

Hauptseminar Mathematische Modellierung SoSe 2019

Johannes-Gutenberg Universität Mainz

## Interpolation von Daten

J. Engel: Anwendungsorientierte Mathematik: Von Daten zur Funktion, Springer (2018)

Kap. 3.1 – 3.3.3, S. 105 – 120

Ausarbeitung zum Vortrag am 27.05.2019

Dozentin: Prof. Dr. Mária Lukáčová-Medvidová

Referentin: Kathrin Langens

# Inhaltsverzeichnis

1 Einleitung .....	1
2 Beispiel Sprinter.....	2
3 Lineare Interpolation .....	3
4 Interpolation durch Polynome.....	5
4.1 Lagrange-Interpolation .....	7
4.1.1 Beispiel: Lagrange-Interpolation für $n = 3$ .....	8
4.1.2 Beispiel: Lagrange-Interpolation 100m-Sprinter .....	9
4.2 Newton-Interpolation .....	9
4.2.1 Beispiel: Newton-Interpolation für $n = 3$ .....	12
4.2.2 Beispiel: Newton-Interpolation 100m-Sprinter.....	12
4.3 Vergleich der zwei Methoden.....	13
5 Beispiel Sprinter.....	13
Literaturverzeichnis .....	16

# 1 Einleitung

Wollen wir funktionale Abhängigkeiten zwischen zwei Variablen modellieren, so reicht unser Kontextwissen oftmals nicht aus, um direkt eine Funktionsklasse festzulegen. Somit können wir dann nur anhand der vorliegenden Daten argumentieren. Um die Daten für sich allein sprechen zu lassen, können Interpolationen verwendet werden.

Doch was ist überhaupt eine Interpolation?

Um diese Frage zu beantworten kann man sich das Spiel „Malen nach Zahlen“ vor Augen führen. Hierbei geht es darum, nummerierte Punkte so zu verbinden, dass ein Bild entsteht. Dafür genügen mehr oder weniger gerade Linien.

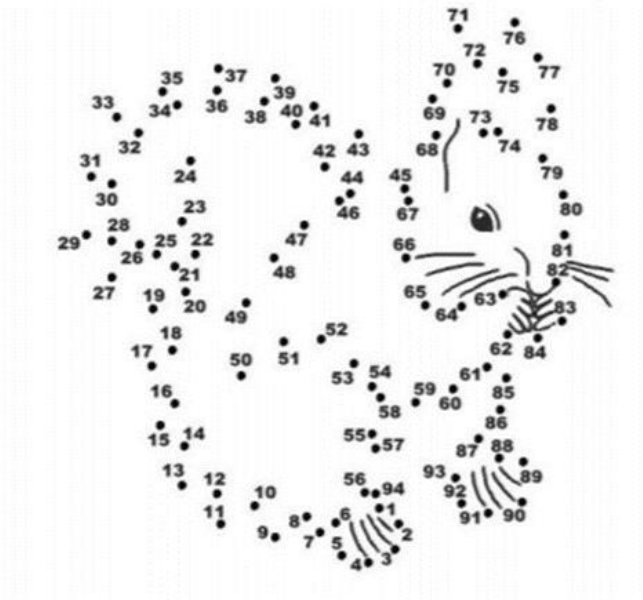


Abbildung 1: Malen nach Zahlen (<https://www.mathi.uni-heidelberg.de/~thaeter/anasem08/Interpolation.pdf>, abgerufen am 14.05.2019)

Bei Interpolationen ist es ähnlich. Wir haben  $n + 1$  Punkte, z.B. durch eine physikalische Messung, in einem Koordinatensystem gegeben und wollen diese Punkte nun verbinden, um Zwischenwerte bestimmen zu können, ohne für jeden Wert eine neue Messung durchführen zu müssen. Dafür verbinden wir die Punkte mithilfe einer Funktion, die durch die Punkte verlaufen soll und die Zwischenwerte möglichst gut annähert. Unser Anspruch an die Funktion ist hierbei noch, dass die Funktion möglichst „glatt“ sein soll, also sich diese möglichst oft ableiten lässt.

## 2 Beispiel Sprinter

Bei einem 100-Meter-Sprinter wurden nach 5m, 15m, 30m, 75m, 85m und 100m die Zeiten und Geschwindigkeiten gemessen. Die gewonnenen Daten sind in der Tabelle 1 aufgeführt.

Tabelle 1: Messdaten eines 100m-Sprinters

Strecke in m	$x_i$	0	5	15	30	75	85	100
Zeit in Sekunden	$t_i$	0	1,5	2,5	5	9,5	11	13,5
Geschwindigkeit in km/h	$v_i$	0	12	21,6	27	28,4	27,81	26,667

Die Herausforderung besteht nun darin, die Geschwindigkeitskurve  $x \rightarrow v(x)$  bzw.  $t \rightarrow v(t)$  zu rekonstruieren. Der Trainer möchte folgende Fragen beantworten:

Wo hatte der Läufer seine maximale Geschwindigkeit?

Wo hatte er eine maximale Beschleunigung?

Um diese Fragen zu beantworten, stellen wir die gemessenen Werte zunächst in einem Koordinatensystem als Streudiagramm dar (siehe Abbildung 2).

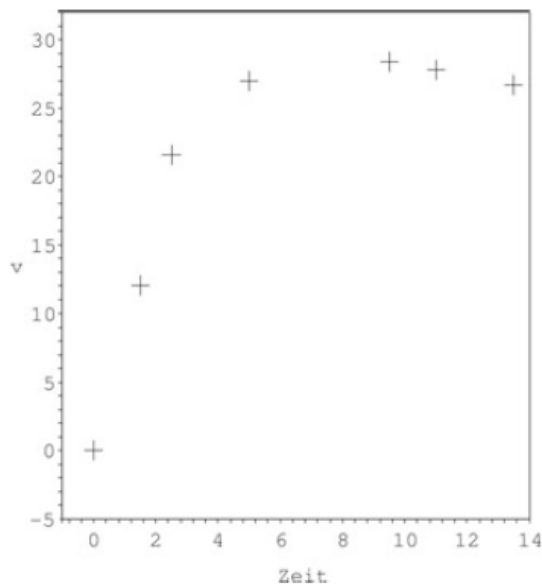


Abbildung 2: Zeit versus Geschwindigkeit eines 100m-Sprinters

Um nun den Geschwindigkeitsverlauf des Läufers genauer beschreiben zu können, verbinden wir die gegebenen Werte mit geraden Linien. Dies nennt man dann auch Liniendiagramm oder Polygonzug und mithilfe dieser linearen Interpolation der Daten können wir mögliche Trends visuell erkennen (siehe Abbildung 3).

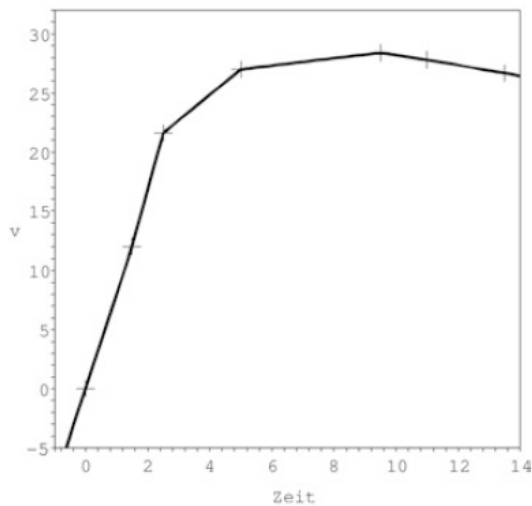


Abbildung 3: Lineare Interpolation der Daten des 100m-Sprinters

Anhand der Abbildung 3 erkennen wir nun den groben Geschwindigkeitsverlauf des Sprinters und können erste Vermutungen aufstellen. Die maximale Beschleunigung des 100m-Sprinters liegt hier in dem Bereich zwischen 0 und 3 Sekunden und die maximale Geschwindigkeit wird vermutlich nach einer Zeit von ca. 9 Sekunden erreicht.

### 3 Lineare Interpolation

Bei der linearen Interpolation werden die Datenpunkte mithilfe von geraden Linien miteinander verbunden. Durch diese Methode ist es möglich, gegebene Trends visuell zu erfassen.

Bei der allgemeinen linearen Interpolation betrachten wir  $n$  Beobachtungspaare  $(x_1, y_1), \dots, (x_n, y_n)$  und dann im Speziellen ein Intervall  $[x_i, x_{i+1}]$ ,  $i = 1, \dots, n - 1$ . Für die einzelnen Intervalle wollen wir nun eine Geradenform aufstellen. Dies geschieht mithilfe der Zweipunkteform der Geradengleichung. Die Herleitung für diese Formel erfolgt über die Steigungsdreiecke (siehe Abbildung 4). Dabei wird mithilfe der Steigungsdreiecke gezeigt, dass  $\frac{y-y_1}{x-x_1} = \frac{y_2-y_1}{x_2-x_1}$ . Lösen wir nun die Gleichung nach  $y$  auf, so erhalten wir die allgemeine Geradengleichung. Dieses lautet dann wie folgt:

$$f[x_i, x_{i+1}](x) = y_i + \frac{y_{i+1}-y_i}{x_{i+1}-x_i}(x - x_i).$$

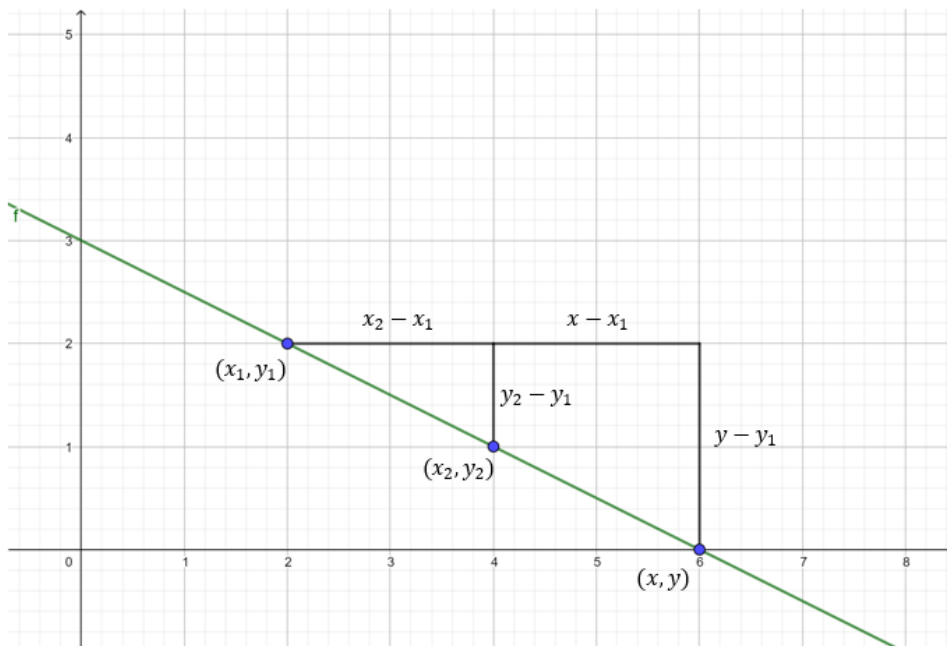


Abbildung 4: Herleitung der Zweipunkteform der Geradengleichung (eigene Darstellung)

Anschließend betrachten wir die Funktionen der Form:

$$H_i(x) := \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}], \quad i > 1, \\ 0 & \text{sonst} \end{cases}$$

und so folgt direkt:  $H_i(x_{i-1}) = 0, H_i(x_i) = 1, H_i(x_{i+1}) = 0$ .

Die Funktionen  $H_i$  heißen **Hutfunktionen**.  $H_i$  ist eine stückweise lineare Funktion auf den Teilintervallen  $[x_{i-1}, x_i]$  und  $[x_i, x_{i+1}]$ , die außerhalb dieser Intervalle als 0 definiert ist. Bis  $x_i$  steigt die Funktion von 0 auf 1 an und danach fällt sie wieder linear auf 0 ab. Das Geradenstück auf dem Intervall  $[x_i, x_{i+1}]$  hat dann die Form:

$$f_{[x_i, x_{i+1}]}(x) = y_i H_i(x) + y_{i+1} H_{i+1}(x),$$

denn  $f_{[x_i, x_{i+1}]}(x_i) = y_i$  und  $f_{[x_i, x_{i+1}]}(x_{i+1}) = y_{i+1}$ .

In der Abbildung 5 sind drei Beispiele für Hutfunktionen dargestellt:  $H_1(x)$  für  $x_1 = 1, x_2 = 3$  und  $x_3 = 4$ ,  $H_2(x)$  für  $x_1 = 3, x_2 = 4$  und  $x_3 = 7$  und  $H_3(x)$  für  $x_1 = 6, x_2 = 7$  und  $x_3 = 8$ .

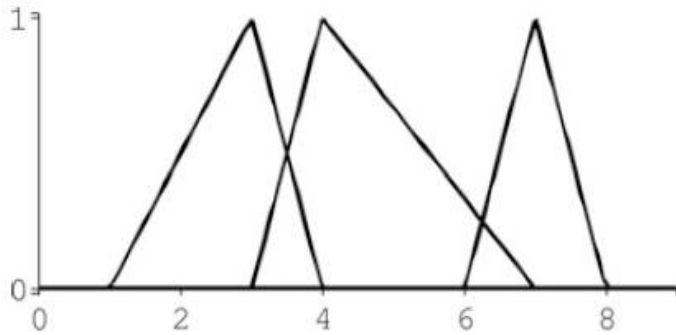


Abbildung 5: Beispiele für Hutfunktionen

Als Formel für die Geschwindigkeit des Sprinters (vgl. Kapitel 2) ergibt sich die linear interpolierende Funktion:

$$f_{\text{Sprinter}}(x) = \sum_{i=1}^7 v_i H_i(x)$$

mit  $v_i = i$ -te gemessene Geschwindigkeit.

Solche Funktionen heißen **linear interpolierende Splines**.

## 4 Interpolation durch Polynome

Wir haben uns in dem Kapitel 3 bereits die lineare Interpolation von Daten angeschaut. Wenn wir nun jedoch versuchen, die zu anfangs gestellten Fragen nach der maximalen Geschwindigkeit und der maximalen Beschleunigung eines 100m-Sprinters (vgl. Kapitel 2) zu beantworten, stellen wir fest, dass dies mithilfe der linearen Interpolation nicht sehr gut geschehen kann. Wir können bisher lediglich eine Vermutung aufstellen und haben eine Funktion mit vielen Knickstellen.

Um eine Funktion zu erhalten, die keine Knickstellen besitzt, wollen wir die Datenpunkte nun mithilfe von Polynomen interpolieren.

Damit ein solches Polynom existiert und eindeutig definiert ist, muss es im Fall von  $n$  vorliegenden Beobachtungen vom Grad  $\leq n - 1$  sein.

In unserem Beispiel des 100m-Sprinters liegen uns 7 Daten vor. Somit suchen wir ein Polynom 6-ten Grades der Form:

$$p_{\text{Sprinter}}(x) = a_0 + a_1x + \dots + a_6x^6.$$

Bevor nun zwei Methoden zur Berechnung des Interpolationspolynoms vorgestellt werden, benötigen wir noch ein paar allgemeine Definitionen.

Gegeben seien  $n$  Punkte  $(x_i, y_i)$ ,  $i = 1, \dots, n$  mit  $x_i \neq x_j$  für  $i \neq j$  (dagegen ist  $y_i = y_j$  erlaubt) und gesucht ist ein Polynom möglichst niedrigen Grades mit der Eigenschaft

$$p(x_i) = y_i, \quad i = 1, \dots, n.$$

Man nennt  $p$  dann das **Interpolationspolynom**.

$p$  ist eindeutig bestimmt mit  $\text{Grad} \leq n - 1$  und hat die folgende Form:

$$p(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}.$$

Dabei heißen die  $x_i$  die **Stützstellen**, die  $y_i$  die **Stützwerte** und  $(x_i, y_i)$  heißen die **Stützpunkte** für  $i = 1, \dots, n$ .

Sind die Stützwerte  $y_i$  die Ordinaten einer gegebenen Funktion  $f$  an den Stützstellen  $x_i$ , also  $y_i = f(x_i)$ , so interpoliert das Interpolationspolynom  $p$  die Funktion  $f$ .

Um das Interpolationsproblem praktisch zu lösen, gibt es verschiedene Möglichkeiten. Der erste Lösungsansatz ist vermutlich das Aufstellen und Lösen eines linearen Gleichungssystems mit  $n$  Gleichungen. Dies sieht dann wie folgt aus:

$$p(x_1) = a_0 + a_1x_1 + \dots + a_{n-1}x_1^{n-1} = y_1$$

$$p(x_2) = a_0 + a_1x_2 + \dots + a_{n-1}x_2^{n-1} = y_2$$

⋮

$$p(x_n) = a_0 + a_1x_n + \dots + a_{n-1}x_n^{n-1} = y_n$$

bzw. in der Matrixschreibweise:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & x_3^3 & \dots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

Die Koeffizientenmatrix hat hierbei eine besondere Struktur. In der zweiten Spalte der Matrix stehen die Stützstellen  $x_1, \dots, x_n$  und die anderen Spalten sind die  $k$ -ten Potenzen ( $k = 0, 2, 3, \dots, n - 1$ ) dieser Werte. Solche Matrizen heißen **Vandermondsche Matrizen**.



Die Vandermondsche Matrix ist regulär, da die Zeilen jeweils genau die Basis des  $n$ -dimensionalen Vektorraums der Polynome darstellen. Es folgt somit, dass die Spalten der Vandermondschen Matrix linear unabhängig sind. Des Weiteren lässt sich zeigen, dass die Determinante der Koeffizientenmatrix ungleich Null ist und somit invertierbar. Praktisch bedeutet die Invertierbarkeit der Matrix, dass das lineare Gleichungssystem eine Lösung besitzt. Diese könnten wir nun zum Beispiel mithilfe der Gauß-Elimination berechnen.

Die Vandermondsche Matrix besitzt jedoch ganz schlechte numerische Eigenschaften, die mit wachsendem  $n$  rapide zunehmen. Bei der Berechnung mit dem Computer können so durch fortgesetztes Runden gravierende numerische Fehler entstehen. Aus diesem Grund ist von dieser Vorgehensweise zur Berechnung des Interpolationspolynoms abzuraten.

Zuletzt noch eine Definition: Die Menge aller Polynome vom Grad kleiner oder gleich  $n - 1$  bezeichnen wir mit  $P_{n-1}$ .

Es gilt der folgende Satz:

Falls es ein Interpolationspolynom  $p \in P_{n-1}$  gibt, dann ist es eindeutig bestimmt.

Beweis: Das ein solches Interpolationspolynom  $p \in P_{n-1}$  existiert, ist dadurch gegeben, da die Koeffizientenmatrix invertierbar ist. Das dieses Polynom auch eindeutig bestimmt ist, geht daraus hervor, dass die Matrix regulär ist, d. h. die Spalten sind linear unabhängig und die Zeilen bilden somit eine Basis des  $n$ -dimensionalen Vektorraums. Folglich ist auch das Interpolationspolynom  $p \in P_{n-1}$  eindeutig bestimmt, denn das lineare Gleichungssystem besitzt eine eindeutige Lösung.

Im Folgenden werden nun zwei Möglichkeiten vorgestellt, wie das Interpolationspolynom  $p$  rechnerisch bestimmt werden kann.

## 4.1 Lagrange-Interpolation

Bei der Lagrange-Interpolation beginnen wir mit speziellen Polynomen  $L_i \in P_{n-1}$ ,  $i = 1, \dots, n$ , die folgendermaßen definiert sind:

$$L_i(x_j) = \delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Die  $L_i$  haben an der Stützstelle  $x_i$  den Wert 1 und an allen anderen Stützstellen  $x_j, j \neq i$  den Wert 0. Die Polynome  $L_i(x)$  heißen **Lagrange Stützpolynome** und sind durch folgende Formel definiert:

$$L_i(x) := \frac{(x-x_1)\times\dots\times(x-x_{i-1})(x-x_{i+1})\times\dots\times(x-x_n)}{(x_i-x_1)\times\dots\times(x_i-x_{i-1})(x_i-x_{i+1})\times\dots\times(x_i-x_n)} = \frac{\prod_{j=1, j \neq i}^n (x-x_j)}{\prod_{j=1, j \neq i}^n (x_i-x_j)}$$

Jedes  $L_i$  ist offensichtlich ein Polynom des Grades  $n - 1$ .

Die Lösung des Interpolationsproblems lässt sich nun einfach mittels der  $L_i$  ausdrücken:

$$p(x) = \sum_{i=1}^n y_i L_i(x).$$

Die erhaltene Funktion  $p(x)$  heißt **Lagrangesches Interpolationspolynom**.

#### 4.1.1 Beispiel: Lagrange-Interpolation für $n = 3$

Für  $n = 3$  gegebene Stützstellen mit  $x_1 = 1, x_2 = 3, x_3 = 4$  berechnen wir die Lagrangeschen Stützpolynome wie folgt:

$$L_1(x) := \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{1}{6}(x^2 - 7x + 12)$$

$$L_2(x) := \frac{(x-1)(x-4)}{(3-1)(3-4)} = -\frac{1}{2}(x^2 - 5x + 4)$$

$$L_3(x) := \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{1}{3}(x^2 - 4x + 3)$$

In der folgenden Abbildung sind die Stützpolynome graphisch dargestellt:

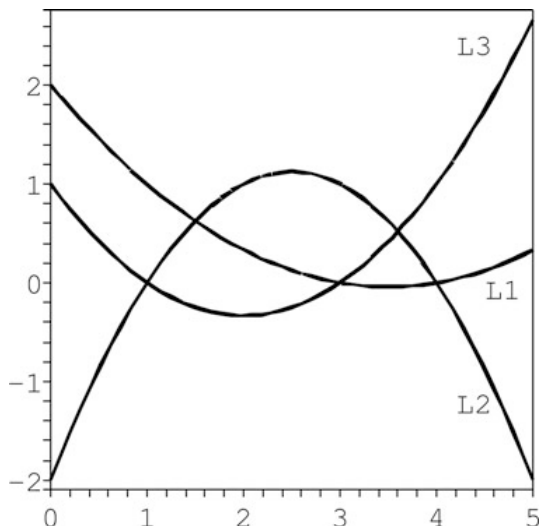


Abbildung 6: Beispiele für Lagrangesche Stützpolynome

Wir suchen jetzt  $p \in P_2$  mit den drei gegebenen Punkten  $(1|3), (3|1), (4|6)$ .

Die Lösung lautet dann:

$$p_2(x) = 3 \times L_1(x) + 1 \times L_2(x) + 6 \times L_3(x) = 2x^2 - 9x + 10.$$

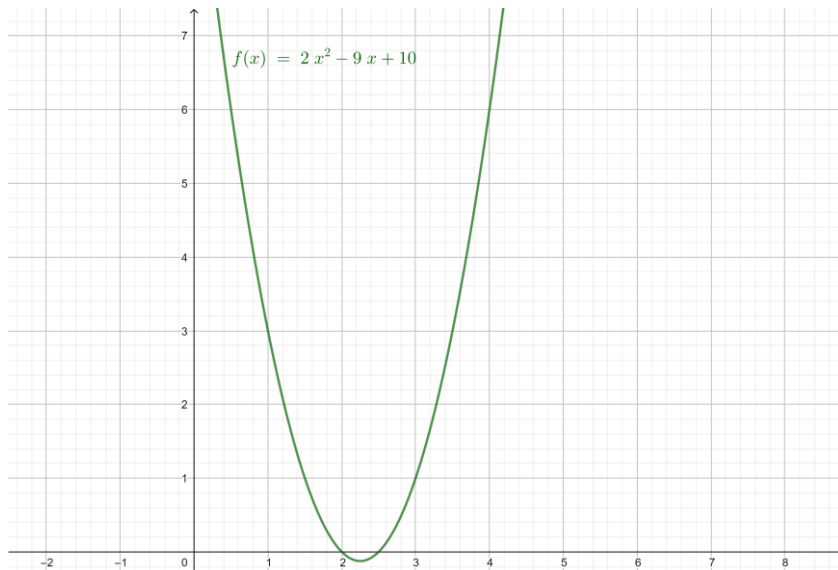


Abbildung 7: Interpolationspolynom (eigene Darstellung)

#### 4.1.2 Beispiel: Lagrange-Interpolation 100m-Sprinter

Bei unserem Anfangsbeispiel aus Kapitel 2 ist die Berechnung der Lagrangeschen Stützpolynome etwas aufwändiger als im vorherigen Beispiel. Da wir nun sieben Datenpunkte gegeben haben, können wir das Lagrange Interpolationspolynom auch mithilfe eines Computerprogramms, z.B. mit Sage (siehe Abbildung 8), berechnen.

```
R = PolynomialRing(RR, 'x')
nodes=([0,0],[1.5,12],[2.5,21.6],[5,27],[9.5,28.4],[11,27.81],[13.5,26.667])
f = R.lagrange_polynomial(nodes)
print f
```

Abbildung 8: Berechnung des Lagrangeschen Interpolationspolynoms mit Sage

Wir erhalten somit das folgende Interpolationspolynom:

$$p(x) = 0,0008467450571x^6 - 0,03456059216x^5 + 0,5247994903x^4 - 3,568914474x^3 + 9,585664294x^2 + 0,048895872x$$

Dieses Polynom ist eindeutig.

#### 4.2 Newton-Interpolation

Die zweite Methode zur Berechnung des Interpolationspolynoms ist die Newton-Interpolation.

Das **Newtonsche Interpolationspolynom** durch  $n$  Stützpunkte  $(x_i, y_i)$ ,  $i = 1, \dots, n$  ist gegeben durch:

$$p(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \dots + a_{n-1}(x - x_1)(x - x_2) \dots (x - x_{n-1}).$$

Die  $n$  Koeffizienten  $a_i$  ergeben sich aus der Interpolationsformel  $p(x_i) = y_i$  mittels der gestaffelten Gleichungen:

$$p(x_1) = a_0 = y_1$$

$$p(x_2) = a_0 + a_1(x_2 - x_1) = y_2 \leftrightarrow a_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

$$p(x_3) = a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) = y_3 \leftrightarrow a_2 = \frac{\frac{y_3 - y_2}{x_3 - x_2} \cdot \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

...

Für  $a_2$  können wir auch schreiben:  $a_2 = \frac{\frac{y_3 - y_1}{x_3 - x_1} \cdot \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_2}$ . Dies lässt sich entweder durch Nachrechnen zeigen oder durch das inhaltliche Argument, dass die Reihenfolge der Daten bei dem Interpolationspolynom keine Rolle spielt.

Bei den obigen Gleichungen sagt man: Das Polynom  $p(x_1) := y_1$  interpoliert somit offenbar bei  $x_1$ . Das Polynom  $p(x_2) = a_0 + a_1(x - x_1) = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$  interpoliert bei  $x_1$  und  $x_2$  etc.

Wir nennen nun

$$p_k(x) = a_0 + a_1(x - x_1) + \dots + a_{k-1}(x - x_1)(x - x_2) \dots (x - x_{k-1})$$

das  $k$ -te **Abschnittspolynom**.

Dabei gelten die drei Eigenschaften:

- Das Abschnittspolynom  $p_k$  interpoliert an den Stellen  $x_1, \dots, x_k$ .
- $p_{k+1}(x) = p_k(x) + a_k(x - x_1)(x - x_2) \dots (x - x_k)$
- $a_{k-1}$  ist im Polynom  $p_k$  der Koeffizient vor der höchsten Potenz  $x^k$  (der sogenannte Leitkoeffizient).

Bei der Berechnung tauchen somit sukzessive Polynome auf, die mit steigendem Grad immer mehr der gegebenen Daten interpolieren.

Kommen wir nun zur effektiven Berechnung. Dafür führen wir zunächst folgende Bezeichnungen ein:

$$\begin{aligned}
[x_i, x_{i+1}; y] &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \\
[x_i, x_{i+1}, x_{i+2}; y] &= \frac{[x_{i+1}, x_{i+2}; y] - [x_i, x_{i+1}; y]}{x_{i+2} - x_i} \\
[x_i, x_{i+1}, x_{i+2}, x_{i+3}; y] &= \frac{[x_{i+1}, x_{i+2}, x_{i+3}; y] - [x_i, x_{i+1}, x_{i+2}; y]}{x_{i+3} - x_i} \\
&\dots \\
[x_i, x_{i+1}, \dots, x_{i+k}; y] &= \frac{[x_{i+1}, \dots, x_{i+k}; y] - [x_i, x_{i+1}, x_{i+k-1}; y]}{x_{i+k} - x_i}
\end{aligned}$$

Wir nennen die Größe  $[x_i, \dots, x_{i+k}; y]$  **Steigung der Ordnung**  $k$  mit den Stützstellen  $x_i, \dots, x_{i+k}$ .

Für die Koeffizienten erhalten wir somit:

$$\begin{aligned}
a_0 &= y_1 \\
a_1 &= [x_1, x_2; y] := \frac{y_2 - y_1}{x_2 - x_1} \\
a_2 &= [x_1, x_2, x_3; y] := \frac{[x_2, x_3; y] - [x_1, x_2; y]}{x_3 - x_1} \\
&\dots \\
a_{n-1} &= [x_1, x_2, \dots, x_n; y] := \frac{[x_2, \dots, x_n; y] - [x_1, \dots, x_{n-1}; y]}{x_n - x_1}
\end{aligned}$$

Die  $a_i$  entstehen also fortlaufend durch die Bildung von Quotienten aus Differenzen von vorher berechneten Ausdrücken. Sie lassen sich mit dem folgenden Schema berechnen:

			Steigungen		
	$x_i$	$y_i$	1. Ordnung	2. Ordnung	3. Ordnung
	$x_1$	$y_1$			
	$x_2 - x_1$		$[x_1, x_2; y]$		
	$x_3 - x_1$	$y_2$		$[x_1, x_2, x_3; y]$	
$x_4 - x_1$	$x_3 - x_2$	$y_3$	$[x_2, x_3; y]$		$[x_1, x_2, x_3, x_4; y]$
	$x_4 - x_2$			$[x_2, x_3, x_4; y]$	
$x_5 - x_2$	$x_4 - x_3$	$y_4$	$[x_3, x_4; y]$		$[x_2, x_3, x_4, x_5; y]$
	$x_5 - x_3$			$[x_3, x_4, x_5; y]$	
$x_6 - x_3$	$x_5 - x_4$	$y_5$	$[x_4, x_5; y]$		$[x_3, x_4, x_5, x_6; y]$
	$x_6 - x_4$			$[x_4, x_5, x_6; y]$	
	$x_6 - x_5$	$y_6$	$[x_5, x_6; y]$		
	$x_6$	$y_6$			

Abbildung 9: Rechenschema Newton-Tableau

Beim Newton-Tableau beginnt man zunächst mit den beiden mittleren Spalten für  $x_i$  und  $y_i$ , also den gegebenen Wertepaaren. Anschließend gewinnt man nach links fortschreitend die  $k$ -

te Spalte aus den Differenzen  $x_{i+k} - x_i, i = 1, 2, \dots$ . Nach rechts fortschreitend gewinnt man eine neue Spalte durch Quotientenbildung. Im Zähler steht immer die Differenz der benachbarten Werte der vorherigen Spalte und im Nenner findet sich immer die Differenz aus den  $x$ -Werten der entsprechenden linken Hälfte des Rechenschemas.

Die Koeffizienten des Newton-Polynoms lassen sich jetzt direkt aus dem Newton-Tableau ablesen (siehe umrandete Werte).

#### 4.2.1 Beispiel: Newton-Interpolation für $n = 3$

Wir betrachten die Punkte:  $(1|3), (3|1), (4|6)$ .

Das zugehörige Newton-Tableau lautet dann:

		1	3		
	2			-1	
3	3	1			2
	1			5	
		4	6		

Für das Newton-Polynom ergibt sich dann folgende Funktion (vgl. Kap. 4.1.1):

$$p(x) = 3 - 1(x - 1) + 2(x - 1)(x - 3) = x^2 - 9x + 10$$

#### 4.2.2 Beispiel: Newton-Interpolation 100m-Sprinter

Das Interpolationspolynom hat die folgende Form:

$$p(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \dots + a_6(x - x_1) \dots (x - x_6)$$

Dabei soll gelten:  $p(x_i) = v_i$  für  $i = 1, \dots, 7$ .

Dies führt zu folgendem gestaffelten linearen Gleichungssystem für die Koeffizienten:

$$p(0) = 0 = a_0$$

$$p(1,5) = 12 = a_0 + a_1(1,5 - 0) = a_0 + 1,5a_1$$

$$p(2,5) = 21,6 = a_0 + a_1(2,5 - 0) + a_2(2,5 - 0)(2,5 - 1,5) = a_0 + 2,5a_1 + 2,5a_2$$

$$p(5) = 27 = a_0 + 5a_1 + 5 \times 3,5a_2 + 5 \times 3,5 \times 2,5a_3$$

$$p(9,5) = 28,4 = a_0 + 9,5a_1 + 9,5 \times 8a_2 + 9,5 \times 8 \times 7a_3 + 9,5 \times 8 \times 7 \times 4,5a_4$$

$$p(11) = 27,81 = a_0 + 11a_1 + 11 \times 9,5a_2 + 11 \times 9,5 \times 8,5a_3 + 11 \times 9,5 \times 8,5 \times 6a_4 + 11 \times 9,5 \times 8,5 \times 6 \times 1,5a_5$$

$$p(13,5) = 26,667 = a_0 + 13,5a_1 + 13,5 \times 12a_2 + 13,5 \times 12 \times 8,5a_3 + 13,5 \times 12 \times 8,5 \times 4a_4 + 13,5 \times 12 \times 8,5 \times 4 \times 2,5a_5 + 13,5 \times 12 \times 11 \times 8,5 \times 4 \times 2,5 \times a_6$$

Durch wiederholtes Einsetzen gelangen wir dann zu den Lösungen:  $a_0 = 0, a_1 = 8, a_2 = 0,64, a_3 = -0,553143, a_4 = 0,082720, a_5 = 0,018896, a_6 = 0,000847$ . Und dies führt uns wiederum zum Interpolationspolynom:

$$p(x) = 0,0008467450571x^6 - 0,03456059216x^5 + 0,5247994903x^4 - 3,568914474x^3 + 9,585664294x^2 + 0,048895872x$$

Das Interpolationspolynom entspricht exakt dem Polynom, welches wir mithilfe der Lagrange-Interpolation erhalten haben.

### 4.3 Vergleich der zwei Methoden

Bei der Methode von Lagrange haben wir den Vorteil, dass die Koeffizientenfunktionen  $L_i(x)$  nur von den Stützstellen abhängen. Diese können explizit angegeben werden und die lineare Abhängigkeit von den Stützwerten  $y_i$  ist besonders deutlich sichtbar.

Als Nachteil haben wir hier jedoch einen erheblichen Rechenaufwand. Wenn Daten hinzukommen, müssen die Stützpolynome immer wieder erneut berechnet werden. Dies ist bei großen Datenmengen sehr mühsam und aufwändig.

Bei der Methode von Newton können wir leicht weitere Interpolationsstellen hinzunehmen, denn hier wird mit einem weiteren Datenpunkt einfach ein zusätzlicher Koeffizient der Interpolationsfunktion berechnet, welche dann um einen Grad ansteigt. Des Weiteren haben wir hier mit dem Newton-Tableau ein einfach zu handhabendes Rechenschema vorliegen.

## 5 Beispiel Sprinter

In diesem Kapitel wollen wir die zwei Anfangsfragen nach der maximalen Geschwindigkeit und der maximalen Beschleunigung des 100m-Sprinters (vgl. Kapitel 2) beantworten. Dafür betrachten wir das Interpolationspolynom:

$$p(x) = 0,0008467450571x^6 - 0,03456059216x^5 + 0,5247994903x^4 - 3,568914474x^3 + 9,585664294x^2 + 0,048895872x$$

In der folgenden Abbildung ist diese Funktion graphisch dargestellt:

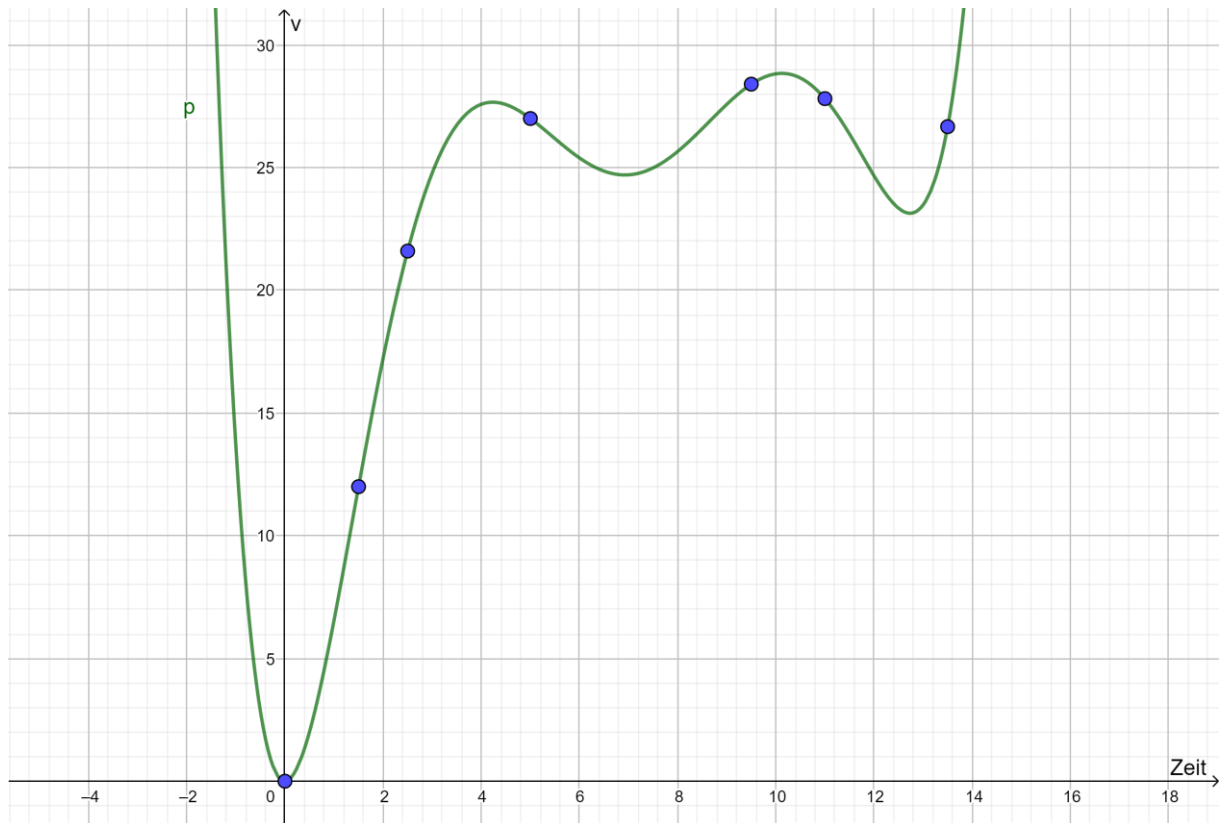


Abbildung 10: Geschwindigkeit des Sprinters interpoliert durch ein Polynom 6-ten Grades (eigene Darstellung)

Wir können im Vergleich zur linearen, stückweisen Interpolation hier von einer globalen Interpolation sprechen. Aufgrund der globalen Interpolation zwischen dem Anfangs- und Endwert oszilliert die Funktion mehr als die Funktion der linearen Interpolation.

Zunächst wollen wir festhalten, dass die Funktion nur für den Bereich von 0 – 13,5 Sekunden betrachtet werden darf. Außerdem ist zu beachten, dass nur an den Stützstellen die exakten Werte der Geschwindigkeit des Läufers wiedergegeben werden. Außerhalb dieser Stützstellen kann der Wert von der wahren (nicht bekannten) Geschwindigkeitskurve abweichen. Eine Interpretation der Daten ist somit immer mit Vorsicht zu genießen.

Doch wo hat der Läufer nun seine maximale Geschwindigkeit und seine maximale Beschleunigung? Um dies zu beantworten müssen wir uns die Nullstellen der ersten und zweiten Ableitung anschauen.

Die erste Ableitung lautet:

$$p'(x) = 0,005080470336x^5 - 0,1728029607x^4 + 2,099197961x^3 - 10,70674341x^2 + 19,17132857x + 0,0488958741$$



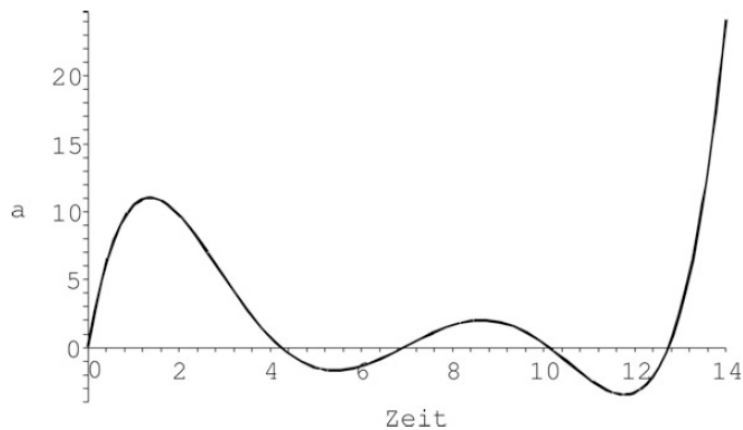


Abbildung 11: 1. Ableitung der Polynominterpolation der Geschwindigkeit des Sprinters: Beschleunigung versus Zeit

Die Nullstellen der ersten Ableitung liegen ungefähr bei  $x_1 = 0$ ,  $x_2 = 4,23$ ,  $x_3 = 6,92$ ,  $x_4 = 10,12$ ,  $x_5 = 12,73$ .

Die zweite Ableitung lautet:

$$p''(x) = 0,02540235167x^4 - 0,6912118424x^3 + 6,297593877x^2 - 21,41348681x + 19,17132857$$

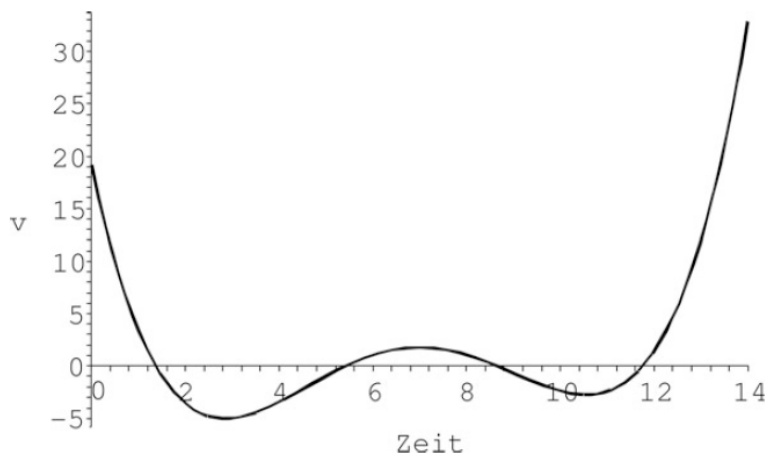


Abbildung 12: 2. Ableitung der Polynominterpolation der Geschwindigkeit des Sprinters: Veränderung der Beschleunigung

Die Nullstellen der zweiten Ableitung liegen ungefähr bei  $x_1 = 1,366$ ,  $x_2 = 5,431$ ,  $x_3 = 8,648$ ,  $x_4 = 11,766$ .

Anhand dieser Graphen und Werte lässt sich der Geschwindigkeitsverlauf des 100m-Sprinters gut beschreiben. Die anfangs hohe Beschleunigung des Sprinters lässt schnell nach. Bei 4,23 Sekunden und 10,12 Sekunden haben wir lokale Maxima. Jeweils nach diesen beiden Zeitpunkten verliert der Läufer an Geschwindigkeit. Nach 6,92 Sekunden setzt er zu einem

Zwischenspur an und nach 12,73 Sekunden dann zum Endspurt. Seine maximale Geschwindigkeit erreicht er nach 10,12 Sekunden. Bei der Beschleunigung ist zu erkennen, dass diese von 1,366 – 5,431 Sekunden abnimmt und anschließend bis 8,646 Sekunden wieder zunimmt. Die maximale Beschleunigung erreicht der Sprinter nach 1,366 Sekunden.

Insgesamt könnte man dem Läufer den Tipp geben, seine Ausdauer zu trainieren, damit er die Geschwindigkeit vom Zwischenspur halten kann und vor dem Ziel nicht mehr an Geschwindigkeit abbaut.

## Literaturverzeichnis

Engel, J. (2018). *Anwendungsorientierte Mathematik: Von Daten zur Funktion. Eine Einführung in die mathematische Modellbildung für Lehramtsstudierende*. 2., vollständig überarbeitete Aufl. Berlin: Springer.

<https://www.mathi.uni-heidelberg.de/~thaeter/anasem08/Interpolation.pdf> abgerufen am 14.05.2019