

7 Interpolation

Sind die Werte einer Funktion $f(x)$ für einen Satz von x -Werten gegeben, etwa $x_0 < x_2 \cdots < x_n$, aber kein analytischer (formelmäßiger) Ausdruck für f , so kann man versuchen, eine geeignete, möglichst einfache Funktion zu finden, welche für die gegebenen x -Werte die entsprechenden Funktionswerte annimmt. Mit Hilfe dieser Funktion lassen sich dann leicht Werte an beliebigen anderen Stellen x berechnen. Sucht man einen Funktionswert für ein x zwischen x_0 und x_n , so spricht man von **Interpolation**; liegt x außerhalb des gegebenen Datenbereiches, heißt die Aufgabe **Extrapolation** und ist beträchtlich riskanter (wie viele ehemalige Börse-Analysten bestätigen können).

Beachten Sie den Unterschied zu den Approximationsaufgaben des vorigen Abschnittes: *Interpolation* legt eine Funktion *durch* gegebene Datenpunkte; *Approximation* findet eine Funktion, die sich an die Datenpunkte möglichst gut *annähert*.

Jede Formelsammlung bringt eine Vielzahl von Interpolationsformeln, verbunden mit klingenden Namen von Leuten, die schon lange tot sind (Newton, Bessel, Lagrange...). Nehmen Sie das als Indiz dafür, dass Interpolationsverfahren früher von großer Bedeutung waren. Zu einer Zeit, als mathematische Funktionen aus Tabellenwerken entnommen werden mussten, waren etwa trigonometrische Berechnungen ständig mit Interpolation in Tabellen verbunden. Winkelfunktionen bietet aber inzwischen jeder bessere Taschenrechner, und auch weniger elementare Funktionen stehen in gängigen Computerprogrammen zur Verfügung. Damit ist die Wichtigkeit der Interpolation in diesem Bereich ziemlich gesunken.

Interpolation und Extrapolation sind noch wichtig als Hilfsmittel zur Lösung anderer mathematischer Probleme (numerische Integration, numerisches Lösen von Differentialgleichungen). Sehr wichtig ist Interpolation bei Computergraphik und computerunterstütztem Design: Lara Croft existiert als ein Satz von Datenpunkten; damit sie am Bildschirm gut zur Geltung kommt, zeichnet der Rechner möglichst attraktive Kurven durch die Daten. Meist handelt es sich dabei um Spline-Interpolationsverfahren.

7.1 Polynomiale Interpolation

Durch zwei Punkte geht genau eine Gerade. Durch drei beliebige Punkte lässt sich eindeutig eine Parabel legen. Allgemein: Durch $n+1$ (verschiedene) Punkte ist ein Polynom n -ten Grades eindeutig bestimmt.

Polynom-Interpolation, Aufgabenstellung:

Gegeben: $n+1$ Wertepaare (x_i, y_i) , $i = 0, \dots, n$, wobei die x_i paarweise verschieden sind.

Gesucht: das eindeutig bestimmte Polynom n -ten Grades p , das durch die gegebenen Datenpunkte verläuft:

$$p(x_i) = y_i \quad \text{für } i = 0, \dots, n$$

Oder: gesucht ist nicht die Form des interpolierenden Polynoms selbst, sondern dessen Wert an einer gegebenen Stelle x .

Lösung mit der Holzhammermethode: Ansatz des Polynoms mit unbestimmten Koeffizienten,

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n .$$

Einsetzen der gegebenen Wertepaare führt auf ein System von $n + 1$ linearen Gleichungen in den $n + 1$ unbekanntem Koeffizienten a_0, a_1, \dots, a_n . Sofern die x_i alle paarweise verschieden sind, ist das System eindeutig lösbar.

Eleganter ist es, das Interpolationspolynom explizit hinzuschreiben. Die klassische Formel dazu:

Lagrangesche Interpolationsformel

Das Interpolationspolynom durch die $n + 1$ Wertepaare $(x_i, y_i), i = 0, \dots, n$ ist gegeben durch

$$p(x) = L_0(x)y_0 + L_1(x)y_1 + \dots + L_n(x)y_n,$$

wobei

$$L_i(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n)}{(x_i - x_0)(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)}$$

Diese Darstellung ist für eine Reihe weiterführender Überlegungen wichtig, aber zum Programmieren oder zur numerischen Auswertung nicht die günstigste.

7.2 Das Verfahren von Neville

Das Verfahren von Neville berechnet den Wert des Interpolationspolynoms an einer gegebenen Stelle x . (Zum Verwechseln ähnlich ist das auch manchmal angegebene Aitken-Verfahren.)

Man setzt für $k = 0, 1, \dots, n$

$$P_i(x) = y_i$$

Dann ist P_i der Wert des eindeutig bestimmten Polynoms vom Grad 0, das durch (x_i, y_i) geht.

Nun sei P_{01} der Wert (an der gegebenen Stelle x) des eindeutig bestimmten Polynoms ersten Grades (der Geraden), das durch die Punkte (x_0, y_0) und (x_1, y_1) geht. Und dementsprechend: P_{12}, P_{23}, \dots . In gleicher Weise sei P_{012} der Wert der Parabel (an der gegebenen Stelle x) durch die Punkte $(x_0, y_0), (x_1, y_1)$ und (x_2, y_2) . Dieses Schema für Polynome höherer Ordnung weiterführend gelangt man zu $P_{0,1,2,\dots,n}$, dem Wert des Polynoms durch alle gegebenen Punkte und damit der gesuchten Antwort.

Der Trick an der Sache ist, dass sich die einzelnen P -Werte leicht aus jeweils zwei vorherigen berechnen lassen. Am einfachsten schreibt sich die Formel mit der Determinante einer 2×2 -Matrix,

$$P_{i(i+1)\dots(i+m)} = \frac{1}{x_{i+m} - x_i} \begin{vmatrix} x - x_i & P_{i(i+1)\dots(i+m-1)} \\ x - x_{i+m} & P_{(i+1)(i+2)\dots(i+m)} \end{vmatrix}$$

Durch Induktion zeigt man leicht, dass die so definierten P genau die behaupteten Eigenschaften haben. Zur tatsächlichen Berechnung bedient man sich dann eines Schemas (Tableaus) der Form

$$\begin{array}{cccc}
x_0 : & y_0 = P_0 & & \\
& & P_{01} & \\
x_1 : & y_1 = P_1 & P_{012} & \\
& & P_{12} & P_{0123} \\
x_2 : & y_2 = P_2 & P_{123} & \\
& & P_{23} & \\
x_3 : & y_3 = P_3 & &
\end{array}$$

Man füllt dieses Tableau spaltenweise von links nach rechts aus. Zum Beispiel ergibt sich der Wert P_{123} aus den beiden linken Nachbarn als

$$P_{123} = \frac{1}{x_3 - x_1} \begin{vmatrix} x - x_1 & P_{12} \\ x - x_3 & P_{23} \end{vmatrix} = \frac{(x - x_1)P_{23} - (x - x_3)P_{12}}{x_3 - x_1}$$

Will man die Interpolationsordnung erhöhen, kann man Werte für x_4 und y_4 unten an das Tableau anfügen und die fehlenden Werte P_{34} , P_{234} , P_{1234} und P_{01234} ergänzen, ohne das restliche Tableau neu berechnen zu müssen (Vorteil gegenüber der Lagrangeschen Formel). Übrigens müssen die x_i weder äquidistant noch irgendwie geordnet sein.

Beispiel: Dichte von Wasser

Die Dichte von Wasser in Abhängigkeit von der Temperatur ist in folgender Tabelle für den Bereich von 0 bis 100 Celsius angegeben

T	ρ	T	ρ
0	999.840	10	999.699
1	999.899	20	998.203
2	999.940	30	995.645
3	999.964	40	992.212
4	999.972	50	988.030
5	999.964	60	983.191
6	999.940	70	977.759
7	999.901	80	971.785
8	999.848	90	965.304
9	999.781	100	958.345

Dichtewerte für Temperaturen, die nicht in der Tabelle angeführt sind, findet man durch Interpolation oder Approximation. Welches Verfahren ist angebracht?

- Einzelne Werte werden benötigt: Neville-Verfahren.
- Ein formelmäßiger Zusammenhang $T = T(\rho)$ in einem engen Temperaturintervall wird benötigt: Newtonsches Interpolationspolynom
- Ein formelmäßiger Zusammenhang über den gesamten Bereich ist gesucht: Approximation durch ein Polynom oder eine rationale Funktion.

Neville

Gesucht sei $\rho(24)$. Ein durchgerechnetes Tableau lautet

10 :	999.699		
		997.605	
20 :	998.203	997.307	
		997.180	997.297
30 :	995.645	997.285	
		997.705	
40 :	992.212		

Die einzelnen Schritte dabei:

$$P_{01} = \frac{(x - x_0)P_1 - (x - x_1)P_0}{x_1 - x_0}, \quad P_{12} = \frac{(x - x_1)P_2 - (x - x_2)P_1}{x_2 - x_1}, \dots$$

Das sind die Werte, die sich aus linearer Interpolation ergeben. In vielen Fällen wird lineare Interpolation ausreichend genau sein; man hätte sich dann mit dem Wert P_{12} zufriedengeben können. Die nächste Spalte entspricht quadratischer Interpolation gemäß den Formeln

$$P_{012} = \frac{(x - x_0)P_{12} - (x - x_2)P_{01}}{x_2 - x_0}, \quad P_{123} = \frac{(x - x_1)P_{23} - (x - x_3)P_{12}}{x_3 - x_1}.$$

Die daraus gewonnenen Werte unterscheiden sich nur mehr um 2 Einheiten in der Hundertstel-Stelle. Der letzte Wert entsteht aus kubischer Interpolation:

$$P_{0123} = \frac{(x - x_0)P_{123} - (x - x_3)P_{012}}{x_3 - x_0},$$

Rechenprogramme, welche Stoffgrößen benötigen, können entweder umfangreiche Tabellen abspeichern und darin linear interpolieren; alternativ können sie Speicherplatz sparen, Tabellen mit weniger Stützstellen und Interpolation höherer Ordnung verwenden. Sehr oft steckt ein beträchtlicher Teil der Rechenzeit in der Auswertung von Stoffgrößen; eine rechengünstige Implementierung ist dann bedeutsam.

7.3 Das Newtonsche Interpolationspolynom

Die Aufgabenstellung zur polynomialen Interpolation lautet: Vorgegeben sind $n+1$ Zahlenpaare $(x_0, y_0), \dots, (x_n, y_n)$, wobei die x_i paarweise verschieden sind. Gesucht ist ein Polynom $p(x)$ vom Grad $\leq n$ durch die Datenpunkte. Das Interpolationspolynom soll also erfüllen: $p(x_i) = y_i$ für $0 \leq i \leq n$.

Häufig sind Werte einer Funktion tabellarisch gegeben, wie im folgenden Beispiel:

Spezifische Wärmekapazität von kohlenstoffarmem Stahl in J/kg K für Temperaturen zwischen 0 und 600C

T	cp
0	460.8
100	471.1
200	496.4
300	537.0
400	593.3
500	666.8
600	760.8

Gesucht sind Funktionswerte zwischen den Stützstellen der Tabelle.

Das Kapitel 7.2 illustriert an einem ähnlichen Beispiel das Verfahren von Neville, das sich gut eignet, wenn ein einziger Zwischenwert interpoliert werden soll und als Hilfsmittel nur Papier, Bleistift und ein billiger Taschenrechner zur Verfügung stehen. Das Newtonsche Interpolationsverfahren ist ein ähnlich rechengünstiges Schema, das nicht nur Funktionswerte, sondern eine formelmäßige Darstellung des Interpolationspolynoms liefert.

Das Interpolationspolynom in der Newtonschen Form verwendet *dividierte Differenzen*. Die erste dividierte Differenz zwischen x_0 und x_1 , symbolisch geschrieben in eckigen Klammern, ist durch

$$[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0}$$

erklärt. Sie ist die Steigung der Geraden durch die beiden Datenpunkte.

Damit lässt sich die Formel für lineare Interpolation (Interpolationspolynom ersten Grades) angeben:

$$p(x) = y_0 + (x - x_0)[x_0, x_1]$$

Die höheren dividierten Differenzen erklärt man durch die niedrigeren. Die zweite Differenz $[x_0, x_1, x_2]$ wird erklärt durch

$$[x_0, x_1, x_2] = \frac{[x_1, x_2] - [x_0, x_1]}{x_2 - x_0}$$

und allgemein ist für $k > i$

$$[x_i, x_{i+1}, \dots, x_{k+1}] = \frac{[x_{i+1}, \dots, x_{k+1}] - [x_i, \dots, x_k]}{x_{k+1} - x_i}$$

Dann hat $p(x)$ folgende, durch Induktion leicht beweisbare Darstellung,

$$p(x) = y_0 + (x - x_0)[x_0, x_1] + (x - x_0)(x - x_1)[x_0, x_1, x_2] + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})[x_0, x_1, \dots, x_n]$$

Die Berechnung der dividierten Differenzen von Hand organisiert man am besten in Tabellenform nach folgendem Schema

		x_0	y_0		
		$x_1 - x_0$		$[x_0, x_1]$	
	$x_2 - x_0$	x_1	y_1		$[x_0, x_1, x_2]$
		$x_2 - x_1$		$[x_1, x_2]$	
\dots	$x_3 - x_1$	x_2	y_2		$[x_1, x_2, x_3] \dots$
		$x_3 - x_2$		$[x_2, x_3]$	
	$x_4 - x_2$	x_3	y_3		$[x_2, x_3, x_4]$
		$x_4 - x_3$		$[x_3, x_4]$	
		x_4	y_4		

Man berechnet zuerst die Einträge in der linken Hälfte der Tabelle. In der rechten Tabellenhälfte erhält man die dividierten Differenzen nach der Regel „linker unterer minus linker oberer Nachbar, dividiert durch den symmetrisch gelegenen Eintrag aus der linken Tabellenhälfte“.

Die Koeffizienten des Newtonschen Interpolationspolynoms liegen in diesem Schema rechts in der oberen Schrägzeile (von der Mitte oben nach rechts schräg abwärts).

Bei äquidistanten x -Werten stehen in der linken Tabellenseite spaltenweise gleiche Werte, und der Rechengang könnte vereinfacht werden. Es gibt (z.B. im Bronstein) eine Vielzahl

spezieller Formeln für Interpolation mit äquidistanten x -Werten; hier beschränken wir uns auf den allgemeinen Fall. Ein durchgerechnetes Tableau für vier Datenpunkte aus dem obigen Beispiel sieht so aus:

		0	460,8		
	100			0,103	
200	100	471,1		0,000750	
300	100			0,253	0,000000050
	200	200	496,4	0,000765	
	100			0,406	
	300	537,0			

Das Newton-Interpolationspolynom hat in diesem Fall die Form

$$p(x) = 460,8 + x \cdot 0,103 + x(x - 100) \cdot 0,000750 + x(x - 100)(x - 200) \cdot 0,000000050$$

Es ist nicht ratsam, diese Formel zu „vereinfachen“, indem man Produkte wie $x(x - 100)(x - 200)$ ausmultipliziert und gleiche Potenzen in x zusammenfasst.

Man setzt x -Werte in die obige Formel direkt ein oder schreibt sie noch rechengünstiger in der Form

$$p(x) = 460,8 + x(0,103 + (x - 100)(0,000750 + (x - 200) \cdot 0,000000050))$$

Umformen auf $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ kann ganz drastische Rundungsfehler hervorrufen!

Die x -Werte in dieser Rechentabelle müssen, wie gesagt, nicht äquidistant sein; sie brauchen auch nicht der Reihe nach geordnet zu sein. Außerdem lässt sich die Tabelle leicht nachträglich ergänzen. Der Rest der Tabelle muss nicht neu berechnet werden, und auch im Interpolationspolynom kommt nur ein Term dazu, die anderen bleiben gleich. Das ist ein Vorteil des Newtonschen (und auch des Nevilleschen) Interpolationsverfahrens gegenüber dem Lagrangeschen Interpolationspolynom, bei dem sich alle Terme ändern, wenn ein Datenpunkt hinzugefügt oder geändert wird.

Will man zum Beispiel die gegebenen Daten im Bereich $200 \leq x \leq 600$ zuerst einmal nur linear approximieren, stellt man das Schema

	200	496,4	
400			0,661
	600	760,8	

auf und erhält $p(x) = 496,4 + (x - 200) \cdot 0,661$. Will man dann doch quadratisch interpolieren, fügt man ein Wertepaar an, ergänzt die Tabelle

	200	496,4		
	400		0,661	
100	600	760,8		0,00085
	-300		0,746	
	300	537,0		

und fügt dem Interpolationspolynom einen Term an: $p(x) = 496,4 + (x - 200) \cdot 0,661 + (x - 200)(x - 600) \cdot 0,00085$.

7.4 Interpolation oder Approximation?

Polynome hohen Grades neigen zu Oszillationen und sind bei der Auswertung in Standardform anfällig gegen Rundungsfehler. Ein Interpolationspolynom durch alle Datenpunkte im Beispiel zur Dichte von Wasser aus Abschnitt 7.2 liefert völlig sinnlose Werte.

Das mit den Matlab-Befehlen `polyfit` und `polyval` berechnete und ausgewertete Interpolationspolynom weicht in diesem Fall selbst an den Original-Datenpunkten (wo es genau stimmen sollte!) deutlich ab: $\rho(0) = 1000.2$ statt 999.8. Zwischen den Datenpunkten liefert es Phantasiwerte im Bereich $\pm 10^8$.

Numerisch genauere Berechnungen des Interpolationspolynoms würden zwar an den Original-Datenpunkten bessere Werte liefern, dazwischen aber trotzdem Schrott.

Eine formelmäßige Darstellung für den gesamten Datenbereich lässt sich mit Regressionspolynomen finden. Eine quadratische Approximation liefert Genauigkeit $\pm 0,5$,

$$\rho(T) = 1000.35 - 0.0614512 T - 0.00364033 T^2$$

ein Regressionspolynom vierten Grades

$$\rho(T) = 999.867 + 0.0545396 T - 0.00765475 T^2 + 0.0000434548 T^3 - 1.38985 \cdot 10^{-7} T^4$$

weicht von den Datenpunkten maximal um 0.03 ab.

Rationale Funktionen sind oft zur Approximation besser geeignet. Beispielsweise zeigt die Näherung

$$\rho(T) = \frac{999.844 + 9.63049 T - 0.0244379 T^2}{1 + 0.00956721 T - 0.0000163504 T^2}$$

einen maximalen Fehler von 0.004. Auch solche Näherungen lassen sich nach der Methode der kleinsten Quadrate finden: Ansatz mit unbestimmten Koeffizienten, Einsetzen der Daten, Umformen auf ein überbestimmtes lineares System und Lösen der Normalgleichungen. (Je nach Ansatz können aber auch *nichtlineare* Ausgleichsprobleme entstehen, die viel schwieriger zu behandeln sind.) Die obige Näherung minimiert das Maximum des Fehlers (sogenannte *Minimax-Näherung*). Für die Konstruktion solcher Näherungen sei auf weiterführende Literatur verwiesen (Das Computer-Algebra-System Mathematica bietet z. B. dafür ein Paket an).

7.5 Weitere Interpolationsverfahren

Nicht alle Funktionen lassen sich in vorteilhafter Weise durch Polynome interpolieren. Andere wichtige Verfahren sind

- Rationale Interpolation
- Spline-Interpolation
- Trigonometrische Interpolation
- Interpolation in zwei oder mehr Dimensionen

Rationale Funktionen sind Quotienten zweier Polynome. Bulirsch und Stoer (siehe deren Buch Numerische Mathematik 1) haben einen Algorithmus gefunden, der mit einem Tableau ähnlich wie bei der Aitken-Neville-Interpolation arbeitet. Die meisten Funktionen lassen sich besser durch rationale Funktionen approximieren als durch Polynome. Auch Ihr Taschenrechner wertet mathematische Funktionen wie Cosinus und Sinus höchstwahrscheinlich in Form rationaler Ausdrücke aus.

Spline-Funktionen spielen eine wichtige Rolle überall dort, wo Computer eine „glatte“ Kurve oder Oberfläche durch gegebene Punkte zeichnen sollen.

Trigonometrische Interpolation verwendet Summen von Sinus- und Cosinusfunktionen, oder komplexen Exponentialfunktionen, um eine Kurve durch einen Satz von Datenpunkten zu legen. Eng verwandt und überaus wichtig ist das Verfahren zur schnellen Fourier-Transformation (*Fast Fourier Transform, FFT*).

7.6 Spline-Interpolation

Interpolationspolynome hohen Grades sind rechnerisch aufwendig, oft sehr empfindlich gegenüber geringfügigen Änderungen der Daten (Koordinaten der Stützpunkte) und neigen zum Überspringen. Will man – etwa für zeichnerische Zwecke – Punkte durch eine „möglichst glatte“ Kurve verbinden, bietet sich die Idee an, einzelne Polynome niedrigeren Grades aneinanderzustückeln. Derartige Funktionen werden allgemein als Splines bezeichnet.

Splines

Unter einer Spline-Funktion versteht man eine stückweise auf Teilintervallen definierte Funktion, deren Teile an den Intervallgrenzen stetig oder sogar ein- bzw. mehrmals stetig differenzierbar aneinanderstoßen.

Kubische (natürliche) Splines

Ein kubischer Spline $s(x)$ durch die $n + 1$ Wertepaare $(x_i, y_i), i = 0, \dots, n$ ist folgendermaßen charakterisiert:

In den einzelnen Intervallen (x_{i-1}, x_i) ist $s(x)$ jeweils ein kubisches Polynom

An den Intervallgrenzen stimmen die Funktionswerte, die ersten und die zweiten Ableitungen rechts- und linksseitig überein.

Zusatzbedingung: (natürlicher Spline) Am linken und rechten Rand ist $s''(x) = 0$.

Die natürlichen kubischen Splines sind unter allen zweimal stetig differenzierbaren, interpolierenden Funktionen $f(x)$ diejenigen mit dem kleinsten

$$\int_{x_0}^{x_n} (f''(x))^2 dx$$

Dieses Integral mißt aber im gewissen Sinn die Welligkeit oder Gesamtkrümmung der Funktion. Dadurch kann es natürlich kaum zum Überschwingen kommen.

Mit natürlichen kubischen Splines lassen sich aber keine schleifen- oder kreisförmigen Kurven zeichnen. Dazu dienen *parametrische* Splines, die sowohl x - als auch y -Werte einer Kurve als Splinefunktion eines Parameters darstellen. Die Methode zum Kurvenzeichnen in Excel arbeitet mit solchen Splines.

Nicht alle Arten von Splines interpolieren Punkte. *Bezier*-Splines verwenden einen Teil der Punkte, um die Gestalt der Kurve zu beeinflussen. Kurvenzeichnen in MS-Paint funktioniert in dieser Weise.