Kapitel 7

Interpolation und Approximation

Interpolation bedeutet, zu einer Funktion, die nur an $n + 1$ diskreten vorgegebenen Stützstellen x_i, $i = 0 \ldots n$, mit den zugehörigen Funktionswerten y_i bekannt ist, eine Interpolationsfunktion $f(x)$ zu finden, für die einerseits die Interpolationsbedingungen

$$f(x_i) = y_i \tag{7.1}$$

gelten, die aber andererseits auch zwischen den x_i definiert ist, im allgemeinen für $x_0 \leq x \leq x_n$.

Vor Einführung des Computers als wichtigstes Rechenhilfsmittel wurden zur Berechnung von Funktionen Rechentafeln (z.B. die berühmten Logarithmentafeln) benutzt. Da in diesen Tabellen nur bestimmte Werte der Funktion aufgelistet waren, war es wichtig, gute Interpolationsverfahren zu kennen, um Zwischenwerte gewinnen zu können. Heute ist diese Anwendnung zwar zurückgedrängt worden, aber weil eine Interpolation häufig numerisch nicht sehr aufwendig durchzuführen ist, findet sie Anwendung bei

1. der Elimination teurer Funktionsberechnungen in zeitkritischen Teilen eines Algorithmus (Kraftberechnung aus Tabellen),
2. bei der numerischen Differentiation,
3. bei der Interpolation zwischen zwei verschiedenen Diskretisierungen,
4. bei der Datenreduktion: eine Bezier-Kurve (Vorsicht: ist keine eigentliche Interpolation, da sie nicht durch die Stützpunkte verläuft) ist bereits durch die Lage weniger Kontrollpunkte festgelegt,
5. bei der *Extrapolation*. Bestimmte Integrationsverfahren (Romberg) beruhen darauf, die Genauigkeit eines Verfahrens dadurch zu verbessern, daß eine Interpolation des Resultates als Funktion des Stützstellenabstandes h für $h \to 0$ durchgeführt wird.

Deshalb sollen im folgenden verschiedene Interpolationsverfahren vorgestellt werden.

7.1 Polynomiale Interpolation

Die einfachsten Interpolationsverfahren beruhen auf der Idee, $f(x)$ als ein Polynom Nter Ordnung mit $N+1$ unbekannten Koeffizienten a_i anzusetzen

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N. \tag{7.2}$$

In der Regel werden dann auch $N+1$ Bedingungen zur Bestimmung der Koeffizienten benötigt, d.h., daß das Interpolationspolynom durch eine an $n+1$ Stellen gegebene Funktion im allgemeinen $N = n$ter Ordnung ist: Durch zwei Punkte läßt sich eine Gerade legen, durch drei Punkte eine Parabel, usw.

Der obige Ansatz führt unter Auswertung der Interpolationsbedingungen (7.1) auf ein *lineares* Gleichungssystem

$$\begin{aligned} P(x_1) = y_0 &= a_0 + a_1 x_0 + a_2 x_0^2 + \ldots a_n x_0^n \\ P(x_1) = y_1 &= a_0 + a_1 x_1 + a_2 x_1^2 + \ldots a_n x_1^n \\ P(x_2) = y_2 &= a_2 + a_1 x_2 + a_2 x_2^2 + \ldots a_n x_2^n \\ \ldots \\ P(x_n) = y_n &= a_0 + a_1 x_n + a_2 x_n^2 + \ldots a_n . x_n^n \end{aligned} \tag{7.3}$$

Wenn die x_i paarweise verschieden sind, sind die Zeilengleichungen linear unabhängig und die Lösung dieses Systems legt eindeutig das Interpolationspolynom fest (die Eindeutigkeit wird auch aus dem folgenden Abschnitt klar werden). Dieses Verfahren ist auch für allgemeinere als polynomiale Interpolationsansätze anwendbar. Für Polynome hingegen müssen wir es gar nicht lösen, sondern können auf eine explizite Darstellung der Lösung zurückgreifen, die wir jetzt besprechen wollen.

7.1.1 Lagrange-Interpolation

Betrachten wir zunächst die sogenannten Lagrange-Polynome [7.1], denen wir schon einmal im Zusammenhang mit der Gauß-Qaudratur begegnet sind (siehe Gl. 6.45),

$$L_i(x) = \prod_{k=1}^{n} \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})\ldots(x_i-x_{i+1})\ldots(x_i-x_n)}. \tag{7.4}$$

Diese Polynome sind vom Grad n und besitzen die Eigenschaft

$$L_i(x_k) = \delta_{ik} = \begin{cases} 1, & \text{falls } i = k \\ 0, & \text{falls } i \neq k. \end{cases} \tag{7.5}$$

Dann ist klar, daß das Polynom

$$P(x) = \sum_{i=0}^{n} y_i L_i(x) \tag{7.6}$$

die Interpolationsbedingungen erfüllt, denn für den speziellen Fall, daß x_k eine Stützstelle ist, gilt

$$P(x_k) = \sum_{i=0}^{n} y_i L_i(x_k) = \sum_{i=0}^{n} y_i \delta_{ik} = y_k. \tag{7.7}$$

Dieses Polynom ist eindeutig, denn gäbe es ein zweites mit den gleichen Eigenschaften, dann wäre auch die Differenz ein Polynom n-ten Grades und müßte an den $n+1$ Stützstellen x_i insgesamt $n+1$ Nullstellen aufweisen. Polynome n-ten Grades können aber maximal n Nullstellen aufweisen, und wir erhalten einen Widerspruch zu unserer Annahme.

Bei dieser Methode werden alle Stützstellen x_i gleich behandelt. Allerdings ist sie für numerische Berechnungen ungeschickt, da mehr als $(n+2)(n+1)$ wesentliche Operationen durchgeführt werden müssen: n Multiplikationen für den Zähler jedes Lagrange-Polynoms sowie eine Division, dann Bestimmung von insgesamt $n+1$ solcher Polynome, und $(n+1)$ weitere Multiplikationen mit den y_i. Ohne weitere Herleitung geben wir hier daher die sogenannte *baryzentrische* Darstellung der Lagrange-Interpolation an, die sich in einigen Schritten aus (7.6) gewinnen läßt:

$$P(x) = \sum_{i=0}^{n} \mu_i y_i / \sum_{i=0}^{n} \mu_i, \quad \mu_i = \frac{1}{x - x_i} \prod_{i \neq j}^{n} \frac{1}{x_i - x_j} \tag{7.8}$$

Bis auf die in den μ_i vorkommenden Faktoren $x - x_i$ sind die vorkommenden Größen konstant und können berechnet werden, sobald die Lage der Stützstellen bekannt ist. Der Berechnungsaufwand für eine Interpolation für $n+1$ Stützstellen beträgt dann noch $2(n+1)$ Multiplikationen zur Bestimmung von Zähler und Nenner, sowie eine Division. Diese Darstellung hat die Form eines gewichteten Mittels der Funktionswerte y_i. Tatsächlich geht aus der Gleichung für die μ_i hervor, daß das Gewicht umso größer wird, je näher x and der Stützstelle x_i liegt.

7.1.2 Neville Schema

Weitere nützliche Darstellungen des Interpolationspolynoms lassen sich aufgrund einer Rekursionsformel gewinnen. Bezeichne $P_{i_0 i_1 \ldots i_k}(x)$ das Interpolationspolynom durch die Punkte (x_i, y_i), $i = i_0, i_1, \ldots i_k$, das also durch eine Teilmenge der Stützstellen x_i festgelegt wird. Dann gilt

$$P_{i_0 i_1 \ldots i_k}(x) = \frac{(x - x_{i_0}) P_{i_1 i_2 \ldots i_k}(x) - (x - x_{i_k}) P_{i_0 i_1 \ldots i_{k-1}}(x)}{(x_{i_k} - x_{i_0})}. \tag{7.9}$$

Dies kann man sehen, indem man einfach die Stützstellen einsetzt und zeigt, daß das Resultat tatsächlich der Funktionswert an der jeweiligen Stelle ist. Beispielsweise überlebt für x_{i_0} nur der zweite Term im Zähler, dessen Vorfaktor sich bis auf das Vorzeichen gegen den Nenner kürzt. Nun ist aber $P_{i_0 i_1 \ldots i_{k-1}}(x_{i_0}) = y_{i_0}$, so daß auch der Gesamtausdruck diesen Wert annimmt. Ähnlich geht man für die übrigen Werte vor.

Auf der linken Seite tauchen nur Polynome k-ten Grades auf, das Gesamtpolynom seinerseits hat $k+1$-ten Grad. Daher kann man man sowohl das Interpolationspolynom, aber auch Werte an bestimmten Stellen x, durch sukzessive Kombination niedergradiger Polynome (bzw. deren Funktionswerte bei x) berechnen. Hierauf beruht das Verfahren von Neville.

Zunächst werden als "Spezialfälle" von (7.9) die Interpolationspolynome zwischen zwei benachbarten Punkten x_i und x_{i+1} bestimmt. Dies ist analog zur Lagrange-Interpolation 1. Ordnung und man erhält eine Gerade:

$$P_{i,i+1}(x) = \frac{1}{x_i - x_{i+1}}((x - x_{i+1})y_i - (x - x_i)y_{i+1}) \tag{7.10}$$

Ein solches Polynom wird für alle benachbarten Punkte x_i, x_{i+1} und die dazugehörigen Werte y_i, y_{i+1} aufgestellt. Die n resultierenden Polynome ersten Grades sind dann $P_{0,1}$, $P_{1,2}, \ldots, P_{n-1,n}$. "Benachbarte" Polynome werden jetzt als Ausgangsbasis in (7.9) für die nächste Generation von Polynomen, jetzt zweiten Grades benutzt. Das Ergebnis sind die $n-1$ Polynome $P_{0,1,2},\ P_{1,2,3}, \ldots, P_{n-2,n-1,n}$. Dieses Verfahren des paarweisen Kombinierens führen wir weiter, bis wir am Interpolationspolynom n-ten Grades angelangt sind.

Dieses rekursive Berechnungsschema läßt sich noch weiter vereinfachen, indem wir nur die jeweils führenden Koeffizienten der linken und rechten Seite in (7.9) miteinander vergleichen. Nennen wir den Vorfaktor von $(x - x_{i_0})...(x - x_{i_k})$ des Polynomes $P_{i_0 i_1 \ldots i_k}(x)$ beispielsweise $f[i_0, i_1, \ldots, i_k]$, dann ist

$$f[i_0, i_1, \ldots, i_k] = \frac{f[i_1, \ldots, i_k] - f[i_0, i_1, \ldots, i_{k-1}]}{x_{i_k} - x_{i_0}}. \tag{7.11}$$

Dies bedeutet, daß wir die Koeffizienten sukzessive als dividierte Differenzen bestimmen können. Als Beispiel ermitteln wir das Interpolationspolynom durch die Punkte $(-2, -37), (0, 1),\ (1, 2),\ (2, 15)$. Man beachte, daß immer durch die Differenz der zu den nicht gemeinsamen Indizes gehörenden x_i dividiert wird:

x_i	y_i			
-2	$\boxed{f[0] = -37}$			
0	$f[1] = 1$	$\boxed{f[0,1] = \frac{1-(-37)}{0-(-2)} = 19}$		
1	$f[2] = 2$	$f[1,2] = (2-1)/(1-0) = 1$	$\boxed{f[0,1,2] = \cdots = -6}$	
2	$f[3] = 15$	$f[2,3] = (15-2)/(2-1) = 13$	$f[1,2,3] = \cdots = 6$	$\boxed{\text{f[0,1,2,3]= 3}}$

An der oberen Diagonale lesen wir das Interpolationspolynom

$$P(x) = -37 + (x+2)(19 + x(-6 + (x-1)3)) = 1 + x - 3x^2 + 3x^3 \tag{7.12}$$

ab.

Als Algorithmus läßt sich unser Vorgehen etwa so formulieren:

```
template<class T>
void pts_neville(int n, const T& x, const T& y, T &result ){
  // x contains the arguments, y the values of the function to be interpolated
  // result will contain the upper diagonal of the Neville scheme,
  // i.e. the coefficients of the interpolating polynomial
  // P(x) = result[0] + result[1]*(x-x0) + result[2]*(x-x1)(x-x0) +...

  // initialize the result with the f(x_i)/y_i
  for (int i=0; i < n; ++i )
    result[i] = y[i];

  // increasing 'order' of the interpolation polynomial
  for (int k=0; k < n; ++k )
    for (int i=n; i> k ; --i){
      // compute new column of divided differences
      result[i] = (result[i]-result[i-1])/(x[i] - x[i-k-1]);
    }
}
```

Eine abgewandelte Form dieser Berechnungsvorschrift läßt sich auch vorteilhaft verwenden, wenn nur ein einzelner zusätzlicher Wert und nicht allgemeiner die Koeffizienten des Polynoms berechnet werden sollen. Dazu wird wieder die rekursive Vorschrift (7.9) angewandt, aber jetzt die Faktoren $(x - x_i)$, die auf der rechten Seite auftreten, bei der Berechnung gleich mit berücksichtigt. Ein derart abgewandeltes Schema läßt sich hervorragend zur Extrapolation nutzen. So wird z.B. beim Romberg-Verfahren der Integration der Wert eines für verschiedene Schrittweiten h berechneten Integrals nach $h \to 0$ extrapoliert [7.1].

7.2 Rationale Interpolation

Polynomiale Interpolationsverfahren eignen sich gut für glatte Funktionen. Hat aber die zu interpolierende Funktion Pole (oder auch weniger starke Singularitäten, wie z.B. $1/\sqrt(x))$, so können diese prinzipiell nicht durch eine polynomiale Interpolation erfaßt werden und in der Nähe der Polstellen sind häufig die auftretenden Fehler unakzeptabel

Eingabe		Interpolation			
x	$\tan x$	x	exakt	rational	polynomial
1.1	1.9647597				
1.2	2.5721516	1.15	2.2344969	$2.2344921(5 \times 10^{-6})$	$2.0980797(1 \times 10^{-1})$
1.3	3.6021024	1.25	3.0095697	$3.0095730(1 \times 10^{-6})$	$3.0862589(1 \times 10^{-1})$
1.4	5.7978837	1.35	4.4552218	$4.4552164(6 \times 10^{-6})$	$4.3438010(1 \times 10^{-1})$
1.5	14.1014200	1.45	8.2380928	$8.2381449(5 \times 10^{-5})$	$8.7133064(5 \times 10^{-1})$

Tabelle 7.1: Werte des $\tan(x)$ und Interpolationswerte, die durch polynomiale bzw. gebrochen-rationale Interpolation erhalten wurden. Die Werte in Klammern sind die absoluten Fehler der jeweiligen Interpolation and dieser Stelle

groß. In diesen Fällen ist es besser, durch eine rationale Funktion (Quotient von Polynomen) zu approximieren. Dazu ist die Methode der *Padé-Approximanten* entwickelt worden: Die rationale Funktion $R(x)$ sei

$$R(x) = \frac{P(x)}{Q(x)} = \frac{p_0 + p_1 x + p_2 x^2 + ... + p_\mu x^\mu}{1 + q_1 x + ... + q_\nu x^\nu} \tag{7.13}$$

Es seien n Werte y_i, $i = 1...n$ gegeben. Die Anzahl der Unbekannten ist also $\mu + \nu + 1$, so daß $n = \mu + \nu + 1$ gelten muß, damit wir die Unbekannten bestimmen können. Üblicherweise wählt man $\mu = \nu$ oder $\mu = \nu - 1$, da sich gezeigt hat, daß die Approximation für diese Wahl häufig gut ist. Um die Unbekannten $p_0, ..., p_\mu,\ q_1, ..., q_\nu$ zu bestimmen, stellt man die N Gleichungen

$$R(x_i) = f_i\,, \quad i = 1...N \tag{7.14}$$

auf und bestimmt daraus die Unbekannten. Man kann so zwar immer eine Lösung erhalten (unter Umständen auch mit verschwindenden Koeffizienten für die führenden Potenzen), jedoch ist nicht immer die Interpolationseigenschaft garantiert. Insbesondere kann eine Nullstelle sowohl im Nenner– als auch im Zählerpolynom auftreten und die nach Herauskürzen übrigbleibende interpolierende Funktion nimmt in der Regel dann an diesen Stellen den Funktionswert nicht mehr an. Man muß dann die Ordnungen des Zähler– und Nennerpolynoms ändern, um ggf. eine gültigen Interpolation zu erhalten.

Die Effizienz dieser Methode im Vergleich zur polynomialen Interpolation für Funktionen mit Polstellen kann man an der Funktion $\tan(x)$ gut sehen. Die Ergebnisse kann man in Tabelle 7.1 miteinander vergleichen. Besonders zu beachten ist der Fehler für Werte nahe $\frac{\pi}{2} \approx 1.57$.

Die bisher behandelten Interpolationsverfahren beschränken sich auf ein Intervall. Will man eine globale Funktion erhalten, so kann man in entsprechend vielen benachbarten Intervallen interpolieren und die Ergebnisse aneinanderreihen. Man erhält zwar eine stetige Funktion, die aber an den Berührstellen der einzelnen Intervalle in der Regel nicht differenzierbar ist, da sie dort einen Knick besitzt, was eigentlich immer unerwünscht ist.

Eine weitere unangenehme Eigenschaft der polynomialen Interpolation ist auch, daß das Interpolationspolynom insbesondere bei äquidistanter Wahl der Stützstellen dazu neigt, an den Intervallenden stark zu “oszillieren.”

Einen Ausweg aus dieser Situation bieten die sogenannten Spline-Funktionen, die an den Anschlußpunkten zwischen den Intervallen garantierte Anschlußbedingungen erfüllen und weiterhin auch eine globale Optimierungseigenschaft haben: Man kann zeigen, daß sie von allen möglichen interpolierenden Funktionen die insgesamt kleinste mittlere Krümmung (Betrag der zweiten Ableitung) aufweisen.

7.3 Spline-Funktionen

Diese Funktionen sind nach dem englischen Begriff für dünne Latten *splines* benannt, die man sich ohne äußere Krafteinwirkung durch die gegebenen Stützpunkte gelegt denkt und die dann eine “glatte” Funktion darstellen. Mit Spline-Funktionen ist es möglich, als bestmögliche Approximation durch n Datenpunkte eine global zweimal stetig differenzierbare Funktion zu legen. Die folgende Darstellung ist i.w. dem Werk *Numerical Recipes* [7.2] entnommen, in dem weitere Details zu finden sind.

Gegeben sei der Datensatz (x_i, y_i), $i = 1, \ldots, n$. Betrachten wir zunächst $x_i < x < x_{i+1}$ und eine lineare Interpolation der Daten in diesem Intervall:

$$f(x) = Ay_i + By_{i+1} = \frac{(x_{i+1} - x)y_i + (x_i - x)y_{i+1}}{x_{i+1} - x_i} \tag{7.15}$$

Die in x linearen Gewichte A und B kann man von der rechten Seite als Vorfaktoren der y_i ablesen, die ansonsten die bekannte Form des interpolierenden Lagrange-Polynomes hat.

Diese Interpolation hat eine verschwindende zweite Ableitung in diesem Interval, jedoch haben die linearen Interpolationen in den Nachbarintervallen im allgemeinen verschiedene erste Ableitungen, so daß diese an den Stützstellen x_i unstetig ist.

Für die weitere Diskussion nehmen wir für einen Moment an, daß wir die Werte y_i'' für die zweiten Ableitungen an den Stellen x_i kennen. Wir addieren jetzt zu unserer linearen Interpolation oben ein kubisches Polynom, dessen zweite Ableitung linear zwischen y_i'' und y_{i+1}'' variiert und das an den Stellen x_i den Wert 0 annimmt. Dies sind vier Bedingungen an ein Polynom dritten Grades, das daher eindeutig bestimmt ist. Wir schreiben das Summenpolynom in der Form

$$y = Ay_i + By_{i+1} + Cy_i'' + Dy_{i+1}'', \tag{7.16}$$

wobei die Koeffizienten von x_i, x_{i+1} und x abhängen und folgende Form haben,

$$\begin{aligned}
A &= (x_{i+1} - x)/(x_{i+1} - x_i), \\
B &= 1 - A = (x - x_i)/x_{i+1} - x_i), \\
C &= (1/6)A(A^2 - 1)(x_{i+1} - x_i)^2, \\
D &= (1/6)B(B^2 - 1)(x_{i+1} - x_i)^2.
\end{aligned}$$

Weil A und B die Werte 0 und 1 bzw. 1 und 0 bei $x = x_i$ und $x = x_{i+1}$ annehmen, sieht man direkt, daß C und D an diesen beiden Punkten verschwinden und das Polynom (7.16) auch weiterhin die Interpolationseigenschaft besitzt. Für die erste und zweite Ableitung von (7.16) nach x errechnet man

$$\begin{aligned}
\frac{d}{dx}y &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{3A^2 - 1}{6}(x_{i+1} - x_i)y_i'' \\
&\quad + \frac{3B^2 - 1}{6}(x_{i+1} - x_i)y_{i+1}'' \qquad (7.17) \\
\frac{d^2}{dx^2}y &= Ay_i'' + By_{i+1}'' \qquad (7.18)
\end{aligned}$$

Wir können jetzt auch sehen, daß (7.16) an den Intervallgrenzen die richtigen Werte für die zweiten Ableitungen annimmt. Wir haben damit einen eindeutigen Ausdruck für das Interpolationspolynom gefunden, ausgedrückt durch die Funktionswerte und die zweiten Ableitungen an den Stützstellen x_i. Im eigentlichen Interpolationsproblem sind die y_i'' allerdings nicht festgelegt (wir hatten das nur der Einfachheit zuliebe eingangs für unsere Argumentation angenommen), wir können daher zwei Bedingungen an die Interpolation stellen. Wir fordern (i) daß an den Verbindungsstellen sowohl die ersten als auch (ii) die zweiten Ableitungen stetig sein sollen.

Formulieren wir diese Bedingung unter Zuhilfenahme von (7.17) und Einsetzen von A bzw. B an den inneren Punkten x_i für die beiden Splines links und rechts von x_i, dann erhalten wir ein Gleichungssystem mit $n-2$ Gleichungen, eine für jeden inneren Stützpunkt x_i, $i = 2, \ldots, n-1$. In diesem System tauchen die y_i'' als Unbekannte auf,

$$\frac{x_i - x_{i-1}}{6}y_{j-1}'' + \frac{x_{i+1} - x_{i-1}}{3}y_i'' + \frac{x_{i+1} - x_i}{6}y_{i+1}'' = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}. \qquad (7.19)$$

Über die restlichen zwei Gleichungen, die für die eindeutige Bestimmung der y_i'' benötigt werden, verfügt man jetzt üblicherweise im Rahmen einer der folgenden Strategien,

1. die *natural spline* Bedingung: festgelegte Werte (üblicherweise = 0) für die zweiten Ableitungen y_1'' und y_n'',

2. feste Werte für die ersten Ableitungen bei x_1 und x_n. Die Werte für die zweite Ableitung an diesen Stellen bestimmt sich dann direkt aus Gl. (7.17),

3. Randbedingungen gemischt aus 1. und 2.,

4. (mit Einschränkungen:) “periodische” Randbedingungen können gefordert werden, falls $y_n = y_0$ gilt. Dann möchte man üblicherweise auch $y'_0 = y'_n$ und $y''_0 = y''_n$ erreichen.

Die ersten drei Zusatzbedingungen führen auf ein sogenanntes tridiagonales System von Gleichungen, das sich durch einmaliges Vorwärtseinsetzen und dann Rückwärtseinsetzen mit geringem Aufwand lösen läßt, siehe z.B. [7.2].

Die letzte Bedingung führt nicht auf ein solch einfaches System und ist daher mit zusätzlichem Aufwand bei der Bestimmung der der y''_i verbunden. Für Routinen zur Ausführung der Spline-Interpolation und zur Bestimmung von Zwischenwerten verweisen wir ebenfalls auf *Numerical Recipes* [7.2].

7.4 Weiterführende Literatur

[7.1] Hans Rudolf Schwarz. *Numerische Mathematik.* B. G. Teubner, Stuttgart, Germany, 1986.
[7.2] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 2nd edition 1992.