Numerik 1

Lennart Cockiwer

October 2022

Inhaltsverzeichnis

Kapitel 1

Grundlegende Konzepte der Numerik

Eine "Mathematische Aufgabe" besteht abstrakt aus der Auswertung einer Abbildung

$$\phi : X \to Y \quad \text{in einem } x \in X \quad \text{mit geeigneten Räumen } X, Y$$

Beispiele

- Berechnung eines Integrals: $\int_a^b f(x)dx$:

$$\phi_{\int}((a,b), f) : X \times L^1 \to \mathbb{R}$$

- Lösung einer DGL

Objekte und Auswertungen können meist nur näherungsweise dargestellt werden, da z.B. nicht jede reelle Zahl auf dem Computer exakt dargestellt werden kann.

- Durch nicht exakte Darstellung entstehen Rundungsfehler
- Durch vereinfachte Beschreibung komplexer Vorgänge können auch Modellfehler enstehen
- Durch ungenaue Messungen können Datenfehler entstehen

Die Numerik befasst sich unter anderem mit folgenden Fragestellungen:

Algorithmik: Angabe von Algorithmen bzw. Berechnungsverfahren zur näherungsweisen Lösung von math. Aufgaben

Konditionierung und Stabilität: Einfluss von Störungen(Fehlern) auf das Ergebnis der math. Aufgabe oder Berechnung

Konvergenz: Abschätzung des Fehlers zwischen berechneter und exakter Lösung

Komplexität: Aufwand des numerischen Verfahrens

1.1 Zahlendarstellung und Rundungsfehler

Computer können Zahlen nur mit endlich vielen Ziffern darstellen, damit sind nicht alle reellen (komplexen) Zahlen exakt darstellbar. Manche Programme können Ganzzahlen mit beliebig vielen Stellen oder Gleitkommazahlen mit beliebig vielen Stellen darstellen (endlich viele, auch begrenzt durch Speicherplatz). Rechnungen damit werden dann jedoch sehr langsam. Meist ist die Anzahl der Stellen also begrenzt, weil nur eine gewisse Anzahl an Bits/Bytes für die Darstellung einer Zahl reserviert ist. 1 Byte = 8 Bits, kann Ganzzahlen zwischen 0 und 255, bzw zwischen -128 bis +127, darstellen.

$\underbrace{\pm}_{1-\text{Bit}} m * b^e$, $b = 2$ Basis $m = 1.\underbrace{m_1 \dots m_{52}}_{52-\text{Bits}}$, $e = \underbrace{c}_{11-\text{Bits}} - 1023$ (double-precision)
Menge aller Gleitkommazahlen =: A (endliche Menge)
$D := [x_{min}, x_{max}] \cup 0 \cup [x_{posmin}, x_{max}$ ist der "darstellbare Zahlenbereich"
Rundung bildet D auf A ab $rd : D \to A$, sodass $|x - rd(x)| = \min_{y \in A} |x - y|$
Rundung zur nächstliegenden Zahl.

IEEE : bei gleichweit entfernten Gleitkommazahlen nehme die, wo $m_{52} = 0$ ist.
Für eine Zahl $x = \pm m * 2^e$ mit $m \in [1, 2)$ ist der absolute Rundungsfehler:

$$|x - rd(x)| \leq \frac{1}{2} * 2^{-52} * 2^e$$

der relative Rundungsfehler

$$\frac{|x - rd(x)|}{|x|} \leq \frac{1}{2} * 2^{-52}$$

ist unabhabgig von der Größe von x.
Mit der "Maschinengenauigkeit" einer Gleitkommadarstellung bezeichnet man den Abstand zwischen 1 und der nächst größeren Gleitkommazahl. bei doppelt genauer Darstellung :

$$\text{"Epsilon" "Eps", "eps"} := 2^{-52} \approx 2,22 * 10^{16}$$

Es gilt immer : $rd(x) = rd(x(1 + \varepsilon))$ für alle $|\varepsilon| \leq \frac{eps}{2}$

Wichtig wird das Runden insbesondere auch bei den arithmetischen Operationen $+, -, *, \div$
Diese werden in Computern duch Maschinenoperationen ersetzt $(\oplus \ominus \otimes \oslash)$ bei denen das Ergebnis wieder eine Maschinenzahl ist.
Für jede Operation $* \in \{+, -, *, \div\}$ und $y, x \in A$
gilt :

$$x \otimes y \in A,\ x \otimes y = (x * y)(1 + \varepsilon) \text{ mit } |\varepsilon| \leq \frac{eps}{2}$$

Im allgemeinen gelten die typischen Geseze nicht, also:

i) $(x \otimes y) \oplus z$ ist nicht assoziativ

ii) $(x \otimes y) \otimes z$ ist nicht distributiv

iii) $x \oplus y = x$ falls $|y| \leq \frac{|x|}{2} eps$

mit iii) kann man eps durch ausprobieren berechnen. (noch was von foto abschreiben)

1.2 Kondition und Stabilität

Einfluss von Störungen oder Fehlern auf das Ergebnis einer mathematischen Aufgabe oder eines Berechnungsverfahrens.

Beispiel 1.2.1. Kleine Unterschiede von Werten können evtl. auf dem Rechner/ in der gewählten Zahlendarstellung gar nicht unterschieden werden. (Berechnungsverfahren)

Beispiel 1.2.2. Mathematische Aufgabe: Beispiel für lineares Gleichungssystems: $x: \ Ax = b$ mit $A = \begin{pmatrix} 1,2969 & 0,8648 \\ 0,2161 & 0,1441 \end{pmatrix}$ für $b = \begin{pmatrix} 0,8642 \\ 0,1220 \end{pmatrix}$ ist die Lösung $x = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$. Für $b = \begin{pmatrix} 0,86419999 \\ 0,12200001 \end{pmatrix}$ ist die Lösung $x = \begin{pmatrix} 0,9911 \\ -0,487 \end{pmatrix}$

Definition 1.2.3. Eine mathematische Aufgabe heißt "schlecht konditioniert" wenn kleine Änderungen in den Daten große relative Fehler verursachen. Andernfalls heißt die Aufgabe "gut konditioniert"

Bemerkung 1.2.4. Eine gute Konditionierung deiner math. Aufgabe ist notwendig, um das Problem numerisch sinnvoll lösen zu können, da Rundungsfehler sonst große Fehler verursachen können.

Sei $\phi : X \to Y$ eine mathematische Aufgabe

Definition 1.2.5. Ein Verfahren oder Algorithmus zur (näherungsweisen) Lösung der math. Aufgabe ϕ ist eine Abbildung $\tilde{\phi} : X \to Y$,
die durch Hintereinanderschaltung endlich vieler (oder abzählbar unendlich vieler) elemetarer, möglicherweise rundungsfehlerbehafteter, Rechenoperation

$$\phi^{(k)}, \ k = 1, 2, 3, \ldots$$

definiert ist, also

$$\tilde{\phi} = \cdots \circ \phi^{(3)} \circ \phi^{(2)} \circ \phi^{(2)} \circ \phi^{(1)}$$

Bemerkung 1.2.6. typicherweise gibt es verschiedene Algorithmen für die gleiche math. Aufgabe ϕ. Von einem "guten" Algorithmus erwartet man, dass die im Verlauf des Algorithmen akkumulierten Fehler den durch die Kondition der math. Aufgabe unvermeidbaren Fahler nicht wesentlich übersteigen.

Definition 1.2.7. Ein Algorithmus $\tilde{\phi}$ heißt "instabil" , wenn es eine Störung $\tilde{x}$ von x gibt so dass der durch den Rundungsfehler und Störungen verursachte relative Fehler erheblich größer ist als der nur durch die Störung verursachte Fehler, d.h. falls $\phi(x) \neq =$ und $\dfrac{|\tilde{\phi}(\tilde{x}) - \phi(x)|}{|\phi(x)|} \gg \dfrac{|\phi(\tilde{x}) - \phi(x)|}{|\phi(x)|}$. Der Algorithmus heißt stabil, falls er nicht instabil ist. (ggf ales "bei x" oder "für kleine $|x|$", große $|x|$, o.ä.)

Beispiel 1.2.8. math. Aufgabe:

$$\phi(x) = \frac{1}{x(x+1)} \ , x \in \mathbb{R}\backslash 0, -1$$

Es gilt:

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

Zwei mögliche verfahren:

$$\tilde{\phi}_1(x) = \frac{1}{x(x+1)}$$

ist stabil für $x \gg 1$

$$\tilde{\phi}_2(x) = \left(\frac{1}{x}\right) - \left(\frac{1}{(x+1)}\right)$$

ist instabil für $x \gg 1$ wegen Auslöschung der Differenzbildung.

1.3 Landau-Symbole, Genauigkeit und Komplexität

Beispiel 1.3.1.

(a) $A \in \mathcal{M}_n(\mathbb{R})$, n-Vektor $x \in \mathbb{R}^n$, $Ax = b \in \mathbb{R}^n$, $b_i = \sum_{j=1}^{n} A_{ij}x_j$
Rechnung von Ax : n^2 Matrixmultiplikationen, $n(n-1)$ Additionen nötig.
$\rightsquigarrow$ Rechenaufwand etwa quadratisch in der Dimension des Gleichungssystems. (bei voll besetzter Matrix)

(b) Genauigkeit der Differenzenquotienten zur Approximation der Ableitung:

$$\left| n'(x) \frac{n(x+h) - n(x)}{h} \right| \leq h \frac{1}{2} \underset{[x,x+h]}{max} |n''|$$

Fehler gleich Größenordnung wie Abstand h.

Definition 1.3.2. Seien $D \subset \mathbb{R}^n$, $f, g : D \to \mathbb{R}$, und $x, x_0 \in D$
Man sagt:

i) Die Funktion f wächst für $x \to x_0$ langsamer als g", geschrieben als: $f = \sigma(g)$ "f ist klein-o von g"

ii) Die Funktion "f wächst für $x \to x_0$ nicht wesentlich schneller als g", geschrieben als $f = \mathcal{O}(g)$, "f ist groß-$\mathcal{O}$ von g" wenn $\exists c > 0 \ \exists \varepsilon > 0 :$ $|f(x)| \leq c|g(x)| \ \forall x \in B_\varepsilon(x_0) |x - x_0| \leq \varepsilon$

iii) analoge Definition für $x \to \pm\infty$

Bemerkung 1.3.3. "Konditionierung schlecht" bzw. "Konditionierung instabil" ist nicht genau definiert.
Was einfacher ist: Aufgabenstellung bzw. Verfahren: Falls Fehlerverstärkung bei verfahren kleiner als bei anderen, dann ist das erste Verfahren "stabiler", bzw. eine aufgabe "besser konditioniert"

1.4 Differentielle Fehleranalyse:

Math. Aufgaben $\phi : X \to Y$. Ist ϕ (unendlich) differenzierbar, dann kann die Kondition auch mit Hilfe der Ableitungen von ϕ bestimmt/berechnet werden:
Sei $\phi : \mathbb{R} \to \mathbb{R}$ eine Abbildung, $x \in \mathbb{R}^m,\ x = (x_1, ..., x_m)^T$

$$\phi(x) = (\phi_1(x_1, ...x_m), ..., \phi_m(x_1, ...x_m))^T$$

Die ϕ_i seinen alle zweimal stetig differenzierbar (partiell).
Damit gilt dann:

$$\begin{aligned}\Delta y_i &= \phi_i(x + \Delta x) - \phi(x) \\ &= \sum_{j=1}^{m} \frac{\partial \phi_i(x)}{\partial x_i} \Delta x_i + R_i(x, \Delta x) \text{ (Taylor)} \\ &= \sum_{j=1}^{m} \frac{\partial \phi_i(x)}{\partial x_i} \Delta x_i + R_i(x, \Delta x) + \mathcal{O}(|\Delta x|^2)\end{aligned}$$

Dann folgt für den relativen Fehler: $\frac{\Delta y}{|y|} = \frac{\Delta y}{|\phi(x)|}$

$\frac{\Delta y}{y_i} = \sum_{j=1}^{m} \frac{\partial \phi_j(x)}{\partial x_,} \frac{x_j}{\phi_i(x)} \frac{\Delta x_j}{x_j}$ Für $x_j \neq 0, y_i \neq 0$
$\frac{\Delta y}{y_i}$ Ist der relative Aufgabenfahler
$\frac{\partial \phi_j(x)}{\partial x_,} =: K_{ij}(x)$
$\frac{\Delta x_j}{x_j}$ ist der relative Datenfehler

Definition 1.4.1. Die $K_{ij}(x), i = 1, ..., n, j = 1, ..., m$ heißen "relative Konditionszahlen" von ϕ in x sie sind ein Maß dafür, wie sich kleine relative Fehler in den Eingangsdaten im ergebnis auswirken.

Die Aufgabe: $y = \phi(x)$ aus x zu berechnen, ist schlecht konditioniert, wenn es ein i, j gibt mit $|K_{ij}(x)| \gg 1$. Ansonsten ist ϕ gut konditioniert.

Beispiel 1.4.2. Grundoperation Addition: $\phi(x_1, x_2) = x_1 + x_2$

$$K_1 = \frac{\partial \phi}{\partial x_1}(x) \frac{x_1}{\phi(x)} = 1 * \frac{x_1}{x_1 + x_2} = \frac{1}{1 + \frac{x_2}{x_1}}$$

$$K_2 = \frac{\partial \phi}{\partial x_2}(x) \frac{x_2}{\phi(x)} = 1 * \frac{x_2}{x_1 + x_2} = \frac{1}{1 + \frac{x_1}{x_2}}$$

Für $\frac{x_1}{x_2} \approx -1$ werden die K_i sehr groß, dort ist die Addition schlecht konditioniert.
Das entspricht $x_1 \approx -x_2$, entspricht Subtraktion von 2 Zahlen, die fest gleich groß sind.
Bei Gleitkommazahlen: Übereinstimmung in den vorderen Mantissenstellen, dadurch Genauigkeit des Resultats geringer als der Daten.

Definition 1.4.3. Unter "Auslöschung" versteht man den Verlust an wesentlichen Dezimalstellen bei der Subtraktion von Zahlen gleichen Vorzeichens. Dies kann zu relativ großen Fehlern führen, falls eine oder beide Zahlen von operationen gerundet ($\Delta x \neq 0$) werden.

Bemerkung 1.4.4. Doese differenzielle Fehleranalyse kann analog für einen Komplexen Algorithmus $\tilde{\phi} = \phi^{(n-1)} \circ ... \circ \phi^{(1)}$, bestehend aus einfachen Rechenoperationen $\phi^{(i)}$, durchgeführt werden, Kettenregel führt auf Ableitung der Hintereinanderschaltungen. Für Komplexe Algorithmen aber nicht sehr Sinnvoll durchzuführen. Man kann stattdessen versuchen statistische Methoden anzuwenden, in denen z.B. Rundungsfehler durch zufalllsvariablen modelliert werden, um damit Wechselwirkungen abschätzen zu können.

Beispiel 1.4.5 (Rekursive Berechnung von Integralen)**.**
Aufgabe: Es sollen die folgenden Untegrale berechnet werden:

$$I_1 := \frac{1}{e} \int_0^1 x^n e^x dx, \ n = 0, 1, 2, ...$$

Berechnung mit integrationsformeln/numerische Integration: Später in Vorlesung.
mit partieller integration sieht man, dass

$$I_n = 1 - nI_{n-1}$$

eine Lösung ist. Für

$$n = 0 \Rightarrow I_0 = \frac{e-1}{e} \approx 0,632...$$

Numerische Berechnung: $I_0 = 0,632...$
$I_5 = 0,1455...$
$I_{10} = 0,0838...$
$I_{15} = 0,059...$
$I_{20} = -30, ...$
$I_{21} = 635,04$
$I_{22} = -13970, ...$
Man sieht leicht:

$$I_n > 0, \ I_n \leq \int_0^1 x^n dx = \frac{1}{n+1}$$

Warum diese Fehler?
In jedem Schritt der Rekursion wird der Fehler aus dem letzten schritt mit Faktor $-n$ multipliziert. Nach n schritten mit gesamtfaktor $(-n)^n * n!$
Die Fakultät wird schnell groß!

Kapitel 2

Interpolation

Aufgabe der Interpolation ist es, diskrete Datenwerte durch eine Kontinuierliche Funktion darzustellen.

BILD!

Dabei sollen die Datenpunkte $(x_i, y_i), i = 1, ..., n$ exakt durch eine "interpolierende" Funktion $f : I \subset \mathbb{R} \to \mathbb{R}$ mit $f(x_i) = y_i$ dargestellt werden. Vorraussetzung: x_i paarweise verschieden!

Idee dahinter: Datenpunkte sind nur Punktauswertungen einer "glatten" Funktion, die durch f approximiert werden soll. Nach der interpolierenden Funktion f wird üblicherweise in einem "einfachen" Funktionenraum gesucht. Z.B. Polynome Trigonometrische Funktionen,..., evntuell nur stückweise definierte aber insgesamt glatte Funktionen.

Zusätzlich zu Funktionen $f(x_i) = y_i$ können auch evntuell Ableitungen $f'(x_i) = z_i$ vorgegeben sein.

Das ganze funktioniert ähnlich auch bei Daten und Funktionen über mehrdimensionalen Gebieten, z.B. Rekonstruktion von 2D- Flächen in 3D (Computergraik, CAD)

2.1 Polynominterpolation

Ein einfacher Ansatz: Interpolation durch Polynome. Ein Polynom vom Grad n ist hier eine Funktion

$$p : \mathbb{R} \to \mathbb{R},\ p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

mit reellen Koeffizienten

$$a_0, \ldots, a_n \in \mathbb{R},\ a_n \neq 0 \rightsquigarrow \text{Grad } n$$

Es bildet

$$\mathbb{P}_n := \{p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \ \mid\ a_i \in \mathbb{R},\ i = 0, \ldots, n\}$$

Die Menge der Polynome vom Grad $\leq n$
$\mathbb{P}_n$ bildet einen $\mathbb{R}$ Vektorraum,

$$(p + q)(x) := p(x) + q(x),\ (\alpha p)(x) := \alpha p(x)\ \forall \alpha \in \mathbb{R},\ p, q \in \mathbb{P}_n$$

Die "Monome" $\{1, x, x^2, \ldots, x^n\}$ bilden eine Basis von $\mathbb{P}_n$, mit $\dim(\mathbb{P}n) = n+1$.

Definition 2.1.1. Die Aufgabe der Polynominterpolation besteht darin, zu $n+1$ paarweise verschiedenen Punkten $x_i, i = 0, \ldots, n$ ("Stützstellen", "Knoten") und gegebenen Knotenwerten $y_i, i = 0, \ldots, n$ ein Polynom $p \in \mathbb{P}$ zu bestimmen, mit der Eigenschaft:

$$p(x_i) = y_i,\ i = 0, \ldots, n$$

Satz 2.1.2. Die Aufgabe der Polynominterpolation ist eindeutig lösbar, d.h. es gibt genau ein $p \in \mathbb{P}_n$, das die Bedingung erfüllt.

Beweis

(a) Eindeutigkeit der Lösung:
angenommen $p \in \mathbb{P}_n$ und $q \in \mathbb{P}_n$ seien zwei Lösungen, $p(x_i) = y_i = q(x_i)$.
Für die Differenz $p - q \in \mathbb{P}_n$ gilt dann:
$p - q$ hat $n + 1$ Nullstellen in den $x_i, i = 0, \ldots, n$, aber ein $\tilde{p} \in \mathbb{P}_n$ kann höchstens n verschiedene Nullstellen haben, oder es gilt $\tilde{p} \equiv 0$ (z.B. über Satz von Rolle). Also ist $p \equiv q$, es gibt demnach höchstens eine Lösung in $\mathbb{P}_n$

(b) Existenz einer Lösung:
$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,\ p(x_i) = y_i,\ i = 0, \ldots, n$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1 n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_o \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_o \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Bedingungen führen auf lineares Gleichungssystem und Matrix V_n (Vandermonde-Matrix)
Man kann zeigen:

$$\det(V_n) = \prod_{i=0}^{n} \prod_{j=i+1}^{n} (x_j - x_i) \neq 0$$

falls alle x_i paarweise verschieden sind. Also ist das lineare Gleichungssystem eindeitig lösbar, wenn $\det(V_n) \neq 0$ bzw. wenn die Stützstellen paarweise verschieden sind. Zu beliebiger recher Seite $(y_0, \ldots, y_n)$ gibt es also Koeffizienten $(a_0, \ldots, a_n)$ für ein interpolation Polynom.

□

Das Lineare Gleichungssystem liefert im Prinzip auch eine Berechnungsmethode, ist aber schlecht konditioniert, und die Lösung ist relativ aufwändig.

2.1.1 Lagrange-Interpolation

Idee

Wähle Basispolynome für $\mathbb{P}_n$ angepasst an die Stützstellen x_i, so dass das Interpolationspolynom damit aus den Werten y_i leicht bestimmt werden kann. Man kann recht einfach Polynome $L_i^{(n)} \in \mathbb{P}_n$ konstruieren, die in genau einem Stützpunkt $x_i = 1$ sind und in allen anderen Stützpunkten $= 0$.

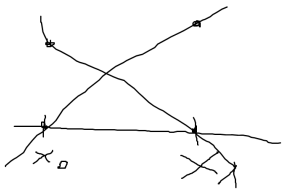
$$L_i^{(n)}(x_i) = \left(\begin{cases} 1 & i = j \\ 0 & \text{sonst} \end{cases} \right) = \delta_{ij}$$

n Nullstellen

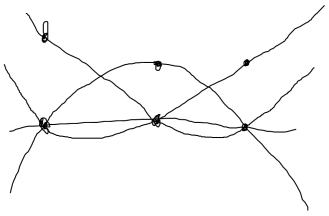
$$x_j, j \neq i \implies L_i^{(n)} = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

Diese $L_i^{(n)}, i = 0, \ldots, n$ heißen "Lagrange-Basispolynome" zu Stützstellen $x_0, \ldots, x_n$.
Linear unabhängig: leicht zu sehen: Alle $L_j^{(n)}(x_i) = 0$. Anzahl ist gleich $\dim \mathbb{P}_n$

Beispiel 2.1.3. $n = 1$ x_0, x_1



$n = 2$



Bemerkung 2.1.4. auch in höheren Dimensionen, $\rightsquigarrow$ Numerik partieller Differentialgleichungen "Finite Elemente Methode"

Definition 2.1.5. Das Polynom $p(x) \coloneqq \sum_{i=0}^{n} y_i \cdot L_i^{(n)}(x)$ heißt Lagrange-Interpolationpolynom zu den Stützstellen $x_0, \ldots, x_n$ und Daten $y_0, \ldots, y_n$.

Vorteil:

Für festes $n \in \mathbb{N}$ und Stützstellen $(x_0, \ldots, x_n)$ relativ leicht zu berechnen, natürliche Darstellung des Interpolationpolynoms mit einfachen Koeffizienten

Nachteil:

- Für viele bzw eng beieinander liegende Stützstellen ist die Formel für $L_i^{(n)}$ relativ schlecht konditioniert (Auslöschungseffekte)
- Will man weitere Stützstellen und Daten dazunehmen, muss komplett neu gerechnet werden

2.1.2 Newton Darstellung

Alternative Darstellung, auch an die Stützstellen angepasst. Eine Polynombasis, die auch zu den Stützstellen passt, aber leicht erweiterbar ist: "Newton-Basis"

$$N_0(x) \coloneqq 1 \text{ konstant}$$
$$N_i(x) \coloneqq \prod_{j=0}^{i-1}(x - x_j),\ i = 1, \ldots, n$$

$\text{Grad}(N_i) = i \implies (N_i)$ linear unabhängig, $\text{span}\{N_0, \ldots, N_i\} = \mathbb{P}_i$, (Basis von $\mathbb{P}_i$)

Damit kann auch das Interpolationspolynom p, $p(x_i) = y_i,\ i = 0, \ldots, n$ in dieser Basis dargestellt werden, $p = \sum_{i=0}^{n} a_i N_i$
Koeffiziernten a_i

$$a_0 = y_0$$
$$y_1 = p(x_1) = a_0 + a_1(x_1 - x_0) + 0 \implies a_1 = \frac{y_1 - a_0}{x_1 - x_0}$$
$$y_n = p(x_n) = a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \cdots + a_n(x_n - x_0)\ldots(x_n - x_{n-1})$$
$$\implies a_n = \frac{y_n - \sum_{i=0}^{n-1} a_i \prod_{j<i} x_n - x_j}{\prod_{j=0}^{n-1} x_n - x_j}$$

Vorteil:

- Man kann beliebig zusätzliche Stützstellen dazunehmen, ohne bisherige Berechnung zu verwerfen
- Reihenfolge/Ordnung der Stützstellen beliebig

Einfacher Algorithmuns zur Berechnung der Koeffizienten:

Satz 2.1.6 (Newton Darstellung mit dividierten Differenzen)**.** Das Interpolationspolynom zu den Punkten $(x_i, y_i), i = 0, \ldots, n$ lässt sich bzgl. der Newton-Basis darstellen als

$$p(x) = \sum_{i=0}^{n} y[x_0, \ldots, x_i] N_i(x)$$

Dabei bezeichnen $y[x_0, \ldots, x_i]$ die zu (x_j, y_j) gehörenden "dividerte Differenzen", rekursiv definiert als

$$y[x_i] \coloneqq y_i, i = 0 \ldots, n$$

$$y[x_i, x_{i+1}, \ldots, x_{i+k}] \coloneqq \frac{y[x_{i+1}, \ldots, x_{i+k}] - y[x_i, \ldots, x_{i+k-1}]}{x_{i+1} - x_i}, \quad i = 0, \ldots, n, \;\; k = 1, \ldots, n-i$$

Beweis
Zu i, n sei $P_{i,i+n}$ das Polynom, das $(x_i, y_i) \ldots (x_{i+l}, y_{i+k})$ interpoliert. $\Longrightarrow$ $p = P_{0,n}$ ist gesucht.

Behauptung.:$P_{i,i+k}(x) = y[x_i] + y[x_i, x_{i+1}](x - x_i) + \ldots y[x_i, x_{i+k}](x - x_i) \ldots (x - x_{i+k-1})$

Per induktion über k: $k = 0: \;\; P_{i,i}(x) = y_i = y[x_i]$
angenommen, es gilt für $k - 1$.
Es ist

$$P_{i,i+k} = P_{i,i+k-1} + a(x - x_i) \ldots (x - x_{i+k-1})$$

mit $a \in \mathbb{R}$
z.z.:

$$a = y[x_i, \ldots, x_{i+k}]$$

. a ist der Koeffizient von x^k in $P_{i,i+k}$
Nach Induktionsannahme gilt:

$$P_{i,i+k-1} = \cdots + y[x_i, \ldots, x_{i+k-1}] \cdot x^{k-1}$$

und

$$P_{i+1,i+k} = \cdots + y[x_{i+1}, \ldots, x_{i+k}] \cdot x^{k-1}$$

Bild

$$P_{i,i+k} = \frac{(x - x_{i+k}) \cdot P_{i,i+k-1} - (x - x_i) P_{i+1,i+k}}{x_1 - x_{i+k}}$$

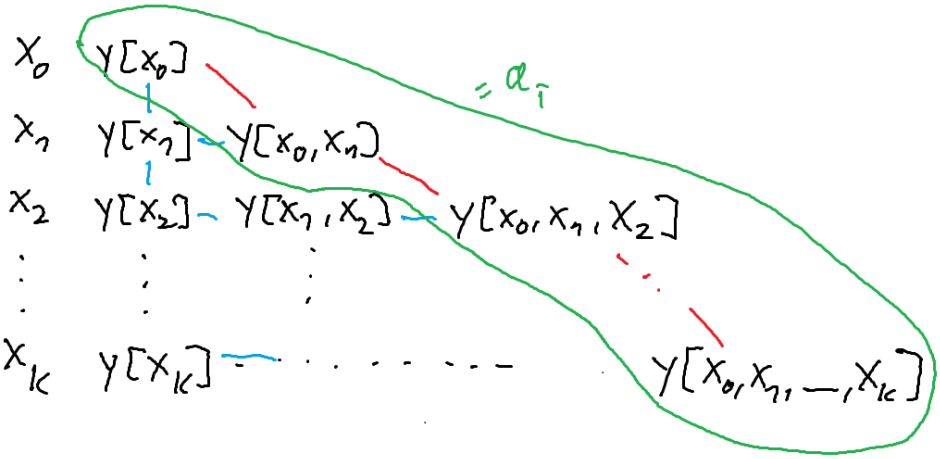
Der Koeffizient der höchsten Potenz x^k in $P_i, i+k$ ist gerade

$$\frac{y[x_i, \ldots, x_{i+k-1}] - y[x_{i+1}, \ldots, x_{i+k}]}{x_i - x_{i+k}} = y[x_i, \ldots, x_{i+k}] = a$$

□

Bemerkung 2.1.7. Das Polynom $P_{i,i+k}$ und die $y[x_i, \ldots, x_{i+k}]$ sind unabhängig von der Reihenfolge der Punkte, also invariant gegenüber Permutation.

Frage (Berechnung der dividierten Differenzen?).



Algorithmus: berechne $P_{i,j}$:

$$\text{Für } i = 0, \ldots, n : P_{i,0} := y_i$$

$$\text{Für } K := 1, \ldots, n,\ i = 0, \ldots, n-k : P_{i,k} := \frac{P_{i+1,k-1} - P_{i,k-1}}{x_{i+k} - x_i}$$

2.1.3 Auswertung von Polynomen

Gegenüber der Auswertung von $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$

$$\#multiplikationen: \ n + 1 + 2 + \cdots + (n-1) = \frac{n(n+1)}{2}$$

Ist die Auswertung der alternativen Darstellung

$$p(x) = a_0 + x(a_1 + x(a_2 + x(\ldots(a_n - 1 + x a_n)\ldots)))$$

nach dem "Horner-Schema"

$$\begin{cases} b_n \coloneqq a_n, & \Rightarrow p(x) = b_0 \qquad \#multiplikationen : n \\ b_k \coloneqq a_k + x b_{k+1}, & k = n-1, \ldots, 0 \end{cases}$$

1. deutlich effizienter
2. oft auch stabiler/besser konditioniert als die Berechnung aller Potenzen

Für die Newton Darstellung:

$$N_i = \prod_{j=0}^{i-1} (x - x_j) = N_{i-1} \cdot (x - x_{i-1})$$

damit gilt für ein Polynom P die alternative Darstellung

$$\begin{aligned} p(x) &= \sum_{i=0}^{n} y[x_0, \ldots, x_i] N_i(x) \\ &= y[x_0] + (x - x_0)(y[x_0, x_1] + (x - x_1)(y[x_0, x_1, x_2] + (x - x_2)(\cdots + (x - x_{n-1})(y[x_0, \ldots, x_n])))) \end{aligned}$$

und ein entsprechendes verallgemeinertes Horner-Schema:

$$\begin{cases} b_n & \coloneqq y[x_0, \ldots, x_n] \\ b_k & \coloneqq y[x_0, \ldots, x_n] + (x - x_k) \cdot b_{k+1}, \quad k = n-1, \ldots, 0 \end{cases}$$

Also $P(x) = b_0$. Dabei ist die Anzahl der Multiplikationen $= n$, somit ist der Aufwand deutlich geringer. Sind die Stützstellen aufsteigend nach dem Abstand zu x sortiert, dann ist das Horner-Schema relativ gut konditioniert.

2.1.4 Interpolationsfehler bei der Interpolation einer gegebenen Funktion

$y_i,\ i = 0, \ldots, n$ nicht (willkürlich) vorgegeben, sondern Funktionswerte einer Funktion $f : I \to \mathbb{R}$, mit $I \subset \mathbb{R}$, alle $x_i \in I,\ i = 0, \ldots, n$
also $y_i = f(x_i)$.

Frage. Wie groß ist der der Unterschied zwischen p und f?

Der Unterschied kann mit einem ähnlichen Ausdruck wie Taylor-Restglied abgeschätzt werden:

Satz 2.1.8. Sei

$$\left[\min_{i=0,\dots,n} x_i, \max_{i=0,\dots,n} x_i\right] \subseteq I \text{ und } f \in C^{n+1}(I)$$

Dann gibt es zu jedem $x \in I$ ein $\xi_x \in \left[\min_i(x_i, x), \max_i(x_i, x)\right]$
mit $\tilde{I} \coloneqq \left[\min_i(x_i, x), \max_i(x_i, x)\right]$ sodass:

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{j=0}^{n}(x - x_j)$$

Beweis Für $x = x_i$, $i \in \{0, \dots, n\}$ ist nichts zu zeigen, da dann

$$f(x) = p(x), \quad \prod(x - x_j) = 0$$

Wir machen einen Ansatz:

$$g(t) \coloneqq \prod_{j=0}^{n}(t - x_j)$$

und

$$c(x) \coloneqq \frac{f(x) - p(x)}{g(x)} \quad \text{(eine Konstante abh. von x bzgl. t)}$$

Setze

$$F(t) \coloneqq f(t) - p(t) - c(x) \cdot g(t)$$

Es ist $F(x) = 0$, und

$$F(x_i) = \underbrace{f(x_i) - p(x_i)}_{=0} - c(x) \cdot \underbrace{g(x_i)}_{=0}$$

$$\begin{aligned}
&\Longrightarrow \quad F \text{ hat mindestens } n+2 \text{ Nullstellen in } \tilde{I} \\
&\Longrightarrow \quad F' \text{ hat mindestens } n+1 \text{ Nullstellen in } \tilde{I} \\
&\Longrightarrow \quad F'' \text{ hat mindestens } n \text{ Nullstellen in } \tilde{I} \\
&\vdots \\
&\Longrightarrow \quad F^{(n+1)} \text{ hat mindestens 1 Nullstelle in } \tilde{I} =: \xi_x
\end{aligned}$$

und es ist

$$0 = F^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - \underbrace{P^{(n+1)}(\xi_x)}_{=0, \text{ da } p \in \mathbb{P}_n} - c(x) \cdot \underbrace{g^{(n+1)}(\xi_x)}_{=(n+1)!}$$

$$\frac{f(x)-p(x)}{g(x)} \cdot (n+1)! = f^{(n+1)}(\xi_x) \implies f(x)-p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \underbrace{g(x)}_{=\prod(x-x_j)}$$

□

Korollar 2.1.9. Ist $f \in C^\infty(I)$ und es gebe ein $M < \infty$ so, dass

$$\left|f^{(n)}(x)\right| \leq M \ \ \forall n \in \mathbb{N} \text{ und } x \in I$$

dann Konvergiert die Folge der Interpolationspolynome $p_n \in \mathbb{P}_n$ zu f mit beliebigen disjunkten Stützpunkten $x_0, \ldots, x_n \in I$ auf I gleichmäßig gegen f

Beweis $\forall x \in [a,b] : |f(x)-p(x)| \leq \frac{1}{(n+1)!} \cdot M \cdot (b-a)^{n+1} \to 0$ für $n \to \infty$ □

Bemerkung 2.1.10. Dies gilt leider nicht für beliebige (auch beliebig glatte) Funktionen

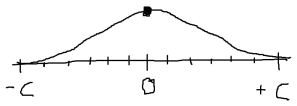
Bemerkung 2.1.11. Satz von Weierstraß sagt aus, dass wir jede stetige Funktion $f \in C^0(I)$ beliebig gut gleichmäßig durch Polynome approximieren können. Dies gilt leider nicht für die (Lagrange)-Interpolationspolynome

Beispiel 2.1.12 (Das Runge-Beispiel)**.**

$$f(x) = \frac{1}{1-x^2} \in C^\infty(\mathbb{R}).$$

Interpolation auf $[-c,+c]$ mit äquidistanten Stützstellen

$$x_i = -c + \frac{2c}{n} i, \ i = 0, \ldots, n$$



Man kann zeigen: ist

$$c \leq \frac{e}{2} \implies ||f-p_n||_\infty \to 0 \text{ für } n \to \infty$$
$$c > \frac{e}{2} \implies ||f-p_n||_\infty \to \infty \text{ für } n \to \infty$$

warum?

$$||f-p||_\infty \coloneqq \max_{x \in [-c,+c]} |f(x)-p(x)|$$
$$\left|f^{(n)}(x)\right| \sim 2^n \cdot n! \cdot \mathcal{O}\left(|x|^{-2-n}\right)$$

Beispiel 2.1.13.

$$f(x) = |x|$$

f ist nur in $C^0([-1,+1])$. Äquidistante Stützstellen:

$$x_i = -1 + \frac{2}{n} \cdot i, \quad i = 0, \ldots, n$$

Man kann zeigen:

$$x \neq x_i, \text{ z.B. } x \text{ irrational, dann } \lim_{n\to\infty} P(x) \neq f(x)$$

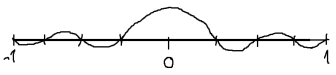
Verhalten gegenüber Störungen in den Daten?

Beispiel 2.1.14.

$$I = [-1,1], \quad x_i = -1 + \frac{2}{n} \cdot i, \quad i = 0, \ldots, n, \quad n \text{ gerade } (x_{n/2} = 0)$$

Wir betrachten Daten

$$y_i = \begin{cases} \varepsilon & i = \frac{n}{2} \\ 0 & \text{sonst} \end{cases}$$



Interpolierende ist

$$p(x) = \varepsilon \cdot L_{n/2}^{(n)}(x) = \varepsilon \cdot \frac{\prod_{j \neq \frac{n}{2}} (x - x_j)}{\prod (0 - x_i)}$$

Die Faktoren im Produkt sind teilweise > 1,

Frage. Kann man bei der Interpolation einer Funktion dem Interpolationsfehler $f - p$ durch geschickte Wahl der Stützstellen kleiner machen?

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot \prod_{j=0}^{n} (x - x_j)$$

$$W(x) := \prod_{j=0}^{n} (x - x_j)$$

Verändern kann man nur den Term

$$W(x) = \prod_{j=0}^{n} (x - x_j).$$

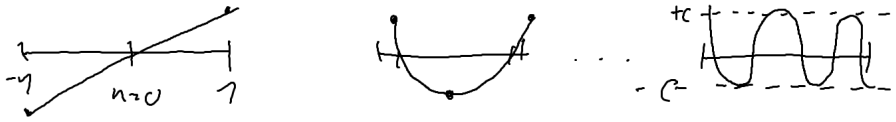
Man kann zeigen, es gilt spezielle paarweise verschiedene Stützstellen, die $|W(x)|$ gleichmäßig minimieren.

Satz 2.1.15.

$$\min_{x_0,\ldots,x_n \in I} \max_{x \in I} \left| \prod_{j=0}^{n} (x - x_j) \right| = \max_{x \in I} \left| \prod_{j=0}^{n} (x - t_j) \right|$$

mit $x_0, \ldots, x_n$ paarweise verschieden und mit den $t_j \in [-1, +1]$ gerade die Nullstellen des "Tschebyscheff- Polynoms" T_{n+1}

Bemerkung 2.1.16 (Optimales $w \in \mathbb{P}_{n+1}$?).



Diese Tschebyscheff-Polynome sind für $k \in \mathbb{N}_0$ definiert als

$$T_k(x) := \cos(k \cdot \arccos x)$$

mit Nullstellen

$$t_j = \cos\left(\frac{2j+1}{2k} \cdot \pi\right), \quad j = 0, \ldots, k-1$$

Es ist nicht direkt klar, dass T_k überhaupt ein Polynom ist (außer für $k = 0, 1$):

$$\begin{cases} T_0 &= \cos(0 \cdot \arccos(x)) = \cos(0) = 1 \\ T_1 &= \cos(1 \cdot \arccos(x)) = x \end{cases}$$

Es gibt folgende 3-Term-Rekursion:

$$T_0(x) \equiv 1,\ T_1(x) = x,\ T_k(x) = 2x \cdot T_{k-1}(x) - T_{k-2}(x), \quad k \geq 2$$

Man sieht für das Runge-Beispiel:
Auch bei Verwendung der Tschebyscheff-Knoten können für relativ großes n, große Fehler am Rand des Intervalles auftreten. Abhilfe gelingt bei diesem Beispiel die Verwendung der "Tschebyscheff-Knoten zweiter Art", das sind gerade die Extremstellen des Tschebyscheff-Polynoms:

$$\tilde{t_j} = \cos\left(\frac{j}{n} \cdot \pi\right), \quad j = 0, \ldots, n$$

Bemerkung 2.1.17 (Kondition/Verhalten bei Störungen:). Sowohl bei Tschebyscheff-Knoten erster oder zweiter Art bleibt die Auswirkung einer Störung bei $x = 0$ auf dem gesamten Intervall $[-1, 1]$ beschränkt, im gegensatz zu äquidistanten Knoten.

2.1.5 Hermite-Interpolation

Nicht nur Funktionswerte vorgegeben, sondern ggf. auch Ableitungen, evtl. auch höhere.

Definition 2.1.18. Die Hermite-Interpolationsaufgabe:
Gegeben:
Stützstellen $x_i,\ i = 0, \dots, m$ paarweise verschieden.
Werte $y_i^{(k)},\ i = 0, \dots, m \quad k = 0, \dots, \mu_i \geq 0$

Gesucht: Polynom $p \in \mathbb{P}_n,\ n \sum_{i=0}^{m} \mu_i$ mit $p^{(k)}(x_i) = y_i^{(x)},\ i = 0, \dots, m \quad k = 0, \dots, \mu_i$
Die x_i werden manchmal auch als μ_i-fache Stützstellen bezeichnet.

Satz 2.1.19. Die Hermite Interpolationsaufgabe ist eindeutig lösbar

Beweis analog zur Lagrange-Interpolation, ähnliche (nicht im Sinne der Äquivalenzrelation) Matrix zur Vandermonde Matrix

$$\begin{aligned} p(x) &= a_0 + a_1 x + \cdots + a_n x^n \\ p'(x) &= a_1 + 2a_2 x + 3a_3 x^2 + \cdots + n a_n x^{n-1} \\ &\vdots \end{aligned}$$

sind z.B. $p(x_0) = b_0$, $p'(x_0) = c_0$, so gilt für die Matrix

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n \\ 0 & 1 & 2x_0 & 3x_0^2 & \dots & n x_0^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} b_0 \\ c_0 \\ \vdots \\ \vdots \end{pmatrix}$$

□

Bemerkung 2.1.20. Eine ähnliche Aufgabe bei der in Stützstellen eventuell nur einige Ableitungen vorgegeben sind, z.B.

$$p \in \mathbb{P}_2 : p(x_0) = y_0, p''(x_1) = y_1^{(2)}, p''(x_2) = y_2^{(2)}$$

ist im allgemeinen nicht oder nicht eindeutig lösbar

Ableitung ist Grenzwert der Differenzenquotienten

$$\frac{f(x+h) - f(x)}{h} \quad \text{für } h \to 0$$

Ähnlich kann man den Grenzwert der dividierten Differenzen anschauen:
Angenommen, die gegebenen Werte y_i sind Werte einer differenzierbaren Funktion $f(x_i) = y_i$, dann ist die erste dividierte Differenz gerade

$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

ist gerade der Differenzenquotient zu f_1
Für $x_j \to x_i$ konvergiert dann $f[x_j, x_i] \to f'(x_i) =: f[x_i, x_i]$
So können Hermite Stützstellen mit vorgegebenen Ableitungen als mehrfache Stützstellen aufgefasst werden:
Damit können wir das Hermite-Interpolationsverfahten wieder in Newton-Darstellung schreiben, wenn man die dividierten Differenzen verallgemeinert:

- Stützstellen mit (höheren) Ableitungen, also $\mu_i > 0$ werden entsprechend dupliziert, statt x_i wird Folge

  $$\underbrace{x_i, \ldots, x_i}_{\mu_i+1-\text{mal}} \quad i = 0, \ldots, m$$

  eingefügt, damit bekommt man eine Folge von Stützstellen $\tilde{x_0}, \tilde{x_1}, \ldots, \tilde{x_n}$.
  Für diese werden nun die modifizierten dividierten Differenzen definiert durch

  $$y[\tilde{x}_i] \coloneqq y_i^{(0)}, \;\; y[\tilde{x}_1, \ldots, \tilde{x}_{i+k}] \coloneqq \begin{cases} y_i^{(k)} \cdot \frac{1}{k!} & \tilde{x}_i = \tilde{x}_{i+k} \\ \dfrac{y[\tilde{x}_{i+1}, \ldots, \tilde{x}_{i+k}] - y[\tilde{x}_i, \ldots, \tilde{x}_{i+k-1}]}{\tilde{x}_{i+k} - \tilde{x}_i} & \tilde{x}_i \neq \tilde{x}_{i+k} \end{cases}$$

  i ist der Original-Index zu $\tilde{x}_i = x_i$

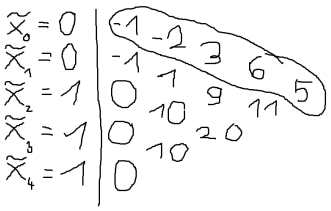
Satz 2.1.21. Damit hat das Hermite-Interpolationspolynom die Darstellung

$$p = \sum_{i=0}^{n} y[\tilde{x}_0, \ldots, \tilde{x}_i] \prod_{j=0}^{i-1} (x - \tilde{x}_j)$$

A8: Stückweise Hermite-Interpolation: Werte $y_i^{(0)}, y_i($ auf $I_i = [x_{i-1}, x_i]$, $i = 1, \ldots, m$:

Beispiel 2.1.22. Gesucht ist ein p mit $p \in \mathbb{P}_4$, $\begin{cases} p(0) = -1, \; p'(0) = -2 \\ p(1) = 0, \; p'(1) = 10, \; p''(1) = 40 \end{cases}$
$m = 1, \; \mu_0 = 1, \; \mu_1 = 2$

Direkte Differenzen dazu?



$$\Rightarrow p(x) = -1 - 2\cdot(x-0) + 3\cdot(x-0)(x-0) + 6\cdot(x-0)(x-0)(x-1) + 5(x-0)(x-0)(x-1)(x-1)$$
$$= -1 - 2x + 3x^2 + 6x^2(x-1) + 5x^2(x-1)^2$$

Ähnlich wie beim Satz über den Fehler der Lagrange-Polynominterpolation kann man zeigen

Satz 2.1.23. Ist

$$I := \left\{ \min_i(x_i), \max_i(x_i) \right\} \text{ und } f \in C^{(n+1)}(I)$$

Dann gibt es zu jedem $x \in I$ ein $\xi \in I$ so, dass für das Hermite-Interpolationspolynom p zu f gilt

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \cdot \prod_{i=0}^{m} (x - x_i)^{\mu_i + 1}$$

Mit der Vorgabe der Ableitungen kann man typicherweise erreichen, dass die oszillationen des interpolierenden Polynoms zwischen den Stützstellen kleiner werden. Eine andere Möglichkeit/Ansatz dafür:

2.1.6 Spline Interpolation

Bei Polynom-Interpolation erhöht sich der Grad des Polynoms mit Anzahl der Stützstellen/Vorgaben, dies kann dann mit höheren $x-$Potenzen zu Oszillationen führen, die nicht gewünscht sind. Ein anderer Ansatz: Verwende nicht global (auf $\mathbb{R}$ bzw. I) definierte Polynome, sondern nur stückweise auf jedem Intervall (z.B. $[x_{i-1}, x_i]$, wenn sortiert) und verwende zusätzliche Übergangsbedingungen: Für $:= x_0 < x_1 < \cdots < x_m = b$ und $I_i =:= [x_{i-1}, x_i],\ i = 1, \ldots, n$ ist der entsprechende Funktionsraum dann:

$$S_{\text{li}}^{(k,r)}[a,b] := \left\{ s \in C^{(r)}[a,b] \colon\ s|_{I_i} \in \mathbb{P}_k,\ i = 1, \ldots, n \right\}$$

(r-mal stetig diffbar, lokale Polynome vom Grad $\leq k$)

Beispiel 2.1.24. Stückweise lineare Interpolation:

$$S_{\text{li}}^{(1,0)}[a,b] = \left\{ s \in C^{(0)}[a,b] \colon s|_{I_i} \in \mathbb{P}_1 \right\}$$

Sind Werte y_i an den Stützstellen x_i vorgegeben, dann ist durch die Vorgabe der Werte in dem Endpunkten jedes Teilintervalls I_i genau ein Polynom $p_i \in \mathbb{P}_1$ festgelegt. Stetigkeit über die Intervallgrenzen ergibt sich dadurch, dass sowohl $p_i(x_i)$ und $p_{i+1}(x_i)$ den Gleichen Wert y_i haben. Der Graph der Funktion s ist gerade der Polygonzug mit Eckpunkten $(x_i, y_i),\ i = 0, \ldots, n$. Auf I_i ist $s|_{I_i}$ gereade das Interpolationspolynom zu $(x_{i-1}, y_{i-1}), (x_i, y_i)$. Also haben wir die Fehlerabschätzung für Interpolation einer Funktion $f \in C^2[x_{i-1}, x_i]$:

$$f(x) - p(x) = \frac{1}{2} f''(\xi_x) \cdot \prod_{j=0}^{1} \underbrace{(x - x_{i-1+j})}_{\leq h_i}$$

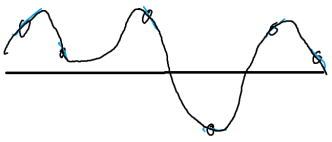
mit $h := \max\limits_{i=1,\ldots,n} |h_i|$ folgt dann

Korollar 2.1.25. Ist $f \in C^2[a, b]$ und s der interpolierende, stückweise lineare Spline, dann gilt:

$$\max_{x \in [a,b]} |f(x) - s(x)| \leq \frac{1}{2} \cdot \max_{x \in [a,b]} |f''(x)| \cdot h^2$$

Beispiel 2.1.26.

$$s \in S_h^{(3,1)}[a, b] : \quad s(x_i) = y_i^{(0)}, s'(x_i) = y_i^{(1)}, \quad i = 0, \ldots, n$$



s gegeben durch die stückweise Hermite Interpolation auf jedem der Teilintervalle. Fehlerabschätzung $|f(x) - s(x)| \leq \ldots$ siehe Übung A8, für $f \in C^4[a, b]$.

"Kubische Splines" : Im gegensatz zu den Hermite-Splines, sollen hier nur die Werte $s(x_i) = y_i,\ i = 0, \ldots, n$ vorgegeben werden, dafür soll der Spline insgesammt glatter sein:

$$s \in S_h^{(3,2)}[a, b] = \left\{ s \in C^2[a, b] \colon s|_{I_i} \in \mathbb{P}_3,\ i = 1, \ldots, n \right\}$$

Frage (Kann das überhaupt Funktionieren?).
$S|I_i \in \mathbb{P}_3$, $s'|_{I_i} \in \mathbb{P}_2$, $s''|_{I_i} \in \mathbb{P}_1$, $s''|_{I_i} \in \mathbb{P}_0$, konstant. Es gilt

$$s \in C^2[a, b] \implies s'' \text{ ist Polygonzug auf } [a, b]$$

Wieviel Bedingungen ergeben sich? Wieviel Freiheitsgerade gibt es?

n Teilintervalle $I_i,\ s\|_{I_i} = P_i \in \mathbb{P}_3 \rightarrow$	$4 \cdot n$ Koeffizienten "frei"	
$s(x_i^-) = y_i,\ i = 0 \ldots, n$	$n+1$	Bedingungen
$s(x_i^-) = s(x_i^+),\ i = 1 \ldots, n-1$	$n-1$	
$s'(x_i^-) = s'(x_i^-),$	$n-1$	
$s''(x_i^-) = s''(x_i^+),$	$n-1$	$4n-2$ Bedingungen

Wir haben also weniger Bedingungen als Koeffizienten, die Bediungngen sollten demnach zu erfüllen sein. Um die Eindeutigkeit zu bekommen sind eventuell noch 2 Bedingungen zusätzlich zu stellen, z.B.:

Steigung in a, b :	$s'(x_0),\ \ s'(x_n)$
Krümmung in a, b :	$s''(x_0),\ \ s''(x_n)$
Periodizität :	$s(x_0) = s(x_n),\ \ s'(x_0) = s'(x_n),$
	$s''(x_0) = s''(x_n),\ \ y_0 = y_n$

Das Wort "Spline" bezeichnet (englisch) eine dünne, biegsame Latte, z.B. Konstruktion der Form eines Schiffrumpfs. Latte versucht (unter der Vorgabe der festen Punkte) ihre elastische Energie zu minimieren. Das entspricht der Minimierung der Gesamtkrümmung. Krümmung eines Graphen $(x, f(x))$:

$$\kappa(x, f(x)) = \frac{f''(x)}{\sqrt{1 + |f'(x)|^2}^{\frac{3}{2}}}$$

Für die Gesamtenergie gilt

$$E(f) \coloneqq \int_a^b |\kappa(x, f(x))|^2 \cdot \sqrt{1 + f'^2} dx \approx \int_a^b \left|f''(x)\right|^2 dx \text{ (nach Linearisierung)}$$

Versucht man, unter allen glatten Funktionen, die in den x_i interpolieren, diese Energie zu minimieren, bekommt man gerade das kubische Spline

Satz 2.1.27. $a = x_0 < x_1 < \cdots < x_n = b, y_i \in \mathbb{R},\ i = 0, \ldots, n$
$A \coloneqq \{f \in C^2[a, b] \colon f(x_i) = y_i,\ i = 0, \ldots, n\}$ Dann gibt es genau eine Lösung $s \in A$ mit

$$E(s) \leq E(f) \quad \forall f \in A$$

mit

$$E(f) \coloneqq \int_a^b (f''(x))^2 dx$$

und dieses $s \in S_h^{(3,2)}[a, b]$ mit $s''(a) = s''(b) = 0$, s ist ein "natürlicher Spline"

Beweis Seien $f, s \in A$, und $s \in S_h^{(3,2)}[a,b]$. Zu zeigen:

$$E(s) \leq E(f)$$

Wir betrachten $f - s$:

$$\begin{aligned} E(f-s) &= \int_a^b (f''(x) - s''(x))^2 dx \\ &= \int_a^b (f'')^2 dx - 2\int_a^b f''s''dx + \int_a^b (s'')^2 dx \\ \text{\includegraphics[height=1em]{runner.png}} &= \int_a^b (f'')^2 dx - 2\int_a^b (f''-s'')s''dx - \int_a^b (s'')^2 dx \end{aligned}$$

Mittlerer Term $= 0$?!?!?!?!?!?!?

$$\begin{aligned} \int_a^b (f''-s'')s''dx &= \sum_{i=1}^n \int_{I_i} (f''-s'') \underbrace{s''|_{I_i}}_{\in \mathbb{P}_3 \subset C^\infty} dx \\ &= \sum_{i=1}^n \left([f'-s']_x i - 1^{x_i} - \underbrace{\int_{I_i} (f'-s') \underbrace{s'''}_{\text{konstant auf } I_i} dx}_{= s'''|_{I_i} \int_{I_i} (f-s)' dx} \right) \\ &= \left((f'-s')\underset{=0}{s''} \right)(x_n) - \left((f'-s')\underset{=0}{s''} \right) = 0, \\ &\Rightarrow \text{\includegraphics[height=1em]{runner.png}} \Rightarrow 0 \leq E(f) - E(s) \Rightarrow E(s) \leq E(f) \end{aligned}$$

Zur Eindeutigkeit:
angenommen, $s, \tilde{s} \in S_h^{(3,2)}[a,b]$ beides Lösungen. Zu zeigen: $s = \tilde{s}$ Damit ist

$$\begin{aligned} E(s-\tilde{s}) = E(s) - E(\tilde{s}) = 0 \text{ (wie oben)} \ &\Rightarrow \int_a^b (s''-\tilde{s}'')^2(x)dx = 0 \\ &\Rightarrow (s-\tilde{s})''(x) \ \forall x \in [a,b] \\ &\to (s-\tilde{s}) \in \mathbb{P}_1[a,b], \ s(x) - \tilde{s}(x) = a_0 + a_1 x \\ &\ s(x_i) = \tilde{s}(x_i), \ i = 0, \ldots, n \\ &\Rightarrow s(x) - \tilde{s}(x) \equiv 0 \end{aligned}$$

Zur Existenz:
mit linearer Algebra: alle Bedingungen sind lineare Gleicuhngen in den Koeffizienten $(a_j(i), \ i = 1, \ldots, n, \ j = 0, 1, 2, 3)$

$$|_{I_i} = a_0^{(i)} + a_0^{(i)} x + a_2^{(i)} x^2 + a_3^{(i)} x^3$$

Also haben wir $4n$ Bedingungen (linear!) für $4n$ Unbekannte und somit ein quadratisches LGS mit linearer Abb. A. Aus der Eindeutigkeit folgt, dass $\operatorname{Ker}(A) = \{0\}$ und $\dim \operatorname{Bild}(A) = 4n$. Dementsprechend haben wir eine eindeutige Lösung für jede rechte Seite □

Beweis (Ein Konstruktiver Existenzbeweis)
Wie oben:

$$s|_{I_i} \in \mathbb{P}_3 \rightsquigarrow s''|_{I_i} \in \mathbb{P}_1,\ s'' \text{ stetig } \rightsquigarrow s'' \text{ Polygonzug.}$$

Sei

$$\begin{aligned}
M_i &:= s''(x_i),\ i = 0, \dots, n \\
\implies s''|_{I_i}(x) &= \frac{M_{i-1}(x_i - x) + M_i(x - x_{i-1})}{x_i - x_{i-1}} \\
\implies s''|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1}(x_i - x) + M_i(x - x_{i-1})) \\
\implies s'|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1}\frac{(x_i - x)^2}{2} + M_i\frac{(x - x_{i-1})^2}{2}) + c_i,\ c_i \in \mathbb{R} \\
\implies s|_{I_i}(x) &= \frac{1}{h_i}(M_{i-1}\frac{(x_i - x)^3}{6} + M_i\frac{(x - x_{i-1})^3}{6}) + c_i(x - x_{i-1}) + d_i,\ d_i \in \mathbb{R}
\end{aligned}$$

Stetigkeit in $x_i,\ i = 1, \dots, n-1 \implies$ Bedingungen für $(\mu_i, c_i, d - i)$

□

Stetigkeit von s' in x_i bedeutet

$$s_i'(x_i) = s_{i+1}'(x_i) : M_i\frac{h_i}{2} + c_i = -M_i\frac{h_{i+1}}{2} + c_{i+1} \quad \left(\text{🚶}\right)$$

Interpolation:

$$\begin{aligned}
s(x_i) = y_i &\implies y_{i-1} = \frac{1}{6}h_i^2 \cdot M_{i-1} + d_i,\ \ y_i = \frac{1}{6}h_i^2 \cdot M_i + c_i h_i + d_i \\
&\implies \frac{y_i - y_{i-1}}{h_i} = \frac{h_i}{6}(M_i - M_{i-1}) + c_i \\
&\implies c_i = y[x_{i-1}, x_i] - \frac{1}{6}h_i(M_i - M_{i-1}),\ \ d_i = y_{i-1} - \frac{1}{6}h_i^2 M_{i-1}
\end{aligned}$$

in $\left(\text{🚶}\right)$ einsetzen.

$$\begin{aligned}
&\frac{1}{2}h_i M_i - \frac{1}{6}h_i(M_i - M_{i-1}) + y[x_{i-1}, x_i] = -\frac{1}{2}h_{i+1}M_i - \frac{1}{6}h_{i+1}(M_{i+1} - M_i) + y[x_i, x_{i+1}] \\
&h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} = 6(y[x_i, x_{i+1}] - y[x_{i-1}, x_i])
\end{aligned}$$

Setze $\mu_i = \frac{h_i}{h_i+h_{i+1}}$ und $\lambda_i = \frac{h_{i+1}}{h_i+h_{i+1}}$

$$\mu_i M_{i+1} + 2M_i + \lambda_i M_{i+1} = y[x_{i-1}, x_i, x_{i+1}]$$

Dies ergibt ein lineares Gleichungssystem für die $M_i, i = 1, \ldots, n-1$ ($M_0 = 0, M_n = 0$, da natürlicher Spline)

$$\begin{pmatrix} 2 & J_1 & & \cdots \\ \mu_2 & 2 & J_2 & \\ & \ddots & \ddots & J_{n-2} \\ & & \mu_{n-1} & 2 \end{pmatrix} \cdot \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{pmatrix} = \begin{pmatrix} 6y[x_0, x_1, x_2] \\ 6y[x_1, x_2, x_3] \\ \vdots \\ 6y[x_{n-1}, x_{n-1}, x_n] \end{pmatrix}$$

Die Matrix des linearen Gleichungssystems ist "strikt diagonaldominant". Also gibt es eine eindeutige Lösung

Definition 2.1.28. Eine Matrix $A \in \mathbb{R}^{m\times m}$ oder $A \in \mathbb{C}^{m\times m}$ heißt

- "strikt diagonaldominant", wenn

$$\forall i = 1, \ldots, m: \quad |a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{m} |a_{ij}|$$

- "schwach diagonaldominant", wenn

$$\forall i = 1, \ldots, m: \quad |a_{ii}| \geq \sum_{\substack{j=1 \\ j\neq i}}^{m} |a_{ij}|$$

  und für mindestens ein i es ">" ist.

Lemma 2.1.29. Ist A strikt diagonaldominant, so gilt

$$\forall z \in \mathbb{R}^n(\mathbb{C}^n) : ||Az||_{\max} \geq c \cdot ||z||_{\max}$$

mit

$$c = \min_i \left(a_{ii} - \sum_{\substack{j=1 \\ j\neq i}} |a_{ij}| \right)$$

Satz 2.1.30. $a = x_0 < x_1 < \cdots < x_n = b$ Die Interpolationsprobleme mit kubischen Splines:

i) Für "natürliche Splines" $s(x_i) = f_i, i = 0, \ldots, n,\ M_0 = M_n = 0$ bzw. $s''(x_0) = s''(x_n) = 0$

ii) Für "vollständige Splines" $s(x_i) = f_i, i = 0, \ldots, n,\ s'(x_0) = f'(x_0),\ s'(x_n) = f'(x_n)$

iii) Für "periodische Splines" $s(x_i) = f_i, i = 0, \ldots, n,$ mit $f_0 = f_n,\ s'(x_0) = s'(x_n),\ s''(x_0) = s''(x_n)$

iv) Für "not-a-Knot-Splines" $s(x_i) = f_i, i = 0, \ldots, n,\ s'''$ stetig in x_1 und $x_{n-1} \Rightarrow s|_{I_1 \cup I_2} \in \mathbb{P}_3,\ S|_{I_{n-1} \cup I_n} \in \mathbb{P}_3$

sind stets eindeutig lösbar.
(kein neuer Beweis, ähnliche Beweise für ii), iii) iv))

Frage (Fehler bei der Splineinterpolation?).

Erinerung. Hermite Interpolation:

$$||f - s||_{\max} \leq \frac{1}{4!} h^4 \left|\left|f^{(4)}\right|\right|_{\max}$$

Satz 2.1.31. $a = x_0 < x_1 < \ldots, < x_n = b$ und $f \in C^4[a,b]$, $h \coloneqq \max\limits_{i=1,\ldots,n} (x_i - x_{i-1})$ Dann ist für den interpolierenden kubischen Spline

$$||f - s||_{\max[a,b]} \leq h^4 \left|\left|f^{(4)}\right|\right|_{\max[a,b]}$$

Beweis $(f-s)(x_i) = 0$. Dann ist auf jeden Teilintervall I_i das Polynom $p_0 \equiv 0$ die lineare Interpolierende zu $f-s$. Die Interpolations-Fehlerabschätzung liefert

$$||(f-s) - p_0||_{\max I_i} = ||f - s||_{\max I_i} \leq \frac{1}{2} h_i^2 \left|\left|(f-s)''\right|\right|_{\max I_i} = \frac{1}{2} h_i^2 \left|\left|f'' - s''\right|\right|_{\max I_i}$$

Sei nun $p_i \in \mathbb{P}_1$ das Polynom, das $f''(x_{i-1})$ und $f''(x_i)$ interpoliert. Dann ist

$$\begin{aligned} \left|\left|f'' - s''\right|\right|_{\max I_i} &\leq \left|\left|f'' - p_i\right|\right|_{\max I_i} + \left|\left|p_i - s''\right|\right|_{\max I_i} \\ &\leq \frac{1}{2} h_i^2 \left|\left|f^{(4)}\right|\right|_{\max I_i} + \max_{j=i-1,i} \left|f''(x_j) - s''(x_j)\right| \end{aligned}$$

Siehe oben:
Die Matrix A ist auf dem Tafelbild ... zu sehen!! ☺

$$A = \begin{pmatrix} 2 & \lambda_i & & \\ \mu_i & \ddots & \ddots & \\ & \ddots & 2 & \end{pmatrix} \implies ||Az|| \geq c||z||_{\max} \text{ mit } c = 1, \text{also } ||z||_{\max} \leq ||Az||_{\max}$$

mit $z_i = f''(x_i) - M_i$ folgt

$$\max_i |f''(x_i) - M_i| \leq \max_i |\mu_i f''(x_{i-i}) + 2f''(x_i) + \lambda_i f''(x_{i+1})|$$

□

2.2 Trigonometrische Interpolation

Ein Ansatz zur Interpolation von periodischen Signalen/Daten mit sin/cos-Funktionen z.B.bei akustischen Signalen.

2.2.1 Zum Hintergrund

"Fourier-Transformation": auch für nicht periodische Funktionen:

$$f\colon \mathbb{R} \to \mathbb{C}\colon \mathcal{F}(f)(y) =:= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{-ixy} dx$$

mit $i^2 = -1$

$$e^{a+ib} := e^a \cdot (\cos(b) + i\sin(b))$$

Dann ist

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} F(x) e^{-ixy} dx$$

z.B. für alle $f \in L^1(\mathbb{R})$

2.2.2 Fourier-Reihen

Für periodische Funktionen, z.B. $f\colon \mathbb{R} \to \mathbb{C}$ 2π-periodisch, also $f(x + 2\pi) = f(x) \quad \forall x \in \mathbb{R}$ Damit

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(kx) + \sin(kx)$$

bzw.

$$f(x) = \sum_{k \in \mathbb{Z}} c_k \cdot e^{ikx}$$

mit

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \cdot e^{-iks} ds \in \mathbb{C}$$

Damit ist

$$a_k = c_k + c_{-k},\ b_k = i(c_k - c_{-k})$$

bzw.

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(s) \cdot \cos(ks)\, ds$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(s) \cdot \sin(ks)\, ds$$

Konvergenz der Fourierreihe für beliebige L^1 oder L^2-Funktionen auf $(-\pi, \pi)$, bzw. stetige, periodishce Funktionen auf $[-\pi, \pi]$
Abgebrochene Reihe/Partialsummen $\to$ Approximation der Funktion f.

2.2.3 Diskrete Fourier-Transformation

Gegeben Werte $a_k, k = 0, \ldots, n-1$ an Punkten $x_k = k\frac{2\pi}{n}$ Mit den Koeffizienten

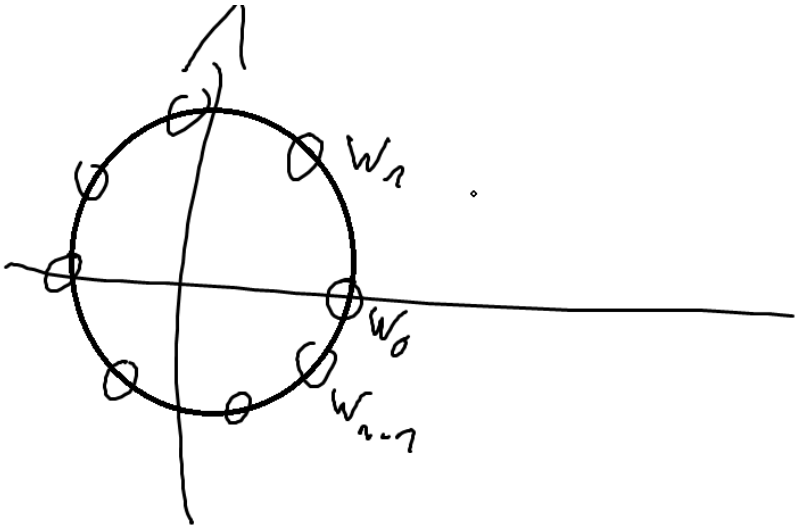
$$\hat{a_j} = \sum_{k=0}^{n-1} a_k \cdot e^{-i \cdot j \cdot k \cdot \frac{2\pi}{n}}$$

ist dann

$$a_k = \frac{1}{n} \sum_{j=0}^{n-1} \hat{a_j} \cdot e^{ijk\frac{2\pi}{n}} = \frac{1}{n} \sum_{j=0}^{n-1} \hat{a_j} \cdot \left(\cos\left(jk\frac{2\pi}{n} \right) + i\sin\left(jk\frac{2\pi}{n} \right) \right)$$

Diskrete Fourier Transformation lässt sich auf Polynom-Interpolation zurückführen mit

$$w := e^{ix}, \quad w_k := e^{ixk} = e^{i2\pi\frac{k}{n}}$$



Dann ist DFT gerade

$$t_n(x) = \sum_{j=0}^{n-1} \frac{\hat{a_j}}{n} \cdot e^{ijx} = \sum_{j=0}^{n-1} \frac{\hat{a_j}}{n} w^j = p(w), \quad \text{mit } p \in \mathbb{P}_{n-1}$$

Polynominterpolation wie in 2.1 - 2.3 geht ganz genau so f+r Komplexwertige Polynome $p \in \mathbb{P}_n[\mathbb{C}]$. Daraus folgt die Existenz und Eindeutigkeit eines interpolierenden Polynoms für Werte a_k an Punkten w_k, $k = 0, \ldots, n-1$. Berechnung der Fourierkoeffizienten $\hat{a_j}$ nicht über Polynom-Methode, sondern entsprechend obiger Formel, bzw. über "Fast Fourier Transformation", mit Aufwand $\mathcal{O}(n \cdot \log n)$ statt$\mathcal{O}(n^2)$.
Anwendung: z.B. MP3-Kompression von Audio-Dateien