,n}(x) = \prod_{\substack{j=0\\ j\neq k}}^{n} \frac{x - x_j}{x_k - x_j}. \tag{2.3}$$

Die Lagrange-Polynome sind gerade so gewählt, dass gilt

$$L_{k,n}(x_i) = \begin{cases} 1 & \text{falls } k = i, \\ 0 & \text{sonst.} \end{cases} =: \delta_{ki}.$$

δ_{ki} ist das Kronecker-Symbol. p_n in (2.3) erfüllt die Interpolationsbedingungen (2.1), denn

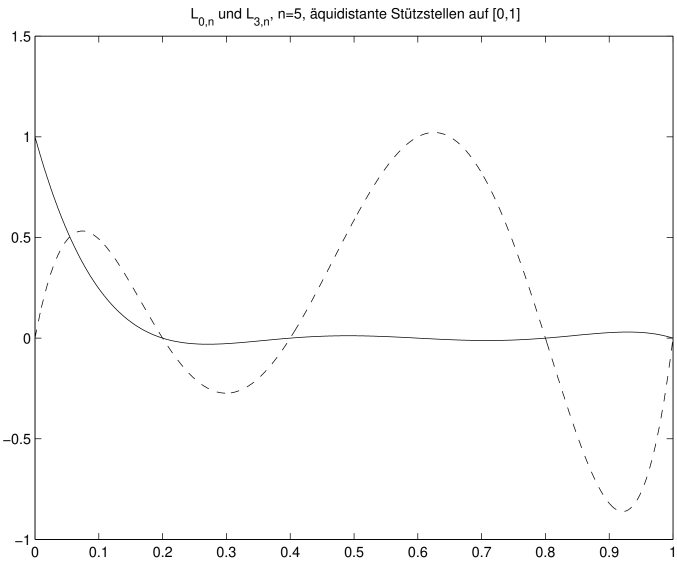


Abbildung 2.1: $L_{0,5}$ und $L_{3,5}$ für äquidistante Stützstellen auf $[0, 1]$.

$$p_n(x_i) = \sum_{k=0}^{n} y_k L_{k,n}(x_i) = \sum_{k=0}^{n} y_k \delta_{ki} = y_i.$$

Tatsächlich ist dies die einzige Lösung der Interpolationsaufgabe:

Satz 2.1.1 *Es gibt genau ein Polynom $p_n(x)$ vom Grad $\leq n$, das die Interpolationsbedingungen* (2.1) *erfüllt, nämlich* (2.3).

Beweis: Das Polynom (2.3) hat Grad $\leq n$ und erfüllt (2.1). Gäbe es eine weitere Lösung $\tilde{p}_n(x)$, dann ist $p_n(x) - \tilde{p}_n(x)$ ein Polynom vom Grad $\leq n$ mit $n+1$ verschiedenen Nullstellen $x_0, \dots, x_n$, muss also identisch 0 sein. $\square$

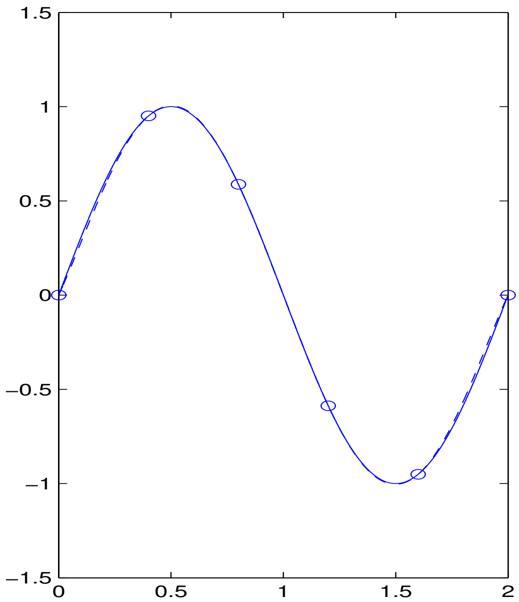
Bemerkung: (2.3) zeigt, dass p_n linear von y_k abhängt. $\square$

Die Darstellung (2.3) von Lagrange ist für theoretische Zwecke sehr nützlich und wird auch in der Praxis oft angewendet.

Vorteile:

- Der Rechenaufwand beträgt:
  Koeffizientenberechnung (Nenner in (2.3)): $O(n^2)$
  Auswertung von $p_n(x)$: $O(n)$
- Intuitive, bequeme Darstellung.

Beispiel: Polynominterpolant von $f(x) = \sin(\pi x)$ auf $[0, 2]$ für $n = 5$ und äquidistante Stützstellen $x_i = \frac{2i}{5}$.



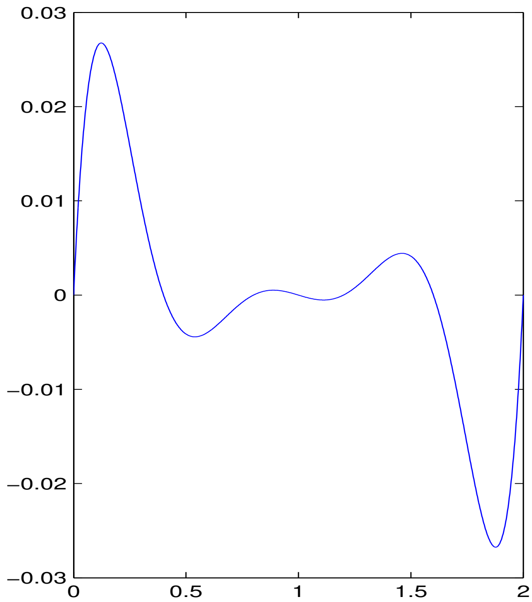


Abbildung 2.2: $\sin(\pi x)$ und $p_5(x)$ (gestrichelt) Fehler $\sin(\pi x) - p_5(x)$.

In der Praxis, insbesondere wenn die effiziente Hinzunahme weiterer Stützstellen möglich sein soll, ist die folgende *Newtonsche Interpolationsformel* angenehmer.

2.1.2 Newtonsche Interpolationsformel

Wir wählen als Ansatz die *Newtonsche Darstellung*

$$p_n(x) = \gamma_0 + \gamma_1(x - x_0) + \gamma_2(x - x_0)(x - x_1) + \ldots + \gamma_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

Einsetzen in (2.1) liefert nun

$$\begin{aligned}
&& \gamma_0 &= y_0 \\
\gamma_0 + \gamma_1(x_1 - x_0) = y_1 &\Longrightarrow & \gamma_1 &= \frac{y_1 - y_0}{x_1 - x_0} \\
\gamma_0 + \gamma_1(x_2 - x_0) + \gamma_2(x_2 - x_0)(x_2 - x_1) = y_2 &\Longrightarrow & \gamma_2 &= \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \\
&& &\vdots
\end{aligned}$$

Man bezeichnet $f_{[x_0,\ldots,x_i]} := \gamma_i$ als die i-te *dividierte Differenz* zu den Stützstellen $x_0, \ldots, x_i$, wobei $f_{[x_0]} = \gamma_0 = y_0$.

Allgemein berechnen sich die dividierten Differenzen zu den Stützstellen $x_j, \ldots, x_{j+k}$ über die Rekursion

$$\begin{aligned}
&j = 0, \ldots, n: \quad f_{[x_j]} = y_j \\
k = 1, \ldots, n: \quad &j = 0, \ldots, n-k: \quad f_{[x_j,\ldots,x_{j+k}]} = \frac{f_{[x_{j+1},\ldots,x_{j+k}]} - f_{[x_j,\ldots,x_{j+k-1}]}}{x_{j+k} - x_j}.
\end{aligned} \tag{2.4}$$

Man erhält das

Newtonsche Interpolationspolynom

$$p_n(x) = \gamma_0 + \sum_{i=1}^{n} \gamma_i(x - x_0)\cdots(x - x_{i-1}), \quad \gamma_i = f_{[x_0,\ldots,x_i]} \tag{2.5}$$

mit den dividierten Differenzen $f_{[x_0,\ldots,x_i]}$ aus (2.4).

Begründung: Für $n = 0$ ist die Darstellung klar. Sind $p_{1,\ldots,i+1}$ und $p_{0,\ldots,i}$ die Interpolanten in $x_1, \ldots, x_{i+1}$ bzw. $x_0, \ldots, x_i$ vom Grad $\leq i$, dann gilt

$$\begin{aligned}
p_{i+1}(x) &= \frac{(x - x_0)p_{1,\ldots,i+1}(x) + (x_{i+1} - x)p_{0,\ldots,i}(x)}{x_{i+1} - x_0} \\
&= \frac{f_{[x_1,\ldots,x_{i+1}]} - f_{[x_0,\ldots,x_i]}}{x_{i+1} - x_0}(x - x_0)\cdots(x - x_i) + \underbrace{\text{Polynom vom Grad } i}_{:=q_i(x)}.
\end{aligned}$$

Da der erste Summand in $x_0, \ldots, x_i$ verschwindet, gilt $q_i(x) = p_i(x)$ wegen (2.1). Vergleich mit (2.5) liefert (2.4). □

Wir erhalten aus (2.4) folgende Vorschrift zur Berechnung der Koeffizienten $\gamma_i = f_{[x_0,\ldots,x_i]}$:

Berechnung der dividierten Differenzen:

Setze $f_{[x_j]} = y_j$, $j = 0, \ldots, n$.

Berechne für $k = 1, \ldots, n$ *und* $j = 0, \ldots, n-k$*:*

$$f_{[x_j,\ldots,x_{j+k}]} = \frac{f_{[x_{j+1},\ldots,x_{j+k}]} - f_{[x_j,\ldots,x_{j+k-1}]}}{x_{j+k} - x_j}.$$

Wir erhalten also das Schema

$$\begin{array}{c|lll}
x_0 & f_{[x_0]} = y_0 \searrow & & \\
 & & f_{[x_0,x_1]} \searrow & \\
x_1 & f_{[x_1]} = y_1 \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & f_{[x_0,x_1,x_2]} \\
 & & f_{[x_1,x_2]} \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & \\
x_2 & f_{[x_2]} = y_2 \nearrow & & \\
\vdots & & &
\end{array}$$

Vorteile:

- Der Rechenaufwand beträgt:
  Berechnung der dividierten Differenzen: $O(n^2)$
  Auswertung von $p_n(x)$: $O(n)$
- Hinzunahme einer neuen Stützstelle erfordert nur die Berechnung von n zusätzlichen dividierten Differenzen.

2.1.3 Fehlerabschätzungen

Nimmt man an, dass die Stützwerte von einer Funktion $f : [a,b] \to \mathbb{R}$ herrühren, also

$$y_i = f(x_i), \quad i = 0, \ldots, n,$$

dann erhebt sich die Frage, wie gut das Interpolationspolynom p_n auf $[a,b]$ mit f übereinstimmt. Es gilt der folgende Satz:

Satz 2.1.2 *Sei f $(n+1)$-mal stetig differenzierbar, kurz $f \in C^{n+1}([a,b])$. Seien $x_0, \ldots, x_n \in [a,b]$ verschiedene Punkte und sei p_n das eindeutige Interpolationspolynom vom Grad $\le n$ zu den Stützwerten $(x_i, f(x_i))$, $i = 0, \ldots, n$. Dann existiert zu jedem $x \in [a,b]$ ein $\xi_x \in [a,b]$ mit*

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0) \cdots (x - x_n).$$

Das Restglied der Interpolation hat also zwei Faktoren: Das sogenannte *Knotenpolynom*

$$\omega(x) = \prod_{i=0}^{n} (x - x_i)$$

und den Faktor $\frac{f^{(n+1)}(\xi_x)}{(n+1)!}$. Durch Abschätzung beider Terme ergibt sich zum Beispiel folgende Fehlerabschätzung.

Korollar 2.1.3 *Unter den Voraussetzungen von Satz 2.1.2 gilt*

$$\max_{x\in[a,b]} |f(x) - p_n(x)| \le \max_{x\in[a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} \max_{x\in[a,b]} |\omega(x)| \le \max_{x\in[a,b]} \frac{|f^{(n+1)}(x)|}{(n+1)!} (b-a)^{n+1}.$$

Achtung: Bei äquidistanter Wahl der Stützpunkte, also $x_i = a + ih$, $h = (b-a)/n$, ist nicht immer gewährleistet, dass gilt

$$\lim_{n\to\infty} f(x) - p_n(x) = 0 \quad \text{für alle } x \in [a,b].$$

Zum Beispiel ist dies für $f(x) = \frac{1}{1+x^2}$ auf $[a,b] = [-5,5]$ der Fall. $\Box$

Als Ausweg kann man x_i als die sog. *Tschebyscheff-Abszissen* wählen, für die $\max_{x\in[a,b]} |\omega(x)|$ minimal wird: Wahl der

Tschebyscheffabszissen

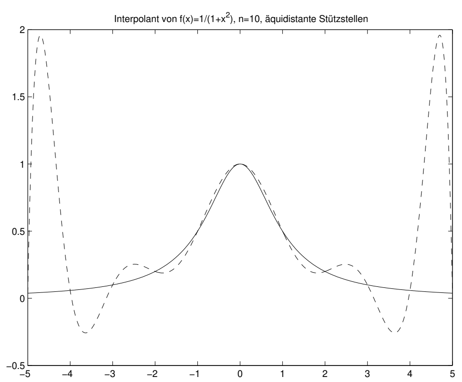
$$x_i = \frac{b-a}{2} \cos\left(\frac{2i+1}{n+1}\frac{\pi}{2}\right) + \frac{b+a}{2}, \quad i = 0, \ldots, n. \tag{2.6}$$

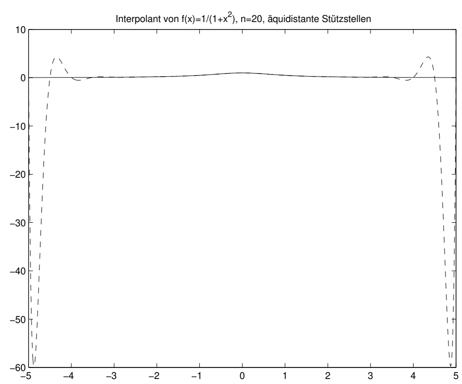
liefert den minimalen Wert für $\max_{x\in[a,b]} |\omega(x)|$, nämlich

$$\max_{x\in[a,b]} |\omega(x)| = \left(\frac{b-a}{2}\right)^{n+1} 2^{-n}.$$

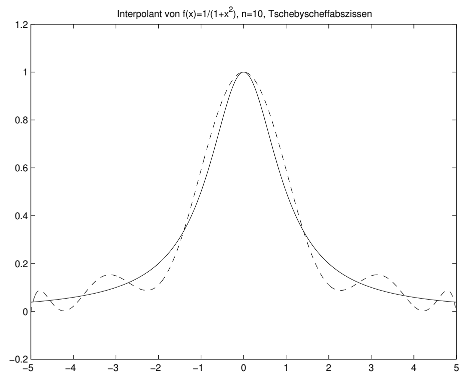
Beipiel: $f(x) = \frac{1}{1+x^2}$ auf $[a,b] = [-5,5]$. Wie bereits erwähnt, geht bei äquidistanten Stützstellen der Fehler $f(x) - p_n(x)$ für $n \to \infty$ nicht an allen Stellen $x \in [a,b]$ gegen 0.

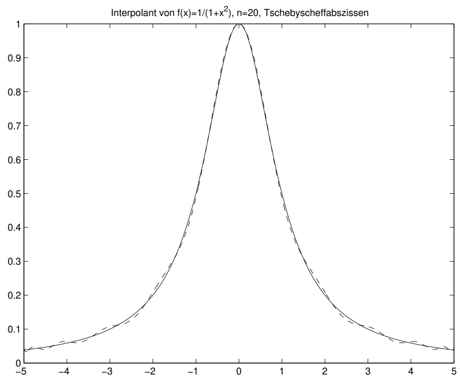
Interpolant bei äquidistanten Stützstellen:





Interpolant bei Tschebyscheffstützstellen:





Allgemein sollte man in der Praxis nicht n sehr groß wählen, sondern besser stückweise in kleinen Intervallen vorgehen, siehe 2.2.

2.1.4 Anwendungen der Polynominterpolation

Wir geben eine Auswahl von Anwendungen für die Polynominterpolation an:

1. **Approximation einer Funktion auf einem Intervall:** Wir haben gesehen, dass hierzu nicht äquidistante Stützstellen sondern die Tschbyscheffabszissen gewählt werden sollten.

2. **Inverse Interpolation:** Sei $f : [a, b] \to \mathbb{R}$ bijektiv, also $f'(x) \neq 0$ auf $[a, b]$. Sind dann (x_i, y_i), $y_i = f(x_i)$, Stützpunkte von f, dann sind (y_i, x_i) wegen $x_i = f^{-1}(y_i)$ Stützpunkte für f^{-1} und eine Approximation von f^{-1} kann durch Interpolation der Stützpunkte (y_i, x_i) gewonnen werden.

3. **Numerische Integration:** (Kapitel 3)
   Zur näherungsweisen Berechnung des Integrals einer Funktion kann man zunächst ein Interpolationspolynom bestimmen, das anschließend einfach integriert werden kann: $\int_a^b f(x)\,dx \approx \int_a^b p_n(x)\,dx.$

4. **Numerische Differentiation:** Mit einem Interpolationspolynom p_n von f ist p_n' eine Approximation von f'.

2.2 Spline-Interpolation

Bei der Polynominterpolation wird die Funktion f auf dem Intervall $[a, b]$ durch *ein* Polynom vom Grad n interpoliert. Wir hatten festgestellt, dass große Genauigkeit nicht immer durch die Wahl vieler Stützstellen sichergestellt werden kann.

Als Ausweg kann man stückweise Interpolation verwenden. Hierbei zerlegt man das Ausgangsintervall $[a, b]$ in kleine Teilintervalle und verwendet auf jedem Teilintervall ein interpolierendes Polynom fester Ordnung. An den Intervallgrenzen sorgt man dafür, dass die Polynome k-mal stetig differenzierbar ineinander übergehen, wobei k fest ist, und die Welligkeit des Interpolanten möglichst klein ist. Dieses Konzept führt auf die Spline-Interpolation.

2.2.1 Grundlagen

Sei

$$\Delta = \{x_i \, : \, a = x_0 < x_1 < \ldots < x_n = b\}$$

eine Zerlegung des Intervalls $[a, b]$. Aus historischen Gründen nennt man die x_i *Knoten*.

Definition 2.2.1 *Eine* Splinefunktion der Ordnung l zur Zerlegung Δ *ist eine Funktion* $s : [a, b] \to \mathbb{R}$ *mit folgenden Eigenschaften*

- *Es gilt* $s \in C^{l-1}([a, b])$, s *ist also stetig und* $l - 1$*-mal stetig differenzierbar.*
- s *stimmt auf jedem Intervall* $[x_i, x_{i+1}]$ *mit einem Polynom* s_i *vom Grad* $\leq l$ *überein.*

Die Menge dieser Splinefunktionen bezeichnen wir mit $S_{\Delta,l}$.

Im Folgenden betrachten wir nur den Fall $l = 1$ (*lineare Splines*) und $l = 3$ (*kubische Splines*).

Wir wollen nun Splines zur Interpolation verwenden und betrachten folgende Aufgabenstellung:

Spline-Interpolation:

Zu einer Zerlegung $\Delta = \{x_i \, : \, a = x_0 < x_1 < \ldots < x_n = b\}$ und Werten $y_i \in \mathbb{R}$, $i = 0, \ldots, n$ bestimme $s \in S_{\Delta,l}$ mit

$$s(x_i) = y_i, \quad i = 0, \ldots, n. \tag{2.7}$$

2.2.2 Interpolation mit linearen Splines

Ein linearer Spline $s \in S_{\Delta,1}$ ist stetig und auf jedem Intervall $[x_i, x_{i+1}]$ ein Polynom s_i vom Grad ≤ 1. Die Interpolationsbedingungen (2.7) erfordern daher $s_i(x_i) = y_i$, $s_i(x_{i+1}) = y_{i+1}$ und legen s_i eindeutig fest zu

$$s(x) = s_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} y_i + \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} \quad \forall\, x \in [x_i, x_{i+1}]. \tag{2.8}$$

Definieren wir die ”Dachfunktionen”

$$\varphi_i(x) = \begin{cases} 0 & \text{falls } x < x_{i-1}, \\ \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{falls } x \in [x_{i-1}, x_i], \\ \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{falls } x \in [x_i, x_{i+1}], \\ 0 & \text{falls } x > x_{i+1}. \end{cases}$$

mit beliebigen Hilfsknoten $x_{-1} < a$ und $x_{n+1} > b$, dann erhalten wir für $s(x)$ auf $[a, b]$ die bequeme Darstellung

$$s(x) = \sum_{i=0}^{n} y_i \varphi_i(x), \quad x \in [a, b].$$

Satz 2.2.2 *Zu einer Zerlegung* $\Delta = \{x_i : a = x_0 < x_1 < \ldots < x_n = b\}$ *von* $[a, b]$ *und Werten* y_i*,* $i = 0, \ldots, n$*, existiert genau ein interpolierender linearer Spline.*

Ferner gilt folgende Fehlerabschätzung.

Satz 2.2.3 *Sei* $f \in C^2([a, b])$*. Dann gilt für jede Zerlegung* $\Delta = \{x_i \,;\, a = x_0 < x_1 < \ldots < x_n = b\}$ *von* $[a, b]$ *und den zugehörigen interpolierenden linearen Spline* $s \in S_{\Delta,1}$ *von* f

$$\max_{x\in[a,b]} |f(x) - s(x)| \leq \frac{1}{8} \max_{x\in[a,b]} |f''(x)| h_{max}^2 \quad \textit{mit } h_{max} = \max_{i=0,\ldots,n-1} x_{i+1} - x_i.$$

Beweis: Auf jedem Intervall $[x_i, x_{i+1}]$ ist s ein interpolierendes Polynom vom Grad ≤ 1. Daher gilt nach Satz 2.1.2

$$|f(x) - s(x)| = \frac{|f''(\xi)|}{2!}(x_{i+1} - x)(x - x_i) \leq \frac{|f''(\xi)|}{2!} \frac{h_{max}^2}{4} \quad \forall\, x \in [x_i, x_{i+1}]$$

mit einem $\xi \in [x_i, x_{i+1}]$. Daraus folgt unmittelbar die Behauptung. □

2.2.3 Interpolation mit kubischen Splines

Kubische Splines sind zweimal stetig differenzierbar aus kubischen Polynomen zusammengesetzt. Wir werden sehen, dass die Interpolation mit kubischen Splines es gestattet, gegebene Punkte durch eine Funktion minimaler Krümmung zu interpolieren.

Berechnung kubischer Spline-Interpolanten

Ist $s \in S_{\Delta,3}$ ein kubischer Spline, dann ist s'' offensichtlich stetig und stückweise linear, also $s'' \in S_{\Delta,1}$. Es bietet sich daher an, s_i durch Integration von s_i'' zu bestimmen.

Seien $M_i = s_i''(x_i)$. Man nennt M_i *Momente*. Dann gilt nach (2.8)

$$s_i''(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i} M_i + \frac{x - x_i}{x_{i+1} - x_i} M_{i+1}.$$

Zweifache Integration ergibt dann den Ansatz

$$s_i(x) = \frac{1}{6}\left(\frac{(x_{i+1} - x)^3}{x_{i+1} - x_i} M_i + \frac{(x - x_i)^3}{x_{i+1} - x_i} M_{i+1}\right) + c_i(x - x_i) + d_i$$

mit Konstanten $c_i, d_i \in \mathbb{R}$. Wir berechnen c_i und d_i aus den Bedingungen

$$s_i(x_i) = y_i, \quad s_i(x_{i+1}) = y_{i+1}.$$

Mit

$$h_i = x_{i+1} - x_i$$

liefert dies

$$d_i = y_i - \frac{h_i^2}{6} M_i, \quad c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(M_{i+1} - M_i).$$

Einsetzen in die Gleichungen $s_i'(x_i) = s_{i-1}'(x_i)$ ergibt schließlich folgende Gleichungen für die Momente M_i:

$$\text{(2.9)} \quad \frac{h_{i-1}}{6} M_{i-1} + \frac{h_{i-1} + h_i}{3} M_i + \frac{h_i}{6} M_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}, \quad i = 1, \ldots, n-1.$$

Dies sind $n - 1$ Gleichungen für $n + 1$ Unbekannte. Der Spline-Interpolant wird eindeutig durch zwei zusätzlich Randbedingungen:

Wichtige Randbedingungen für kubische Splines:

a) Natürliche Randbedingungen: $s''(a) = s''(b) = 0$, also $M_0 = M_n = 0$

b) Hermite-Randbedingungen: $s'(a) = f'(a)$, $s'(b) = f'(b)$, also

$$\frac{h_0}{3} M_0 + \frac{h_0}{6} M_1 = \frac{y_1 - y_0}{h_0} - f'(a), \quad \frac{h_{n-1}}{3} M_n + \frac{h_{n-1}}{6} M_{n-1} = f'(b) - \frac{y_n - y_{n-1}}{h_{n-1}}.$$

Für jeden der Fälle a)-b) ergibt sich zusammen mit (2.9) eine eindeutige Lösung für $M_0, \ldots, M_n$.

Für a) und b) erhält man ein strikt diagonaldominantes tridiagonales Gleichungssystem der Form

$$\text{(2.10)} \quad \begin{pmatrix} \mu_0 & \lambda_0 & & & & \\ \frac{h_0}{6} & \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{h_{i-1}}{6} & \frac{h_{i-1}+h_i}{3} & \frac{h_i}{6} & \\ & & & \ddots & \ddots & \ddots \\ & & & & \lambda_n & \mu_n \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} b_0 \\ \frac{y_2-y_1}{h_1} - \frac{y_1-y_0}{h_0} \\ \vdots \\ \frac{y_{i+1}-y_i}{h_i} - \frac{y_i-y_{i-1}}{h_{i-1}} \\ \vdots \\ b_n \end{pmatrix}.$$

Für a) kann man zum Beispiel $b_0 = b_n = \lambda_0 = \lambda_n = 0$ und $\mu_0 = \mu_n = 1$ wählen. Wegen der strikten Diagonaldominanz ist nach dem Satz von Gershgorin 0 kein Eigenwert und daher ist die Koeffizientenmatrix invertierbar.

Minimaleigenschaften kubischer Splines

Es zeigt sich, dass der kubische Spline-Interpolant mit Randbedingung a) oder b) unter allen zweimal stetig differenzierbaren minimale Krümmung im folgenden Sinne hat:

Satz 2.2.4 *Gegeben sei eine beliebige Funktion $f \in C^2([a,b])$ und eine Unterteilung Δ von $[a,b]$. Dann gilt für den kubischen Spline-Interpolanten $s \in S_{\Delta,3}$ mit Randbedingungen a) oder b)*

$$\int_a^b f''(x)^2\,dx = \int_a^b s''(x)^2\,dx + \int_a^b (f''(x) - s''(x))^2\,dx \geq \int_a^b s''(x)^2\,dx.$$

Beweis: Siehe zum Beispiel [St94], [Pl00]. □

Fehlerabschätzung für kubische Spline-Interpolation

Unter Verwendung der Tatsache, dass die Momente $\hat{M}_i = f''(x_i)$ das Gleichungssystem (2.10) auf $O(h_{max}^3)$ mit $h_{max} = \max_{0\leq i<n} h_i$ erfüllen und die Norm der Inversen der Koeffizientenmatrix in (2.10) von der Ordnung $O(1/h_{min})$ ist mit $h_{min} = \min_{0\leq i<n} h_i$, kann man folgendes Resultat zeigen.

Satz 2.2.5 *Sei $f \in C^4([a,b])$ mit $f''(a) = f''(b) = 0$. Dann gilt für jede Unterteilung Δ mit dem kubischen Spline-Interpolanten $s \in S_{\Delta,3}$ zu Randbedingungen a)*

$$|f(x) - s(x)| \leq \frac{h_{max}}{h_{min}} \sup_{\xi\in[a,b]} |f^{(4)}(\xi)| h_{max}^4,$$

$$|f^{(k)}(x) - s^{(k)}(x)| \leq \frac{2h_{max}}{h_{min}} \sup_{\xi\in[a,b]} |f^{(4)}(\xi)| h_{max}^{4-k}, \quad k = 1,2.$$

Beweis: Siehe zum Beispiel [Pl00]. □

Für Hermite-Randbedingungen lässt sich der Satz verschärfen:

Satz 2.2.6 *Sei $f \in C^4([a,b])$. Dann gilt für jede Unterteilung Δ mit dem kubischen Spline-Interpolanten $s \in S_{\Delta,3}$ zu Randbedingungen b)*

$$|f(x) - s(x)| \leq \frac{5}{384} \sup_{\xi\in[a,b]} |f^{(4)}(\xi)| h_{max}^4,$$

$$|f^{(k)}(x) - s^{(k)}(x)| \leq \frac{2h_{max}}{h_{min}} \sup_{\xi\in[a,b]} |f^{(4)}(\xi)| h_{max}^{4-k}, \quad k = 1,2.$$

Beweis: Siehe zum Beispiel [DB02, Pl00, TS90]. □

Literaturverzeichnis

[DB02] P. Deuflhard, F. Hohmann. *Numerische Mathematik I.* de Gruyter, Berlin, 2008.

[DB02] P. Deuflhard, F. Bornemann. *Numerische Mathematik II.* de Gruyter, Berlin, 2002.

[Pl00] R. Plato. *Numerische Mathematik kompakt.* Vieweg Verlag, Braunschweig, 2000.

[St94] J. Stoer. *Numerische Mathematik 1.* Springer Verlag, Berlin, 1994.

[TS90] W. Törnig, P. Spellucci. *Numerische Mathematik für Ingenieure und Physiker 2.* Springer Verlag, Berlin, 1990.

[We92] J. Werner. *Numerische Mathematik 2.* Vieweg Verlag, Braunschweig, 1992.