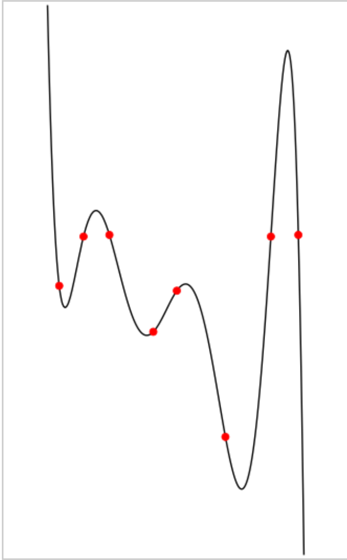
Polynominterpolation

aus Wikipedia, der freien Enzyklopädie

In der numerischen Mathematik versteht man unter **Polynominterpolation** die Suche nach einem Polynom, welches exakt durch vorgegebene Punkte (z. B. aus einer Messreihe) verläuft. Dieses Polynom wird **Interpolationspolynom** genannt und man sagt, es interpoliere die gegebenen Punkte.



Interpolationspolynom 7. Grades

Inhaltsverzeichnis

Anwendungen

Polynome lassen sich sehr leicht integrieren und ableiten. Deswegen tauchen interpolierende Polynome an vielen Stellen in der numerischen Mathematik auf, beispielsweise bei der numerischen Integration und entsprechend bei Verfahren zur numerischen Lösung gewöhnlicher Differentialgleichungen.

Problemstellung

Für $n+1$ gegebene Wertepaare $(x_i,\, f_i)$ mit paarweise verschiedenen Stützstellen x_i wird ein Polynom P maximal n-ten Grades gesucht, das alle Gleichungen

$$P(x_i) = f_i, \quad i = 0, \dots, n$$

erfüllt. Ein solches Polynom existiert stets und ist eindeutig bestimmt, wie im Folgenden gezeigt wird.

Beim Interpolationsproblem ist also P im Vektorraum Π_n der Polynome mit Grad n oder kleiner zu suchen, kurz $P \in \Pi_n$. Ist $\phi_0, \dots, \phi_n$ eine Basis von Π_n, so ergeben die Gleichungen $P(x_i) = f_i$ ein lineares Gleichungssystem für die Koeffizienten der Basisdarstellung $P = \sum_{k=0}^{n} a_k \phi_k$. Da sich ein und dasselbe Polynom aber unterschiedlich darstellen lässt, je nachdem welche Basis für den Vektorraum Π_n gewählt wird, kann man ganz verschiedene Gleichungssysteme erhalten. Wählt man für Π_n die Standardbasis $\left\{x^k \mid 0 \le k \le n\right\}$, also für P die Darstellung $P(x) = \sum_{k=0}^{n} a_k x^k$, so erhält man ein Gleichungssystem mit der Vandermonde-Matrix:

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ f_n \end{pmatrix}.$$

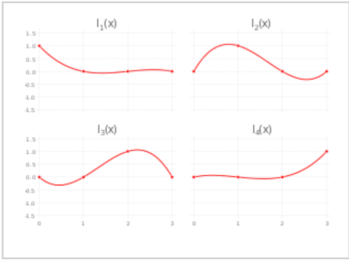
Diese ist regulär, wenn die Stützstellen x_i paarweise verschieden sind, das Gleichungssystem lässt sich dann eindeutig lösen. Somit ist die Existenz und Eindeutigkeit des gesuchten Polynoms $P \in \Pi_n$ immer sichergestellt. Trotz der theoretischen Machbarkeit wird diese Art der Interpolation in der Praxis nicht durchgeführt, da die Berechnung der Vandermonde-Matrix aufwendig ist ($\mathcal{O}(n^2)$, siehe Landau-Symbole) und zudem schlecht konditioniert bei einer ungeeigneten Wahl der Stützstellen.

Lösungsverfahren

Obiges Gleichungssystem ließe sich beispielsweise mit dem Gaußschen Eliminationsverfahren lösen. Der Aufwand dafür ist mit $\mathcal{O}(n^3)$ allerdings vergleichsweise groß. Bei Wahl einer anderen Basis als der Standardbasis zur Beschreibung des Polynoms P kann der Aufwand verringert werden.

Lagrangesche Interpolationsformel

Eher für theoretische Betrachtungen günstig ist eine Darstellung in der Lagrange-Basis. Die Basisfunktionen sind die Lagrange-Polynome



Beispielhafte lagrangesche Basisfunktionen für $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$ ($n = 3$)

Fehler beim Parsen (MathML mit SVG- oder PNG-Rückgriff (empfohlen für moderne Browser und Barrierefreiheitswerkzeuge): Ungültige Antwort („Math extension cannot connect to Restbase.“) von Server „/mathoid/local/v1/“:): \ell_i(x) = \prod_{\begin{smallmatrix}j=0\\j\neq i\end{smallmatrix}}^n \frac{x-x_j}{x_i-x_j}=\frac{x-x_0}{x_i-x_0}\cdots\frac{x-x_{i-1}}{x_i-x_{i-1}}\cdot\frac{x-x_{i+1}}{x_i-x_{i+1}}\cdots\frac{x-x_n}{x_i-x_n},

die so definiert sind, dass

$$\ell_i(x_k) = \delta_{ik} = \begin{cases} 1 & \text{falls } i = k \\ 0 & \text{falls } i \neq k \end{cases}$$

gilt, wobei δ_{ik} das Kronecker-Delta darstellt. Damit entspricht die Matrix $(\ell_i(x_j))_{i,j=0,1,\dots,n}$ genau der Einheitsmatrix. Die Lösung des Interpolationsproblems lässt sich dann einfach angeben als

$$P(x) = \sum_{i=0}^{n} f_i \ell_i(x)$$

mit den Stützwerten f_i. Dies wird häufig benutzt, um die Existenz der Lösung des Interpolationsproblems zu beweisen. Ein Vorteil der Lagrange-Basis ist somit, dass die Basisfunktionen ℓ_i von den Stützwerten f_i unabhängig sind. Dadurch lassen sich verschiedene Sätze von Stützwerten f_i mit gleichen Stützstellen x_i schnell interpolieren, wenn die Basisfunktionen ℓ_i einmal bestimmt worden sind. Ein Nachteil dieser Darstellung ist jedoch, dass alle Basisvektoren bei Hinzunahme einer einzelnen Stützstelle komplett neu berechnet werden müssen, weshalb dieses Verfahren für die meisten praktischen Zwecke zu aufwendig ist.

Newtonscher Algorithmus

In diesem Verfahren wird das Polynom P in Newton-Basis dargestellt, so dass die Koeffizienten effizient mit dem Schema der dividierten Differenzen bestimmt werden können. Eine effiziente Auswertung des Polynoms kann dann mithilfe des Horner-Schemas erfolgen.

Ansatz: Newton-Basis

Als Ansatz für das gesuchte Interpolationspolynom P wählt man die Newton-Basisfunktionen $N_0(x) = 1$ und $N_i(x) = \prod_{j=0}^{i-1}(x - x_j) = (x - x_0)\cdots(x - x_{i-1})$ mit $i = 1, \dots, n$, so dass P dargestellt wird mit der Newtonschen Interpolationsformel

$$P(x) = \sum_{i=0}^{n} c_i \cdot N_i(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots + c_n(x - x_0)\cdots(x - x_{n-1})$$

Das Gleichungssystem der Gleichungen $P(x_i) = f_i$ hat dann die Form

$$\begin{pmatrix} 1 & & & & 0 \\ 1 & (x_1 - x_0) & & & \\ 1 & (x_2 - x_0) & (x_2 - x_0)(x_2 - x_1) & & \\ \vdots & \vdots & & \ddots & \\ 1 & (x_n - x_0) & \ldots & & \prod_{i=0}^{n-1}(x_n - x_i) \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ f_n \end{pmatrix}$$

Im Gegensatz zur komplizierten Vandermonde-Matrix bei Wahl der Standardbasis $\{x^k | 0 \leq k \leq n\}$ erhält man bei Wahl der Newton-Basis also eine einfach strukturierte untere Dreiecksmatrix und das Gleichungssystem lässt sich einfach lösen.

Bestimmung der Koeffizienten: Schema der dividierten Differenzen

Die Koeffizienten c_i werden aber nicht direkt aus dem obigen Gleichungssystem bestimmt, sondern effizienter mithilfe der dividierten Differenzen. Durch Induktion beweist man mit der Rekursionsformel von Aitken, dass für die Koeffizienten c_i gilt

$$c_i = [x_0, \cdots, x_i]f.$$

Dabei sind für $i < j$ die dividierten Differenzen **Fehler beim Parsen (MathML mit SVG- oder PNG-Rückgriff (empfohlen für moderne Browser und Barrierefreiheitswerkzeuge): Ungültige Antwort („Math extension cannot connect to Restbase.“) von Server „/mathoid/local/v1/“:): [x_i, \dotsc, x_j]f** rekursiv definiert durch

$$\begin{aligned} [x_i]f &= f_i \\ [x_i, \ldots, x_j]f &= \frac{[x_{i+1}, \ldots, x_j]f - [x_i, \ldots, x_{j-1}]f}{x_j - x_i}. \end{aligned}$$

Die Notation mit angehängtem f erklärt sich dadurch, dass oft eine unbekannte Funktion f angenommen wird, die bei bekannten Funktionswerten $f_i = f(x_i)$ interpoliert werden soll.

Die rekursive Berechnung der dividierten Differenzen lässt sich wie folgt veranschaulichen. Dabei sind die gesuchten Koeffizienten c_i genau die oberste Schrägzeile:

$$\begin{array}{cccccccccc}
[x_0]f & & & & & & & & & \\
& \searrow & & & & & & & & \\
[x_1]f & \rightarrow & [x_0, x_1]f & & & & & & & \\
& \searrow & & \searrow & & & & & & \\
[x_2]f & \rightarrow & [x_1, x_2]f & \rightarrow & [x_0, x_1, x_2]f & & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & & \\
& \searrow & & \searrow & & & \searrow & & & \\
[x_{n-1}]f & \rightarrow & [x_{n-2}, x_{n-1}]f & \rightarrow & [x_{n-3}, x_{n-2}, x_{n-1}]f & \cdots & \rightarrow & [x_0 \ldots x_{n-1}]f & & \\
& \searrow & & \searrow & & & \searrow & & \searrow & \\
[x_n]f & \rightarrow & [x_{n-1}, x_n]f & \rightarrow & [x_{n-2}, x_{n-1}, x_n]f & \cdots & \rightarrow & [x_1 \ldots x_n]f & \rightarrow & [x_0 \ldots x_n]f
\end{array}$$

Offensichtlich ist bei Ergänzung der $n+1$ Wertepaare (x_i, f_i) um einen weiteren Punkt (x_{n+1}, f_{n+1}) in obigem Schema nur eine weitere Zeile hinzuzufügen, um den zusätzlichen Koeffizienten $c_{n+1} = [x_0, \ldots, x_{n+1}]f$ zu berechnen. Die zuvor bestimmten Koeffizienten $c_0, \ldots, c_n$ müssen nicht neu berechnet werden.

Alternativ zur obigen rekursiven Definition wird zum Beispiel in einem der Artikel von Marsden[1] die *dividierte Differenz* $[x_0, \ldots, x_n]f$ einer hinreichend oft differenzierbaren Funktion f als der eindeutige Koeffizient zur höchsten Potenz von x eines Polynoms n-ten Grads $p(x)$ definiert, das f an den Stellen $x_0, \ldots, x_n$ interpoliert. Tritt dabei ein Wert in der Sequenz $x_0, \ldots, x_n$ mit der Vielfachheit $\nu \geq 1$ auf, so sollen die Ableitungen des Polynoms die Ableitungen der Funktion f an dieser Stelle bis zur Ordnung $\nu - 1$ interpolieren.

Auswertung des Polynoms: Horner-Schema

Wenn die Koeffizienten c_i des Interpolationspolynoms P einmal bekannt sind, kann man es effizient mithilfe des Horner-Schemas auswerten. Dazu schreibt man P in der Form (einfache Umformung der Newtonschen Interpolationsformel)

$$P(x) = (\cdots(c_n(x - x_{n-1}) + c_{n-1})(x - x_{n-2}) + \cdots + c_1)(x - x_0) + c_0 \ ,$$

so dass $P(x)$ rekursiv berechnet werden kann durch

$$\begin{aligned} b_n &= c_n \\ b_i &= b_{i+1}(x - x_i) + c_i, \qquad i = n-1, \ldots, 0 \\ P(x) &= b_0 \end{aligned}$$

Dies erfordert einen Aufwand von $O(n)$.

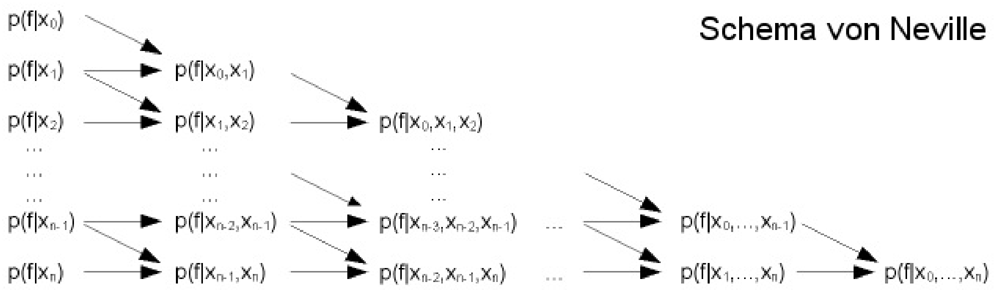
Algorithmus von Neville-Aitken

Ähnlich wie im Newtonschen Algorithmus wird beim Algorithmus von Neville-Aitken die Lösung rekursiv berechnet. Dazu bezeichne $p\left(f|x_i \ldots x_j\right)$ das eindeutig bestimmte Interpolationspolynom k-ten Grades zu den $k+1$ Stützpunkten $(x_i, f_i), \ldots, (x_j, f_j)$, wobei $k = j - i$ ist. Es gilt dann die Rekursionsformel von Aitken:

$$\begin{aligned} p(f|x_i)(x) &\equiv f_i, \\ p(f|x_i, \ldots, x_j)(x) &\equiv \frac{(x - x_i)p(f|x_{i+1}, \ldots, x_j)(x) - (x - x_j)p(f|x_i, \ldots, x_{j-1})(x)}{x_j - x_i}. \end{aligned}$$

Beweisen lässt sie sich durch Einsetzen von x_i, wodurch man verifiziert, dass die rechte Seite der Gleichung die Interpolationsbedingung erfüllt. Die Eindeutigkeit des Interpolationspolynoms liefert dann die Behauptung.

Mit dem Schema von Neville kann die Auswertung von $p(f|x_0,\dots,x_n)(x) = P(x)$ dann rekursiv erfolgen:

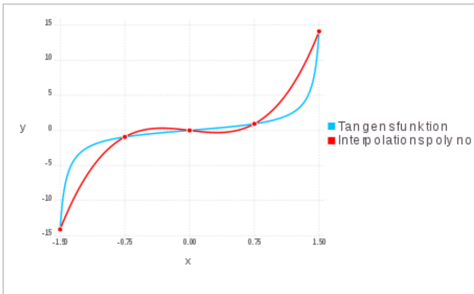


Vergleich der Lösungsverfahren

Möchte man alle Koeffizienten des Interpolationspolynoms P bestimmen, so bietet der Newtonsche Algorithmus hierfür den geringsten notwendigen Aufwand von $\mathcal{O}(n^2)$. Das so bestimmte Polynom lässt sich dann mit $\mathcal{O}(n)$ Operationen an einer Stelle auswerten. Darum ist der Newtonsche Algorithmus gut geeignet, wenn das Interpolationspolynom an vielen Stellen ausgewertet werden soll. Auch lassen sich effizient weitere Stützpunkte hinzufügen. Liegen die Stützstellen oder die Stützwerte allerdings zu nahe beieinander, so besteht die Gefahr der Auslöschung bei der Bestimmung der dividierten Differenzen.

Der Neville-Aitken-Algorithmus ist dagegen gut geeignet, wenn ein Interpolationspolynom nur an ganz wenigen Stellen ausgewertet werden soll, dabei ist er weniger anfällig gegen Auslöschung. Auch im Neville-Aitken-Algorithmus lassen sich effizient neue Stützpunkte hinzufügen. So kann z. B. eine gewünschte Genauigkeit der Interpolation an einer Stelle durch Hinzufügen immer weiterer Stützstellen erreicht werden.

Beispiel: Interpolation der Tangensfunktion



Tangensfunktion (blau) und ihre Polynominterpolante dritten Grades (rot)

Interpoliere die Funktion $f(x) = \tan(x)$ bei gegebenen Punkten

$x_0 = -1{,}5$	$f(x_0) = -14{,}101420$
$x_1 = -0{,}75$	$f(x_1) = -0{,}931596$
$x_2 = 0$	$f(x_2) = 0$
$x_3 = 0{,}75$	$f(x_3) = 0{,}931596$
$x_4 = 1{,}5$	$f(x_4) = 14{,}101420$

Lösung mit Lagrange

Die Lagrange-Basisfunktionen sind

$$\begin{aligned}
\ell_0(x) &= \frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2}\cdot\frac{x-x_3}{x_0-x_3}\cdot\frac{x-x_4}{x_0-x_4} = \frac{1}{243}x(2x-3)(4x-3)(4x+3)\\
\ell_1(x) &= \frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2}\cdot\frac{x-x_3}{x_1-x_3}\cdot\frac{x-x_4}{x_1-x_4} = -\frac{8}{243}x(2x-3)(2x+3)(4x-3)\\
\ell_2(x) &= \frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1}\cdot\frac{x-x_3}{x_2-x_3}\cdot\frac{x-x_4}{x_2-x_4} = \frac{3}{243}(2x+3)(4x+3)(4x-3)(2x-3)\\
\ell_3(x) &= \frac{x-x_0}{x_3-x_0}\cdot\frac{x-x_1}{x_3-x_1}\cdot\frac{x-x_2}{x_3-x_2}\cdot\frac{x-x_4}{x_3-x_4} = -\frac{8}{243}x(2x-3)(2x+3)(4x+3)\\
\ell_4(x) &= \frac{x-x_0}{x_4-x_0}\cdot\frac{x-x_1}{x_4-x_1}\cdot\frac{x-x_2}{x_4-x_2}\cdot\frac{x-x_3}{x_4-x_3} = \frac{1}{243}x(2x+3)(4x-3)(4x+3)
\end{aligned}$$

also ist das Interpolationspolynom

$$\begin{aligned}
P_{\text{Lagrange}}(x) = &\tfrac{1}{243}\big(f(x_0)x(2x-3)(4x-3)(4x+3)\\
&\quad - 8f(x_1)x(2x-3)(2x+3)(4x-3)\\
&\quad + 3f(x_2)(2x+3)(4x+3)(4x-3)(2x-3)\\
&\quad - 8f(x_3)x(2x-3)(2x+3)(4x+3)\\
&\quad + f(x_4)x(2x+3)(4x-3)(4x+3)\big)\\
= &-1{,}477474x + 4{,}834848x^3
\end{aligned}$$

Lösung mit Newton

Die dividierten Differenzen sind hier

x_i	$f(x_i)$				
$-1{,}50$	$-14{,}10140$				
$-0{,}75$	$-0{,}931596$	$\frac{-0{,}931596-(-14{,}1014)}{-0{,}75-(-1{,}5)}=17{,}5597$			
$0{,}00$	$0{,}00000$	$\frac{0-(-0{,}931596)}{0-(-0{,}75)}=1{,}24213$	$\frac{1{,}24213-17{,}5597}{0-(-1{,}5)}=-10{,}8784$		
$0{,}75$	$0{,}931596$	$\frac{0{,}931596-0}{0{,}75-0}=1{,}24213$	$\frac{1{,}24213-1{,}24213}{0{,}75-(-0{,}75)}=0{,}00000$	$\frac{0-(-10{,}8784)}{0{,}75-(-1{,}5)}=4{,}83484$	
$1{,}50$	$14{,}10140$	$\frac{14{,}10140-0{,}931596}{1{,}5-0{,}75}=17{,}5597$	$\frac{17{,}5597-1{,}24213}{1{,}5-0}=10{,}8784$	$\frac{10{,}8784-0}{1{,}5-(-0{,}75)}=4{,}83484$	$\frac{4{,}83484-4{,}83484}{1{,}5-(-1{,}5)}=0$

und das Interpolationspolynom ist

$$\begin{aligned}P_{\text{Newton}}(x) =& -14{,}1014+17{,}5597(x+1{,}5)-10{,}8784(x+1{,}5)(x+0{,}75)\\ &+4{,}83484(x+1{,}5)(x+0{,}75)x+0(x+1{,}5)(x+0{,}75)x(x-0{,}75)\\ =& -0{,}00005-1{,}4775x-0{,}00001x^2+4{,}83484x^3\end{aligned}$$

Verwendet man genauere Startwerte $f(x_i)$, verschwinden der erste und der dritte Koeffizient.

Interpolationsgüte

Fehlerabschätzung

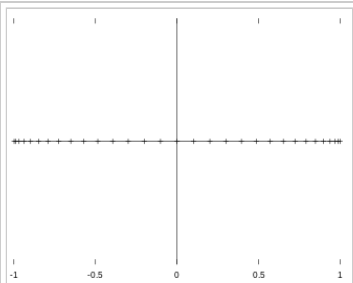
Gegeben sei eine Funktion f, deren $n+1$ Funktionswerte f_i an den Stellen x_i durch das Polynom P interpoliert werden. Mit I sei das kleinste Intervall bezeichnet, das die Stützstellen x_i und eine Stelle x enthält. Ferner sei f $(n+1)$-mal stetig differenzierbar auf I. Dann existiert ein $\xi \in I$, für das gilt:

$$f(x)-P(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x-x_i)$$

Insbesondere ist also bezüglich der Maximumsnorm auf $[a,b]$ und mit $w_n(x)=\prod_{i=0}^{n}(x-x_i)$:

$$|f(x)-P(x)|\le\frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\prod_{i=0}^{n}|x-x_i|\le\frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|w_n\|_\infty$$

Fehleroptimierung nach Tschebyschow[2]



Für größere n clustern die Tschebyschow-Punkte an den Intervallrändern.

Der Fehler hängt also von einer Ableitung von f ab und von dem Produkt **Fehler beim Parsen (MathML mit SVG- oder PNG-Rückgriff (empfohlen für moderne Browser und Barrierefreiheitswerkzeuge): Ungültige Antwort („Math extension cannot connect to Restbase.“) von Server „/mathoid/local/v1/“:): w_n(x):=\prod_{i=0}^n (x-x_i)** , also den Stützstellen x_i. Manchmal ist man in der Position, dass man sich Stützstellen selbst wählen kann; etwa, wenn man ein physikalisches Experiment durchführt, oder aber auch bei einigen Verfahren zur numerischen Lösung von Differentialgleichungen. In diesem Fall ist die Frage interessant, für welche Stützstellen die Maximumsnorm $\|w_n\|_\infty$ minimal wird.

Tschebyschow hat gezeigt, dass die Nullstellen der Tschebyschow-Polynome („Tschebyschow-Punkte“) optimale Stützstellen sind. Die Polynome $T_{n+1}(x)=\cos((n+1)\arccos(x))$ haben die Nullstellen $t_k=\cos\left(\frac{2k+1}{2n+2}\pi\right)\ k\in\{0,1,\dots,n\}$. Die ersten Tschebyschow-Polynome sind:

$$T_2(x)=2x^2-1,\qquad T_3(x)=4x^3-3x,\qquad T_4(x)=8x^4-8x^2+1$$

Man kann dann beweisen, dass jedes Polynom der Form w_n auf dem Intervall $[-1,1]$ durch ein normiertes Tschebyschow-Polynom beschränkt bleibt. Diese Aussage kann dann mit der Transformation

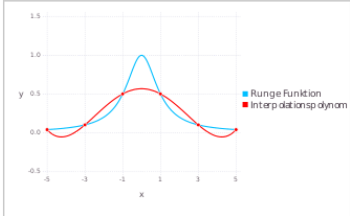
$$\begin{aligned}\xi\in[-1,1]\rightsquigarrow x&=\frac{a+b}{2}+\frac{b-a}{2}\xi &&\in[a,b]\\ x\in[a,b]\rightsquigarrow \xi&=\frac{2x-a-b}{b-a} &&\in[-1,1]\end{aligned}$$

auf den Fall eines allgemeinen Intervalls $[a,b] \subset \mathbb{R}$ übertragen werden. Der Beweis liefert auch die Abschätzung

$$\|w_n\|_{[a,b],\infty} = \max_{x\in[a,b]} |w_n(x)| = 2\left(\frac{b-a}{4}\right)^{n+1}.$$

Runges Phänomen

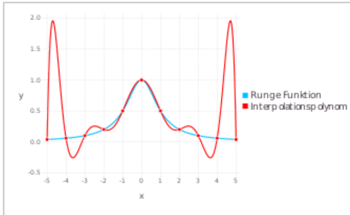
Verbessert sich die Interpolationsgüte, wenn mehr Stützpunkte hinzugefügt werden? Im Allgemeinen nicht: Bei hohem Grad des Polynoms kann es vorkommen, dass die Polynomfunktion kaum noch der zu interpolierenden Funktion ähnelt, was auch als *Runges Phänomen* bekannt ist. Polynome streben im Grenzfall $x \to \pm\infty$ gegen $\pm\infty$. Verhält sich die zu interpolierende Funktion anders, etwa periodisch oder asymptotisch konstant, treten starke Oszillationen in der Nähe der Intervallgrenzen auf. Für solche Funktionen sind Polynominterpolationen über das gesamte Intervall relativ ungeeignet.



Interpolation der Rungefunktion bei 6 äquidistanten Stützstellen (rote Punkte)

Tschebyschow-Stützstellen, die an den Intervallgrenzen dichter liegen, können zwar den Gesamtfehler der Interpolation verkleinern, dennoch empfiehlt sich ein Wechsel des Interpolationsverfahrens, etwa zur Spline-Interpolation. Runge gab für dieses Phänomen ein Beispiel an, die nach ihm benannte Runge-Funktion:

$$f(x) = \frac{1}{1+x^2}\,, \quad x \in [-5;5]$$



Interpolation der Rungefunktion bei 11 äquidistanten Stützstellen (rote Punkte)

Konvergenzverhalten

Es gibt aber Bedingungen, unter denen sich die Interpolationsgüte mit steigender Anzahl von Stützpunkten verbessert: Wenn das Stützstellengitter immer „feiner" wird und eine analytische Funktion interpoliert wird. Genauer: Sei f eine analytische Funktion auf dem Intervall $I = [a,b] \subset \mathbb{R}$. Für eine Intervallteilung

$$\Delta_m = \{a = x_0^{(m)} < x_1^{(m)} < \cdots < x_{n_m}^{(m)} = b\}, \qquad m \in \mathbb{N}$$

sei ihre Norm definiert durch

$$\|\Delta_m\| = \max_i |x_{i+1}^{(m)} - x_i^{(m)}|.$$

Zu jeder Intervallteilung Δ_m gibt es ein eindeutig bestimmtes Polynom P_{Δ_m}, das f an den Stützstellen Δ_m interpoliert. Gilt für eine Folge von Intervallteilungen $\|\Delta_m\| \to 0$, so folgt $P_{\Delta_m} \to f$ gleichmäßig.

Allerdings lässt sich zu *jeder* Folge $\{\Delta_m\}_{m\in\mathbb{N}}$ auch eine auf I stetige Funktion f finden, so dass $\{P_{\Delta_m}\}_{m\in\mathbb{N}}$ *nicht* gleichmäßig gegen f konvergiert (Satz von Faber[3]).

Verallgemeinerung

Bisher wurden die Stützstellen x_i des Interpolationspolynoms P als paarweise verschieden angenommen. Bei der Hermiteinterpolation ist das nicht der Fall. An mehrfach vorkommenden Stützstellen werden dabei nicht nur die Funktionswerte, sondern auch die Werte der Ableitungen des Interpolationspolynoms vorgegeben.

Literatur

- Hans R. Schwarz, Norbert Köckler: *Numerische Mathematik*. 5. Aufl. Teubner, Stuttgart 2004, ISBN 3-519-42960-8
- Stoer, Bulirsch: *Numerische Mathematik 1*. 10. Auflage. Springer Verlag, Berlin, Heidelberg, New York 2007, ISBN 978-3-540-45389-5, 2.1 Interpolation durch Polynome, S. 39–57 (Behandelt die Verfahren nach Lagrange, Neville-Aitken und Newton, Hermite-Interpolation und Fehlerabschätzung jeweils mit Beispielen und Beweisen.).
- Press, Teukolsky, Vetterling, Flannery: *Numerical Recipes*. The Art of Scientific Computing. 3. Auflage. Cambridge University Press, Cambridge 2007, ISBN 978-0-521-88407-5, 3.2 Polynomial Interpolation and Extrapolation, S. 118–120 (Neville-Aitken-Algorithmus mit C++-Implementation).
- Carl Runge: *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*. In: *Zeitschrift für Mathematik und Physik*. Band 46. B. G. Teubner, Leipzig 1901, S. 224–243 (iris.univ-lille1.fr (http://iris.univ-lille1.fr/handle/1908/2014) Runge-Phänomen).

Weblinks

Wikibooks: Dividierte Differenzen & Horner-Schema – Implementierungen in der Algorithmensammlung

- Seite zu Newton, Lagrange und Cubic Spline (http://www.mathe-online.at/materialien/Andreas.Pester/files/applets/Interpolation/) mit Java-Applet

- Erläuterungen und Beispiel zur Lagrange-Interpolation (http://matheplanet.com/default3.html?call=article.php?sid=977)

Einzelnachweise

1. Martin J. Marsden: An Identity for Spline Functions with Applications to Variation-Diminishing Spline Approximation. Journal of Approximation Theory 3, 7-49 (1970).
2. Jochen Werner: *10.4*. In: *Numerische Mathematik*, 1, Vieweg Studium, Nr.32, Vieweg Verlagsgesellschaft, 1992, ISBN 3-528-07232-6.
   - Auch hier (http://www.sam.math.ethz.ch/~hiptmair/tmp/NCSE_full.pdf) (4.1.3.; PDF-Datei; 11,68 MB)
3. Andrei Nikolajewitsch Kolmogorow, et al.: *4*. In: *Mathematics of the 19th Century*, 1, Birkhäuser, 1998, ISBN 3-7643-5845-9.
   - Auch hier (http://books.google.de/books?id=Mw6JMdZQO-wC&pg=PA277&lpg=PA277&dq=satz+faber+1914+(1877-1966)&source=bl&ots=N-yWl0pxi2&sig=lBBUKHD9uQdvmUR_oJ9FicSKNAg&hl=de&ei=pu90TrzLJNHAtAbr3-mSCw&sa=X&oi=book_result&ct=result&resnum=5&ved=0CEIQ6AEwBA#v=onepage&q&f=false) (Google Books)