Kapitel 7

Interpolation

Idee: Zu einer Funktion $f(x)$ finde man ein Polynom (oder eine andere gut handhabbare Funktion), das mit $f(x)$ an gewissen vorgegebenen Stellen übereinstimmt.

Anwendung: Konstruktion von Zwischenwerten für eine Funktion, von der nur einige Funktionswerte bekannt sind.

7.1 Algebraische Interpolation

Sei Π_n die Menge aller Polynome p vom Grad $\leq n$ der Form

$$p(x) = a_0 + a_1 x + \ldots + a_n x^n, \quad a_j \in \mathbb{R}(\mathbb{C}).$$

Interpolationsproblem:

Gegeben: $n+1$ paarweise verschiedene Punkte $x_0, \ldots, x_n \in \mathbb{R}$, die **Stützstellen** oder **Knoten** genannt werden, sowie $n+1$ zugehörige Werte $y_0, \ldots, y_n \in \mathbb{R}$, **Funktions-** oder **Stützwerte** genannt.
Gesucht: $p \in \Pi_n$, das die *Interpolationsbedingungen*

$$p(x_k) = y_k, \qquad k = 0, \ldots, n, \tag{7.1}$$

erfüllt.

Satz 7.1:

Das Interpolationsproblem (7.1) besitzt genau eine Lösung.

Beweis:

1) Existenz: Das Polynom $p \in \Pi_n$ hat die Form

$$p(x) = \sum_{j=0}^{n} a_j x^j$$

und ist durch $a_0, \ldots, a_n$ eindeutig bestimmt. Aus den Interpolationsbedingungen (7.1) folgt

$$p(x_k) = \sum_{j=0}^{n} a_j x_k^j = y_k, \quad k = 0, \ldots, n.$$

Wir erhalten ein lineares Gleichungssystem (LGS) der Form

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}}_{\text{Vandermonde-Matrix } \mathbf{V}=(x_j^k)_{j,k=1}^n} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Wegen $\det \mathbf{V} = \prod_{0\le j<k\le n}(x_k - x_j) \neq 0$ besitzt das System eine eindeutige Lösung.

2) Die Eindeutigkeit folgt aus der Tatsache, dass $1, x, \dots, x^n$ eine Basis von Π_n bilden. Denn angenommen, für $p_1, p_2 \in \Pi_n$ ist

$$p_1(x_k) = p_2(x_k) = y_k, \qquad k = 0, \dots, n,$$

erfüllt. Das bedeutet

$$p_1(x_k) - p_2(x_k) = 0, \quad k = 0, \dots, n,$$

d.h., das Polynom $q(x) := p_1(x) - p_2(x)$ hat mindestens $n + 1$ Nullstellen. Wegen $q \in \Pi_n$ folgt $q \equiv 0$, und damit ist $p_1 \equiv p_2$. □

Definition 7.2: (Lagrange-Grundpolynome)

Es seien $n + 1$ paarweise verschiedene Punkte $x_k \in \mathbb{R}, k = 0, \dots, n$, gegeben. Dann heißen

$$l_j(x) := \prod_{\substack{k=0\\k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \quad \text{für } j = 0, \dots, n,$$

die zu diesem Knoten gehörenden **Lagrange-Grundpolynome**.

Es gilt

$$l_j(x_k) = \delta_{jk} = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases}.$$

Setzen wir $w_{n+1}(x) := \Pi_{k=0}^n (x - x_k)$, so erhalten wir eine neue Darstellung von $l_j(x)$:

$$\begin{aligned} l_j(x) &= \frac{w_{n+1}(x)}{(x - x_j)} \frac{1}{\prod\limits_{\substack{k=0\\k\neq j}}^{n} (x_j - x_k)} = \frac{w_{n+1}(x)}{(x - x_j)} \lim_{x\to x_j} \frac{(x - x_j)}{w_{n+1}(x)} \\ &= \frac{w_{n+1}(x)}{(x - x_j)} \lim_{x\to x_j} \frac{1}{w'_{n+1}(x)} = \frac{w_{n+1}(x)}{(x - x_j) w'_{n+1}(x_j)}. \end{aligned}$$

Satz 7.3:

Die eindeutige Lösung des Interpolationsproblems (7.1) lässt sich in der **Lagrange-Form**

$$p(x) = \sum_{j=0}^{n} y_j \; l_j(x) = \sum_{j=0}^{n} y_j \prod_{\substack{k=0\\k\neq j}}^{n} \frac{x - x_k}{x_j - x_k} \tag{7.2}$$

darstellen.

Beweis:

Wegen $l_j(x_k) = \delta_{jk},\ 0 \le j,k \le n$, folgt

$$p(x_k) = \sum_{j=0}^{n} y_j l_j(x_k) = y_k,\ k = 0,\dots,n. \qquad \square$$

Die Lagrange-Form ist für numerische Berechnungen kaum geeignet falls n groß ist.

Beispiel:

Sei $n = 2$. Gegeben:

x_k	0	1	3
y_k	1	3	2

Gesucht ist $p(2)$, wobei $p(x_k) = y_k$ für $k = 0,1,2$.

Wir erhalten:

$$l_0(x) = \frac{(x-1)(x-3)}{(0-1)(0-3)},\ l_1(x) = \frac{(x-0)(x-3)}{(1-0)(1-3)},\ l_2(x) = \frac{(x-0)(x-1)}{(3-0)(3-1)}$$

Also folgt

$$p(2) = 1 \cdot l_0(2) + 3 \cdot l_1(2) + 2 \cdot l_2(2) = 1 \cdot \frac{(-1)}{3} + 3 \cdot 1 + 2 \cdot \frac{1}{3} = \frac{10}{3}.$$

Rekursive Berechnung:

Gegeben sind $(x_k, y_k), k = 0,1,\dots,n$, mit $x_k \in [a,b]$ paarweise verschieden.
Wähle $k_0,\dots,k_r \in \{0,1,\dots,n\}$ paarweise verschieden. Sei $p_{k_0,\dots,k_r} \in \Pi_r$ das Interpolationspolynom, das die Interpolationsbedingungen an den Stellen $x_{k_0},\dots,x_{k_r}$ erfüllt, d.h.

$$p_{k_0,\dots,k_r}(x_{k_j}) = y_{k_j}, \qquad j = 0,\dots,r.$$

Lemma 7.4:

Es gilt die Rekursionsformel

(i) $p_k(x) \equiv y_k, \qquad k \in \{0,\dots,n\}$,

(ii) $p_{k_0,\dots,k_r}(x) = \dfrac{(x - x_{k_0})p_{k_1,\dots,k_r}(x) - (x - x_{k_r})p_{k_0,\dots,k_{r-1}}(x)}{x_{k_r} - x_{k_0}}$.

Beweis:

(i) Wegen $p_k \in \Pi_0$ ist p_k eine konstante Funktion mit

$$p_k(x_k) = y_k \quad \Rightarrow\ p_k(x) \equiv y_k.$$

(ii) Sei

$$p(x) = \frac{(x - x_{k_0})p_{k_1,\dots,k_r}(x) - (x - x_{k_r})p_{k_0,\dots,k_{r-1}}(x)}{(x_{k_r} - x_{k_0})}.$$

Dann folgt $p(x) \in \Pi_r$, denn $p_{k_1,\ldots,k_r}, p_{k_0,\ldots,k_{r-1}} \in \Pi_{r-1}$. Weiter gilt

$$
\begin{aligned}
p(x_{k_0}) &= \frac{0 - (x_{k_0} - x_{k_r})\overbrace{p_{k_0,\ldots,k_{r-1}}(x_{k_0})}^{y_{k_0}}}{(x_{k_r} - x_{k_0})} = y_{k_0}, \\
p(x_{k_r}) &= \frac{(x_{k_r} - x_{k_0})p_{k_1,\ldots,k_r}(x_r) - 0}{(x_{k_r} - x_{k_0})} = y_{k_r},
\end{aligned}
$$

und für $j = 1, 2, \ldots, r-1$,

$$
p(x_{k_j}) = \frac{(x_{k_j} - x_{k_0})y_{k_j} - (x_{k_j} - x_{k_r})y_{k_j}}{(x_{k_r} - x_{k_0})} = y_{k_j}.
$$

Somit folgt $p(x_{k_j}) = y_{k_j}$ für $j = 0, \ldots, r$. Wegen 7.3. folgt $p = p_{k_0,\ldots,k_r}$. □

Algorithmus von Neville:

Gegeben: $(x_k, y_k),\ k = 0, \ldots, n$, mit $x_k \in [a,b]$ paarweise verschieden, $x \in [a,b]$ fest.

Gesucht: $p(x)$ wobei $p \in \Pi_n$ die Interpolationsbedingungen $p(x_k) = y_k,\ k = 0, \ldots, n$ erfüllt.

Symmetrisches Dreiecksschema:

$$
\begin{array}{llll}
y_0 = p_0(x) & & & \\
y_1 = p_1(x) & p_{01}(x) & & \\
y_2 = p_2(x) & p_{12}(x) & p_{012}(x) & \\
y_3 = p_3(x) & p_{23}(x) & p_{123}(x) & p_{0123}(x)
\end{array}
$$

Für $p_{123}(x)$ gilt z.B.

$$
p_{123}(x) = \frac{(x - x_1)p_{23}(x) - (x - x_3)p_{12}(x)}{x_3 - x_1}.
$$

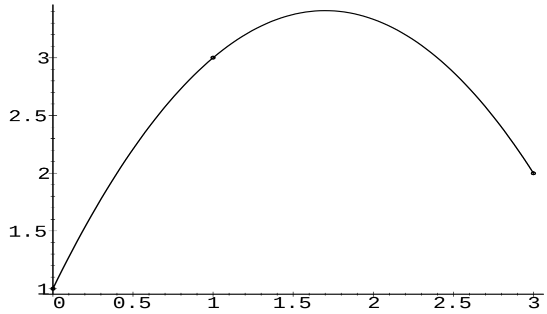
Beispiel:

Gegeben:

x_k	0	1	3
y_k	1	3	2

Gesucht: Funktionswert an der Stelle $x = 2$.

$$
\begin{array}{c|lclcl}
 & r = 0 & & r = 1 & & r = 2 \\
\hline
x_0 = 0 & p_0(2) = 1 & & & & \\
 & & \ddots & & & \\
x_1 = 1 & p_1(2) = 3 & \ldots & p_{01}(2) = 5 & & \\
 & & \ddots & & \ddots & \\
x_2 = 3 & p_2(2) = 2 & \ldots & p_{12}(2) = \frac{5}{2} & \ldots & p_{012}(2) = \frac{10}{3}
\end{array}
$$



Quadratisches Interpolationspolynom (siehe Beispiel)

$$\begin{aligned} p_{01}(2) &= \tfrac{(2-0)\cdot 3-(2-1)\cdot 1}{1-0} = 5, \quad p_{12}(2) = \tfrac{(2-1)\cdot 2-(2-3)\cdot 3}{3-1} = \tfrac{5}{2}, \\ p_{012}(2) &= \tfrac{(2-0)\cdot\frac{5}{2}-(2-3)\cdot 5}{3-0} = \tfrac{10}{3} \end{aligned}$$

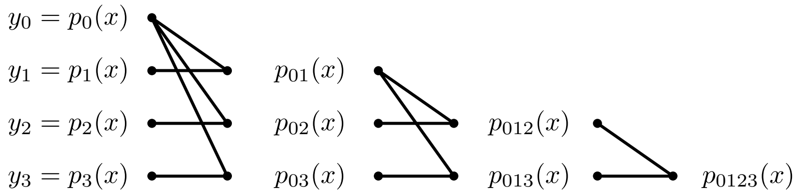
Aufwand: 4 Subtraktionen, 2 Multiplikationen, 1 Division pro "Zwischenpolynom".
Für $n+1$ Knoten benötigen wir $\sum_{j=1}^{n} j = \frac{(n+1)n}{2}$ „Zwischenpolynome", d.h., insgesamt sind also $2(n+1)n$ Additionen/Subtraktionen und $\frac{3}{2}n(n+1)$ Multiplikationen/Divisionen zur Berechnung eines Wertes des Interpolationspolynoms nötig.

MAPLE-Prozedur für den Neville-Algorithmus.

```
> restart;
  with(LinearAlgebra):
  Digits:=20;
  # Neville-Algorithmus
  neville:=proc(x, f, wert)
  # x ist der Vektor der Stuetzstellen
  # f ist der Vektor der Stuetzwerte
  # wert ist die x-Koordinate, an der das Interpolationspolynom ausgewertet werden
  # soll
  # option trace;
  local y, n, j, k;
  n:=Dimension(f);
  y:=Vector(n);
  for j from 1 to n do y[j] := f[j] end do;
  for k from 1 to n-1 do
          for j from n by (-1) to k + 1 do
                  y[j]:=evalf(((wert-x[j-k])*y[j]+(x[j]-wert)*y[j-1])/(x[j]-x[j-k]))
          end do
  end do;
  RETURN(y[n])
  end proc:
```

Bemerkung:

(i) Alternativ kann man die **Methode von Aitken** verwenden, die auf einem unsymmetrischen Tableau basiert:



(ii) Der Neville- und der Aiken-Algorithmus eignen sich nicht gut zur Berechnung der Koeffizienten des Interpolationspolynoms.

7.2 Newton-Darstellung des Interpolationspolynoms

Idee: Wählt man die Monome $\{1, x, \ldots, x^n\}$ als Basis von Π_n, so liefert das Interpolationsproblem (7.1) ein LGS mit der Vandermonde-Matrix $\mathbf{V} = (x_k^j)_{k,j=0}^n$ als Koeffizientenmatrix. Wählt man die Lagrange-Grundpolynome als Basis von Π_n, so liefert das Interpolationsproblem (7.1) ein LGS mit der Einheitsmatrix als Koeffizientenmatrix. $((l_j(x_k))_{k,j=0}^n = \mathbf{I}_{n+1})$.

Gesucht: Basis $\{u_j, j = 0, \ldots, n\}$ so, dass $(u_j(x_k))_{k,j=0}^n$ eine Dreiecksmatrix ist.

Definition 7.4: (Newton-Grundpolynome)

Die **Newton-Grundpolynome** seien rekursiv durch

$$\begin{aligned} u_0(x) &:= 1 \\ u_j(x) &:= (x - x_{j-1})u_{j-1}(x), \quad j = 1, \ldots, n, \end{aligned}$$

bzw. explizit durch $u_j(x) := \prod_{k=0}^{j-1}(x - x_k),\ j = 1, \ldots, n$, definiert.

Wir erhalten

$$u_j(x_k) = \begin{cases} 0, & k < j, \\ \prod_{r=0}^{j-1}(x_k - x_r), & k \geq j. \end{cases}$$

Mit dem Ansatz $p(x) = \sum_{j=0}^n a_j u_j(x) \in \Pi_n$ erhalten wir aus den Interpolationsbedingungen $p(x_k) = y_k, k = 0, \ldots, n$ das LGS

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & x_1 - x_0 & 0 & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 1 & x_n - x_0 & \ldots & \ldots & \Pi_{l=0}^{n-1}(x_n - x_l) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

Die Berechnung der Koeffizienten a_j kann nun durch Vorwärtselimination erfolgen. Wir zeigen, dass die a_j sogenannte dividierte Differenzen sind.

Definition 7.5: (Dividierte Differenz)

Sei $p_j \in \Pi_j$ das Interpolationspolynom zu den paarweise verschiedenen Stützstellen $x_0, \ldots, x_j$

und $p_j(x_k) = y_k$ für $k = 0, \ldots, j$. Dann heißt der Koeffizient der höchsten Potenz x^j von p_j **dividierte Differenz** j**-ter Ordnung**.
Bezeichnung: $[y_0, \ldots, y_j] = [p_j(x_0), \ldots, p_j(x_j)] = p_j[x_0, \ldots, x_j]$.
(Falls für eine Funktion f gilt $f(x_k) = y_k,\ k = 0, \ldots, j$, schreibt man auch $[f(x_0), \ldots, f(x_j)]$ oder $f[x_0, \ldots, x_j]$).

Satz 7.6:

Für die dividierte Differenz $[y_0, \ldots, y_j]$ zu den paarweise verschiedenen Stützstellen x_k und den Stützwerten $y_k, k = 0, \ldots, j$, gilt:

(i) $$[y_0, y_1, \ldots, y_j] = \sum_{r=0}^{j} y_r \prod_{\substack{k=0 \\ k \neq r}}^{j} \frac{1}{x_r - x_k}.$$

(ii) $[y_0, \ldots, y_j]$ ist invariant gegenüber Permutation der Wertepaare $(x_0, y_0), \ldots, (x_j, y_j)$.

(iii) $[y_0, \ldots, y_j]$ lässt sich rekursiv berechnen:

$$[y_i] := y_i, \qquad i = 0, \ldots, j,$$
$$[y_i, \ldots, y_{i+l}] := \frac{[y_{i+1}, \ldots, y_{i+l}] - [y_i, \ldots, y_{i+l-1}]}{x_{i+l} - x_i}, \quad i = 0, 1, \ldots, j-l;\ l = 1, \ldots, j.$$

(iv) Es gilt $[y_0, \ldots, y_j, y_{j+1}] = p_j[x_0, \ldots, x_{j+1}] = 0$, falls $p_j \in \Pi_j$ und $p_j(x_k) = y_k,\ k = 0, \ldots, j+1$.

Beweis:

(i) Aus der Lagrange-Darstellung des Interpolationspolynoms $p_j \in \Pi_j$ mit $p_j(x_k) = y_k$ für $k = 0, \ldots, j$,

$$p_j(x) = \sum_{l=0}^{j} y_l \prod_{\substack{k=0 \\ k \neq l}}^{j} \frac{x - x_k}{x_l - x_k}$$

ergibt sich für den Koeffizienten der höchsten Potenz x^j:

$$a_j = \sum_{l=0}^{j} y_l \prod_{\substack{k=0 \\ k \neq l}}^{j} \frac{1}{x_l - x_k}.$$

Aus Definition 7.5 folgt damit die Behauptung.

(ii) Die Aussage ist klar, da das Interpolationspolynom p_j und damit auch sein Höchstkoeffizient eindeutig bestimmt ist.

(iii) Sei $p \in \Pi_{l-1}$ das Interpolationspolynom durch die l Punkte $(x_r, y_r), r = i, \ldots, i + l - 1$, und $q \in \Pi_{l-1}$ das Interpolationspolynom mit $q(x_r) = y_r,\ r = i+1, \ldots, i+l$. Die Leitkoeffizienten von p und q sind $[y_i, \ldots, y_{i+l-1}]$ und $[y_{i+1} \ldots, y_{i+l}]$. Betrachte das Polynom $w \in \Pi_l$,

$$w(x) := \frac{1}{(x_{i+l} - x_i)}[(x - x_i)q(x) - (x - x_{i+l})p(x)], \qquad x \in \mathbb{R}.$$

Dann folgt

$$w(x_i) = \frac{-(x_i - x_{i+l})}{(x_{i+l} - x_i)} p(x_i) = y_i, \quad w(x_{i+l}) = \frac{x_{i+l} - x_i}{(x_{i+l} - x_i)} q(x_{i+l}) = y_{i+l},$$

und

$$w(x_j) = y_j, \quad j = i+1, \dots, i+l-1.$$

Also ist $[y_i, \dots, y_{i+l}]$ Höchstkoeffizient von $w(x)$. Somit hat der Leitkoeffizient von $w(x)$ die Form $\frac{1}{(x_{i+l}-x_i)}([y_{i+1}, \dots y_{i+l}] - [y_i \dots y_{i+l-1}])$.

(iv) $[y_0, \dots, y_{j+1}]$ ist Höchstkoeffizient des Interpolationspolynoms $p \in \Pi_{j+1}$ mit $p(x_k) = y_k, k = 0, \dots, j+1$. Wegen Satz 7.1 ist jedoch $p = p_j$ und daher der Koeffizient der höchsten Potenz x^{j+1} gleich Null. □

Satz 7.7:

Das Interpolationspolynom $p_n \in \Pi_n$, das die Bedingungen $p_n(x_k) = y_k,\ k = 0, \dots, n$, erfüllt, hat die Darstellung

$$p_n(x) = [y_0] + [y_0, y_1](x - x_0) + \ldots + [y_0, \dots, y_n](x - x_0) \dots (x - x_{n-1}).$$

Beweis:

Sei das Interpolationspolynom p_n in der Form

$$p_n(x) = \sum_{j=0}^{n} a_j u_j(x), \qquad u_j(x) := \prod_{r=0}^{j-1} (x - x_r),$$

gegeben. Dann ist a_n der Höchstkoeffizient von p_n, also $a_n = [y_0, \dots, y_n]$. Wegen $u_n(x_k) = 0,\ k = 0, \dots, n-1$, gilt für das Polynom $p_{n-1}(x) = \sum_{j=0}^{n-1} a_j u_j(x)$ nun

$$p_{n-1}(x_k) = p_n(x_k) - a_n u_n(x_k) = y_k, \quad k = 0, \dots, n-1,$$

d.h., $p_{n-1} \in \Pi_{n-1}$ ist Interpolationspolynom für $(x_k, y_k),\ k = 0, \dots, n-1$.
Also gilt $a_{n-1} = [y_0, \dots, y_{n-1}]$ usw. □

Beispiel:

Sei $n = 2$.

x_k	0	1	3
y_k	1	3	2

$$\begin{array}{r|lllll}
 & k=0 & & k=1 & & k=2 \\
\hline
x_0 = 0 & [y_0] = 1 & & & & \\
 & & \ddots & & & \\
x_1 = 1 & [y_1] = 3 & \dots & [y_0, y_1] = 2 & & \\
 & & \ddots & & \ddots & \\
x_2 = 3 & [y_2] = 2 & \dots & [y_1, y_2] = -\frac{1}{2} & \dots & [y_0, y_1, y_2] = -\frac{5}{6}
\end{array}$$

$$[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{3-1}{1-0} = 2, \quad [y_1, y_2] = \frac{y_2 - y_1}{x_2 - x_1} = \frac{(-1)}{2}, \quad \text{usw..}$$

$$\Rightarrow \quad p_{012}(x) = 1 + 2(x-0) - \tfrac{5}{6}(x-0)(x-1) = 1 + (x-0)\cdot(2 - \tfrac{5}{6}(x-1))$$

$$\Rightarrow \quad p_{012}(2) = 1 + (2-0)\cdot(2 - \tfrac{5}{6}\cdot(2-1)) = \tfrac{10}{3}.$$

Numerische Berechnung:

Unter Verwendung der Newtonschen Interpolationsformel kann das Interpolationspolynom mit einer Art Hornerschema berechnet werden:

$$p_n(x) = [y_0] + (x - x_0)([y_0, y_1] + (x - x_1) \cdot ([y_0, y_1, y_2] + \ldots + \\ + (x - x_{n-1})[y_0, \ldots, y_n]) \ldots))$$

Die dividierten Differenzen werden mittels des Dreiecksschemas rekursiv berechnet:

$$\begin{array}{lllllllll}
x_0 & y_0 = & [y_0] & & & & & & \\
 & & & \ddots & & & & & \\
x_1 & y_1 = & [y_1] & \ldots & [y_0, y_1] & & & & \\
 & & & \ddots & & \ddots & & & \\
x_2 & y_2 = & [y_2] & \ldots & [y_1, y_2] & \ldots & [y_0 y_1 y_2] & & \\
\vdots & & & & \vdots & & & \ddots & \\
x_n & y_n = & [y_n] & \ldots & [y_{n-1}, y_n] & & & \ldots & [y_0, \ldots, y_n] \quad .
\end{array}$$

MAPLE-Prozedur zur Berechnung der Newtonkoeffizienten.

```
> restart;
  with(LinearAlgebra);
  Newtonkoeff:=proc(x, f)
  # x ist der Vektor der Stuetzstellen
  # f ist der Vektor der Stuetzwerte
  local y, n, i, k;
  n:=Dimension(x);
  y:=Vector(n);
  y:=f;
  for k from 1 to n − 1 do
          for i from n by −1 to k + 1 do
                  y[i]:=evalf((y[i] − y[i − 1])/(x[i] − x[i − k]))
          end do
  end do;
  evalm(y);
  end proc:
```

Anwendung:

```
> x:=Vector([0, 1, 3]):
  y:=Vector([1, 3, 2]):
  w:=Newtonkoeff(x, y);
```

ergibt

$$w := [1,\ 2.,\ -.8333333333].$$

Zur Berechnung von Funktionswerten des Interpolationspolynoms verwenden wir das Horner-Schema:

```
> Newtoninterpol:=proc(x, f, z)
  # f ist der Vektor der Stuetzwerte
  # x ist der Vektor der Stuetzstellen
```

```
# z ist der Vektor der Stuetzstellen, fuer den ein Interpolationswert gesucht ist
local Newtonkoeff, y, p, m, n, j, k;
Newtonkoeff:=proc(x, f) ... end proc;
# Berechnung der Funktionswerte des Interpolationspolynoms
y:=Newtonkoeff(x, f);
n:=Dimension(y);
m:=Dimension(z);
p:=Vector(m);
for j from 1 to m do
        p[j] := y[n];
        for k from n - 1 by (-1) to 1 do p[j]:=evalf(y[k] + (z[j] - x[k]) * p[j]) end do;
end do;
evalm(p);
end proc:
```

Anwendung:

```
> n:=100: z:=Vector(n):
  a := 0 : b := 3:
  for j from 1 to n do z[j]:=evalf(a + (j - 1) * (b - a)/(n - 1)) end do:
  p:=Newtoninterpol(x, y, z):
  with(plots):
  listplot([seq([z[t], p[t]],t = 1..n)]);
```

Aufwand für $n+1$ **Knoten:** Zur Berechnung der dividierte Differenzen: benötigen wir $(n+1)n$ Additionen und $(n+1)\frac{n}{2}$ Multiplikationen. Für die Auswertung des Polynoms mit dem Hornerschema brauchen wir $2n$ Additionen und n Multiplikationen pro Polynomwert. Damit ist Newton-Darstellung zur numerischen Auswertung weit günstiger als der Neville-Algorithmus.

Bemerkung:

Wähle die Stützstellen x_k äquidistant, d.h.

$$x_k = x_0 + kh, \quad k = 0, \ldots, n, \quad \text{mit } h > 0.$$

Die Vorwärtsdifferenzen seien durch

$$\Delta^0 y_k := y_k, \quad \Delta^r y_k := \Delta^{r-1} y_{k+1} - \Delta^{r-1} y_k$$

rekursiv definiert, d.h. z.B.

$$\begin{aligned} \Delta^1 y_k &= y_{k+1} - y_k, \\ \Delta^2 y_k &= \Delta^1 y_{k+1} - \Delta^1 y_k = (y_{k+2} - y_{k+1}) - (y_{k+1} - y_k) = y_{k+2} - 2y_{k+1} - y_k. \end{aligned}$$

Dann gilt: $[y_k, \ldots, y_{k+r}] = \frac{1}{h^r r!} \Delta^r y_k$.

Beweis:

Übungsaufgabe. □

Das Interpolationspolynom hat dann die Newtondarstellung

$$\begin{aligned}
p_n(x) &= \sum_{j=0}^{n} \frac{1}{h^j} \frac{\Delta^j y_0}{j!} \cdot \prod_{k=0}^{j-1} (x - x_0 - kh) \\
&\underset{t=\frac{x-x_0}{h}}{=} \sum_{j=0}^{n} \frac{1}{h^j} \frac{\Delta^j y_0}{j!} \prod_{k=0}^{j-1} h(t-k) \\
&= \sum_{j=0}^{n} \Delta^j y_0 \cdot \underbrace{\frac{1}{j!} \cdot t(t-1)\dots(t-j+1)}_{:=\binom{t}{j}} = \sum_{j=0}^{n} \Delta^j y_0 \binom{t}{j}.
\end{aligned}$$

Satz 7.8: (Leibnizregel für dividierte Differenzen)

Ist $f = g \cdot h$ das Produkt zweier Funktionen, so gilt für die dividierte Differenz $f[x_0, \dots, x_n]$ zu den Knoten $x_0 < \dots < x_n$

$$f[x_0, \dots, x_n] = \sum_{i=0}^{n} g[x_0 \dots x_i]\ h[x_i \dots x_n].$$

Beweis:

Setze

$$u_i(x) := \prod_{k=0}^{i-1} (x - x_k), \qquad \tilde{u}_j(x) := \prod_{l=j+1}^{n} (x - x_l).$$

Nach Satz 7.7 hat das Interpolationspolynom $p \in \Pi_n$ von g mit $p(x_k) = g(x_k)$, $k = 0, \dots, n$, die Darstellung

$$p(x) = \sum_{i=0}^{n} g[x_0, \dots, x_i] \cdot u_i(x).$$

Analog folgt für das Interpolationspolynom $q \in \Pi_n$ von h mit $q(x_k) = h(x_k),\ k = 0, \dots, n$,

$$q(x) = \sum_{j=0}^{n} h[x_j, \dots, x_n] \tilde{u}_j(x).$$

Dann interpoliert

$$p(x)q(x) = \sum_{i=0}^{n} \sum_{j=0}^{n} g[x_0, \dots, x_i] h[x_j, \dots, x_n] u_i(x) \tilde{u}_j(x)$$

die Funktion f in $x_0, \dots, x_n$. Aber für $i > j$ ist $u_i(x_k)\tilde{u}_j(x_k) = 0,\ k = 0, \dots, n$. Das Polynom

$$w(x) = \sum_{i=0}^{n} \sum_{j=i}^{n} g[x_0, \dots, x_i] h[x_j, \dots, x_n] \underbrace{u_i(x)}_{\text{Grad}\, i} \ \underbrace{\tilde{u}_j(x)}_{\text{Grad}(n-j)}$$

interpoliert f ebenfalls in $x_0, \dots, x_n$. Da $w \in \Pi_n$, hat $w(x)$ nach Definition 7.5 den Leitkoeffizient

$$f[x_0, \dots, x_n] = \sum_{i=0}^{n} g[x_0, \dots, x_i] h[x_i, \dots, x_n].$$

□

Wir wollen nun für die dividierte Differenz eine Integraldarstellung herleiten:

Satz 7.9: (Integraldarstellung für dividierte Differenzen)

Ist $f \in C^n[x_0, x_n]$, so hat die dividierte Differenz $f[x_0, \dots, x_n]$ zu den Knoten $x_0 < \dots < x_n$ die Integraldarstellung

$$f[x_0, \dots, x_n] = \int_{x_0}^{x_n} f^{(n)}(t) G_{n-1}(t) dt$$

mit dem **Peano-Kern**

$$G_{n-1}(t) := \sum_{k=0}^{n} w_k \cdot \frac{(x_k - t)_+^{n-1}}{(n-1)!}, \quad t \in \mathbb{R},$$

wobei

$$w_k := \prod_{\substack{l=0 \\ l \neq k}}^{n} \frac{1}{x_k - x_l},$$

und $x_+^n := \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases}$ die so genannte abgeschnittene Potenz (truncated power function) ist.

Beweis:

Für $f \in C^n[x_0, x_n]$ und $x \in [x_0, x_n]$ gilt die Taylorformel

$$\begin{aligned} f(x) &= \underbrace{\sum_{j=0}^{n-1} \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j}_{:= p_{n-1}(x)} + \underbrace{\int_{x_0}^{x} \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t) dt}_{\text{Restglied}} \\ &= p_{n-1}(x) + \int_{x_0}^{x_n} \frac{(x-t)_+^{n-1}}{(n-1)!} f^{(n)}(t) dt. \end{aligned}$$

Nach Satz 7.6 (i) war

$$\begin{aligned} f[x_0, \dots, x_n] &= \sum_{k=0}^{n} w_k f(x_k) \quad \text{mit } w_k = \prod_{\substack{l=0 \\ l \neq k}}^{n} \frac{1}{x_k - x_l} \\ \Rightarrow f[x_0, \dots, x_n] &= \sum_{k=0}^{n} w_k \left[p_{n-1}(x_k) + \int_{x_0}^{x_n} \frac{(x_k - t)_+^{n-1}}{(n-1)!} f^{(n)}(t) dt \right]. \end{aligned}$$

Wegen $p_{n-1} \in \Pi_{n-1}$ gilt mit Satz 7.6 (i) und (iv)

$$\sum_{k=0}^{n} w_k p_{n-1}(x_k) = p_{n-1}[x_0, \dots, x_n] = 0.$$

Daraus folgt

$$\begin{aligned} f[x_0, \dots, x_n] &= \sum_{k=0}^{n} w_k \int_{x_0}^{x_n} \frac{(x_k - t)_+^{n-1}}{(n-1)!} f^{(n)}(t) dt \\ &= \int_{x_0}^{x_n} f^{(n)}(t) \cdot \underbrace{\sum_{k=0}^{n} w_k \cdot \frac{(x_k - t)_+^{n-1}}{(n-1)!}}_{G_{n-1}(t)} dt. \end{aligned}$$

$\square$

Beispiel:

Für $n = 1$ erhalten wir den Peano-Kern

$$G_0(t) = \frac{1}{(x_0 - x_1)} \frac{(x_0 - t)_+^0}{0!} + \frac{1}{(x_1 - x_0)} \frac{(x_1 - t)_+^0}{0!} = \begin{cases} (x_1 - x_0)^{-1} & t \in (x_0, x_1], \\ 0 & \text{sonst.} \end{cases}$$

Für $n = 2$ finden wir

$$\begin{aligned} G_1(t) &= \frac{(x_0 - t)_+^1}{(x_0 - x_1)(x_0 - x_2)} + \frac{(x_1 - t)_+^1}{(x_1 - x_0)(x_1 - x_2)} + \frac{(x_2 - t)_+^1}{(x_2 - x_0)(x_2 - x_1)} \\ &= \begin{cases} \frac{(t - x_0)}{(x_2 - x_0)(x_2 - x_1)} & t \in (x_0, x_1], \\ \frac{(-t + x_2)}{(x_2 - x_0)(x_2 - x_1)} & t \in (x_1, x_2], \\ 0 & \text{sonst.} \end{cases} \end{aligned}$$

7.3 Interpolationsfehler

Sei f eine Funktion und p_n das zugehörige Interpolationspolynom, d.h.,

$$f(x_k) = y_k = p_n(x_k), \; k = 0, \ldots, n.$$

Frage: Wie groß ist der Interpolationsfehler

$$r_n(x) := f(x) - p_n(x) \;?$$

Satz 7.10:

Ist f in einem Intervall I, das $x, x_0, \ldots, x_n$ enthält, $(n+1)$-mal stetig differenzierbar, so gilt

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} u_{n+1}(x),$$

wobei $u_{n+1}(x) := (x - x_0) \cdot \ldots \cdot (x - x_n)$ und $\xi_x \in I$ eine Zwischenstelle ist.

Beweis:

1) Für $x = x_k$ ist $f(x) - p_n(x) = 0$ und die Behauptung gilt.
2) Sei $x \neq x_k \; \forall \, k = 0, \ldots, n$, fest. Wir betrachten die Funktion

$$g_x(t) = g(t) := f(t) - p_n(t) - \left(\frac{f(x) - p_n(x)}{u_{n+1}(x)} \right) \cdot u_{n+1}(t).$$

Dann hat die Funktion g im Intervall I mindestens die Nullstellen $x_0, \ldots, x_n, x$, und $g \in C^{n+1}(I)$. Wir wenden den Satz von Rolle $(n+1)$-mal an. Dann existiert ein $\xi_x \in I$, so dass $g^{(n+1)}(\xi_x) = 0$ gilt. Aus

$$g^{(n+1)}(t) = f^{(n+1)}(t) - \underbrace{p_n^{(n+1)}(t)}_{=0} - \left(\frac{f(x) - p_n(x)}{u_{n+1}(x)} \right) \cdot (n+1)!$$

folgt

$$\begin{aligned} & g^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - \left(\frac{f(x) - p_n(x)}{u_{n+1}(x)} \right) (n+1)! = 0 \\ \Leftrightarrow \quad & \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \cdot u_{n+1}(x) = f(x) - p_n(x). \end{aligned}$$

$\square$

Da die Lage von ξ_x nicht bekannt ist, schätzen wir $f^{(n+1)}(\xi_x)$ durch die Maximumnorm $\|f^{(n+1)}\|_\infty$ ab.

Folgerung 7.11:

Unter den Voraussetzungen von Satz 7.10 gilt die Abschätzung

$$|f(x) - p_n(x)| \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} |u_{n+1}(x)| \quad \forall\, x \in I,$$

wobei

$$\|f^{(n+1)}\|_\infty := \max_{t \in I} |f^{(n+1)}(t)|.$$

Eine andere Fehlerdarstellung erhält man mit Hilfe dividierter Differenzen. Die Voraussetzung der Differenzierbarkeit ist hier nicht notwendig!

Satz 7.12:

Bei der Interpolation von $f \in C^1(I)$ durch ein Polynom $p_n \in \Pi_n$ in den paarweise verschiedenen Stützstellen $x_k, k = 0, \ldots, n$ gilt

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x]\; u_{n+1}(x).$$

Beweis:

1) Sei zunächst $x \in I$ mit $x \neq x_k, k = 0, \ldots, n$, fest. Es war

$$p_n(t) = f[x_0] + f[x_0, x_1](t - x_0) + \ldots + f[x_0, x_1, \ldots, x_n](t - x_0) \ldots (t - x_{n-1}).$$

Betrachte nun das Interpolationspolynom für die Stützstellen $x_0, \ldots, x_n, x$ und die Stützwerte $f(x_0), \ldots, f(x_n), f(x)$,

$$\begin{aligned} p_{n+1}(t) \;\; = \;\; & f[x_0] + f[x_0, x_1](t - x_0) + \ldots + f[x_0, \ldots, x_n](t - x_0) \ldots (t - x_{n-1}) \\ & + f[x_0, \ldots, x_n, x](t - x_0) \ldots (t - x_{n-1})(t - x_n). \end{aligned}$$

Dann ergibt sich

$$p_{n+1}(t) - p_n(t) = f[x_0, \ldots, x_n, x](t - x_0) \ldots (t - x_n).$$

Setze nun $t = x$. Dann folgt $p_{n+1}(x) = f(x)$, und damit erhalten wir

$$f(x) - p_n(x) = f[x_0, \ldots, x_n, x] \cdot \underbrace{(x - x_0) \ldots (x - x_n)}_{u_{n+1}(x)}.$$

2) Für $x = x_k,\ k = 0, \ldots, n$, lässt sich $f[x_0, \ldots, x_n, x]$ folgendermaßen definieren:

$$f[x_k, x_k] = \lim_{x \to x_k} \frac{f(x) - f(x_k)}{x - x_k} = f'(x_k),$$

sonst, wie bisher

$$f[x_i, \ldots, x_{i+l}] = \frac{f[x_i \ldots x_{i+l-1}] - f[x_{i+1} \ldots x_{x+l}]}{x_i - x_{i+l}}.$$

Die Behauptung folgt nun mit $u_{n+1}(x_k) = 0, \quad k = 0, \ldots, n.$ □

Durch Vergleich mit der Cauchy-Darstellung des Interpolationsfehlers finden wir

Folgerung 7.13:

Ist $f \in C^n[x_0, x_n]$, so hat die dividierte Differenz zu den Knoten $x_0 < \ldots < x_n$ die Darstellung

$$f[x_0, \ldots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$$

für ein $\xi \in [x_0, x_n]$.

Optimale Stützstellenwahl

Problem: Wie können wir durch günstige Wahl der Stützstellen den Interpolationsfehler $f(x) - p(x)$ minimieren?
Erinnerung: Es war $|f(x) - p(x)| \leq \frac{\|f^{n+1}\|_\infty}{(n+1)!} |u_{n+1}(x)|$.

Idee: Wähle $x_0, \ldots, x_n \in [a, b]$ so, dass $\max_{x \in [a,b]} |u_{n+1}(x)|$ möglichst klein wird.

Beispiel:

Sei $[a, b] = [-1, 1]$ mit äquidistanter Knotenfolge $x_0 = -1, x_1 = -1 + \frac{2}{n}, \ldots, x_n = 1$, d.h. $x_k = -1 + \frac{2k}{n}$ für $k = 1, \ldots, n$. Sei $x \in [x_j, x_{j+1}]$.
1) Für $k < j$: $\underbrace{x_j - x_k}_{2(j-k)/n} \leq x - x_k \leq \underbrace{x_{j+1} - x_k}_{2(j+1-k)/n}$.
2) Für $k \geq j + 1$: $\underbrace{x_k - x_j + 1}_{2(k-j-1)/n} \leq |x_k - x| \leq \underbrace{x_k - x_j}_{2(k-j)/n}$.
Daraus folgt einerseits

$$\begin{aligned} |u_{n+1}(x)| &\leq \prod_{k=0}^{j-1} 2\frac{(j+1-k)}{n} \cdot \prod_{k=j+2}^{n} 2\frac{(k-j)}{n} \cdot |x_j - x||x_{j+1} - x| \\ &= (\frac{2}{n})^{n-1} \cdot (j+1)! \cdot (n-j)! |x - x_j||x_{j+1} - x| \end{aligned}$$

und andererseits

$$|u_{n+1}(x)| \geq (\frac{2}{n})^{n-1} j!(n-j-1)! |x - x_j||x_{j+1} - x|.$$

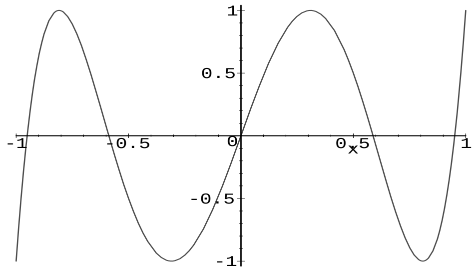
Daher kann $|u_{n+1}(x)|$ für x in der Nähe des Intervallrandes recht groß werden, z.B.

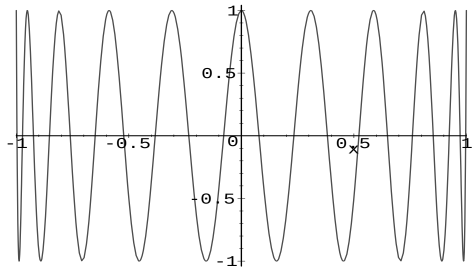
$$x = \frac{x_{n+1} + x_n}{2} \Rightarrow |u_{n+1}(x)| \geq (\frac{2}{n})^{n-1} \cdot (n-1)! \cdot \frac{1}{n} \cdot \frac{1}{n}$$

$$\begin{aligned} n = 5: \quad & |u_6(0,825)| &= \; & 0.113462959 \approx \max_{x \in [-1,1]} |u_6(x)|, \\ n = 6: \quad & |u_7(0.865)| &= \; & 0.0692257 \approx \max_{x \in [-1,1]} |u_7(x)|. \end{aligned}$$

Ist eine andere Knotenwahl günstiger? Ja!
Wir wählen die Stützstellen $x_k := \cos \frac{(2k+1)\pi}{2(n+1)}, \; k = 0, \ldots, n.$
Dann ist

$$T_{n+1}(x) := 2^n \prod_{k=0}^{n} (x - \cos \frac{(2k+1)\pi}{2(n+1)})$$





Tschebyscheff-Polynome vom Grad 5 (links) und vom Grad 20 (rechts)

das **Tschebyscheff-Polynom** vom Grad $n+1$.

Satz 7.14:

Das Tschebyscheff-Polynom lässt sich für $x \in [-1,1]$ in der Form

$$T_n(x) = \cos n(\arccos x), \quad n = 0,1,2,\ldots$$

darstellen, d.h. $T_n(\cos\varphi) = \cos(n\varphi)$. $\quad (x := \cos\varphi \in [-1,1]$ mit $\varphi \in [0,\pi))$.
Die Tschebyscheff-Polynome genügen der Rekursionsformel

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1$$

mit $T_0(x) = 1, T_1(x) = x$.

Beweis:

Sei $\tilde{T}_n(x) := \cos n(\arccos x)$ für $x \in [-1,1]$, d.h., $\tilde{T}_n(\cos\varphi) = \cos(n\varphi)$ für $x := \cos\varphi$. Insbesondere folgt $\tilde{T}_0(x) = \cos 0 = 1$, und $\tilde{T}_1(x) = x$. Da sich $\cos(n\varphi)$ als Linearkombination von $\cos^j\varphi,\ j = 0,\ldots,n$, darstellen lässt, ist $\tilde{T}_n(x) \in \Pi_n$. Es gilt

$$\cos(n\varphi) = \cos((n-1)\varphi + \varphi) = \cos(n-1)\varphi\cos\varphi - \sin(n-1)\varphi\sin\varphi.$$

Außerdem gilt $\sin\alpha\,\sin\beta = \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)]$. Damit folgt

$$\sin(n-1)\varphi\sin\varphi = \frac{1}{2}\cos(n-2)\varphi - \frac{1}{2}\cos n\varphi$$

Einsetzen ergibt

$$\frac{1}{2}\cos n\varphi = \cos\varphi\cos(n-1)\varphi - \frac{1}{2}\cos(n-2)\varphi.$$

Also erfüllt $\tilde{T}_n$ die Rekursionsformel und es gilt $\tilde{T}_n \in \Pi_n$. Weiter erhalten wir

$$\begin{aligned}\tilde{T}_n(\cos\tfrac{(2k+1)\pi}{2n}) &= \cos n(\arccos(\cos\tfrac{(2k+1)\pi}{2n})) = \cos(n\tfrac{(2k+1)\pi}{2n}) \\ &= \cos\tfrac{(2k+1)\pi}{2} = 0.\end{aligned}$$

Das bedeutet $\tilde{T}_n = T_n$. Der Höchstkoeffizient 2^{n-1} folgt aus der Rekursionsformel. $\square$

Beachte: Bei Knotenwahl $x_k = \cos\frac{(2k+1)\pi}{2(n+1)}, \quad k = 0,\ldots,n$, ist $u_{n+1}(x) = \frac{T_{n+1}(x)}{2^n}$.

Satz 7.15:

Das Polynom $2^{-n}T_{n+1}(x)$ ist vom Grad $n+1$ mit höchstem Koeffizienten Eins, und es gilt

$$2^{-n} = \max_{x\in[-1,1]} \frac{|T_{n+1}(x)|}{2^n} \le \max_{x\in[-1,1]} |u_{n+1}(x)|$$

für jedes Polynom $u_{n+1}(x)$ vom Grad $n+1$ mit höchstem Koeffizient 1.

Beweis:

Das Polynom $T_{n+1}(x)$ hat den Höchstkoeffizient 2^n. Dies folgt induktiv aus der Rekursionsformel in Satz 7.14.

Aus der Definition von $T_{n+1}(x)$ folgt $|T_{n+1}(x)| \le 1$ für $x \in [-1,1]$. Die Extremalwerte werden an den $n+2$ Punkten $x_k = \cos\frac{k\pi}{n+1}, k = 0,\ldots,n+1$, angenommen:

$$T_{n+1}(\cos\frac{k\pi}{n+1}) = \cos\left((n+1)\frac{k\pi}{(n+1)}\right) = \cos(k\pi) = \begin{cases} 1 & , k \text{ gerade} \\ -1 & , k \text{ ungerade.} \end{cases}$$

Sei $u_{n+1} \in \Pi_{n+1}$ beliebig mit Höchstkoeffizient 1. Angenommen $|u_{n+1}(x)| < 2^{-n}\ \forall\ x \in [-1,1]$. Dann ist $p(x) = u_{n+1}(x) - \frac{1}{2^n}T_{n+1}(x)$ alternierend positiv bzw. negativ an den Stellen $\frac{k\pi}{n+1}$, $k = 0,\ldots,n+1$, hat also mindestens $n+1$ Nullstellen. Da aber $p \in \Pi_n$ folgt damit $p \equiv 0$. Dies ist ein Widerspruch zur Annahme. □

Beispiel:

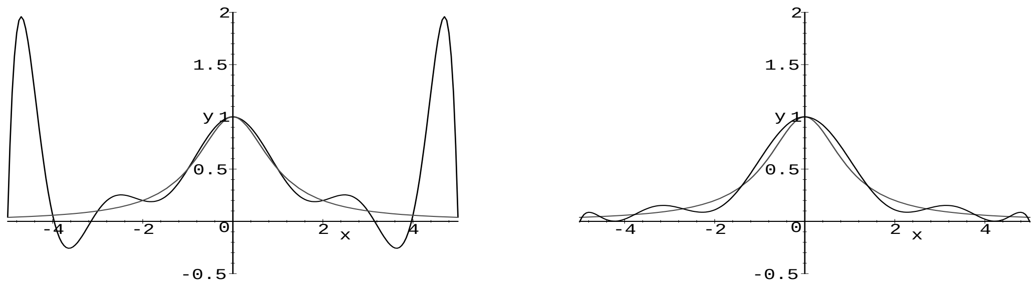
Für die Knotenwahl $x_k = \cos\frac{(2k+1)\pi}{2(n+1)}$, $k = 0,\ldots,n$ erhalten wir nun

$$\begin{array}{lcl} n=5: & \max_{x\in[-1,1]}|u_6(x)| = & 2^{-5} = 0.03125, \\ n=6: & \max_{x\in[-1,1]}|u_7(x)| = & 2^{-6} = 0.015625 \end{array}$$

Beispiel: (Runge)

Es sei $f(x) = \frac{1}{x^2+1}$. Wir betrachten die Interpolation auf dem Intervall $[-5,5]$.

1. Äquidistante Knoten: $x_k = -5 + \frac{10k}{n}$, $\quad k = 0,\ldots,n$, $\Rightarrow$ Divergenz für $n\to\infty$.
2. Tschebyscheff-Knoten: $x_k = 5\cos\frac{(2k+1)\pi}{2(n+1)}$, $\quad k = 0,\ldots,n$, $\Rightarrow$ Konvergenz für $n\to\infty$.



Interpolation für äquidistante Knoten (links) und Tschebyscheff-Knoten (rechts) für $n=10$.

Einfluss von Störungen der Funktionswerte

Die Stützwerte $y_k, k = 0, \dots, n$, seien fehlerbehaftet. Es gelte:

$$\tilde{y}_k = y_k + \varepsilon_k, \quad |\varepsilon_k| < \varepsilon \text{ für } k = 0, \dots, n.$$

Das mit den korrekten Stützwerten y_k berechnete Interpolationspolynom p sowie das mit $\tilde{y}_k$ berechnete Polynom $\tilde{p}$ haben folgende Lagrangedarstellung

$$p = \sum_{k=0}^{n} y_k l_k, \qquad \tilde{p} = \sum_{k=0}^{n} \tilde{y}_k l_k.$$

Sei

$$\eta = \tilde{p} - p = \sum_{k=0}^{n} (\tilde{y}_k - y_k)\, l_k.$$

Dann folgt

$$|\eta(x)| \leq \sum_{k=0}^{n} |\varepsilon|\; |l_k(x)| \leq \varepsilon \cdot L_n(x),$$

wobei $L_n(x) := \sum_{k=0}^{n} |l_k(x)|$. Wir erhalten für $a = x_0 < x_1 \dots < x_n = b$

$$\max_{x \in [a,b]} |\tilde{p}(x) - p(x)| = \max_{x \in [a,b]} |\eta(x)| \leq \varepsilon \cdot \underbrace{\max_{x \in [a,b]} L_n(x)}_{=:\Lambda_n}.$$

Die Konstante $\Lambda_n := \max_{x \in [a,b]} L_n(x)$ ist nur abhängig von den Stützstellen $x_0 < x_1 \dots < x_n$. Sie ist ein Maß dafür, wieweit sich der Eingangsfehler verstärkend auf das Endresultat auswirkt.

Beispiel:

Sei $[a, b] = [-1, 1]$.

n	Λ_n für äquidistante Knoten	Λ_n für Tscheyscheff-Knoten
5	$3,106292$	$2,104398$
10	$29,890695$	$2,489430$
15	$512,052451$	$2,727778$
20	$10986,533993$	$2,900825$

(Λ_n divergiert für beide Knotensequenzen gegen ∞, jedoch unterschiedlich schnell!)

Kapitel 8

Numerische Integration

8.1 Einführung

Bei der Berechnung von bestimmten Integralen ist man im Allgemeinen auf numerische Näherungen angewiesen, wenn keine analytische, geschlossene Darstellung dieser Integrale möglich ist.

Definition 8.1: (Quadraturformel)

Unter einer **Quadraturformel** Q zur Berechnung des bestimmten Integrals $I(f) := \int_a^b f(x)\,dx$ für $f \in C[a,b]$ verstehen wir die Summe

$$Q(f) := \sum_{j=0}^{n} a_j f(x_j), \qquad a_j \in \mathbb{R}, x_j \in [a,b],$$

mit den **Knoten** $x_0, \ldots, x_n$ und den **Gewichten** $a_0, \ldots, a_n$. Die Differenz

$$R(f) := \int_a^b f(x)\,dx - Q(f)$$

bezeichnet man als **Quadraturfehler**. Wenn $R(p) = 0$ für alle Polynome $p \in \Pi_d$ gilt, so heißt Q **exakt** auf der Menge der Polynome Π_d vom Höchstgrad d.

Beispiel:

$$\begin{array}{lll} n = 0: & Q(f) = f(a)\,(b-a) & \text{Rechteckregel} \\ & Q(f) = f(\frac{a+b}{2})\,(b-a) & \text{Mittelpunktregel} \\ n = 1: & Q(f) = (f(a) + f(b))\,\frac{b-a}{2} & \text{Trapezregel} \end{array}$$

Idee: Zur Berechnung von $\int_A^B f(x)\,dx$ wende man eine einfache Quadraturformel auf möglichst vielen Teilintervallen von $[A,B]$ an:

Wähle eine Zerlegung $A = z_0 < z_1 < z_2 < \ldots < z_N = B$ und wende die Quadraturformel auf jedem Teilintervall $[z_j, z_{j+1}]$, $j = 0, \ldots, N-1$, an:

a) zusammengesetzte Rechteckregel: $I(f) \approx \sum_{j=0}^{N-1} f(z_j)\,(z_{j+1} - z_j)$

b) zusammengesetzte Mittelpunktregel: $I(f) \approx \sum_{j=0}^{N-1} f(\frac{z_j + z_{j+1}}{2})\,(z_{j+1} - z_j)$

c) zusammengesetzte Trapezregel:

$$
\begin{aligned}
I(f) &\approx \sum_{j=0}^{N-1} \frac{1}{2}(f(z_j) + f(z_{j+1}))\,(z_{j+1} - z_j) \\
&= f(z_0)\,(\frac{z_1 - z_0}{2}) + \sum_{j=1}^{N-1} f(z_j)\,(\frac{z_{j+1} - z_{j-1}}{2}) + f(z_N)\,(\frac{z_N - z_{N-1}}{2}).
\end{aligned}
$$

Wählt man die Zerlegung des Intervalls $[A, B]$ äquidistant, d.h. $z_j = A + j\,h$, mit $h = \frac{B-A}{N}$, $j = 0, \ldots, N$, so folgt z.B. für die zusammengesetzte Mittelpunktregel:

$$
I(f) \approx \sum_{j=0}^{N-1} f(\frac{A + hj + A + h(j+1)}{2})\,h = h \sum_{j=0}^{N-1} f(A + \frac{2j+1}{2}h).
$$

Um verschiedene Quadraturformeln miteinander vergleichen zu können, benötigen wir eine Darstellung des Quadraturfehlers $R(f) = I(f) - Q(f)$. Dazu betrachten wir im Folgenden ein "kleines" Teilintervall $[a, b]$.

Satz 8.2:

Der Fehler $R(f)$ einer $(n+1)-$punktigen Quadraturformel vom Exaktheitsgrad d

$$
\int_a^b f(x)\,dx = \sum_{j=0}^{n} a_j f(x_j) + R(f), \quad a \le x_0 < \ldots < x_n \le b,
$$

hat für $f \in C^{m+1}[a, b]$ und $0 \le m \le d$ die Darstellung

$$
R(f) = \int_a^b f^{(m+1)}(t) G_m(t)\,dt,
$$

wobei

$$
G_m(t) := \frac{1}{m!} R_x[(x-t)_+^m] := \frac{1}{m!}(\int_a^b (x-t)_+^m dx - \sum_{j=0}^{n} a_j (x_j - t)_+^m).
$$

Hierbei bedeutet $R_x((x-t)_+^m)$, dass R auf die Funktion $(x-t)_+^m$ als Funktion in x anzuwenden ist. G_m heißt **Peano-Kern** von R.

Beweis:

Sei $f \in C^{m+1}[a, b]$ gegeben. Eine Taylor-Entwicklung von f in a ergibt

$$
f(x) = \underbrace{\sum_{j=0}^{m} \frac{1}{j!} f^{(j)}(a)\,(x-a)^j}_{\text{Polynom vom Grad } m \le d} + r_m(x)
$$

mit dem Restglied

$$
r_m(x) = \frac{1}{m!} \int_a^x f^{(m+1)}(t)(x-t)^m\,dt = \frac{1}{m!} \int_a^b f^{(m+1)}(t)(x-t)_+^m\,dt.
$$

Wegen $R(p) = 0$ für alle $p \in \Pi_d$ und $m \le d$ folgt

$$
R(f) = R(r_m).
$$

Nun ist

$$\begin{aligned}
R(r_m) &= \int_a^b r_m(x)\,dx - \sum_{j=0}^n a_j r_m(x_j) \\
&= \frac{1}{m!}\int_a^b\int_a^b f^{(m+1)}(t)(x-t)_+^m\,dtdx - \sum_{j=0}^n \frac{a_j}{m!}\int_a^b f^{(m+1)}(t)(x_j-t)_+^m\,dt \\
&= \frac{1}{m!}\int_a^b f^{(m+1)}(t)\cdot\left(\int_a^b (x-t)_+^m dx - \sum_{j=0}^n a_j(x_j-t)_+^m\right)dt \\
&= \frac{1}{m!}\int_a^b f^{(m+1)}(t)\cdot R_x[(x-t)_+^m]\,dt.
\end{aligned}$$

□

Folgerung 8.3:

Ist $f \in C^{m+1}[a,b]$, so ist

$$|R(f)| \le \max_{x\in[a,b]} |f^{(m+1)}(x)| \int_a^b |G_m(t)|\,dt.$$

Beispiel:

Die Mittelpunktregel: $\int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2}) = Q(f)$ hat den Exaktheitsgrad $d=1$, denn für $p(x) = mx+n$ ist

$$\begin{aligned}
I(p) = \int_a^b (mx+n)dx &= \left[\frac{mx^2}{2}+nx\right]_a^b = m\frac{(b^2-a^2)}{2} + n(b-a) \\
&= (b-a)(m\Big(\frac{b+a}{2}\Big)+n) = (b-a)p(\frac{a+b}{2}) = Q(p).
\end{aligned}$$

Wähle $m=0$. Dann erhalten wir den Peano-Kern

$$\begin{aligned}
G_0(t) &= \int_a^b (x-t)_+^0\,dx - (b-a)(\frac{a+b}{2}-t)_+^0 \\
&= \begin{cases} (b-t)-(b-a) = a-t, & a \le t \le \frac{a+b}{2} \\ (b-t)-0, & \frac{a+b}{2} < t \le b \end{cases}.
\end{aligned}$$

Für $f \in C^1[a,b]$ folgt

$$|R(f)| \le \int_a^b |f'(t)||G_0(t)|\,dt \le \max_{x\in[a,b]} |f'(x)| \underbrace{\int_a^b |G_0(t)|\,dt}_{\frac{(b-a)^2}{4}}.$$

Wähle nun $m = 1$:

$$\begin{aligned}
G_1(t) &= \int_a^b (x-t)_+^1 dx - (b-a)(\frac{a+b}{2} - t)_+^1 \\
&= \begin{cases} \int_t^b (x-t)dx - (b-a)(\frac{a+b}{2} - t), & a \le t \le \frac{a+b}{2} \\ \int_t^b (x-t)dx, & \frac{a+b}{2} < t \le b \end{cases} \\
&= \begin{cases} \frac{(a-t)^2}{2}, & a \le t \le \frac{a+b}{2} \\ \frac{(b-t)^2}{2}, & \frac{a+b}{2} < t \le b \end{cases}
\end{aligned}$$

da

$$\int_t^b (x-t)dx = \left[\frac{(x-t)^2}{2}\right]_t^b = \frac{(b-t)^2}{2}.$$

Dann gilt aus Symmetriegründen

$$\int_a^b |G_1(t)|\, dt = 2\int_a^{\frac{a+b}{2}} \frac{(t-a)^2}{2} dt = 2\left[\frac{(t-a)^3}{6}\right]_a^{\frac{a+b}{2}} = 2\frac{(b-a)^3}{48} = \frac{(b-a)^3}{24},$$

und damit folgt für $f \in C^2[a,b]$ also

$$|R(f)| \le \max_{x\in[a,b]} |f''(x)| \int_a^b |G_1(t)|\, dt = \frac{(b-a)^3}{24} \max_{x\in[a,b]} |f''(x)|.$$

Für die zusammengesetzte Mittelpunktregel folgt daraus für $f \in C^2[A,B]$

$$\begin{aligned}
|I(f) - Q(f)| &= |\int_A^B f(x)\, dx - \sum_{j=0}^{N-1} f(A + \frac{2j+1}{2} h)\, h| \\
&\le \sum_{j=0}^{N-1} |\int_{A+jh}^{A+(j+1)h} f(x)\, dx - f(A + \frac{2j+1}{2} h)\, h| \\
&\le \sum_{j=0}^{N-1} \max_{x\in[A+jh, A+(j+1)h]} |f''(x)|\, \frac{h^3}{24} \\
&\le \max_{x\in[A,B]} |f''(x)|\, n\, \frac{(B-A)^3}{24N^3} = \mathcal{O}(\frac{1}{N^2}).
\end{aligned}$$

8.2 Interpolatorische Quadraturformeln

Definition 8.4: (interpolatorische Quadraturformel)

Eine $(n+1)-$punktige Quadraturformel $Q(f) := \sum_{j=0}^n a_j f(x_j)$ heißt **interpolatorisch**, wenn Q auf der Menge der Polynome Π_n exakt ist.

Satz 8.5:

Zu beliebig vorgegebenen Stützstellen $a \le x_0 < x_1 < \ldots < x_n \le b$ existiert genau eine interpolatorische Quadraturformel $Q(f) = \sum_{j=0}^{n} a_j f(x_j)$. Ihre Gewichte erhält man durch Integration der Lagrange-Grundpolynome $l_j, j = 0, \ldots, n$, d.h.

$$a_j = \int_a^b l_j(x)\, dx \quad \text{mit} \quad l_j(x) := \prod_{\substack{k=0\\k\neq j}}^{n} \frac{(x-x_k)}{(x_j-x_k)}.$$

(Idee: Interpoliere f durch $p_n \in \Pi_n$ und integriere dann p_n exakt statt f.)

<u>**Beweis:**</u>

1) Existenz: Da jedes Polynom $p \in \Pi_n$ eindeutig in der Lagrangeform $p(x) = \sum_{j=0}^{n} p(x_j) l_j(x)$ darstellbar ist (siehe Satz 7.3), folgt

$$\int_a^b p(x)\, dx = \int_a^b \sum_{j=0}^{n} p(x_j) l_j(x)\, dx = \sum_{j=0}^{n} p(x_j) \underbrace{\int_a^b l_j(x) dx}_{=a_j} = Q(p).$$

2) Eindeutigkeit: Seien $Q(f)$ und $\tilde{Q}(f) = \sum_{j=0}^{n} b_j f(x_j)$ zu den Stützstellen $a \leq x_0 < \ldots < x_n \leq b$ interpolatorisch. Dann gilt für alle $p \in \Pi_n$: $Q(p) - \tilde{Q}(p) = 0$. Das bedeutet $\sum_{j=0}^{n} (a_j - b_j) p(x_j) = 0$ für alle $p \in \Pi_n$. Wählen wir $p_k(x) := x^k, k = 0, \ldots, n$, so erhalten wir das LGS

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_n \\ x_0^2 & x_1^2 & \ldots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 - b_0 \\ a_1 - b_1 \\ \vdots \\ a_n - b_n \end{bmatrix} = \mathbf{0}.$$

Da die Koeffizientenmatrix invertierbar ist, folgt $a_j = b_j,\ j = 0, \ldots, n$. □

<u>Bemerkung:</u>

Für äquidistante Stützstellen $x_j = a + h\,j,\ j = 0, \ldots, n,\ h = \frac{b-a}{n}$, heißen die interpolatorischen Quadraturformeln **Newton-Cotes-Formeln**. Wir erhalten in diesem Fall mit der Substitution $t = \frac{x-a}{h}$

$$\begin{aligned} a_j &= a_{j,n} = \int_a^b \prod_{\substack{k=0\\k\neq j}}^{n} \frac{x-x_k}{x_j-x_k}\, dx = \int_a^b \prod_{\substack{k=0\\k\neq j}}^{n} (\frac{x-a-hk}{hj-hk})\, dx \\ &= \int_0^n \prod_{\substack{k=0\\k\neq j}}^{n} \frac{h(t-k)}{h(j-k)} h\, dt = \frac{h}{\prod\limits_{\substack{k=0\\k\neq j}}^{n} (j-k)} \int_0^n \prod_{\substack{k=0\\k\neq j}}^{n} (t-k)\, dt \\ &= \frac{h(-1)^{n-j}}{j!(n-j)!} \int_0^n \prod_{\substack{k=0\\k\neq j}}^{n} (t-k)\, dt. \end{aligned} \tag{8.1}$$

<u>Beispiel:</u>

Für $n = 1$ erhält man die Trapezregel: $x_0 = a, x_1 = b$,

$$Q(f) = a_0 f(x_0) + a_1 f(x_1) = a_0 f(a) + a_1 f(b)$$

mit

$$\begin{aligned} a_0 &= \int_a^b l_0(x)\,dx = \int_a^b \frac{(x-b)}{(a-b)}\,dx = \left[\frac{\frac{x^2}{2}-bx}{a-b}\right]_a^b = \frac{b-a}{2}, \\ a_1 &= \int_a^b l_1(x)\,dx = \frac{b-a}{2}. \end{aligned}$$

Satz 8.6:

Der Quadraturfehler für die $(n+1)-$punktige interpolatorische Quadraturformel hat die Form

$$R(f) = \int_a^b f(x)\,dx - \int_a^b p_n(x)\,dx = \int_a^b f[x_0,\ldots,x_n,x]\,u_{n+1}(x)\,dx$$

mit $u_{n+1}(x) = (x-x_0)(x-x_1)\cdot\ldots\cdot(x-x_n)$, wobei $p_n \in \Pi_n$ das Interpolationspolynom von f zu den Stützstellen $x_0 < x_1 < \ldots < x_n$ ist.

Beweis:

Für die Polynom-Interpolation war nach Satz 7.11:

$$f(x) - p_n(x) = f[x_0,\ldots,x_n,x]u_{n+1}(x),$$

woraus die Behauptung folgt. □

Beispiel:

1. Für $n=1$ folgt wegen $(x-a)(x-b) \le 0 \quad \forall\, x \in [a,b]$

$$\begin{aligned} |R(f)| &= |\int_a^b f[a,b,x](x-a)(x-b)\,dx| \\ &\le \max_{\eta\in[a,b]} |f[a,b,\eta]| \int_a^b |(x-a)(x-b)|\,dx = \max_{\eta\in[a,b]} f[a,b,\eta]\,\frac{(b-a)^3}{6}. \end{aligned}$$

   Für $f \in C^2[a,b]$ war $f[a,b,\eta] = \frac{1}{2}f''(\xi)$ für eine Zwischenstelle $\xi \in [a,b]$ (siehe Folgerung 7.12). Damit erhalten wir $|R(f)| \le \frac{(b-a)^3}{12} \max_{\xi\in[a,b]} |f''(\xi)|$.

2. Für $n=2$ nennt man die entsprechende Newton-Cotes-Formel **Simpson-Regel** oder **Keplersche Fassregel.** Wir erhalten

$$Q(f) = a_0 f(a) + a_1 f(\frac{a+b}{2}) + a_2 f(b)$$

   und mit (8.1)

$$\begin{aligned} a_0 &= \int_a^b l_0(x)\,dx = \frac{(\frac{b-a}{2})\cdot(-1)^{2-0}}{0!(2-0)!}\int_0^2 (t-1)(t-2)\,dt \\ &= \frac{b-a}{4}\left[\frac{t^3}{3} - \frac{3t^2}{2} + 2t\right]_0^2 = (\frac{b-a}{4})\,(\frac{8}{3} - 6 + 4) = \frac{b-a}{6}, \\ a_1 &= \int_a^b l_1(x)\,dx = \frac{(\frac{b-a}{2})\cdot(-1)}{1!1!}\int_0^2 t\,(t-2)\,dt = \frac{4(b-a)}{6}, \\ a_2 &= \int_a^b l_2(x)\,dx = \frac{b-a}{6}. \end{aligned}$$

Also folgt

$$Q(f) = \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)].$$

Quadraturfehler

Die Simpson-Regel ist sogar für Polynome vom Grad 3 exakt. Wir nutzen die Abschätzung aus Folgerung 8.3 (bzw. Satz 8.2) mit $m = d = 3$ und erhalten

$$|R(f)| \le \max_{x\in[a,b]} |f^{(4)}(x)| \int_a^b |G_3(t)|\, dt$$

für $f \in C^4[a,b]$, wobei für $t \in [a,b]$,

$$\begin{aligned}
G_3(t) &= \frac{1}{3!}\left\{\int_a^b (x-t)_+^3\, dx - (\frac{b-a}{6})\left[(a-t)_+^3 + 4(\frac{a+b}{2}-t)_+^3 + (b-t)_+^3\right]\right\} \\
&= \frac{1}{6}\left\{\int_t^b (x-t)^3\, dx - \frac{b-a}{6}\left[4(\frac{a+b}{2}-t)_+^3 + (b-t)^3\right]\right\} \\
&= \begin{cases} \frac{1}{6}\left\{\frac{(x-t)^4}{4}\Big|_t^b - \frac{b-a}{6}\left[4(\frac{a+b}{2}-t)^3 + (b-t)^3\right]\right\}, & a \le t \le \frac{a+b}{2}, \\ \frac{1}{6}\left\{\frac{(b-t)^4}{4} - \frac{b-a}{6}\cdot(b-t)^3\right\}, & \frac{a+b}{2} < t \le b, \end{cases} \\
&= \begin{cases} \frac{(t-a)^3}{72}(3t-a-2b), & a \le t \le \frac{a+b}{2}, \\ \frac{(b-t)^3}{72}(2a+b-3t), & \frac{a+b}{2} < t \le b. \end{cases}
\end{aligned}$$

Also folgt insgesamt

$$\int_a^b |G_3(t)|\, dt = 2\int_a^{\frac{a+b}{2}} \frac{(t-a)^3(a+2b-3t)}{72}\, dt = \frac{1}{2880}(b-a)^5.$$

Wir erhalten für $f \in C^4[a,b]$:

$$|R(f)| \le \frac{1}{2880}(b-a)^5 \max_{x\in[a,b]} |f^{(4)}(x)|.$$

Newton-Cotes-Formeln höherer Ordnung werden kaum benutzt. Besser ist es, Formeln niedriger Ordnung zusammenzusetzen.

Zusammengesetzte Simpson-Regel

Für die Zerlegung $A = z_0 < z_1 < \ldots < z_N = B$ in äquidistante Stützstellen $z_j = A + j\,h,\ j = 0,\ldots,N,\ h = \frac{B-A}{N}$, erhalten wir

$$\begin{aligned}
\int_a^b f(x)\, dx &\approx \sum_{j=0}^{N-1} \frac{h}{6}\left[f(z_j) + 4f(\frac{z_j+z_{j+1}}{2}) + f(z_{j+1})\right] \\
&= \frac{h}{3}\left[\frac{f(A)}{2} + 2\sum_{j=0}^{N-1} f(\frac{z_j+z_{j+1}}{2}) + \sum_{j=1}^{N-1} f(z_j) + \frac{f(B)}{2}\right].
\end{aligned}$$

Fehler für zusammengesetzte Regel:

$$
\begin{aligned}
|R^z(f)| &\leq \sum_{j=0}^{N-1} \frac{1}{2880}\, h^5 \max_{\xi \in [z_j, z_{j+1}]} |f^{(4)}(\xi)| \\
&\leq \max_{x \in [A,B]} |f^{(4)}(x)|\, \frac{(B-A)^5}{N^5}\, \frac{N}{2880} = \mathcal{O}\left(\frac{1}{N^4}\right).
\end{aligned}
$$