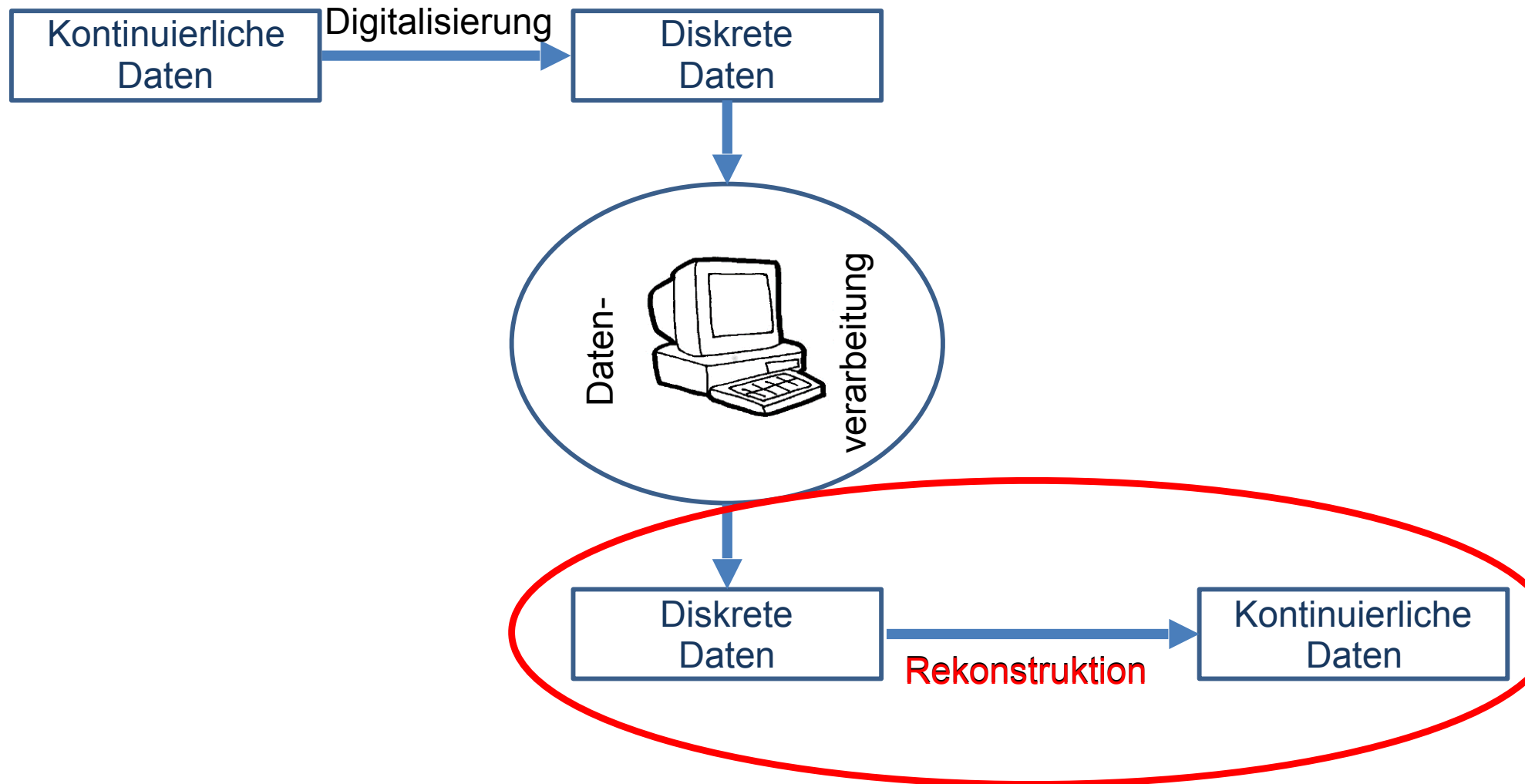


# Algorithmik kontinuierlicher Systeme

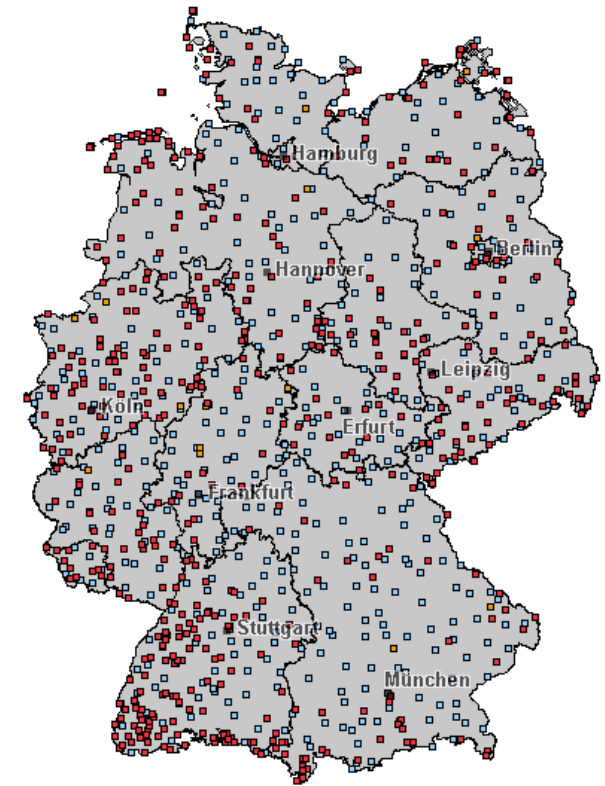
## *Rekonstruktion kontinuierlicher Daten – Interpolation 1D*



- Rekonstruktion kontinuierlicher Daten aus diskreten Daten



- Digital-Analog-Wandlung : D/A Wandler
- Äquidistante Abtastung: Abtast-Theorem
  - Falls Voraussetzungen erfüllt, Rekonstruktion mit sinc-Funktion
  - Die Voraussetzungen sind in konkreten Anwendungen meist nicht gegeben
  - Beispiel Wetterkarte



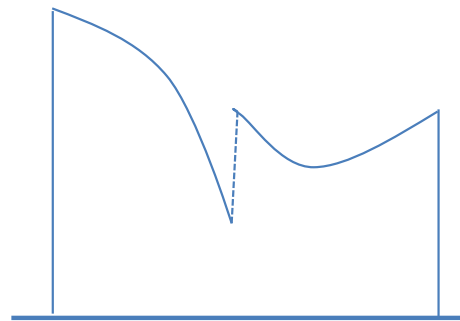
Wetterstationen  
Deutschland

- Alternativen: Interpolation und Approximation

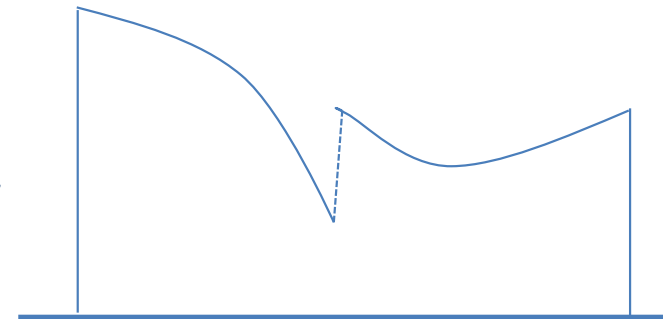
- Rekonstruktion „kontinuierlicher Daten“ aus diskreten
  - Gegeben „Punkte“
  - Gesucht stetige (glatte) „Kurve“ (oder Fläche, oder Körper)
  
- Beispiele:
  - CAD-Daten aus Abtastwerten eines Modells errechnen
  - Vermessung: Geo-Informationssysteme (Digitale Höhenmodelle, Erstellung digitaler Landkarten)
  - Rückwandlung digitalisierter Signale in analoge Signale (Bilder, Audio, Video, Umrechnung von Bildauflösungen, Abtastraten, etc.),
  - „In-Betweening“ bei Animationen – Key frame Animation
  - „Zoomen“ digitaler Daten

- Zoomen eines digitalen Bildes bei fester Auflösung
- anschaulich: Grauwertbild, nur eine Bildzeile

kontinuierlich



original:  $f(x)$

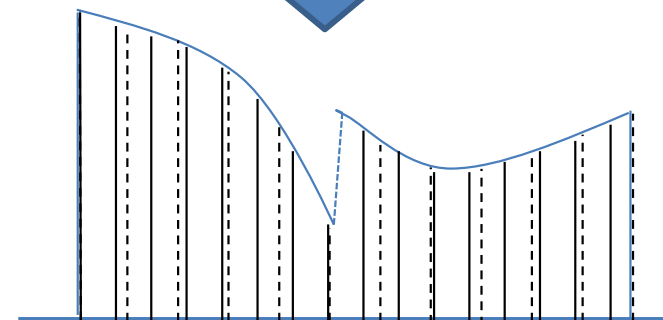
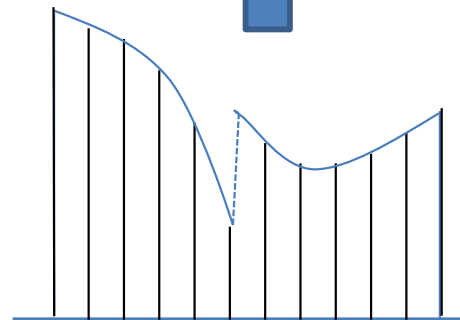


Zoomfaktor: 1.7

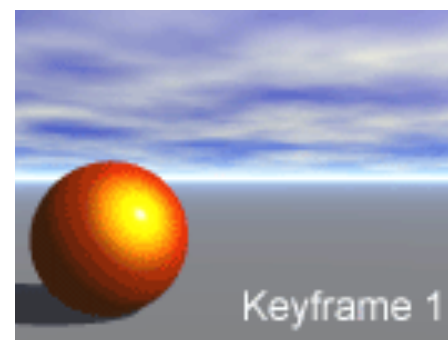
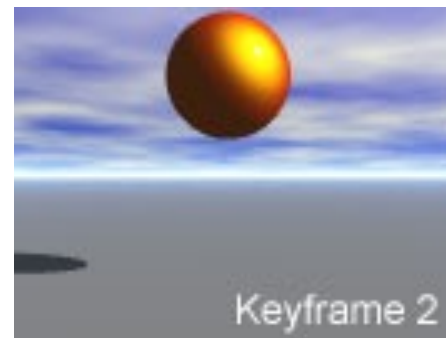
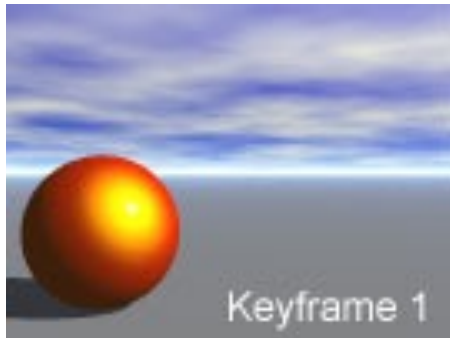
$$g(x) = f(x/1.7)$$



diskret (10dpi)

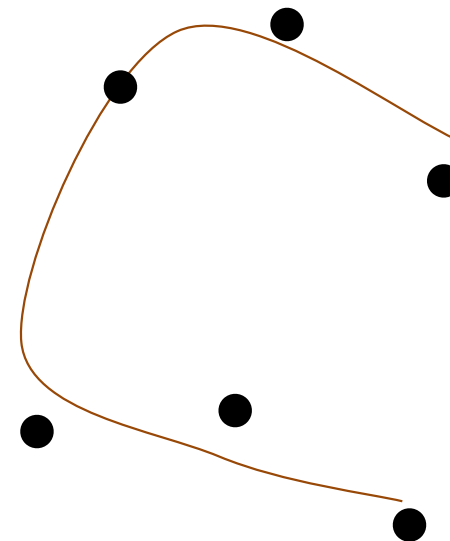
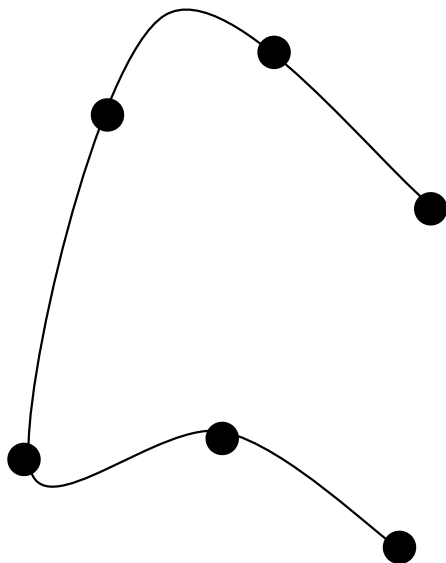


1. rekonstruieren
2. resampeln



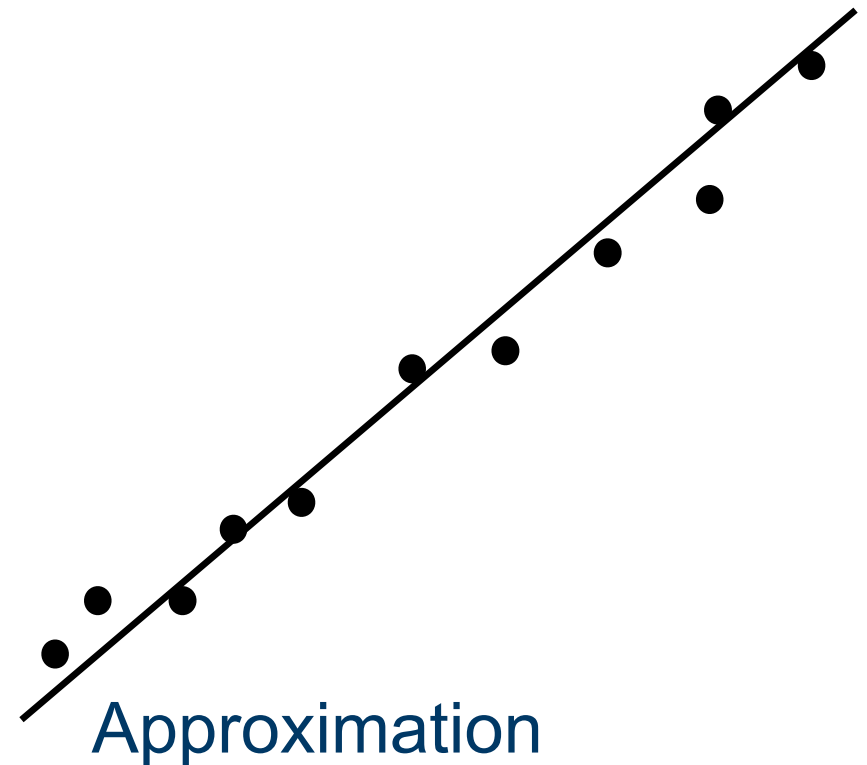
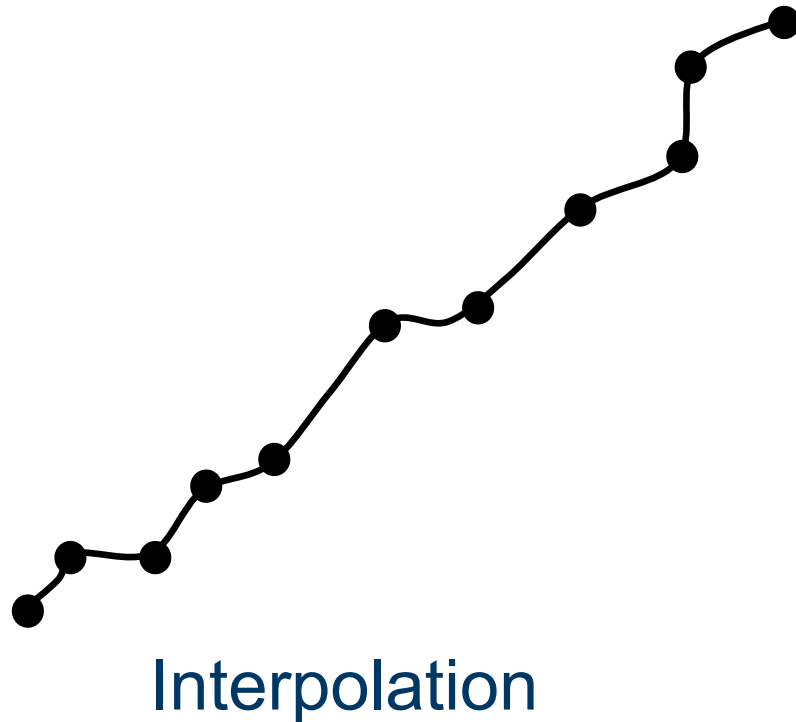
## • Interpolation vs. Approximation

- Soll die rekonstruierte Funktion (Kurve, Fläche, ...) exakt durch die Punkte verlaufen, so sprechen wir von **Interpolation**. Das behandeln wir im Folgenden.
- Soll die Kurve (Fläche) nur näherungsweise durch die Punkte verlaufen, so sprechen wir von **Approximation**. Das ist verwandt mit den Least-Squares-Problemen, (Ausgleichsgerade, etc. s.o.)





- Interpolation ist nicht in jedem Fall das genauere, bessere Verfahren
  - z.B. bei Messfehlern

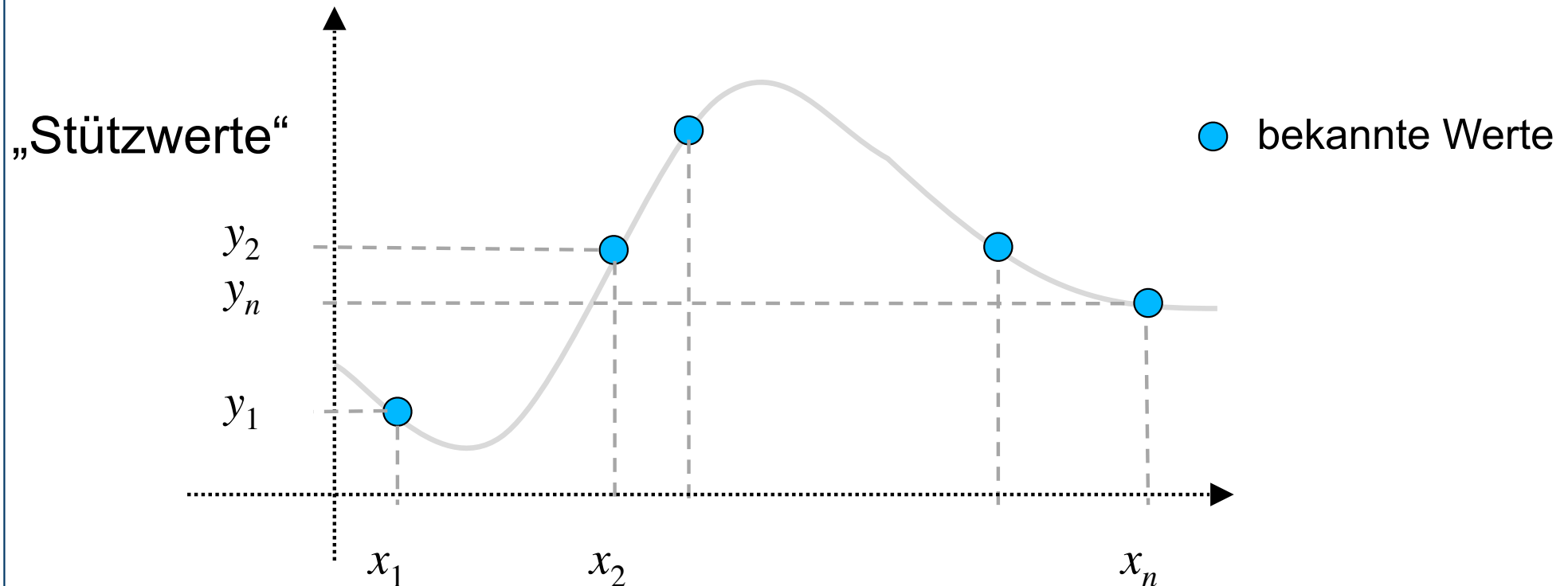




- Verwandte Aufgabenstellungen:
  - Interpolation/Approximation komplizierter Funktionen durch einfachere (z.B. Approximation der exp-Funktion durch Polynome)
    - ★ „Komplizierte“ (transzendente) Funktionen werden intern mit Interpolation approximiert.
- Kontinuierliche Daten sind eigentlich Relationen, d.h. allgemeiner als „Funktionen“
  - Rekonstruktion von „Kurven“ anstelle von Funktionen
  - Aus Gründen der Einfachheit werden wir meistens trotzdem über Funktionen sprechen.

- Von einer unbekannten Funktion  $f : \mathbb{R} \rightarrow \mathbb{R}$  kennt man nur die Werte an endlich vielen Punkten:

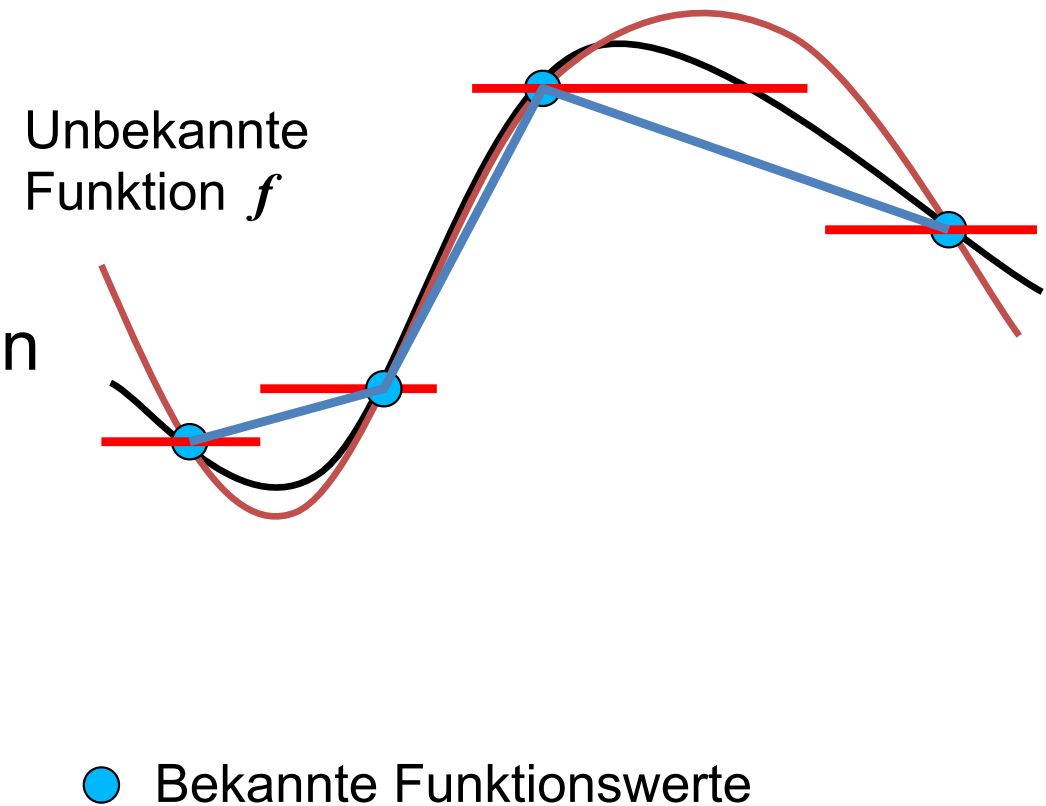
$$y_i = f(x_i) \quad \text{für } i = 1, 2, \dots, n$$

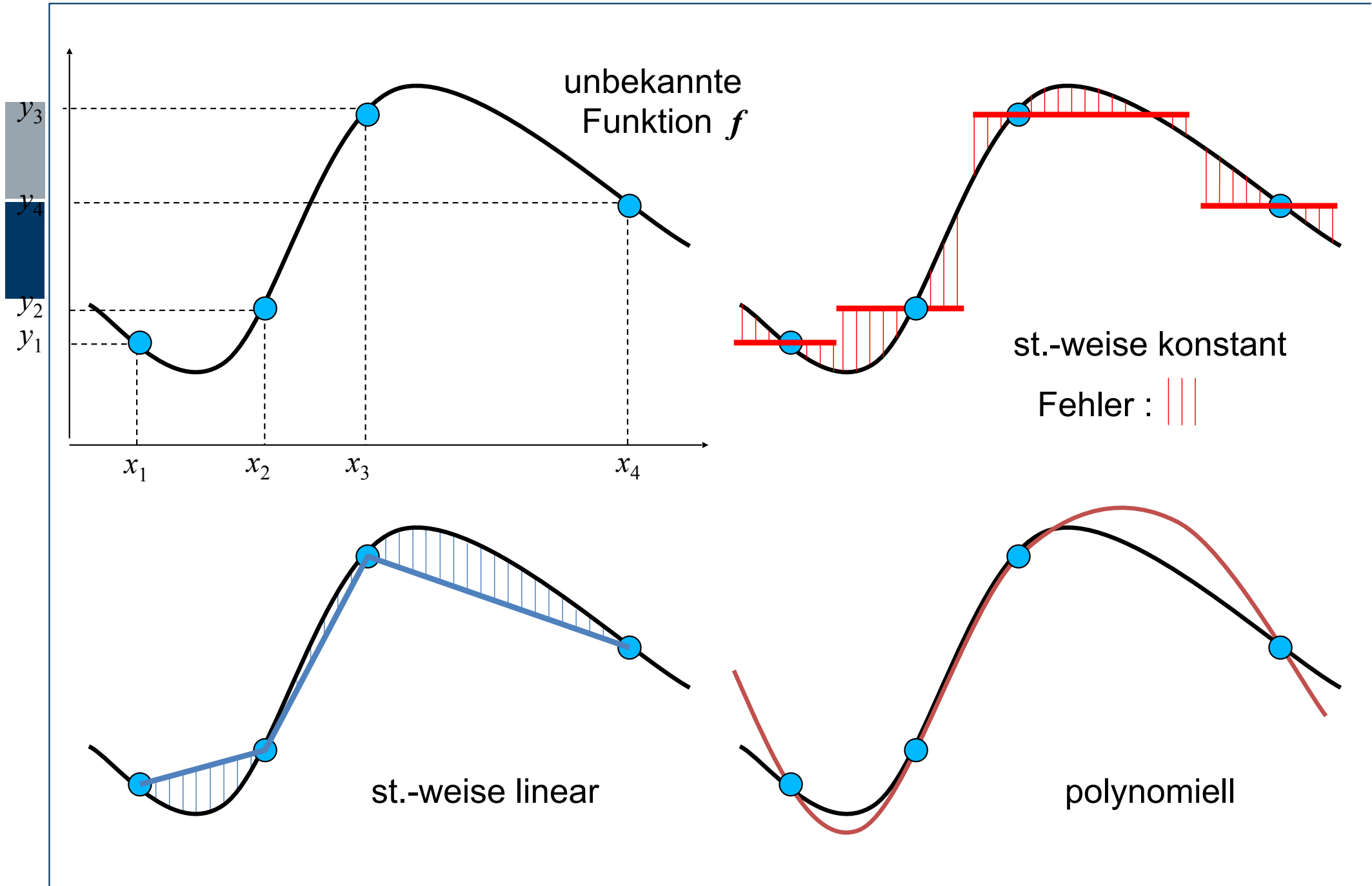


- Annahme:  $x_1 < x_2 < \dots < x_{n-1} < x_n$  „Stützstellen“

- Aus der Anwendung kennen wir gewisse Grundeigenschaften von  $f$ , z.B.
  - $f$  ist eine Gerade, oder ein Polynom
  - $f$  ist stetig (oder glatt, d.h. differenzierbar).
- Wir wählen eine Klasse (Menge)  $K$  von Funktionen, in der wir  $f$  aus den endlich vielen Punkten näherungsweise rekonstruieren.
- Kriterien:
  - $f$  muss in der Funktionsklasse  $K$  gut approximierbar sein,
  - die Funktionsklasse  $K$  muss auf dem Rechner gut (d.h. effizient) darstellbar und manipulierbar sein.  
Gut darstellbar sind z.B, Polynome
  - Polynome werden oft „gestückelt“, d.h. stückweise definiert verwendet

- Unbekannte Funktion
- Rekonstruktion durch stückweise konstante Funktion
- Rekonstruktion durch stückweise lineare Funktion
- Rekonstruktion durch (kubisches) Polynom





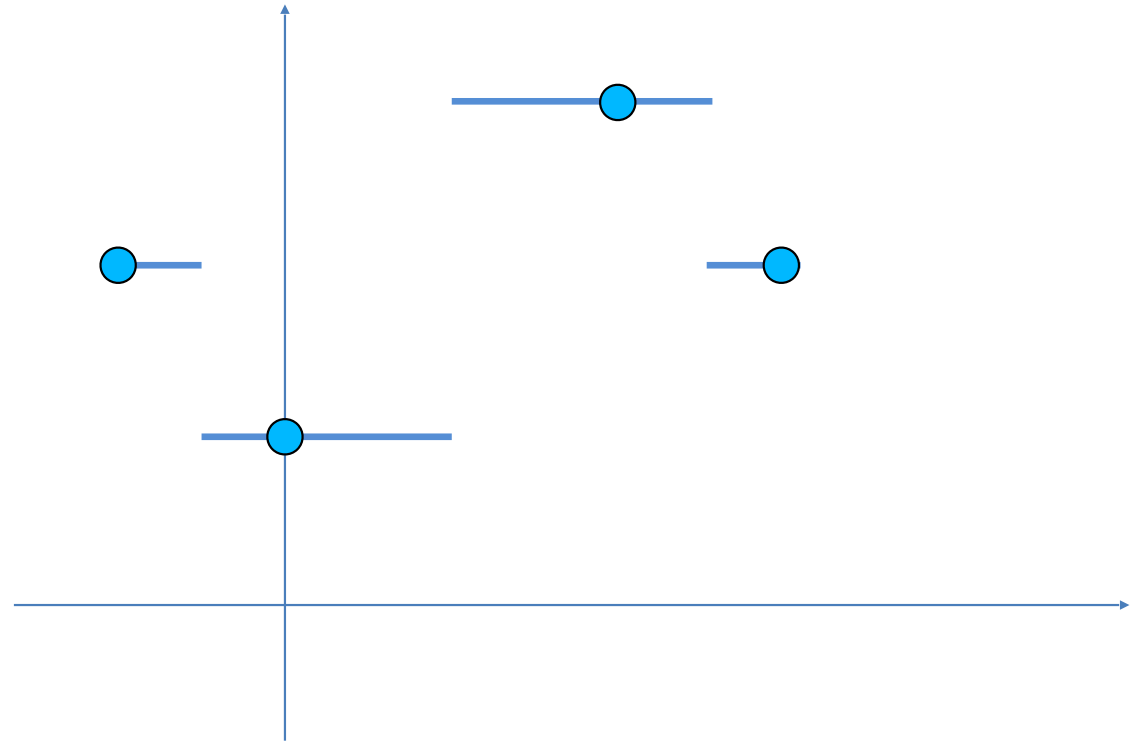
- Unterscheidung zwischen **lokalen** und **globalen** Verfahren
  - Lokal: interpolierter Wert  $p(x)$  hängt nur von den „benachbarten“ Werten ab. Z.B.:  $x_i < x < x_{i+1}$  : nur  $y_i$  und  $y_{i+1}$  gehen ein.
  - Global: alle Werte  $y_i$  gehen ein
- Lokale Verfahren
  - Nearest neighbor (stückweise konstant)
  - Linear (stückweise linear)
  - Catmull-Rom (stückweise kubisches Polynom)
- Globale Verfahren
  - Polynom-Interpolation
  - B-Spline-Interpolation

- Gegeben
  - **Stützstellen**  $\{ x_1, x_2, \dots, x_n \}$  und **Stützwerte**  $\{ y_1, y_2, \dots, y_n \}$  (Skalare oder Vektoren) (Abtastwerte einer unbekannten, zu rekonstruierenden Funktion  $f$  )
  - Gesucht: kontinuierliche Rekonstruktion  $p(x)$  (Näherung von  $f(x)$  )
- Stückweise konstante Interpolation:
  - Um den Wert an der Stelle  $x$  anzunähern, suche den *nearest neighbor* d.h. die nächst gelegene Stützstelle  $x_i$
  - Interpolierter Wert ist:  $p(x) = y_i$
- Bezeichnung: **nearest neighbor interpolation**



- Beispiel:

$x_i$	-1	0	2	3
$y_i$	2	1	3	2



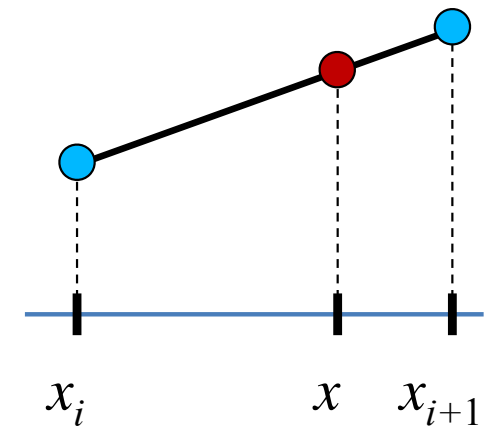
- Allgemeine Formel:

$$p(x) = \begin{cases} y_1 & \text{falls } x_1 \leq x \leq \frac{1}{2}(x_1 + x_2) \\ y_2 & \text{falls } \frac{1}{2}(x_1 + x_2) < x \leq \frac{1}{2}(x_2 + x_3) \\ \vdots & \vdots \\ y_n & \text{falls } \frac{1}{2}(x_{n-1} + x_n) < x \leq x_n \end{cases}$$

- Kein (Rechen-)Aufwand, nur Suchen der Nachbarn
  - äquidistant:  $x_i = a + i h$  ( $h$  Schrittweite)
  - einfache Suche  $i = \text{int}((x-a)/h)$   $O(1)$
  - Komplexität für die Suche
  - sind die Stützstellen schon geordnet:  
Aufwand für Suche  $O(\log(n))$
  - wenn nicht, dann  $O(n * \log(n))$
- in vielen Fällen zu ungenau, z.B.  
bei bekanntermaßen glattem  $f$
- Fehler bei glattem (d.h. differenzierbarem)  $f$  :

$$|p(x) - f(x)| \leq \frac{h}{2} \cdot \max_{x_1 \leq \xi \leq x_n} \{|f'(\xi)|\} \quad \text{wobei} \quad h = \max_{1 \leq i \leq n} \{x_{i+1} - x_i\}$$

- Das am weitesten verbreitete Interpolationsverfahren.
  - Warum sind Graphikkarten so leistungsfähig?
  - Lineare Interpolation in Hardware + höchst parallel!
- Gegeben
  - **Stützstellen**  $\{x_1, x_2, \dots, x_n\}$  und **Stützwerte**  $\{y_1, y_2, \dots, y_n\}$   
(Abtastwerte einer unbekannten zu rekonstruierenden Funktion  $f$ )
  - Gesucht: kontinuierliche Rekonstruktion  $p(x)$   
(Näherung von  $f(x)$ )
- Stückweise lineare Interpolation:
  - Um den Wert an der Stelle  $x$  anzunähern, suche die nächst gelegene linke und rechte Stützstelle  $x_i \leq x \leq x_{i+1}$
  - Interpoliere im Intervall  $[x_i, x_{i+1}]$  linear



- Stückweise lineare Interpolation:

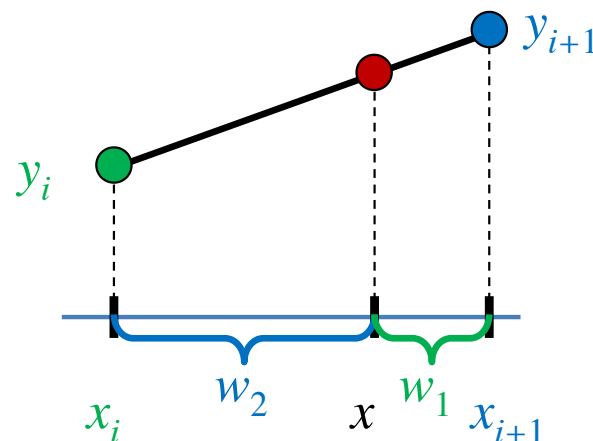
- Um den Wert an der Stelle  $x$  anzunähern, suche die nächst gelegenen Stützstellen  $x_i \leq x \leq x_{i+1}$

- Interpolierter Wert ist:

$$p(x) = m_i(x - x_i) + y_i \quad \text{mit} \quad m_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

- Alternativ: gewichtetes Mittel von  $y_i$  und  $y_{i+1}$  :

$$p(x) = w_1 y_i + w_2 y_{i+1} \quad \text{mit} \quad w_1 = \frac{x_{i+1} - x}{x_{i+1} - x_i} \quad \text{und} \quad w_2 = \frac{x - x_i}{x_{i+1} - x_i} = 1 - w_1$$



- Gegeben:  
Stützstellen  $\{x_1, x_2, \dots, x_n\}$  und -werte  $\{y_1, y_2, \dots, y_n\}$
- Idee:
  1. Schätze an jeder Stelle die erste Ableitung (Steigung):  
 $\rightarrow \{y_1', y_2', \dots, y_n'\}$
  2. Finde auf jedem Teilintervall  $[x_i, x_{i+1}]$  das (eindeutige) kubische Polynom  $p_i$  welches in den beiden Endpunkten die Stützwerte und die geschätzten Ableitungen interpoliert:

$$p(x) = a_0(x_{i+1} - x)^3 + a_1(x_{i+1} - x)^2(x - x_i) +$$

$$a_2(x_{i+1} - x)(x - x_i)^2 + a_3(x - x_i)^3$$

$$a_0 = \frac{y_i}{(x_{i+1} - x_i)^3}, \quad a_1 = 3a_0 + \frac{y_i'}{(x_{i+1} - x_i)^2},$$

$$a_3 = \frac{y_{i+1}}{(x_{i+1} - x_i)^3}, \quad a_2 = 3a_3 - \frac{y_{i+1}'}{(x_{i+1} - x_i)^2},$$

## From Wikipedia

- Ed Catmull ist ein amerikanischer Informatiker, der zu vielen wichtigen Entwicklungen in der Computergraphik beigetragen hat.
- Er ist vierfacher Oscar-Preisträger.
- Er studierte Physik und Computer Science an der University of Utah
- 1979 Arbeit für George Lucas bei Lucasfilm
- 1986 Steve Jobs übernimmt *Lucasfilm's digital division* und *Pixar*.
- Catmull ist Chef-Entwickler, u.a das rendering system RenderMan
- Animationsfilme Filme *Toy Story*, *Finding Nemo*, ...
- zur Zeit Präsident von *Pixar* und *Disney Animation Studios*



[https://de.wikipedia.org/wiki/Edwin\\_Catmull](https://de.wikipedia.org/wiki/Edwin_Catmull)

## Wie schätzt man die Ableitungen?

- Mittels Differenzenquotienten:

- Vorwärts-Differenz

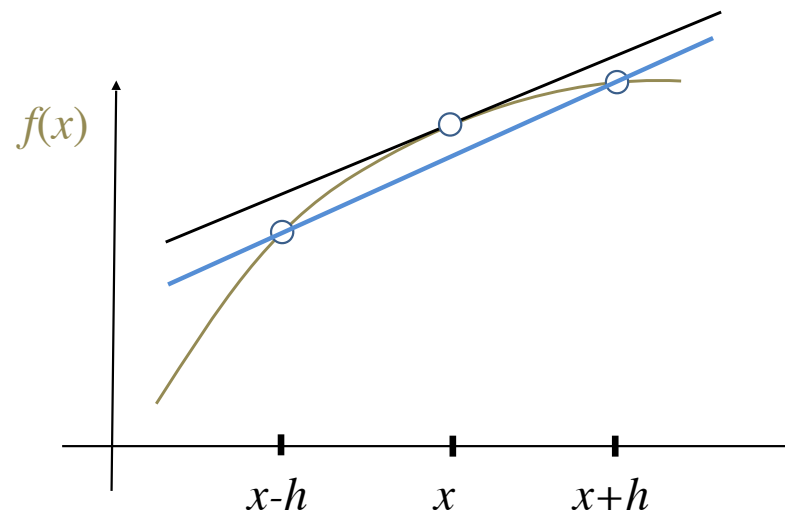
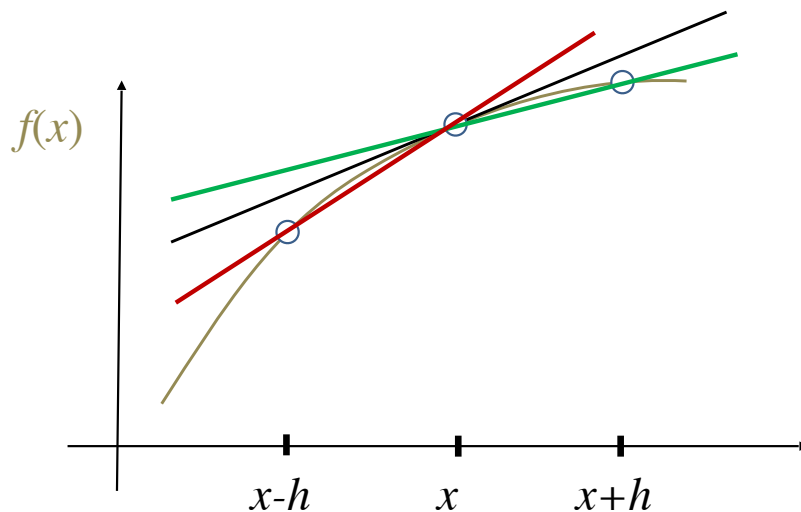
$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

- Rückwärts-Differenz

$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h)$$

- Zentrale Differenz

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$





Wie schätzt man die Ableitungen (allgemein)?

- Beispiel zu Differenzenquotienten.

$$f(x) = \exp(x), \quad x_0 = 0.5, \quad f'(x_0) = 1.648721271 \dots$$

- Tabelle der absoluten Fehler für Schrittweiten  $h=2^{-3}, 2^{-4}, \dots$

Schrittweite	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$	$2^{-10}$
Vorwärts-D.	.1075	.0526	.0260	$.129 \cdot 10^{-1}$	$.646 \cdot 10^{-2}$	$.322 \cdot 10^{-2}$	$.161 \cdot 10^{-2}$	$.804 \cdot 10^{-3}$
Rückwärts-D.	.0908	.0505	.0255	$.128 \cdot 10^{-1}$	$.642 \cdot 10^{-2}$	$.321 \cdot 10^{-2}$	$.161 \cdot 10^{-2}$	$.804 \cdot 10^{-3}$
Zentrale D.	.0043	.0011	.00026	$.671 \cdot 10^{-4}$	$.168 \cdot 10^{-4}$	$.411 \cdot 10^{-5}$	$.905 \cdot 10^{-6}$	$.137 \cdot 10^{-6}$

- Erkenntnis:
  - Vorwärts- und Rückwärts-Differenz halbiert sich der Fehler
  - Zentraler Differenz wird er Fehler jeweils geviertelt

## Schätzen der Ableitung bei Catmull-Rom (1. Schritt):

- ~~Vorwärtsdifferenz~~  $y_i'_{fw} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$
- ~~Rückwärtsdifferenz~~  $y_i'_{bw} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$
- Zentrale Differenz (einfach)  $y_i' = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}$
- Zentrale Differenz (exakt):  
(gewichtetes Mittel von Vorwärts- und Rückwärts-D.):

$$y_i' = \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}} \cdot y_i'_{fw} + \frac{x_{i+1} - x_i}{x_{i+1} - x_{i-1}} \cdot y_i'_{bw}$$

Im äquidistanten Fall sind die beiden zentralen D. identisch.

## Zusammenfassung

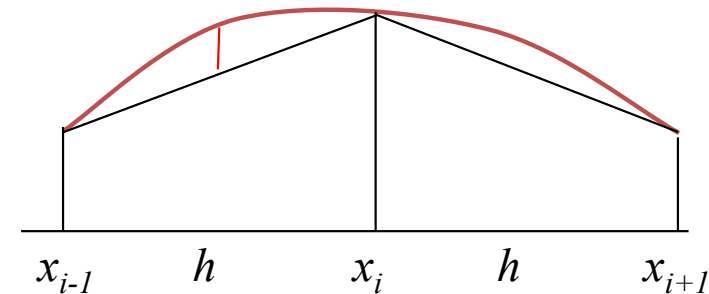
Gegeben: Stützstellen  $\{x_1, x_2, \dots, x_n\}$  und -werte  $\{y_1, y_2, \dots, y_n\}$

1. Schätze an inneren Stützstellen die Ableitung durch zentrale Differenz  $\rightarrow \{y_2', y_3', \dots, y_{n-1}'\}$
2. Schätze die Ableitung in den beiden Endpunkten  $y_1', y_n'$ 
  - ▶ zB Vorwärts- bzw. Rückwärts-Differenz
  - ▶ NB: es gibt bessere Verfahren
3. Finde auf jedem Teilintervall  $[x_i, x_{i+1}]$  das (eindeutige) kubische Polynom  $p_i$  welches in den beiden Endpunkten die Stützwerte und die geschätzten Ableitungen interpoliert

Formeln siehe oben!

- Annahme äquidistante Stützstellen Schrittweite  $h$  und zu rekonstruierte Funktion  $f$  ist genügend differenzierbar
- Nearest Neighbor:  $O(h)$  genauer:  $\leq |f'(\xi)|/2 \cdot h$   
(konstante Interpolation)
- Lineare Interpolation:  $O(h^2)$  genauer:  $\leq |f''(\xi)|/8 \cdot h^2$
- Catmull Rom:  $O(h^3)$  genauer:  $\leq |f'''(\xi)|/24 \cdot h^3$
- B-Spline:  $O(h^4)$
- Auch im allgemeinen Fall gültig, wenn  $h$  = maximaler Abstand aufeinander folgender Stützstellen

- $|f(x) - p(x)| \leq h^2/8 \max_{\xi} \{ |f''(\xi)| \}$



- Anwendung:
  - Tabellierung von Funktionswerten
  - Look-up-tables
- Beispiel: Tabelle (look-up-table) für  $\sin(x)$   
 Gewünschte Genauigkeit:  $\varepsilon_0 = 10^{-3}$ ,  $\varepsilon_1 = 10^{-6}$   
 Aus Fehlerabschätzung folgt für Schrittweite  $h$  :  
 $h_0 \leq 9 \cdot 10^{-2}$  bzw.  $h_1 \leq 2.8 \cdot 10^{-3}$

- Die Funktion

$$f(x) = \sin(\pi x), I = [0,1]$$

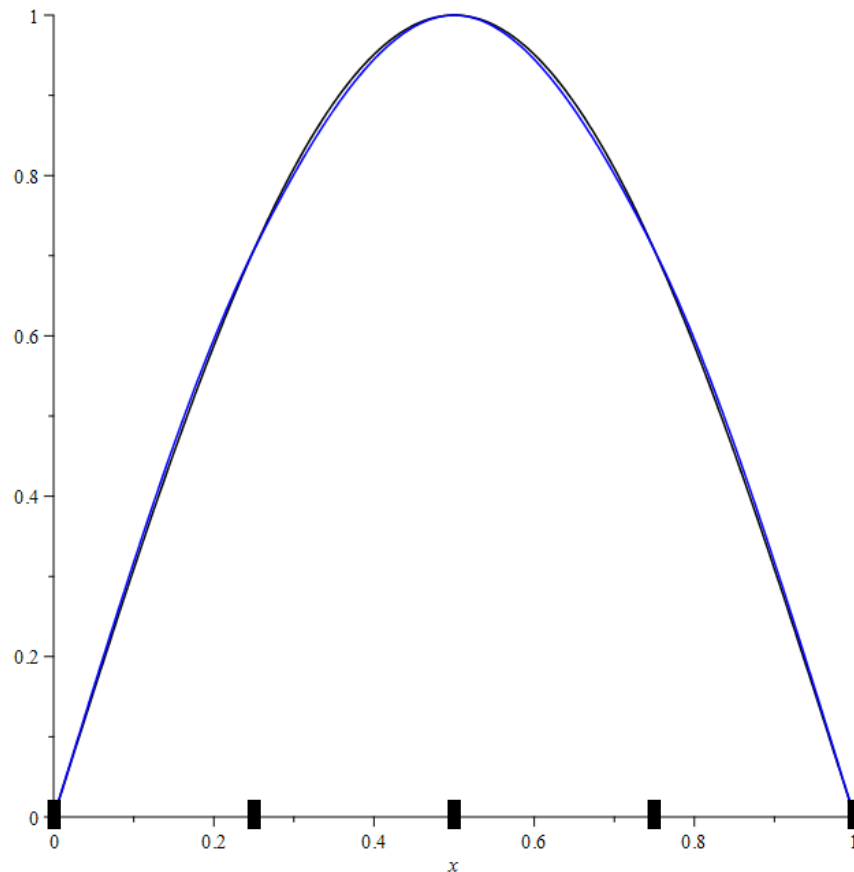
wird äquidistant abgetastet mit Schrittweite  $h$

- Der Interpolationsfehler für linearen, Catmull-Rom- und B-Spline-Interpolanten

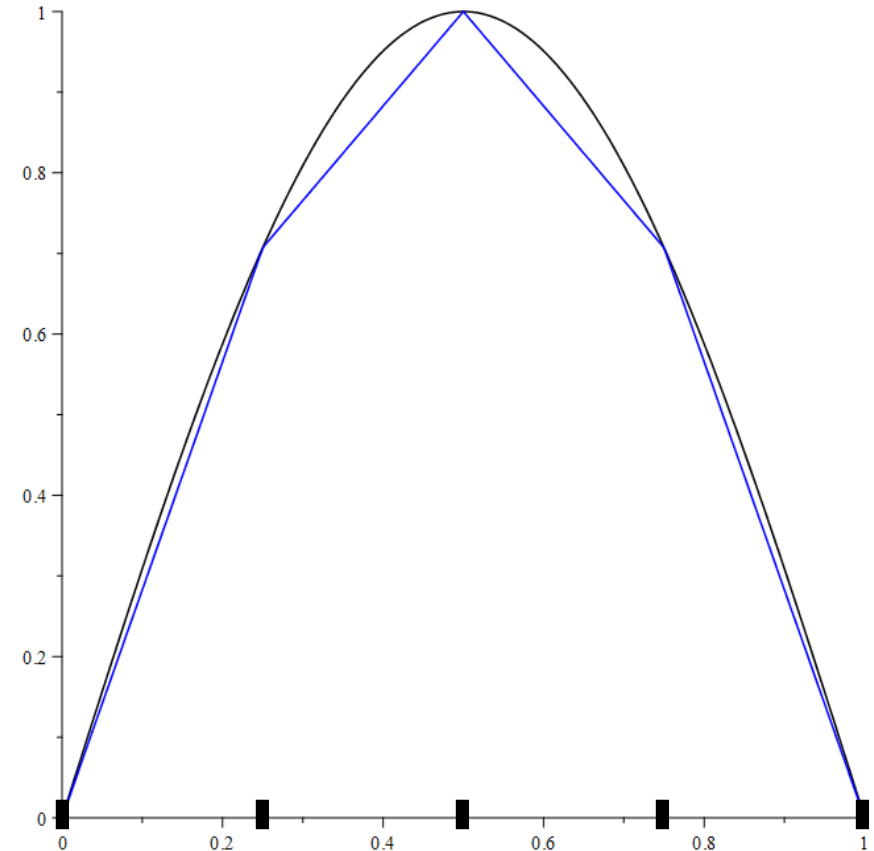
$h$	<i>linear</i>	<i>EOC</i>	<i>Catmull-R.</i>	<i>EOC</i>	<i>B-Spline</i>	<i>EOC</i>
	$O(h^2)$	4	$O(h^3)$	8	$O(h^4)$	16
0.25	0.0703		8.80E-03		1,06E-03	
0.125	0.0188	3.73	9.90E-04	8.88	6.31E-05	16.86
0.0625	4.79E-3	3.93	1.21E-04	8.13	3.89E-06	16.22
0.03125	1.20E-3	3.98	1.52E-05	8.03	2.42E-07	16.06
0.015625	3.01E-4	4.00	1.89E-06	8.01	1.51E-08	16.02

- Die Funktion  $f(x) = \sin(\pi x)$ ,  $I = [0,1]$   
samples:  $\{0.0, 0.25, 0.5, 0.75, 1.0\}$  ( $h = 1/4$ )

## Catmull-Rom-Interpolant



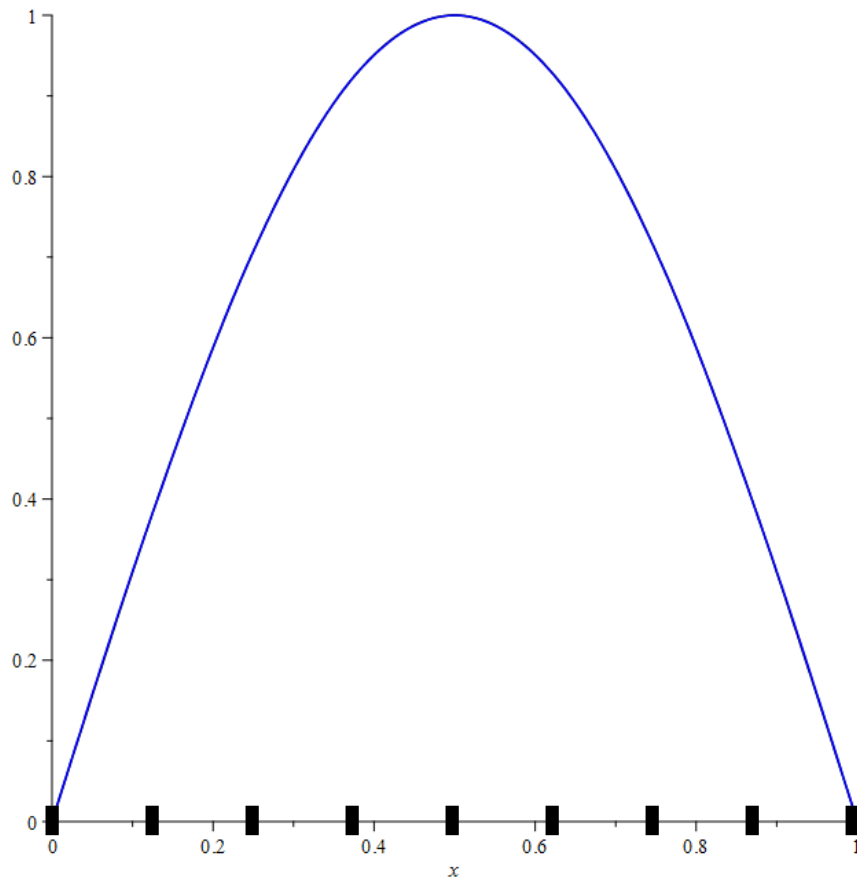
## Linearer Interpolant



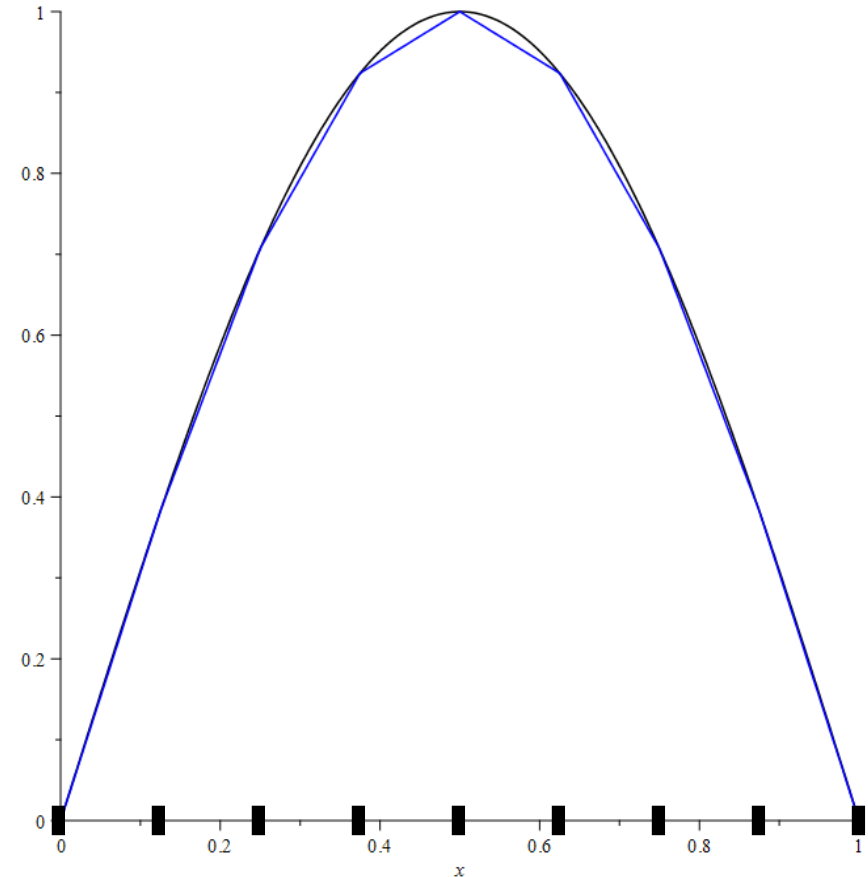


- Die Funktion  $f(x) = \sin(\pi x)$ ,  $I = [0,1]$   
 samples:  $\{ 0.0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0 \}$  ( $h = 1/8$ )

## Catmull-Rom-Interpolant

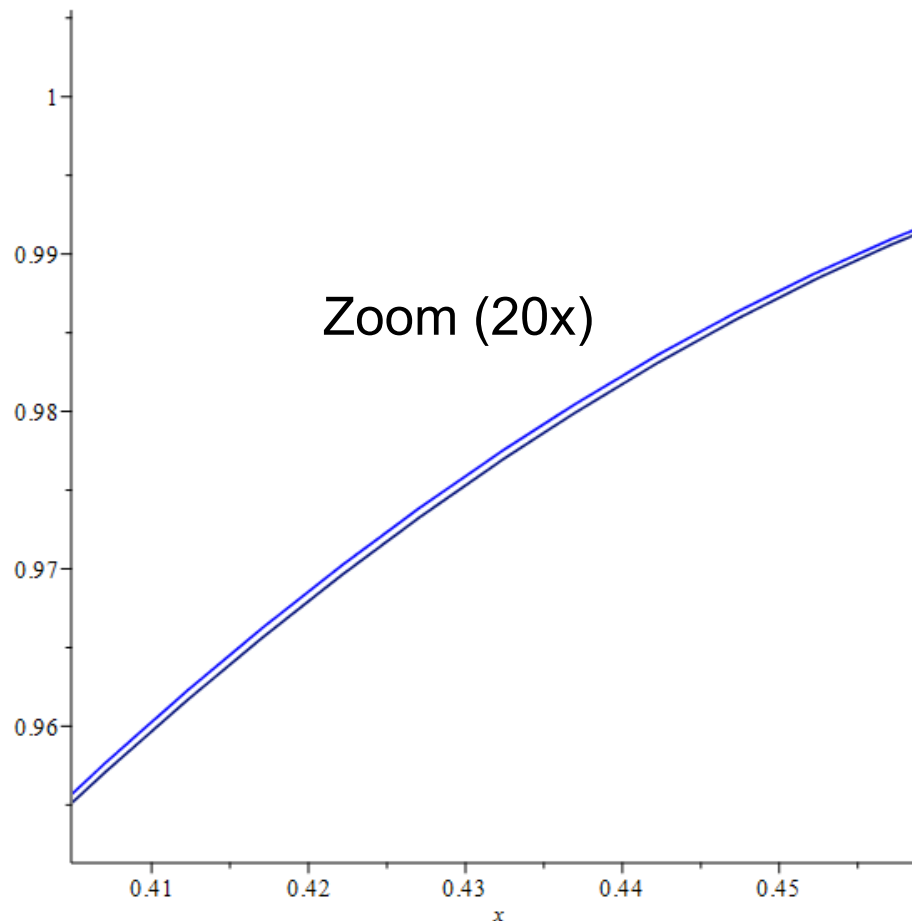


## Linearer Interpolant

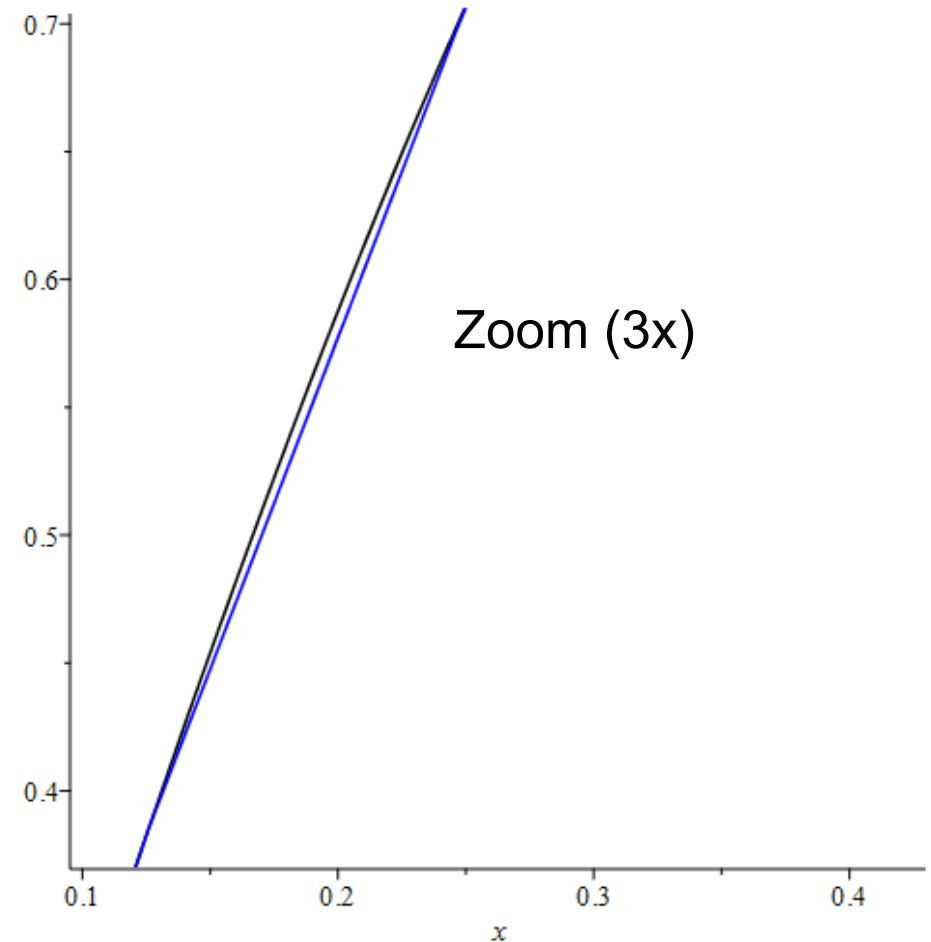


- Die Funktion  $f(x) = \sin(\pi x)$ ,  $I = [0,1]$   
**samples:**  $\{ 0.0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0 \}$  ( $h = 1/8$ )

## Catmull-Rom-Interpolant



## Linearer Interpolant



Die Funktion

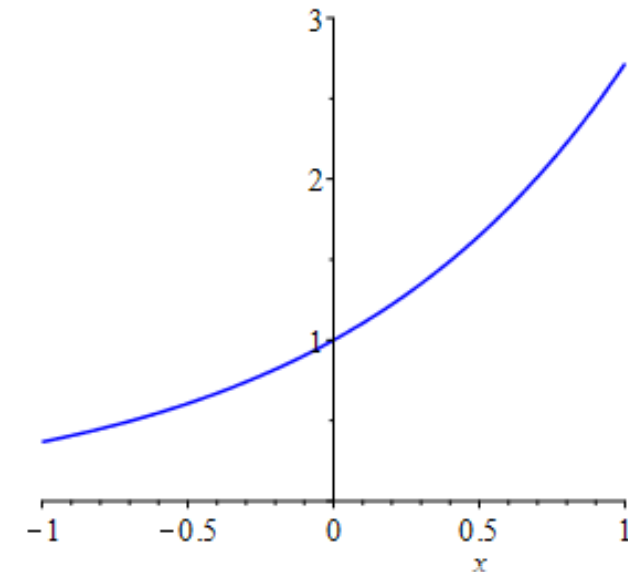
$$f(x) = \sin(\pi x), I = [0,1]$$

wird äquidistant abgetastet mit Schrittweite  $h$

Interpolationsfehler  
für linearen, Catmull-Rom- und B-Spline-Interpolanten

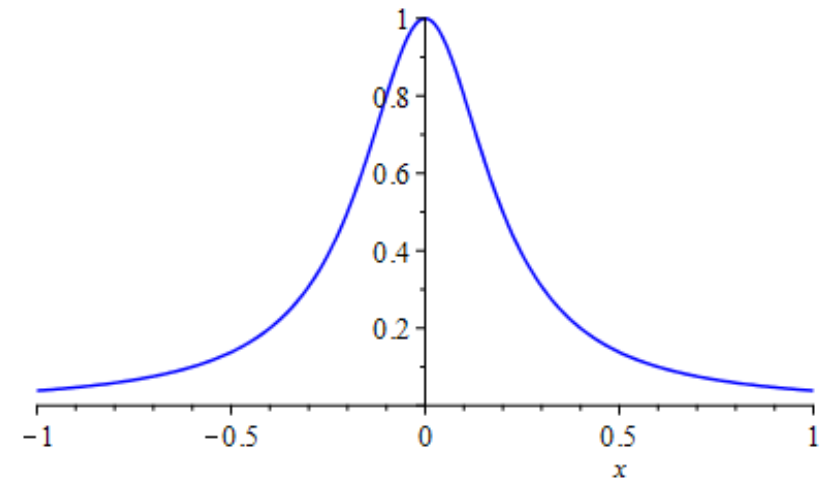
$h$	<i>linear</i>	<i>EOC</i>	<i>Catmull-R.</i>	<i>EOC</i>	<i>B-Spline</i>	<i>EOC</i>
	$O(h^2)$	4	$O(h^3)$	8	$O(h^4)$	16
$2^{-2}$	0.07		0.010		6,96E-05	
$2^{-3}$	1.9E-02	3.68	1.75E-03	5.7	4,21E-06	16,54
$2^{-4}$	4.8E-03	3.96	2.4E-04	7.3	2,60E-07	16,17
$2^{-5}$	1.2E-03	4.0	3.0E-05	8,0	1,75E-08	14,85
$2^{-6}$	3.0E-04	4.0	3.75E-06	8.0		

- $f(x) = \exp(x)$ ,  $I = [-1, 1]$
- wird äquidistant abgetastet mit Schrittweite  $h$
- Interpolationsfehler:  
für linearen, Catmull-Rom-  
und B-Spline-Interpolanten



$h$	<i>linear</i>	<i>EOC</i>	<i>Catmull-R.</i>	<i>EOC</i>	<i>B-Spline</i>	<i>EOC</i>
	$O(h^2)$	<b>4</b>	$O(h^3)$	<b>8</b>	$O(h^4)$	<b>16</b>
0.4			2,20E-03		1,67E-04	
0.2	1,24E-02		3,11E-04	7,07	1,09E-05	15,32
0.1	3,24E-03	3,83	4,12E-05	7,55	6,94E-07	15,71
0.05	8,30E-04	3,90	5,30E-06	7,77	4,40E-08	15,77
0.025	2,10E-04	3,95	6,74E-07	7,86		

- $f(x) = 1/(1+25x^2)$ ,  $I = [-1, 1]$   
wird äquidistant abgetastet  
mit Schrittweite  $h$
- Interpolationsfehler  
für linearen, Catmull-Rom-  
und B-Spline-Interpolanten



$h$	<i>linear</i>		<i>Catmull-R.</i>		<i>B-Spline</i>	
	$O(h^2)$	<b>4</b>	$O(h^3)$	<b>8</b>	$O(h^4)$	<b>16</b>
0.4	0.5		4.50E-01		0.422	
0.2	6.75E-02	7.41	1.82E-02	24.78	2.20E-02	19.29
0.1	4.18E-02	1.61	1.12E-02	1.63	3.18E-03	6.90
0.05	1.41E-02	2.98	1.74E-03	6.43	2.77E-04	11.49
0.025	3.80E-03	3.69	1.58E-04	11.02	1.61E-05	17.21

- Lokale Verfahren vs. globale Verfahren
- Lokale Verfahren
  - *nearest neighbor* : unstetig,  $O(h)$
  - linear: stetig,  $O(h^2)$
  - Catmull-Rom: glatt (diff.-bar,  $C^1$ ),  $O(h^3)$
  - kubische Splines: glatt (diff.-, bar,  $C^2$ ),  $O(h^4)$
- Schätzen von Ableitungen : vorwärts, rückwärts, zentral
- Rechenaufwand:
  - Aufwand für Suche ...
  - der Rest ist  $O(1)$
- Globale Verfahren
  - Polynominterpolation
  - B-Spline-Interpolation