Approximation mit Polynomen

Approximation durch Polynome

Anwendungen:

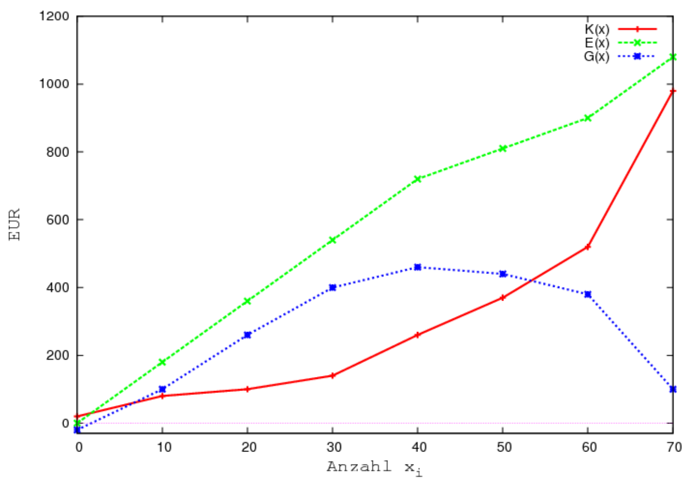
- zur Vereinfachung einer gegebenen Funktion durch einen Polynomausdruck. Dann sind übliche Rechenoperation $+, -, \cdot, /$ möglich.
- zur Interpolation von Daten einer Tabelle

Beispiel

Trotz guter Verkaufslage verringern sich die Gewinne $G(x)$ ab einer bestimmten Stückzahl x aufgrund steigender Produktionskosten.

Anzahl x_i	Kosten $K(x_i)$	Erlös $E(x_i)$	Gewinn $G(x_i)$
0	20	0	- 20
10	80	180	100
20	100	360	260
30	140	540	400
40	260	720	460
50	370	810	440
60	520	900	380
70	980	1080	100

Kurvenverlauf von $K(x)$, $E(x)$, $G(x)$ als Geradenstücke



Frage:

- Wo liegt die Gewinnzone ?
- Wo liegt das Gewinnmaximum?

Antwort:

- Gewinnzone: Die Bestimmung der Nullstelle(n) der Funktion notwendig, d.h. $G(x) = 0$.
- Gewinnmaximum: Die notwendige Bedingung für das Funktionsmaximum: $G'(x) = 0$.

Fazit:

Man benötigt eine Funktionsdarstellung für $G(x)$, z.B. durch Approximation der Daten über eine Polynominterpolation.

Polynome

besonders geeignet in der Numerik, da die Auswertung nur einfache Addition und Multiplikation bedeutet.

Definition 1

Eine Funktion $f(x)$ für $f : \mathbb{R} \to \mathbb{R}$ der Form

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

mit $a_n \neq 0$ heißt Polynom vom Grad n. Die reellen Zahlen $a_0, a_1, \ldots, a_n$ heißen Koeffizienten des Polynoms.

Beispiel

$f(x) = 4x^8 - x^5 - 10$

Rechnen mit Polynomen

Definition 2

Gegeben sind zwei Polynome $f(x)$ und $g(x)$ der Form
$f(x) = a_m x^m + \ldots + a_1 x + a_0$ und
$g(x) = b_n x^n + \ldots + b_1 x + b_0$
vom Grad m und n mit $m \leq n$.
Dann ist:

$$\begin{aligned} f(x) + g(x) &= b_n x^n + \ldots + b_{m+1} x^{m+1} + (a_m + b_m) x^m \\ &\quad + \ldots + (a_1 + b_1) x + (a_0 + b_0) \end{aligned}$$

und

$$\begin{aligned} f(x) \cdot g(x) &= (a_m b_n) x^{n+m} + \ldots + (a_1 b_n + \ldots \\ &\quad + a_m b_{n-m+1}) x^{n+1} + (a_0 b_n + \ldots + a_m b_{n-m}) x^n \\ &\quad + \ldots + (a_0 b_1 + a_1 b_0) x + a_0 b_0 \end{aligned}$$

Teilbarkeit durch einen Linearfaktor

Satz 3

Für jedes Polynom $f(x)$ und jeden Wert x_0 gilt:

$$\begin{aligned} f(x) &= a_n x^n + \dots . a_1 x + a_0 \\ &= (x - x_0)(b_n x^{n-1} + \dots + b_2 x + b_1) + f(x_0) \end{aligned}$$

mit $f(x_0) = r$.

Beweis.

(Herleitung des Horner - Schemas)

$$\begin{aligned} &(x - x_0) \cdot (b_n x^{n-1} + \dots + b_2 x + b_1) + r \\ &\qquad = b_n x^n + b_{n-1} x^{n-1} + \ldots + b_2 x^2 + b_1 x \\ &\qquad\quad - x_0 b_n x^{n-1} - \ldots - x_0 b_2 x - x_0 b_1 + r \\ &\qquad = a_n x^n + \ldots + a_1 x + a_0 \end{aligned}$$

$\square$

Beweis.

Ein Koeffizientenvergleich nach Potenzen von x ergibt

$$\begin{aligned} b_n &= a_n \\ b_{n-1} &= a_{n-1} + x_0 b_n \\ b_{n-2} &= a_{n-2} + x_0 b_{n-1} \\ \vdots & \quad \vdots \\ b_1 &= a_1 + x_0 b_2 \\ r &= a_0 + x_0 b_1 \end{aligned}$$

$\square$

Horner - Schema

Verfahren zur systematischen Auswertung von Polynomen:

$$\begin{array}{ccccccc}
 & a_n & a_{n-1} & \dots & a_2 & a_1 & a_0 \\
+ & & x_0 b_n & \dots & x_0 b_3 & x_0 b_2 & x_0 b_1 \\
\hline
 & b_n \nearrow & b_{n-1} \nearrow & \dots & b_2 \nearrow & b_1 \nearrow & r = f(x_0)
\end{array}$$

Beispiel

Gegeben ist die Funktion $f(x) = 2x^4 - 6x^3 - 35x + 10$.
Gesucht ist der Funktionswert an der Stelle $x_0 = 4$ über das Horner-Schema.

$$
\begin{array}{c|cccccc}
 & 2 & -6 & 0 & -35 & 10 & \\
+ & & 2\cdot 4 & 2\cdot 4 & +8\cdot 4 & -3\cdot 4 & \\
\hline
 & 2 & 2 & +8 & -3 & -2 & = f(4)
\end{array}
$$

$\Longrightarrow f(4) = -2.$

Bemerkung:
Das Horner-Verfahren ist besonders effektiv, da es Potenzieren vermeidet; Daher ist es für numerische Anwendungen geeignet.

Nullstellenproblem

Satz 4

Wenn x_1 eine Nullstelle des Polynoms n-ten Grades $f(x)$ ist, dann gilt:

$$f(x) = (x - x_1)(b_n x^{n-1} + \ldots + b_1 x)$$

Bemerkung:

Das Horner - Schema liefert die Zerlegung von $f(x)$.

Beispiel

Gegeben ist die Funktion $f(x) = x^3 + 2x^2 - 13x + 10$. Gesucht ist das Restpolynom bei Abspaltung des Linearfaktors um $x_1 = 1$.

$$\begin{array}{cccccc} & 1 & 2 & -13 & 10 \\ + & & 1 & 3 & -10 \\ \hline & 1 & 3 & -10 & 0 \end{array}$$

$$\Longrightarrow f(x) = (x-1) \cdot (x^2 + 3x - 10).$$

Polynomdivision

Dasselbe Ergebnis ergibt sich durch Polynomdivision:

$$
\begin{array}{l}
x^3 + 2x^2 - 13x + 10 : x - 1 = x^2 + 3x - 10 \\
\underline{-(x^3 - x^2)} \\
\qquad\quad 3x^2 - 13x \\
\qquad \underline{-(3x^2 - 3x)} \\
\qquad\qquad\quad -10x + 10 \\
\qquad\qquad\quad \underline{-10x + 10} \\
\qquad\qquad\qquad\qquad 0
\end{array}
$$

Bemerkung:

Dividiert man $f(x)$ durch $(x - x_1)$ ohne Rest, so ist das Resultat ein Polynom vom Grad $n - 1$.

Satz 5

Jedes Polynom n-ten Grades hat höchstens n verschiedene Nullstellen.

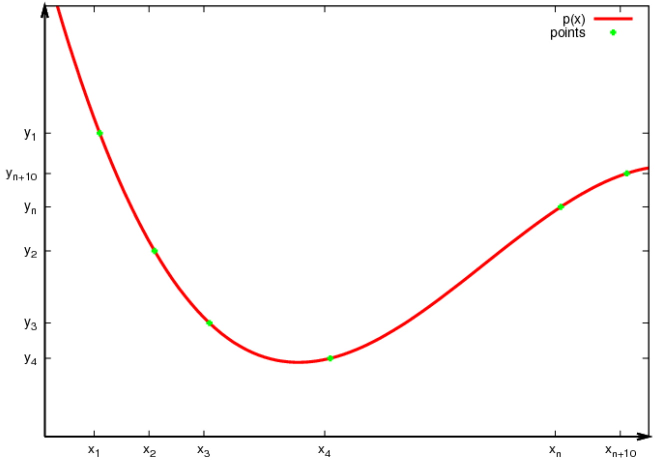
Interpolationspolynom

Satz 6

Zu $n+1$ Wertepaaren $(x_1, y_1), (x_2, y_2), \ldots, (x_{n+1}, y_{n+1})$ mit paarweise verschiedenen x_i gibt es genau eine Polynomfunktion $f(x)$ mit $f(x_i) = y_i, i = 1, \ldots, n+1$, deren Grad nicht größer als n ist. Das Polynom heißt Interpolationspolynom.

Bemerkung:
Aus den diskreten Werten (x_i, y_i) lassen sich durch das Interpolationspolynom näherungsweise beliebige Zwischenwerte berechnen (interpolieren).

Schematische Darstellung des Interpolationspolynoms



Vandermondsche Matrix

Die Bestimmung des Interpolationspolynoms erfolgt durch Einsetzen der diskreten Werte (x_i, y_i) in die Funktion $f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ und durch Bestimmung der Koeffizienten $a_0, \ldots, a_n$ aus den $n+1$ Bedingungen $f(x_i) = y_i$:

$$\begin{array}{ccccccccccc} a_n x_1^n & + & \ldots & + & a_2 x_1^2 & + & a_1 x_1 & + & a_0 & = & y_1 \\ \vdots & & & & \vdots & & \vdots & & \vdots & & \vdots \\ a_n x_{n+1}^n & + & \ldots & + & a_2 x_{n+1}^2 & + & a_1 x_{n+1} & + & a_0 & = & y_{n+1} \end{array}$$

Eigenschaften:

- zu lösen ist ein lineares Gleichungssystem mit den Unbekannten $a_0, \ldots, a_n$
- das Verfahren ist numerisch ungünstig, insbesondere für große n

Methode 2: Lagrange Formel

Ansatz:

$$
\begin{aligned}
f(x) \quad = \quad & y_1 \frac{(x-x_2)(x-x_3)....(x-x_{n+1})}{(x_1-x_2)(x_1-x_3)...(x_1-x_{n+1})} + \\
& y_2 \frac{(x-x_1)(x-x_3)....(x-x_{n+1})}{(x_2-x_1)(x_2-x_3)...(x_2-x_{n+1})} + \\
& \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
& y_{n+1} \frac{(x-x_1)(x-x_2)....(x-x_n)}{(x_{n+1}-x_1)(x_{n+1}-x_2)..(x_{n+1}-x_n)}
\end{aligned}
$$

Eigenschaften:

- $\Rightarrow f(x_i) = y_i$
- das Verfahren ist ebenfalls für große Datenmengen rechenzeitaufwendig

Methode 3: Newton - Algorithmus

(numerisch geschickteste Methode)

Ansatz:

$$\begin{aligned} f(x) &= a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \ldots \\ &\quad + a_n(x - x_1)(x - x_2)\cdots(x - x_n) \end{aligned}$$

Die Koeffizienten $a_0, a_1, \ldots, a_n$ können iterativ bestimmt werden. Einsetzen von x_1 in die Funktion $f(x)$ ergibt:

$$\begin{aligned} y_1 &= f(x_1) = a_0 \quad \Rightarrow a_0 = y_1 \\ y_2 &= f(x_2) = a_0 + a_1(x_2 - x_1) \quad \Rightarrow a_1 = \frac{y_2 - y_1}{x_2 - x_1} \\ y_3 &= f(x_3) = a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) \\ &\ldots\ldots \quad \Rightarrow a_2 = \frac{y_3 - a_0 - a_1(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} \end{aligned}$$

Newton - Algorithmus (fortgesetzt)

Algebraisches Umstellen ergibt:

$$\begin{aligned} a_2 &= \frac{y_3 - a_0 - a_1(x_3 - x_1)}{(x_3 - x_1)(x_3 - x_2)} \\ &= \dots \\ &= \frac{1}{x_3 - x_1}\left(\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}\right) \\ \Rightarrow a_2 &= \frac{D_{3,2} - D_{2,1}}{x_3 - x_1} \end{aligned}$$

mit den Abkürzungen:

$$D_{2,1} = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{und} \quad D_{3,2} = \frac{y_3 - y_2}{x_3 - x_2}$$

Newton - Algorithmus (fortgesetzt)

Mit den Abkürzungen:

$$D_{4,3,2} = \frac{D_{4,3} - D_{3,2}}{x_4 - x_2} \quad \text{und} \quad D_{3,2,1} = \frac{D_{3,2} - D_{2,1}}{x_3 - x_1}$$

und über vollständige Induktion kann man zeigen, dass gilt:

$$\boxed{a_{k-1} = D_{k,\dots,1}}$$

das sogenannte Verfahren der dividierten Differenzen.

Berechnungsschema: Newton - Algorithmus

$$
\begin{array}{c|c|c|lllll}
k & x_k & y_k & & & & & \\
\hline
1 & x_1 & y_1 & = \color{red}{a_0} & & & & \\
 & & & \searrow & & & & \\
2 & x_2 & y_2 & \to D_{2,1} = \frac{y_2 - y_1}{x_2 - x_1} & = \color{red}{a_1} & & & \\
 & & & \searrow & \searrow & & & \\
3 & x_3 & y_3 & \to D_{3,2} = \frac{y_3 - y_2}{x_3 - x_2} & \to & D_{3,2,1} = \frac{D_{3,2} - D_{2,1}}{x_3 - x_1} & & \\
 & & & \searrow & \searrow & & \searrow & \\
4 & x_4 & y_4 & \to D_{4,3} = \frac{y_4 - y_3}{x_4 - x_3} & \to & D_{4,3,2} = \frac{D_{4,3} - D_{3,2}}{x_4 - x_2} & \to & \\
\vdots & \vdots & \vdots & \quad\quad\vdots & & \quad\quad\vdots & &
\end{array}
$$

wobei $a_0 = y_1, a_1 = D_{2,1}, a_2 = D_{3,2,1}, \ldots, a_n = D_{n+1,\ldots,1}$ die gesuchten Koeffizienten des Interpolationspolynoms sind.

Beispiel

Gegeben ist die folgende Wertetabelle:

k	1	2	3
x_k	0	2	5
y_k	-12	16	28

Gesucht ist das Interpolationspolynom $f(x)$ der Form
$f(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$.

Lösung über den Newton-Algorithmus:

k	x_k	y_k		
1	0	-12		
2	2	16	$D_{2,1} = \frac{16+12}{2} = 14$	
3	5	28	$D_{3,2} = \frac{28-16}{3} = 4$	$D_{3,2,1} = \frac{4-14}{5} = -2$

$$\begin{aligned} \Rightarrow f(x) &= -12 + 14x - 2x(x-2) \\ &= -2x^2 + 18x - 12 \end{aligned}$$

Beispiel

Gegeben ist ein zusätzlicher Messpunkt $(x_4, y_4) = (7, -54)$. Gesucht ist wieder das Interpolationspolynom, das durch alle Messpunkte verläuft.

Lösung über den Newton-Algorithmus:

k	x_k	y_k			
1	0	-12			
2	2	16	$D_{2,1} = 14$		
3	5	28	$D_{3,2} = 4$	$D_{3,2,1} = -2$	
4	7	-54	$D_{4,3} = -41$	$D_{4,3,2} = -9$	$D_{4,3,2,1} = -1$

$$\begin{aligned} \Rightarrow f(x) &= -12 + 14x - 2x(x-2) - x(x-2)(x-5) \\ &= \quad \ldots \quad = -x^3 + 5x^2 + 8x - 12 \end{aligned}$$

Taylorreihen

Fast jede elementare Funktion (wie z.B. $e^x, \sin x, \sqrt{x}, \ln x$ usw.) lässt sich in der Umgebung eines Punktes x_0 durch einen Polynomausdruck $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$ beliebig genau annähern.

Beispiel

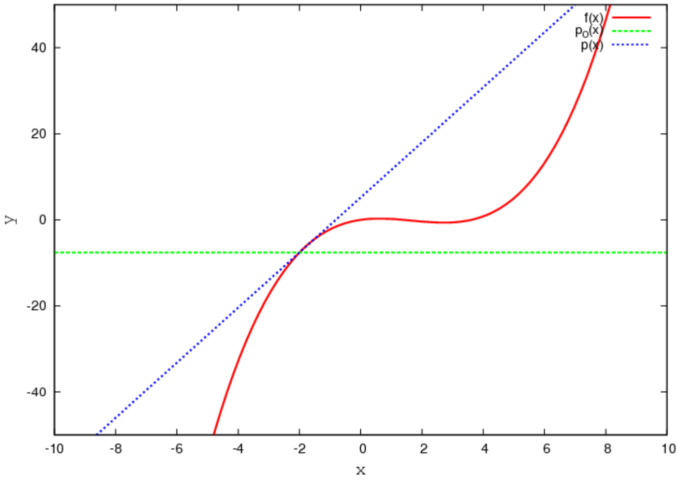
Gegeben ist die Funktion $f(x) = \frac{1}{1-x}$ für $|x| < 1$. Gesucht ist ein Polynomausdruck, der $f(x)$ annähert.

Lösung

$$\frac{1}{1-x} = 1 + x + x^2 + \ldots + x^n + \ldots = \sum_{n=0}^{\infty} x^n$$

Herleitung der Taylorformel

Gegeben ist eine Funktion $f(x)$ und gesucht ist eine Näherung in der Umgebung von $x_0 \in D$.



Herleitung der Taylorformel (fortgesetzt)

Nullte Näherung $p_0(x)$

$$p_0(x) = f(x_0)$$

zwischen $p_0(x)$ und $f(x)$ stimmt nur der Funktionswert an der Stelle x_0 überein.

Erste (lineare) Nährung $p_1(x)$

$$p_1(x) = f(x_0) + f'(x_0)(x - x_0)$$

(Tangente an die Funktion $f(x)$ im Punkt x_0)
der Funktionswert und die erste Ableitung sind im Punkt x_0 identisch.

Herleitung der Taylorformel (fortgesetzt)

Zweite (quadratische) Nährung $p_2(x)$

$$p_2(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2$$

$p_2(x)$ hat zusätzlich die gleiche Krümmung wie $f(x)$ im Punkt x_0.

Hieraus lässt sich eine Formel für das n-te Näherungspolynom, das sogenannte Taylorpolynom n-ten Grades entwickeln.

Taylorpolynom

Satz 7

(Satz von Taylor 1685 - 1731)
Gegeben sei eine in $x_0 \in D$ $(n+1)$*-mal stetig differenzierbare Funktion* $f(x)$.
Dann gilt die Taylor-Formel:

$$\begin{aligned}(p_n(x) =) f(x) &= f(x_0) + f'(x_0)(x - x_0) + \dots \\ &\quad \dots + \frac{1}{n!} f^{(n)}(x_0)(x - x_0)^n + R_n(x)\end{aligned}$$

mit dem Restglied

$$R_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\zeta)(x - x_0)^{n+1} \quad \textit{mit} \quad x \in D$$

und ζ *einem Zwischenwert zwischen* x *und* x_0.

Bemerkungen zur Taylorformel

- der Wert von ζ ist nicht näher bekannt
- das Restglied $R_n(x)$ gibt die Abweichung zwischen der Nährungsfunktion $p_n(x)$ und der Funktion $f(x)$ an; d.h. $R_n(x)$ ist ein Maß für den Fehler
- für $R_n(x) \to 0$ erhält man die sogenannte Taylorreihe für die Funktion $f(x)$ im Entwicklungspunkt x_0.

Beispiel

Gegeben ist die Funktion $f(x) = e^x$. Gesucht ist die Taylorentwicklung im Punkt $x_0 = 0$.

Zur Bestimmung der Taylorreihe wird der Funktionswert $f(x_0)$ und die Werte der Ableitungen $f'(x) = e^x, f''(x) = e^x, \ldots$ im Punkt x_0 benötigt.
Mit

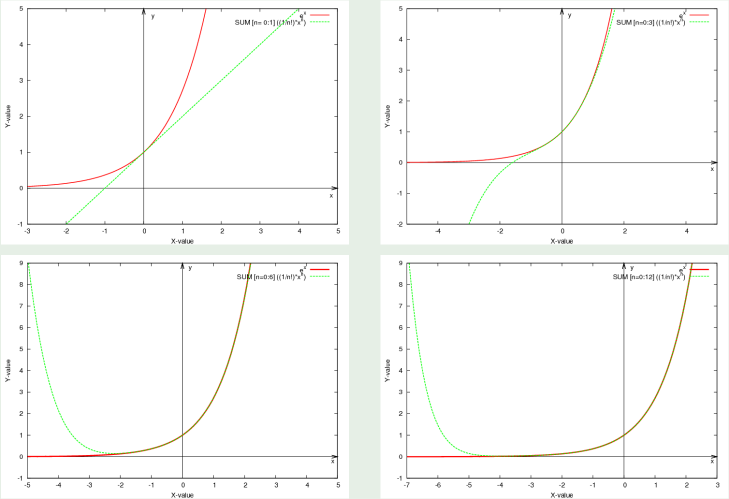
$$f(0) = 1, f'(0) = 1, \ldots, f^{(n)}(0) = 1$$

folgt

$$\begin{aligned} f(x) &= 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \ldots + \frac{1}{n!}x^n + \ldots \\ \Rightarrow e^x &= \sum_{n=0}^{\infty} \frac{1}{n!}\, x^n \end{aligned}$$

Taylorreihe der Funktion e^x für alle $x \in \mathbb{R}$

Beispiel (fortgesetzt)



Beispiel (fortgesetzt)

An der Stelle $x = 1$ ergibt sich die Eulersche Zahl e. Einsetzen in die Taylorformel für $f(x) = e^x$ liefert:

$$e^1 \cong p_n(1) = 1 + 1 + \frac{1}{2!}1 + \ldots + \frac{1}{n!}1^n$$

Mit $n = 9$:

$$e^1 \cong \sum_{n=0}^{9} \frac{1}{n!} = 2.7182815$$

Bemerkung:

Für eine Genauigkeit von 6 Dezimalstellen werden nur neun Summationsglieder benötigt.

Beispiel

Gegeben ist die Funktion $f(x) = \sin x$. Gesucht ist die Taylorentwicklung im Punkt $x_0 = 0$.

Die Ableitungen von $f(x) = \sin x$ sind:

$$\begin{aligned} f'(x) &= \cos x, \quad f''(x) = -\sin x, \\ f'''(x) &= -\cos x, \quad f^{(4)}(x) = \sin x, \quad \ldots \end{aligned}$$

Damit ergeben sich an der Stelle $x_0 = 0$ die Werte:

$$f'(0) = 1, f''(0) = 0, f'''(0) = -1, f^{(4)}(0) = 0, \ldots$$

Beispiel (fortgesetzt)

Kürzer formuliert:

$$f^{(2n)}(0) = 0 \quad \text{und} \quad f^{(2n+1)}(0) = (-1)^n$$

Daraus ergibt sich für die Funktion $f(x) = \sin x$ folgende Taylorreihe

$$f(x) = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \ldots - \ldots$$

und in Summenschreibweise

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

Splines - stückweise definierte Polynome

Eigenschaft der Interpolationspolynome:
Je mehr Messdaten vorhanden sind, umso höher ist der Grad des interpolierenden Polynoms. Daher ist es bei großen Datensätzen in der Praxis nicht möglich, diese beliebig genau durch eine einzelne Funktion $f(x)$ zu approximieren.

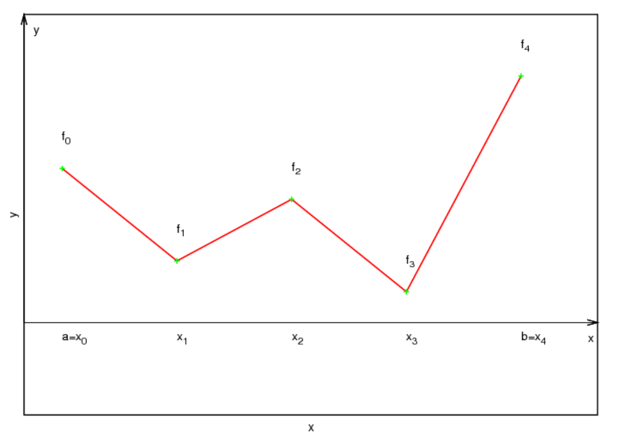
Alternative:
Die Funktion $f(x)$ wird durch stückweise definierte Polynome niedrigen Grades, sogenannte Splines definiert.

Vorgehen:
Die Funktion $f(x)$ soll in einem Intervall $[a,b]$ bestimmt werden. Dazu wird das Intervall $[a,b]$ in n Teilintervalle $[x_i, x_{i+1}]$ unterteilt mit

$$a = x_0 < x_1 < \ldots < x_n = b$$

Dann definieren wir eine Funktion $g(x)$, die sich in jedem Teilinterwall $[x_i, x_{i+1}]$ aus Polynomausdrücken zusammensetzt. $g(x)$ ist eine stückweise definierte Polynomfunktion.

Beispiel



Quadratische Splines

Beispiel

Approximation von Messwerten durch stückweise quadratische Polynome:

Gegeben ist ein Intervall $I = [0, 1]$ und die Wertetabelle

x	0	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{5}{6}$	1
$f(x)$	1	3	2	1	0	2	1

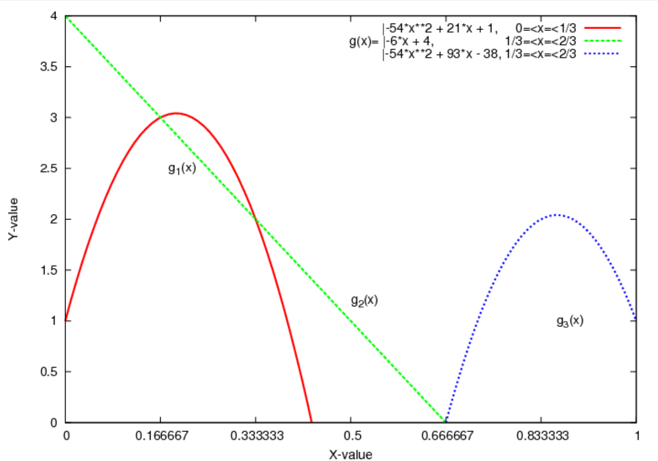
Lösung

$$g(x) = \begin{cases} -54x^2 + 21x + 1 & 0 \leq x \leq \frac{1}{3} \\ -6x + 4 & \frac{1}{3} \leq x \leq \frac{2}{3} \\ -54x^2 + 93x - 38 & \frac{2}{3} \leq x \leq 1 \end{cases}$$

$g(x)$ ist eine stückweise definierte Funktion mit $g(x_i) = f(x_i)$.

Lösung (fortgesetzt)

$$g(x) = \begin{cases} -54x^2 + 21x + 1 & 0 \leq x \leq \frac{1}{3} \\ -6x + 4 & \frac{1}{3} \leq x \leq \frac{2}{3} \\ -54x^2 + 93x - 38 & \frac{2}{3} \leq x \leq 1 \end{cases}$$



Problem: $g(x)$ ist nicht überall differenzierbar an den Schnittstellen.

$\Rightarrow$ Splines: Im gesamten Intervall differenzierbare, stückweise definierte Funktion.

Beispiel: Herleitung quadratischer Splines

Vorgehensweise zur Bestimmung quadratischer Splines am Beispiel für $n = 4$ Messpunkte.

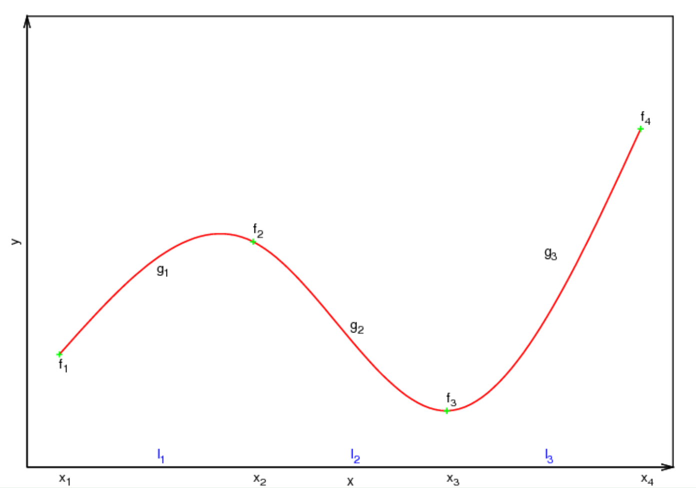
Es gibt drei Teilintervalle $I_i = [x_i, x_{i+1}], i = 1, 2, 3$

und entsprechend drei quadratische Funktionen

$$g_i(x) = a_{i2}x^2 + a_{i1}x + a_{i0}, \quad i = 1, 2, 3.$$

Gesucht ist eine Funktion $g(x)$ mit $g(x) = g_i(x)$ für $x \in I_i, i = 1, 2, 3.$

Beispiel: Herleitung quadratischer Splines (fortgesetzt)



Zur Bestimmung von $g(x)$ müssen die Funktionen $g_i(x), i = 1, 2, 3$ definiert werden;

d.h. es müssen die neun Koeffizienten $a_{12}, a_{11}, a_{10}, a_{22}, a_{21}, a_{20}, a_{32}, a_{31}, a_{30}$ bestimmt werden. Hierzu sind 9 Bedingungen erforderlich:

1. Stetigkeit von $g(x)$ an den Knotenpunkten:

$$\begin{array}{lll} g_1(x_1) = f_1, & g_2(x_2) = f_2, & g_3(x_3) = f_3 \\ g_1(x_2) = f_2, & g_2(x_3) = f_3, & g_3(x_4) = f_4 \end{array}$$

2. Differenzierbarkeit von $g(x)$ an den Knotenpunkten:

$$g_1'(x_2) = g_2'(x_2) \quad g_2'(x_3) = g_3'(x_3)$$

3. Zusätzliche 9. Bedingung, die vorgegeben ist:
z.B. $g_1'(x_1) = c_1$ mit $c_1 \approx f'(x_1)$ (Approximation)

$\Longrightarrow g(x)$ festgelegt.

Allgemeiner:

n Knoten $\Rightarrow n-1$ Intervalle und $n-1$ Parabeln $g_i(x)$

Bedingungen:

1. Stetigkeit:

$$\begin{aligned} g_i(x_i) &= f_i, \quad i = 1, \ldots, n-1 \\ g_i(x_{i+1}) &= f_{i+1}, \quad i = 1, \ldots, n-1 \end{aligned}$$

2. Differenzierbarkeit:

$$g_i'(x_{i+1}) = g_{i+1}'(x_{i+1}), \quad i = 1, \ldots, n-2$$

$\Rightarrow$ Das sind $3n-4 = (n-1)+(n-1)+(n-2)$ lineare Gleichungen für die Bestimmung der $3n-3$ unbekannten Koeffizienten der Polynome $g_1(x), \ldots, g_{n-1}(x)$.

3. Zusätzliche Bedingung:

$$g'(x_1) = c_1$$

Zu lösen ist ein System aus $3n - 3$ linearen Gleichungen.

Beispiel

Gegeben sind folgende Messpunkte

x	1	2	3
$f(x)$	2	3	5

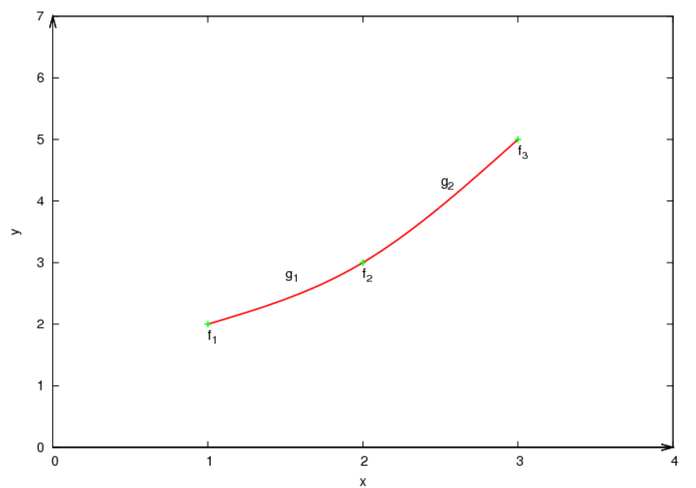
und die Nebenbedingung $g_1'(1) = 0$.

Gesucht ist die quadratische Spline-Funktion:

$$g(x) = \begin{cases} g_1(x) = a_{12}x^2 + a_{11}x + a_{10} & 1 \leq x \leq 2 \\ g_2(x) = a_{22}x^2 + a_{21}x + a_{20} & 2 \leq x \leq 3 \end{cases}$$

Beispiel (fortgesetzt)

Quadratische Spline-Funktion anschaulich:



Lösung

Es werden die Ableitungen $g_1'(x)$ und $g_2'(x)$ benötigt:

$$g_1'(x) = 2a_{12}x + a_{11} \quad \text{und} \quad g_2'(x) = 2a_{22}x + a_{21}$$

1. Stetigkeit:

$$\begin{aligned} g_1(1) &= 2 \qquad & g_2(2) &= 3 \\ g_1(2) &= 3 \qquad & g_2(3) &= 5 \end{aligned}$$

2. Differenzierbarkeit:

$$g_1'(2) = g_2'(2)$$

3. Zusätzliche Bedingung:

$$g_1'(1) = 0$$

Lösung (fortgesetzt)

Aus drei der Bedingungen kann ein Gleichungssystem für die Koeffizienten a_{12}, a_{11} und a_{10} der Funktion $g_1(x)$ aufgestellt werden.

$$\begin{array}{ccccccc} a_{12} & + & a_{11} & + & a_{10} & = & 2 \\ 4a_{12} & + & 2a_{11} & + & a_{10} & = & 3 \\ 2a_{12} & + & a_{11} & & & = & 0 \end{array}$$

Mit dem Gaußalgorithmus:

$$\begin{array}{ccc|c} 1 & 1 & 1 & 2 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & 0 & 0 \\ \hline & \vdots & & \vdots \end{array}$$

$\Rightarrow a_{10} = 3,\ a_{11} = -2,$ und $a_{12} = 1$

$\Rightarrow g_1(x) = x^2 - 2x + 3$

Lösung (fortgesetzt)

Aus den übrigen drei Bedingungen kann ein Gleichungssystem für die Koeffizienten a_{22}, a_{21} und a_{20} der Funktion $g_2(x)$ aufgestellt werden.

$$\begin{array}{ccccccc} 4a_{22} & + & 2a_{21} & + & a_{20} & = & 3 \\ 9a_{22} & + & 3a_{21} & + & a_{20} & = & 5 \\ 4a_{22} & + & a_{21} & & & = & 2 \end{array}$$

Mit dem Gaußalgorithmus:

$$\begin{array}{ccc|c} 4 & 2 & 1 & 3 \\ 9 & 3 & 1 & 5 \\ 4 & 1 & 0 & 2 \\ \hline & \vdots & & \vdots \end{array}$$

$\Rightarrow a_{20} = -1,\ a_{21} = 2,$ und $a_{22} = 0$

$\Rightarrow g_2(x) = 2x - 1$

Kubische Splines

Bemerkung:

Die quadratischen Splines besitzen an den Punkten, an denen die Teilintervalle zusammenstoßen, im allgemeinen keine einstellbare zweite Ableitung.

Will man bestimmte Werte für die zweite Ableitung an den Übergangspunkten, benötigt man Polynome dritten Grades, sogenannte kubische Splines. Diese sind auf dem ganzen Intervall zweimal stetig differenzierbar.

Vorgehensweise:

analog wie im Fall der quadratischen Splines, d.h. für die Koeffizienten der kubischen Polynome wird ein Gleichungssystem aufgestellt.

Alternative:

Die gesuchte Funktion wird als Linearkombination aus kubischen Basis-Splines (B-Splines) ausgedrückt.

Kubische Splines

Kubische Splines finden unter anderem in der graphischen Datenverarbeitung Einsatz, da sie vom Menschen subjektiv als glatt empfunden werden.

Computermodell eines menschlichen Kopfes

Kubische Splines

Gegeben sind n Knoten $x_1, \dots, x_n$. Ein kubischer Spline ist eine glatte Kurve, die durch die gegebenen Punkte verläuft und in jedem Teilstück durch eine kubische Parabel

$$g_i(x) = a_{i3}x^3 + a_{i2}x^2 + a_{i1}x + a_{i0}$$

mit geeigneten Koeffizienten a_{i3}, a_{i2}, a_{i1} und a_{i0} definiert ist.

Zu jedem Abschnitt gibt es also 4 Unbekannte. Bei $n-1$ Abschnitten (d.h. n Knoten) müssen $4(n-1)$ Unbekannte durch Lösen eines Gleichungssystems bestimmt werden.

Vorgehensweise: An den Knoten werden folgende Bedingungen formuliert: die Funktionswerte sowie die Werte der ersten und zweiten Ableitungen der angrenzenden Teilkurven müssen übereinstimmen.

Bedingungen:

Stetigkeit:

$$\begin{aligned} g_i(x_i) &= f_i, \quad i = 1, \ldots, n-1 \\ g_i(x_{i+1}) &= f_{i+1}, \quad i = 1, \ldots, n-1 \end{aligned}$$

Differenzierbarkeit 1. Ableitung: Gleiche Steigung

$$g_i'(x_{i+1}) = g_{i+1}'(x_{i+1}), \quad i = 1, \ldots, n-2$$

Differenzierbarkeit 2. Ableitung: Gleiche Krümmung

$$g_i''(x_{i+1}) = g_{i+1}''(x_{i+1}), \quad i = 1, \ldots, n-2$$

$\Rightarrow$ Das sind $(4n-6) = (n-1) + (n-1) + (n-2) + (n-2)$ lineare Gleichungen für die Bestimmung der $(4n-4)$ unbekannten Koeffizienten der kubischen Polynome $g_1(x), \dots, g_{n-1}(x)$.

Also ist die Interpolationsaufgabe für kubische Splines unterbestimmt, so dass zwei zusätzliche Freiheitsgrade zur Verfügung stehen. Durch weitere Zusatzbedingungen lässt sich die Eindeutigkeit herstellen.

Randbedingungen für kubische Splines

Natürliche Randbedingungen:

Die zweite Ableitung des Splines am Anfangs- und Endpunkt wird Null gesetzt. Das bewirkt, dass der Spline eine minimale Gesamtkrümmung hat.

$$g_1''(x_1) = 0 \quad \text{und} \quad g_{n-1}''(x_n) = 0$$

Vorgabe von Randableitungen:

Die erste Ableitung am Anfangs- und Endpunkt wird vorgegeben.

$$g_1'(x_1) = c_l \quad \text{und} \quad g_{n-1}'(x_n) = c_r$$

Spline-Interpolation

Erstellen einer Geschwindigkeitskurve: Gemessen sind die Zeiten eines Sprintläufers an verschiedenen charakteristischen Messpunkten, z.B. nach $5m$, $15m$, $30m$, $75m$, $85m$ und $100m$.

Strecke in m	Zeit in sec
5	1,5
15	2,5
30	4,6
75	9,5
85	11
100	13,5

Durch eine Spline-Interpolation läßt sich eine Geschwindigkeitskurve aufstellen und berechnen, nach wieviel Sekunden der Läufer z.B. die 50-Meter-Marke erreicht hat.

Spline-Interpolation (fortgesetzt)

Beispielprogramm

Beispiel

Gegeben sind folgende Messpunkte

x	0	1	2
$f(x)$	4	1	2

und es soll die natürliche Randbedingung angesetzt werden, d.h. $g_1''(x_1) = 0$ und $g_{n-1}''(x_n) = 0$.

Gesucht ist die kubische Spline-Funktion:

$$g(x) = \begin{cases} g_1(x) = a_{13}x^3 + a_{12}x^2 + a_{11}x + a_{10} & 0 \leq x \leq 1 \\ g_2(x) = a_{23}x^3 + a_{22}x^2 + a_{21}x + a_{20} & 1 \leq x \leq 2 \end{cases}$$

Lösung

Es werden die Ableitungen $g_1'(x), g_2'(x)$ und $g_1''(x), g_2''(x)$ benötigt:

$$\begin{aligned} g_1'(x) &= 3a_{13}x^2 + 2a_{12}x + a_{11}, g_1''(x) = 6a_{13}x + 2a_{12} \\ g_2'(x) &= 3a_{23}x^2 + 2a_{22}x + a_{21}, g_2''(x) = 6a_{23}x + 2a_{22} \end{aligned}$$

1. Stetigkeit:

$$\begin{array}{ll} g_1(0) = 4 & g_2(1) = 1 \\ g_1(1) = 1 & g_2(2) = 2 \end{array}$$

2. Differenzierbarkeit:

$$g_1'(1) = g_2'(1) \quad \text{und} \quad g_1''(1) = g_2''(1)$$

3. Natürliche Randbedingungen:

$$g_1''(0) = 0 \quad \text{und} \quad g_2''(2) = 0$$

Lösung (fortgesetzt)

Aus diesen 8 Bedingungen kann ein Gleichungssystem für die 8 Koeffizienten $a_{13}, a_{12}, \ldots, a_{21}, a_{20}$ der beiden kubischen Funktionen $g_1(x)$ und $g_2(x)$ aufgestellt werden:

$$
\begin{array}{cccccccc|c}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 4 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 8 & 4 & 2 & 1 & 2 \\
3 & 2 & 1 & 0 & -3 & -2 & -1 & 0 & 0 \\
6 & 2 & 0 & 0 & -6 & -2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 12 & 2 & 0 & 0 & 0 \\
\hline
 & \vdots & & & \vdots & & & \vdots & \vdots
\end{array}
$$

$\Rightarrow a_{13} = 1, a_{12} = 0, a_{11} = -4, a_{10} = 4$ und $a_{23} = -1, a_{22} = 6, a_{21} = -10, a_{20} = 6$

$\Rightarrow g_1(x) = x^3 - 4x + 4$ und $g_2(x) = -x^3 + 6x^2 - 10x + 6.$

Alternative: Kubische Basis-Splines (B-Splines)

Eine kubische Spline-Funktion wird als Linearkombination der Basis-Splines dargestellt.

Definition 8

Seien $x_1, \ldots, x_n$ äquidistante Punkte mit Abstand h. Dann ist der um x_i zentrierte kubische B-Spline B_i definiert durch:

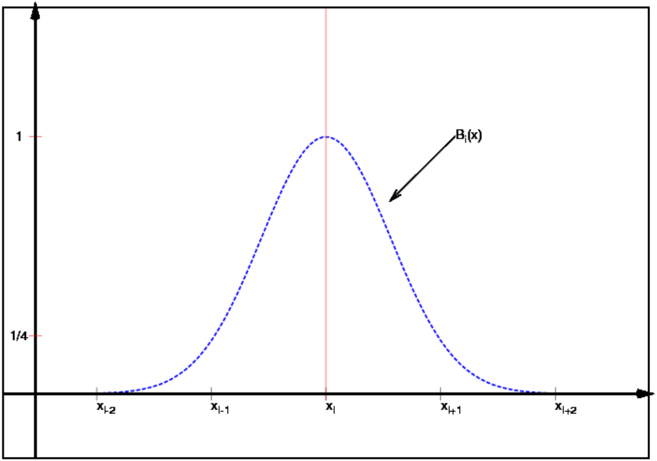
$$\frac{1}{4h^3}(x - x_{i-2})^3, x_{i-2} \leq x \leq x_{i-1}$$

$$\frac{1}{4} + \frac{3}{4h}(x - x_{i-1}) + \frac{3}{4h^2}(x - x_{i-1})^2 - \frac{3}{4h^3}(x - x_{i-1})^3, x_{i-1} \leq x \leq x_i$$

$$\frac{1}{4} + \frac{3}{4h}(x_{i+1} - x) + \frac{3}{4h^2}(x_{i+1} - x)^2 - \frac{3}{4h^3}(x_{i+1} - x)^3, x_i \leq x \leq x_{i+1}$$

$$\frac{1}{4h^3}(x_{i+2} - x)^3, x_{i+1} \leq x \leq x_{i+2}$$

$$0, \quad \text{sonst}$$



Darstellung des kubischen B-Splines

Kubische B-Splines

Eigenschaften:

$$B_i(x_i) = 1, \quad B_i(x_{i\pm1}) = \frac{1}{4} \quad \text{und} \quad B_i(x_{i\pm2}) = 0$$

Vorgehensweise:

Gesucht ist ein kubischer Spline $c(x)$, der durch die gegebenen Knoten verläuft, d.h.

$$c(x_i) = f_i, \quad i = 1, \ldots, n$$

Lösungsansatz: $c(x)$ wird als Linearkombination der B-Splines dargestellt

$$c(x) = \sum_{i=1}^{n} \alpha_i B_i.$$

Kubische B-Splines (fortgesetzt)

Da $B_i(x_j) = 0$ für $|i-j| \geq 2$, führen die Bedingungen $c(x_i) = f_i, i = 1, \ldots, n$ auf:

$$
\begin{aligned}
\alpha_1 B_1(x_1) + \alpha_2 B_2(x_1) &= f_1 \\
\alpha_1 B_1(x_2) + \alpha_2 B_2(x_2) + \alpha_3 B_3(x_2) &= f_2 \\
\vdots \qquad &\quad \vdots \\
\alpha_{n-2} B_{n-2}(x_{n-1}) + \alpha_{n-1} B_{n-1}(x_{n-1}) + \alpha_n B_n(x_{n-1}) &= f_{n-1} \\
\alpha_{n-1} B_{n-1}(x_n) + \alpha_n B_n(x_n) &= f_n
\end{aligned}
$$

Dies ist ein lineares Gleichungssystem zur Berechnung der Linearfaktoren α_i.

Mit $B_i(x_i) = 1$, und $B_i(x_{i\pm1}) = \frac{1}{4}$ ergibt sich die Matrixschreibweise:

$$\frac{1}{4}\begin{bmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{bmatrix}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

Dieses System besitzt eine eindeutige Lösung für die Koeffizienten α_i. Somit gibt es eine Lösung für die kubische Splinefunktion $c(x)$ mit $c(x_i) = f_i$.

$c(x)$ ist ein kubisches Polynom, das es sich als Linearkombination kubischer Funktionen ergibt.