Interpolation

Stephan Hering

Universität Rostock - Institut für Mathematik

Wintersemester 2024/2025

Gliederung

Begriffsklärung

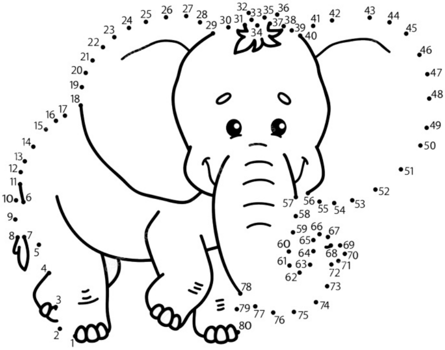

Abbildung 1: Malen nach Zahlen

Begriffsklärung

- lat. inter = dazwischen und polire = glätten, schleifen
- Prozess des Definierens einer Funktion, welche spezielle Werte an speziellen Stellen annimmt
- Ziel: Werte an Stellen zwischen gegebenen Punkten bestimmen
- die Funktion sollte möglichst ”glattßein, mit dem Ziel, dass sich diese möglichst oft ableiten lässt

Stückweise lineare Interpolation

- Datenpunkte werden mithilfe von geraden Linien miteinander verbunden
- Ausgangsituation: n Beobachtungspaare $(x_1, y_1), \ldots, (x_n, y_n)$
- Betrachtung: spezielles Intervall $[x_i, x_{i+1}]$ mit $i = 1, \ldots, n-1$ wobei $x_i \leq x < x_{i+1}$
- für die einzelnen Intervalle wird die Geradenform hergestellt

Stückweise lineare Interpolation

- Definieren der Ortsvariable

$$s = x - x_i \tag{1}$$

- Berechnung der Steigung

$$\delta_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{2}$$

- resultierende Interpolationsformel

$$L(x) = y_i + (x - x_i)\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \tag{3}$$

$$\Leftrightarrow L(x) = y_i + s\delta_i$$

Stückweise lineare Interpolation

beispielhafte Umsetzung in MATLAB

- Definieren der x-Werte
$$x = 1{:}6$$
- Definieren der dazugehörigen y-Werte
$$y = [10, 15, 9, 6, 2, 0]$$
- Plotten des Graphen `plot(x,y,'o',x,y,'-')`

Stückweise lineare Interpolation

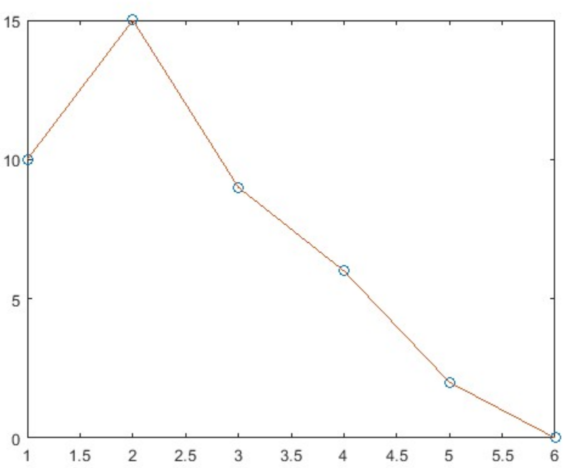

Abbildung 2: beispielhafte stückweise Lineare Interpolation

Stückweise lineare Interpolation

Aus Abb. 2 können wir auf auf folgende Vor- und Nachteile der linearen Interpolation schließen:

- \+ simple graphische Veranschaulichung der gegebenen Wertepaare
- \+ entstandener Graph ist stetig
- \- nicht jeder Verlauf von Messwerten verläuft linear
- \- Ableitung des Graphen ist nicht mehr stetig

Polynominterpolation

Wir wissen:

- Zwei Punkte (x_1,y_1) und (x_2,y_2) in der Fläche mit $x_1 \neq x_2$ definieren eine Gerade
- die Punkte definieren ein Polynom ersten Grades in x, dessen Gerade beide Punkte durchläuft

Für mehr als zwei Punkte bedeutet dies:

- n Punkte in der Fläche (x_i, y_i) mit $i = 1, \dots, n$ und unterschiedlichen x_i erzeugen ein Polynom vom Grad kleiner n
- das Polynom muss allerdings nicht vom Grad n-1 sein

Ziel: eine Funktion zu erhalten, die keine Knickstellen besitzt

Polynominterpolation

- Ein Polynom wird als Interpolationspolynom bezeichnet, wenn gilt:

$$P(x_i) = y_i, i = 1, ..., n \tag{4}$$

- P hat die Form:

$$P(x) = c_{n-1}x^{n-1} + ... + c_1 x + c_0 \tag{5}$$

- x_i werden als Stützstellen bezeichnet und y_i als Stützwerte

Polynominterpolation

Ein Lösungsansatz für das Interpolationsproblem ist über Aufstellen und Lösen eines linearen Gleichungssystems der Form:

$$P(x_1) = c_{n\text{-}1} {x_1}^{n\text{-}1} + \ldots + c_1 x_1 + c_0 = y_1$$

$$\vdots$$

$$P(x_n) = c_{n\text{-}1} {x_n}^{n\text{-}1} + \ldots + c_1 x_n + c_0 = y_n$$

Polynominterpolation

Das LGS kann in Matrixschreibweise formuliert werden, indem die sog. Vandermonde-Matrix mit einem Vektor multipliziert wird, welcher die Koeffizienten c_1 bis c_n beinhaltet. Dieses Produkt liefert dann den Ergebnisvektor, welcher die Werte y_1 bis $_n$ umfasst

$$\begin{pmatrix} x_1^{n-1} & x_1^{n-2} & \cdots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \cdots & x_2 & 1 \\ \cdots & \cdots & \cdots & \cdots & 1 \\ x_n^{n-1} & x_n^{n-2} & \cdots & x_n & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Abbildung 3: LGS als Matrix-Vektor-Produkt mittels Vandermonde-Matrix (links)

In MATLAB kann eine solche Vandermonde-Matrix über den Befehl *vander(x)* erstellt werden.

Polynominterpolation

Vandermonde-Matritzen besitzten schlechte numerische Eigenschaften. Darunter zählen u.A.:

- teils hohe Konditionszahlen
- verursachen gravierende Fehler durch fortgesetztes Runden bei Computerberechnungen

Polynominterpolation

Eines der meistgenutzen Interpolationspolynome ist das Lagrange-Polynom mit der Form:

$$P(x) = \sum_{i} (\prod_{j \neq i} \frac{x - x_{\mathrm{j}}}{x_{\mathrm{i}} - x_{\mathrm{j}}}) y_{\mathrm{i}} \tag{6}$$

Im Fall $x = x_i$

- alle Produkte außer das i-te ergeben 0
- das i-te Produkt ergibt 1, somit entspricht die Summe gleich y_i und erfüllt somit Gleichung (4)

Polynominterpolation

Beispiel für Polynominterpolation

- x = 0:3
- y = [-5 -6 -1 16]

Polynominterpolation

Beispiel für Polynominterpolation

- x = 0:3
- y = [-5 -6 -1 16]

Einsetzen in Lagrange-Interpolationsformel

$$P(x) = \frac{(x-1)(x-2)(x-3)}{(-6)}(-5) + \frac{x(x-2)(x-3)}{(2)}(-6) + \frac{x(x-1)(x-3)}{(-2)}(-1) + \frac{x(x-1)(x-2)}{(6)}(16)$$

Polynominterpolation

Beispiel für Polynominterpolation

- x = 0:3
- y = [-5 -6 -1 16]

Einsetzen in Lagrange-Interpolationsformel

$$P(x) = \frac{(x-1)(x-2)(x-3)}{(-6)}(-5) + \frac{x(x-2)(x-3)}{(2)}(-6) + \frac{x(x-1)(x-3)}{(-2)}(-1) + \frac{x(x-1)(x-2)}{(6)}(16)$$

Ergibt zusammengefasst:

$$x^3 - 2x - 5$$

Polynominterpolation

Nachteil von Polynominterpolation sind z.B.:

- teilweise große Polynomschwankungen innerhalb des Definitionsbereichs (Abb. 4)
- mit steigendem Polynom starke Oszillation außerhalb des Definitionsbereichs (Abb. 5)

Polynominterpolation

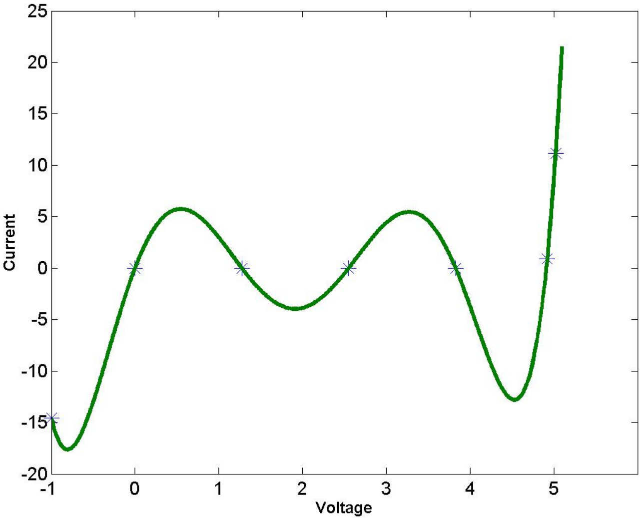

Abbildung 4: Spannungs-Strom-Kennlinie durch Polynom fünften Grades

Polynominterpolation

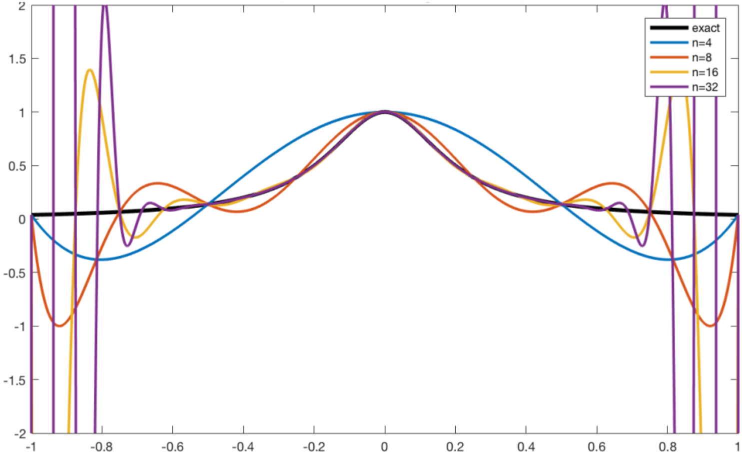

Abbildung 5: Graph der Runge-Funktion $f(x) = \frac{1}{(1+25x^2)}$ mit äquidistanten Stützstellen für n = 4, 8, 16, 32

Newton-Interpolation

- nutzt das mathematisches Konzept der dividierten Differenzen
- kann schrittweise erweitert werden, ohne die vorherigen Berechnungen zu beeinflussen
- jede Stufe der Berechnung baut auf den Ergebnissen der vorherigen Stufen auf
- Newtonsches Interpolationspolynom durch n Stützpunkte (x_i, y_i), i = 1, ... , n ist gegeben durch:

$$P(x) = c_0 + c_1(x - x_0) + ... + c_n(x - x_0)...(x - x_{n-1}) \tag{7}$$

Newton-Interpolation

Dividierte Differenzen lassen sich als Schema wie folgt darstellen:

$$
\begin{array}{cccccccc}
 & & & x_0 & y_0 & & & \\
 & & x_1 - x_0 & & & \mathrm{f}[x_0, x_1] & & \\
 & x_2 - x_0 & & x_1 & y_1 & & \mathrm{f}[x_0, x_1, x_2] & \\
 & & x_2 - x_1 & & & \mathrm{f}[x_1, x_2] & & \\
\ldots & x_3 - x_1 & & x_2 & y_2 & & \mathrm{f}[x_1, x_2, x_3] & \ldots \\
 & & x_3 - x_2 & & & \mathrm{f}[x_2, x_3] & & \\
 & x_4 - x_2 & & x_3 & y_3 & & \mathrm{f}[x_2, x_3, x_4] & \\
 & & x_4 - x_3 & & & \mathrm{f}[x_3, x_4] & & \\
 & & & x_4 & y_4 & & &
\end{array}
$$

Abbildung 6: Abb. 6: Schema der dividierten Differenzen

Newton-Interpolation

Aus Abb. 4 erkennt man, dass sich die dividierten Differenzen

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

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

allgemein formulieren lassen als

$$f[x_\mathrm{i}, x_\mathrm{i+1}, \dots, x_\mathrm{k+1}] = \frac{f[x_\mathrm{i+1}, \dots, x_\mathrm{k+1}] - f[x_\mathrm{i}, \dots, x_\mathrm{k}]}{x_\mathrm{k+1} - x_\mathrm{i}} \tag{8}$$

Die n Koeffizienten c_i lassen sich der Reihe nach berechnen:

$$P(x_0) = y_0 = c_0 \tag{9}$$

$$P(x_1) = y_1 = c_0 + c_1(x_1 - x_0)$$
$$\Leftrightarrow c_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

Newton-Interpolation

Beispiel für Newton-Interpolation

x_i	y_i
1	2
3	6
4	5

Newton-Interpolation

Beispiel für Newton-Interpolation

x_i	y_i
1	2
3	6
4	5

- Ermitteln der ersten dividierten Differenz:

$$f[1,3] = \frac{f(3) - f(1)}{3 - 1} = \frac{6 - 2}{3 - 1} = \frac{4}{2} = 2$$

Newton-Interpolation

Beispiel für Newton-Interpolation

x_i	y_i
1	2
3	6
4	5

- Ermitteln der ersten dividierten Differenz:

$$f[1,3] = \frac{f(3) - f(1)}{3 - 1} = \frac{6 - 2}{3 - 1} = \frac{4}{2} = 2$$

- Ermitteln der zweiten dividierten Differenz:

$$f[1,3,4] = \frac{f[1,4] - f[1,3]}{4 - 1} = \frac{(5 - 2) - (6 - 2)}{4 - 1} = -\frac{1}{3}$$

Newton-Interpolation

Beispiel für Newton-Interpolation

x_i	y_i
1	2
3	6
4	5

Das resultierende Polynom lautet dann:

$$P(x) = 2 + 2(x-1) - \frac{1}{3}(x-1)(x-3)$$

Newton-Interpolation

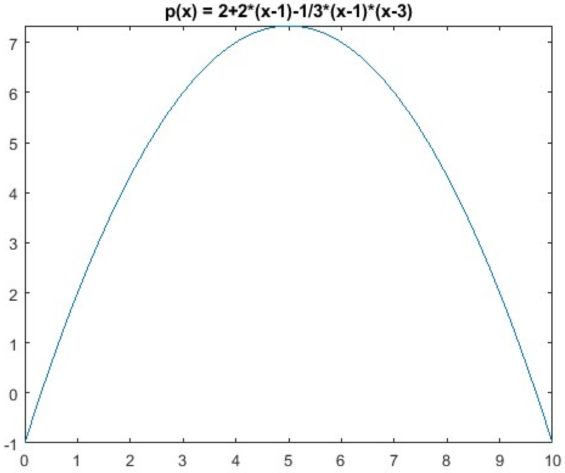

Abbildung 7: Graph von p(x) nach Newton-Interpolation

Hermite-Interpolation

- baut auf dem Interpolationsprinzip von Newton auf
- beinhaltet außerdem die Ableitungen an speziellen Stellen, sodass gilt:

$$P^{(j)}(x_i) = f_i^{(j)} \tag{10}$$

- für die dividierten Differenzen folgt

$$f[x_i, x_{i+1}] = f'(x_i)$$

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f''(x_i)}{2!}$$

$$\ldots$$

Hermite-Interpolation

Beispiel für eine Hermite-Interpolation
- gegeben sind die Informationen:

- $P(1) = 1$
- $P'(1) = 4$
- $P(2) = 3$
- $P'(2) = 1$
- $P''(2) = 2$

Hermite-Interpolation

Vorgehen:

- Die Anzahl an Informationen zu einer Stelle entspricht der Anzahl, wie oft diese Stelle in der zur Ermittlung der Koeffizienten mittels Tabelle betrachtet wird
  - in diesem Beispiel bezieht es sich zweimal auf die Stelle 1 (Ausgangsfunktion und 1. Ableitung) und dreimal auf die Stelle 2 (Ausgangsfunktion sowie erste und zweite Ableitung)
-

x_i	$P(x_i)$	$\frac{P''(x_i)}{1!}$	$\frac{P''(x_i)}{2!}$	$\frac{P'''(x_i)}{3!}$	
1	1	-	-	-	-
1	1	-	-	-	-
2	3	-	-	-	-
2	3	-	-	-	-
2	3	-	-	-	-

Hermite-Interpolation

Vorgehen:

- Werden gleiche Stellen betrachtet, wie z.B. bei der ersten dividierten Differenz, wird die dazugehörige Ableitung eingesetzt, welche jedoch durch n! geteilt wird wobei n dem Grad der Ableitung entspricht
-

x_i	$P(x_i)$	$P'(x_i)$	$\frac{P''(x_i)}{2!}$	$\frac{P'''(x_i)}{3!}$	
1	1	-	-	-	-
1	1	4	-	-	-
2	3	-	-	-	-
2	3	1	-	-	-
2	3	1	-	-	-

Hermite-Interpolation

Vorgehen:

- im Anschluss werden die verbleibenden dividierten Differenzen berechnet
-

x_i	$P(x_i)$	$\frac{P''(x_i)}{1!}$	$\frac{P''(x_i)}{2!}$	$\frac{P'''(x_i)}{3!}$	
1	1	-	-	-	-
1	1	4	-	-	-
2	3	2	-2	-	-
2	3	1	-1	1	-
2	3	1	1	2	1

Hermite-Interpolation

x_i	$P(x_i)$	$\frac{P''(x_i)}{1!}$	$\frac{P''(x_i)}{2!}$	$\frac{P'''(x_i)}{3!}$	
1	1	-	-	-	-
1	1	4	-	-	-
2	3	2	-2	-	-
2	3	1	-1	1	-
2	3	1	1	2	1

- Die Koeffizienten des Polynoms sind nun die Werte, welcher der Hauptdiagonalen der $P(x_i)$-Spalte entsprechen, also 1, 4, -2, 1, 1

$$P(X) = 1 + 4(x-1) - 2(x-1)^2 + 1(x-1)^2(x-2) + 1(x-1)^2(x-2)^2$$

Spline-Interpolation

- ermöglicht komplexere Kurven und Flächen mit hoher Präzision zu modellieren
- Ziel: Punkte durch Polynome niedrigen Grades so zu verbinden, dass eine glatte und steitge Kurve entsteht
- üblicherweise werden Polynome vom Grad drei (also kubisch) für die einzelnen Abschnitte verwendet
- an jedem Punkt, an dem zwei Polynome aufeinandertreffen, stimmen die erste und zweite Ableitung der Polynome überein

Spline-Interpolation

Berechung eines Splines zwischen zwei Punkten x_j und x_{j+1} über:

$$S(X)_{[x_j,\, x_{j+1}]} = f_j \frac{x_{j+1} - x}{x_{j+1} - x_j} + f_{j+1} \frac{x - x_j}{x_{j+1} - x_j} - \frac{1}{6} S''(x_j) \frac{(x_j - x)(x - x_j)}{x_{j+1} - x_j} \tag{11}$$

$$((x_{j+1} - x) + (x_{j+1} - x_j)) - \frac{1}{6} S''(x_{j+1}) \frac{(x_{j+1} - x)(x - x_j)}{x_{j+1} - x_j}$$

$$((x - x_j) + (x_{j+1} - x_j))$$

Spline-Interpolation

S″(x_0, ... , S″(x_n) werden über ein LGS berechnet.
Für die Randbedingungen ergeben sich die Fälle:

- Fall 1: natürlicher Spline $=>$ $S''(x_0) = S''(x_n) = 0$
- Fall 2: Hermite-Splines $=>$ Habe $S'(x_0) = c$ und $S'(x_n) = d$ gegeben mit

$$S'(x_0) = c = \frac{f(x_1) - f(x_0)}{x_1 - x_0} - (\frac{1}{3}S''(x_0) + \frac{1}{6}S''(x_1))(x_1 - x_0)$$

$$S'(x_n) = d = \frac{f(x_n) - f(x_{n\text{-}1})}{x_n - x_{n\text{-}1}} + (\frac{1}{6}S''(x_{n\text{-}1}) + \frac{1}{3}S''(x_n))(x_n - x_{n\text{-}1})$$

Spline-Interpolation

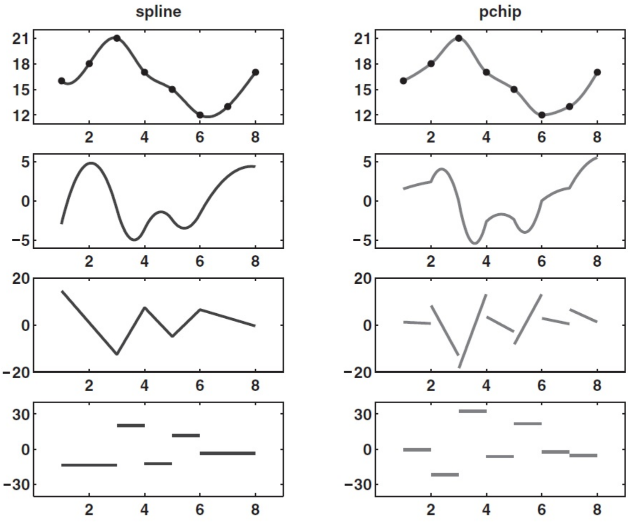

Abbildung 8: Vergleich Spline-Interpolation mit Hermite-Interpolation einschließlich der ersten 3 Ableitungen

Quellen

Literatur

- Moler, Numerical Computing with Matlab
- Stoer, Numerische Mathematik 1
- Engel, Anwendungsorientierte Mathematik: Von Daten zur Funktion. Eine Einführung in die mathematische Modellbildung für Lehramtsstudierende

Bildquellen

- Abb. 1: razukraski.com
- Abb. 4: dmpeli-math-mcmaster-ca.translate.goog/Matlab/Math4Q3/Lecture2-1/
- Abb. 5: katana.iwr.uni-heidelberg.de
- Abb. 6: unileoben.ac.at
- Abb. 8: Moler, Numerical Computing with Matlab

Gibt es noch Fragen?