

# I. Numerische Quadratur

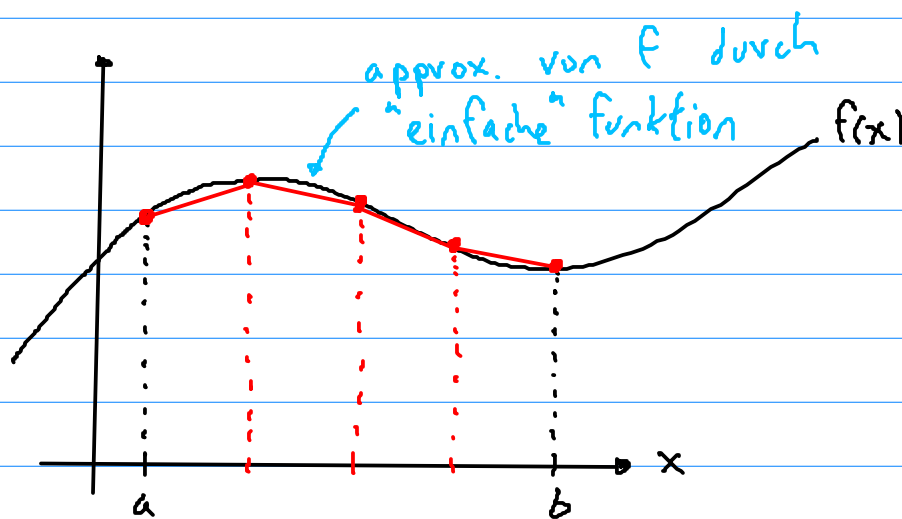
Ziele: - approximieren von bestimmten Integralen

$$Q[f] \approx \int_a^b f(x) dx$$

- Genauigkeit der Approximation abschätzen
- fundamentale Konzepte der Numerik kennenlernen
- Newton-Cotes, Gauss, adaptive Quadratur
- zwei-dimensionale Quadratur

Wozu: Oft ist  $\int_a^b f(x) dx$  nicht exakt berechenbar

Idee:



$$I[f] = \int_a^b f(x) dx \approx \sum_j w_j \cdot f(x_j) = Q[f]$$

# I.1 Polynomiale Interpolation

Gegeben  $n+1$  paarweise verschiedene Stützstellen / Knoten  $x_0, x_1, \dots, x_n$  und zugehörige Stützwerte  $y_0, y_1, \dots, y_n$

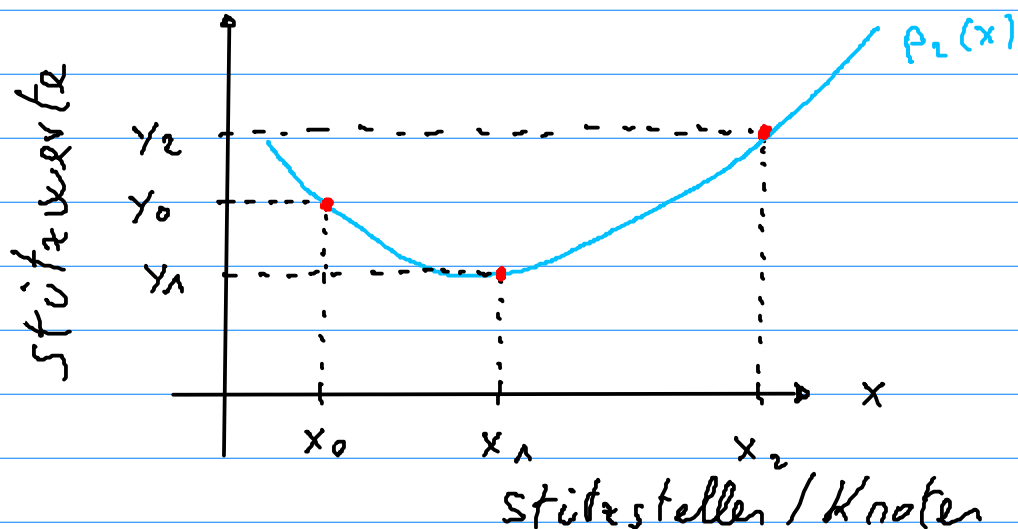
finde das Polynom  $n$ -ten Grades

$$p_n(x) = a_0 + a_1 \cdot x + \dots + a_n \cdot x^n \in \mathbb{P}_n$$

welches die Interpolationsbedingungen (IB) erfüllt

$$p_n(x_j) = y_j \quad (j=0, 1, \dots, n)$$

Die  $n+1$  Koeffizienten  $a_0, a_1, \dots, a_n$  des sog. Interpolationspolynom (IP) ergeben sich aus den  $n+1$  IB (nur lineares Gleichungssystem) (LGS)



Bsp.: (1) Finde  $p_2(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2$

mit  $(x_0, y_0) = (1, 2)$ ,  $(x_1, y_1) = (3, 5)$  und

$(x_2, y_2) = (4, 4)$

Die IB lauten

$$p_2(x_0) = p_2(1) = a_0 + a_1 + a_2 = 2 = y_0$$

$$p_2(x_1) = p_2(3) = a_0 + 3a_1 + 9a_2 = 5 = y_1$$

$$p_2(x_2) = p_2(4) = a_0 + 4a_1 + 16a_2 = 4 = y_2$$

Oder als LGS

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ 4 \end{pmatrix}$$

Lösen ...  $a_0 = -2$ ,  $a_1 = \frac{23}{6}$ ,  $a_2 = -\frac{5}{6}$

MATLAB: -  $p = \text{polyfit}(x, y, n)$

- Einfache Auswertung mit  $\text{polyval}$

Das IP kann man auch direkt mittels der Lagrange'schen Interpolationsformel (LI) bestimmen

$$p_n(x) = \sum_{j=0}^n y_j \cdot L_j^{\wedge}(x)$$

wobei

$$L_j^{\wedge}(x) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i} \quad (j = 0, 1, \dots, n)$$

die sog. Lagrange-Polynome (LP) sind.

Die LP haben folgende Eigenschaften

(LP1)  $L_j^{\wedge}(x)$  sind Polynome  $n$ -ten Grades

$$(LP2) \quad L_j^{\wedge}(x_k) = \delta_{jk} = \begin{cases} 1, & j=k \\ 0, & j \neq k \end{cases}$$

(LP2) ist der Grund wieso LI die IB erfüllt:

$$p_n(x_i) = \sum_{j=0}^n y_j \cdot L_j^{\wedge}(x_i)$$

$$= 0 + \dots + 0 + y_i \cdot \underbrace{L_i^{\wedge}(x_i)}_1 + 0 + \dots$$

$$= y_i \cdot 1$$

Bsp.: (2) Finde das IP durch  $(x_0, y_0) = (1, 2)$ ,  
 $(x_1, y_1) = (3, 5)$  und  $(x_2, y_2) = (4, 4)$

↪ wie Bsp. (1)!

Berechne die LP:

$$\begin{aligned} L_0^2(x) &= \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x - 3}{1 - 3} \cdot \frac{x - 4}{1 - 4} \\ &= \frac{1}{6} (x - 3)(x - 4) \end{aligned}$$

$$\begin{aligned} L_1^2(x) &= \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} = \frac{x - 1}{3 - 1} \cdot \frac{x - 4}{3 - 4} \\ &= -\frac{1}{2} (x - 1)(x - 4) \end{aligned}$$

$$\begin{aligned} L_2^2(x) &= \frac{x - y_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} = \frac{x - 1}{4 - 1} \cdot \frac{x - 3}{4 - 3} \\ &= \frac{1}{3} (x - 1)(x - 3) \end{aligned}$$

Damit

$$p_2(x) = 2 \cdot L_0^2(x) + 5 \cdot L_1^2(x) + 4 \cdot L_2^2(x)$$

$$= \dots = -2 + \frac{29}{6}x - \frac{5}{6}x^2$$

(≡ Bsp. (1) ✓)

## I.2 Interpolationsfehler

Nun sollen die Stützwerte  $y_j$  Werte einer Funktion  $f$  an den paarweise verschiedenen Stützstellen  $x_j$  sein und wir fragen uns wie gut das IP die Funktion  $f$  zwischen den Stützstellen approximiert.

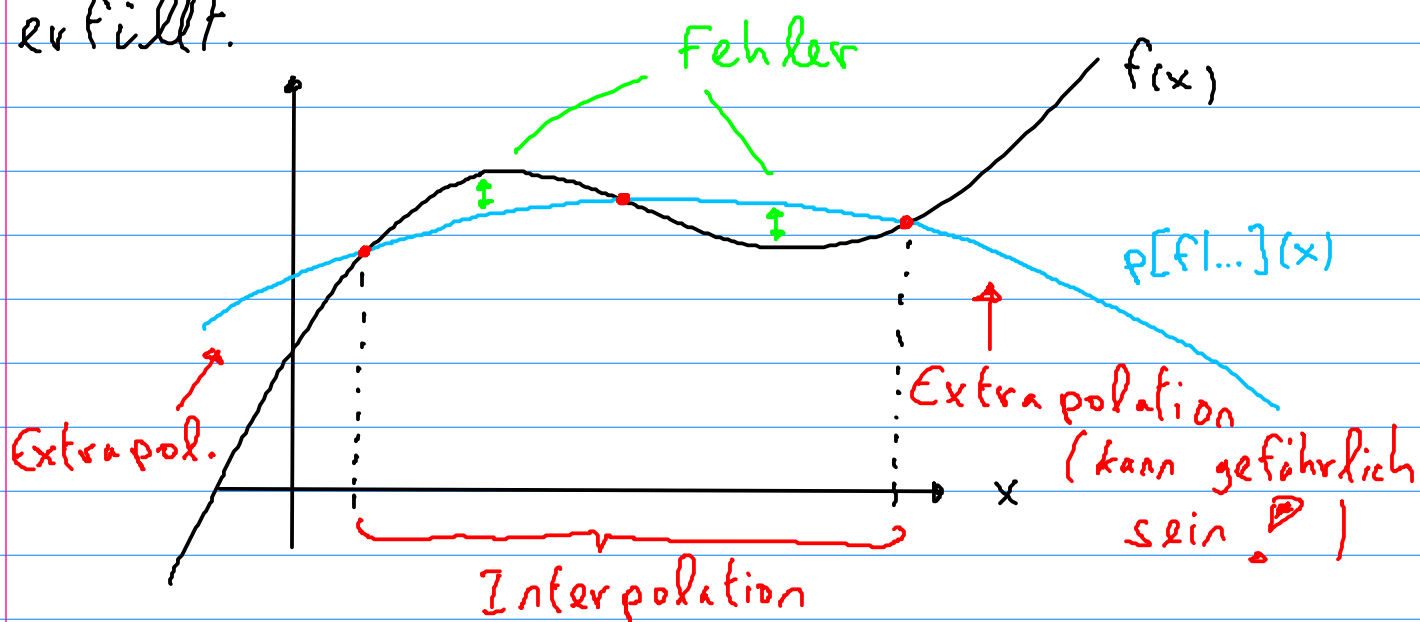
Sei also  $f: I = [a, b] \rightarrow \mathbb{R}$  und wir bezeichnen mit

$$p[f|x_0, \dots, x_n](x) \in \mathbb{P}_n$$

das IP welches die IB

$$p[f|x_0, \dots, x_n](x_j) = f(x_j) \quad (j=0, 1, \dots, n)$$

erfüllt.



Für  $f$   $(n+1)$ -mal stetig differenzierbar lässt sich zeigen, dass es für jedes  $x \in I$  ein  $\xi = \xi(x) \in I$  gibt mit

↑ hängt von  $x$  ab!

↑  $(n+1)$ -te Ableitung

$$e(x) = f(x) - p[f|x_0, \dots, x_n](x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x-x_j)$$

hängt von  $f$  ab      den Stützstellen ab

$e(x)$  ist eine Fehlerfunktion über das ganze Intervall  $I$ . Oft ist man (nur) am grössten Fehler über  $I$  interessiert:

$$\|e\|_\infty = \max_{x \in I} |e(x)| \quad (\text{Maximumsnorm})$$

$$= \max_{x \in I} \left| \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{j=0}^n (x-x_j) \right|$$

Abschätzung

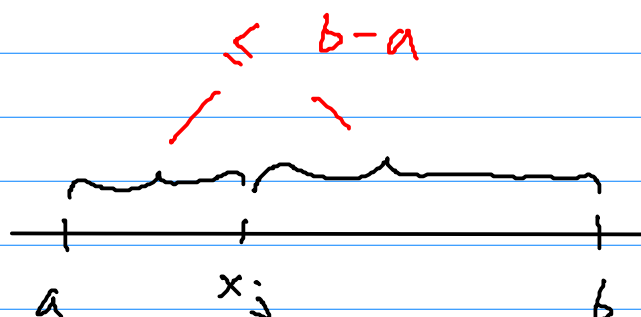
$$\leq \max_{x \in I} \left| \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \right| \cdot \max_{x \in I} \left| \prod_{j=0}^n (x-x_j) \right|$$

$$= \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \left\| \prod_{j=0}^n (x-x_j) \right\|_\infty$$

≤  $b-a$

$$\leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} (b-a)^{n+1}$$

Die letzte Abschätzung kann man am besten graphisch verstehen:



Die Aussage "für  $f$   $(n+1)$ -mal stetig differenzierbar" werden wir noch oft sehen.

Man kann auch sagen: -  $f \in C^{n+1}[I]$

weniger  
präzise ...

- $f$  genügend glatt (smooth)
- $f$  genügend oft stetig differenzierbar



## I.3 Numerische Integration = Quadratur

9

Ziel: Approximation von

$$I[f] = \int_a^b f(x) dx$$

berechnen

Idee: Verwende Polynomiale Interpolation um  $f(x)$  zu approximieren und integriere

$$p[f/x_0, \dots, x_n]$$

(... Polynome sind einfach zu integrieren...)

Def.: Eine endliche Rechenvorschrift der Form

$$Q[f] = \sum_{j=0}^n w_j \cdot f(x_j)$$

zur Approx. von  $I[f]$  nennt man

Quadraturregel (QR) oder Quadraturformel.

Die  $x_j \in I = [a, b]$  nennt man (Quadratur) Knoten oder Integrationsstützstellen und die  $w_j$  (Quadratur) Gewichte.

Quadraturregeln können nun ganz einfach hergeleitet werden.

Seien  $x_0, x_1, \dots, x_n \in I$  uns gegebene Knoten.

Dann ist das IP einer Funktion  $f$

$$p[f|x_0, \dots, x_n](x) = \sum_{j=0}^n f(x_j) \cdot L_j^n(x)$$

Da das IP die Funktion approx., so gilt

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p[f|x_0, \dots, x_n](x) dx \\ &= \int_a^b \sum_{j=0}^n f(x_j) \cdot L_j^n(x) dx \\ &= \sum_{j=0}^n \int_a^b f(x_j) L_j^n(x) dx \\ &= \sum_{j=0}^n f(x_j) \underbrace{\int_a^b L_j^n(x) dx}_{\text{Konstant!}} \\ &= \sum_{j=0}^n f(x_j) \cdot w_j = Q_n[f] \end{aligned}$$

Die Quadratur Gewichte können also ganz einfach berechnet werden:

$$w_j = \int_a^b L_j^n(x) dx$$

Beachte: Die  $w_j$  sind unabhängig von  $f$ !

D.h. für gegebene Knoten  $x_j$  kann man sie ein für alle Mal berechnen und tabellieren.

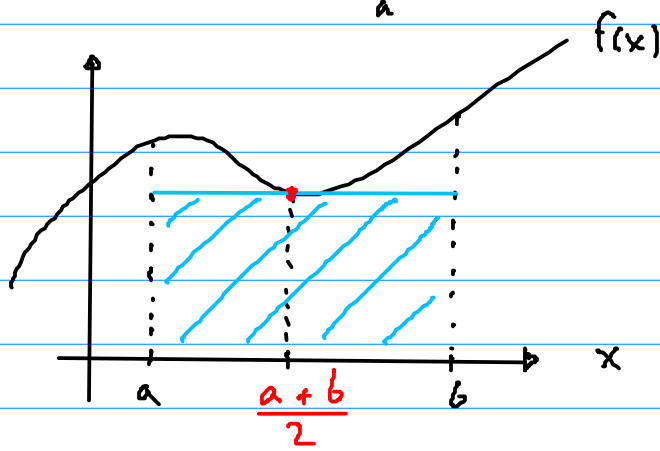
Wichtige Beispiele ...

Bsp.: (3) Mittelpunktsregel (MR) ( $n=0$ )

$$\text{Knoten: } x_0 = \frac{a+b}{2}$$

$$\text{LP: } L_0^0(x) = 1$$

$$\text{Gewichte: } w_0 = \int_a^b L_0^0(x) dx = b-a$$



Damit

$$Q_0[f] = (b-a) \cdot f\left(\frac{a+b}{2}\right)$$

(4) Trapezregel (TR) ( $n=1$ )

Knoten:  $x_0 = a$ ,  $x_1 = b$

LP :  $L_0^1(x) = \frac{x-b}{a-b}$

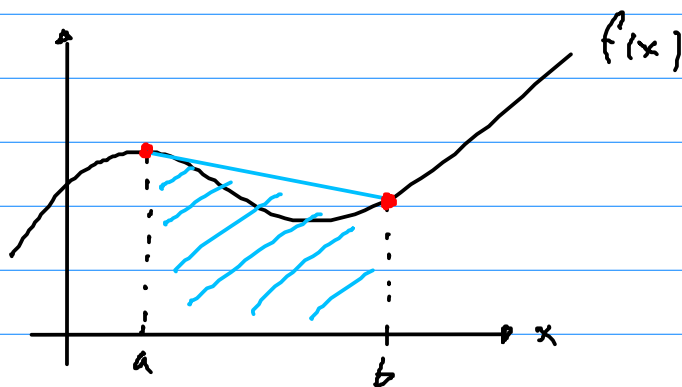
$$L_1^1(x) = \frac{x-a}{b-a}$$

Gewichte:  $w_0 = \int_a^b L_0^1(x) dx = \int_a^b \frac{x-b}{a-b} dx$

$$= \frac{b-a}{2}$$

$$\int_a^b \frac{(x-b)^2}{2(a-b)} dx = \frac{(x-b)^3}{6(a-b)} \Big|_a^b = \frac{0 - (a-b)^3}{6(a-b)} = -\frac{(a-b)^2}{6}$$

$$w_1 = \dots = \frac{b-a}{2}$$



Damit

$$Q_1[f] = \frac{b-a}{2} (f(a) + f(b))$$

(s) Simpson-Regel (SR) ( $n=2$ )

Knoten:  $x_0 = a$ ,  $x_1 = \frac{a+b}{2}$ ,  $x_2 = b$

$$LP : L_0^2(x) = \frac{x - \frac{a+b}{2}}{a - \frac{a+b}{2}} \cdot \frac{x - b}{a - b}$$

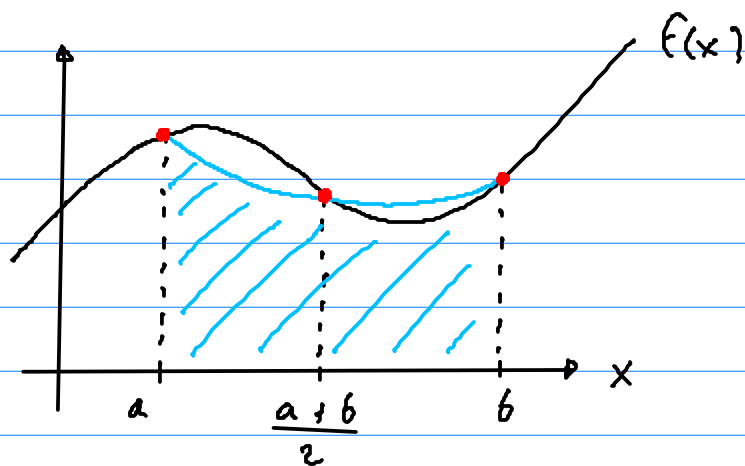
$$L_1^2(x) = \dots$$

$$L_2^2(x) = \dots$$

Gewichte:  $w_0 = \int_a^b L_0^2(x) dx = \dots = \frac{b-a}{6}$

$$w_1 = \dots = \frac{4(b-a)}{6}$$

$$w_2 = \dots = \frac{b-a}{6}$$



Damit

$$Q_2[f] = \frac{b-a}{6} \left( f(a) + 4 \cdot f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Die MR, TR und SR sind sog.  
Newton-Cotes (NC) QR<sub>n</sub>.

Bei diesen QR verteilt man die Knoten  $x_j$   
 äquidistant über das Intervall  $I = [a, b]$

$$n=0: x_0 = \frac{a+b}{2}$$

$$n>0: x_j = a + \frac{b-a}{n} j \quad \text{für } j=0, 1, \dots, n$$

Bem.: (i) TR und SR gehören zu den  
 "populärsten" QR

(ii) NC QR mit  $n > 6$  werden numerisch  
 unbrauchbar (da negative Gewichte  $w_j$   
 auftreten)

## I.4 Quadraturfehler

Nun interessieren wir uns für die Güte  
 von QR<sub>n</sub>.

Def.: Wir nennen  $E[F] = |Q[F] - I[F]|$   
 den Quadraturfehler (QF).

Im Prinzip könnten wir den QF mit Hilfe  
 des Interpolationsfehler untersuchen... Dies ist  
 jedoch (relativ) "mühsam".

Als ein Maß der Genauigkeit einer QR definieren wir:

Def.: Eine QR hat Genauigkeitsgrad (GG)  $q \in \mathbb{N}$ , falls sie alle Polynome bis und mit zum Grad  $q$  exakt integriert und  $q$  die größtmögliche Zahl mit dieser Eigenschaft ist.  
Manchmal auch Exaktheitsgrad.

Def.: Die Ordnung  $s$  einer QR ist definiert durch  $s = q + 1$ .

Dank der Linearität von  $I[f]$  und  $Q[f]$  kann man den GG einfach bestimmen durch

$$Q[x^k] = I[x^k] \quad k = 0, 1, \dots, q$$

$$Q[x^{q+1}] \neq I[x^{q+1}]$$

$$\begin{aligned} \Gamma \quad p \in \mathcal{P}_q : \quad I[p] &= \int_a^b a_0 + a_1 \cdot x + \dots + a_q \cdot x^q \, dx \\ &= a_0 \int_a^b 1 \, dx + a_1 \int_a^b x \, dx + \dots + a_q \int_a^b x^q \, dx \\ &= a_0 \cdot I[1] + a_1 \cdot I[x] + \dots + a_q \cdot I[x^q] \end{aligned}$$

$$Q[p] = \dots \sim \text{gleich wie für } I[p] \quad \checkmark$$

Eine weitere willkommene Vereinfachung bei der Bestimmung des GG ist, dass man es nur für das Referenz-Intervall (RI)  $I = [-1, 1]$  überprüfen muss. Wieso?

Weil sich jedes Intervall  $[a, b]$  durch die Variablen substitution

$$x = \frac{b-a}{2} t + \frac{a+b}{2}, \quad t \in [-1, 1]$$

in das RI transformieren lässt:

$$\begin{aligned} I[f] &= \int_a^b f(x) dx \\ &= \int_{-1}^1 f\left(\frac{b-a}{2} t + \frac{a+b}{2}\right) \underbrace{\frac{b-a}{2} dt}_{dx} \\ &= \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2} t + \frac{a+b}{2}\right) dt \end{aligned}$$



Es ist klar, dass NCs  $QR_n$  mindestens den GG des zugrundeliegenden IPs haben. Falls der Grad  $n$  des IPs aber gerade ist, so gewinnt man einen GG gratis dazu aus Symmetriegründen:

Bsp.: (6) MR ( $n=0$ , also gerade)

$$Q_0[f] = (b-a) f\left(\frac{a+b}{2}\right)$$

$$Q_0[1] = 2 \cdot 1 = I[1]$$

$$Q_0[x] = 2 \cdot 0 = I[x]$$

$$Q_0[x^2] = 2 \cdot 0^2 \neq \frac{2}{3} = I[x^2]$$

$$\leadsto \text{GG } q=1 = n+1$$

(7) TR ( $n=1$ , also ungerade)

$$Q_1[f] = \frac{b-a}{2} (f(a) + f(b))$$

$$Q_1[1] = \frac{2}{2} \cdot (1 + 1) = 2 = I[1]$$

$$Q_1[x] = \frac{2}{2} \cdot (1 + 1) = 0 = I[x]$$

$$Q_1[x^2] = \frac{2}{2} \cdot (1 + 1) = 2 \neq I[x^2]$$

$$\leadsto \text{GG } q = 1 = n$$

(8) SR ( $n=2$ , also gerade)

$$Q_2[f] = \frac{b-a}{6} \left( f(a) + 4 \cdot f\left(\frac{a+b}{2}\right) + f(b) \right)$$

$$Q_2[1] = \dots$$

⋮

Übung

$$Q_2[x^k] = \dots$$

$$\leadsto \text{GG } q = 3 = n + 1$$

Für den QF lässt sich zeigen

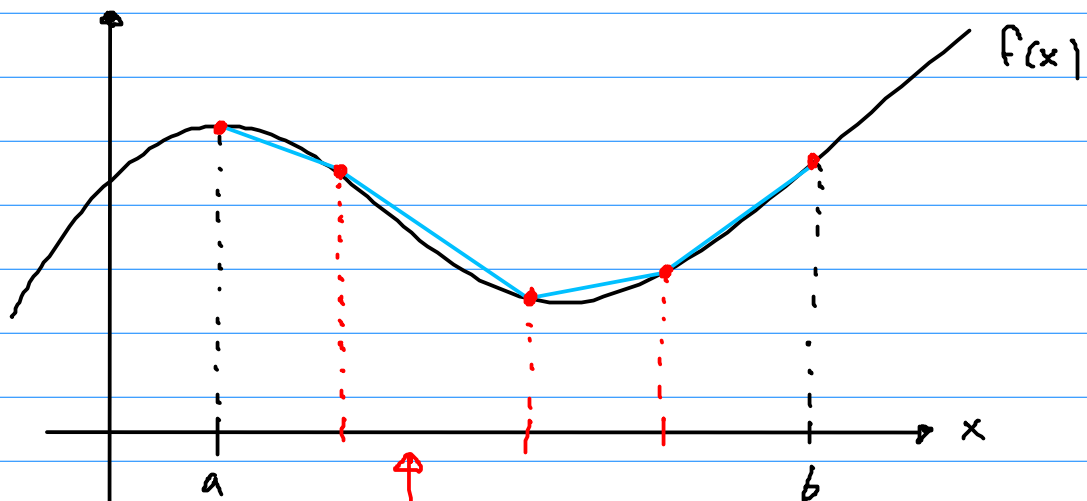
$$E[f] \leq \frac{\|f^{(q+1)}\|_{\infty}}{(q+1)!} (b-a)^{q+2} \quad \left( \begin{array}{l} \text{S} \\ \text{S} \end{array} \right) \quad \left( \begin{array}{l} \text{S}+1 - \text{Ordnung} \\ \text{GG} \end{array} \right) \quad (\text{QFA})$$

Bsp.: (9)  $\leadsto$  Slides

Zusammengefasst: Je grösser der GG, desto genauer ist eine QR, vorausgesetzt, das IP ist eine gute Approx. der Funktion  $f$

## I.5 Summierte Quadraturregeln

Um bessere Approximationen von  $I[f]$  zu erhalten benutzt man i.A. eine gegebene QR nicht über das gesamte Intervall  $[a, b]$ . Sondern man zerlegt  $[a, b]$  in eine Reihe kleinere Teil-Intervalle und wendet die QR auf diese an und summiert die so erhaltenen Näherungen für die Teil-Integrale. Die so erhaltenen Formeln nennt man summierte QR (SQR) oder zusammengesetzte QR.



TR auf jedem Teil-Intervall  
und summierte TR (STR)

Das Intervall  $I = [a, b]$  wird in  $N$  Teil-Intervalle  $I_j = [x_{j-1}, x_j]$  ( $j=1, \dots, N$ ) zerlegt mit

$$x_j = a + j \cdot h, \quad j=0, 1, \dots, N$$

und

$$h = \frac{b-a}{N}.$$

Nun wendet man eine gegebene QR auf die Teil-Intervalle an und summiert

Bsp.: (10) Summierte MR (SMR)

$$\begin{aligned} Q_0^N[F] &= \sum_{j=1}^N Q_0[F \text{ auf } I_j] \\ &= \sum_{j=1}^N h \cdot f\left(\frac{x_{j-1} + x_j}{2}\right) \end{aligned}$$

(11) Summierte TR (SMR)

$$\begin{aligned} Q_1^N[F] &= \sum_{j=1}^N Q_1[F \text{ auf } I_j] \\ &= \sum_{j=1}^N \frac{h}{2} (f(x_{j-1}) + f(x_j)) \\ &= \frac{h}{2} f(x_0) + h \sum_{j=1}^{N-1} f(x_j) + \frac{h}{2} f(x_N) \\ &= \frac{h}{2} \left( f(a) + 2 \sum_{j=1}^{N-1} f(x_j) + f(b) \right) \end{aligned}$$

(12) Summierte SR (SSR)

$$Q_2^N[f] = \sum_{j=1}^N Q_2 [f \text{ auf } I_j]$$

$$= \sum_{j=1}^N \frac{h}{6} \left( f(x_{j-1}) + 4 \cdot f\left(\frac{x_{j-1} + x_j}{2}\right) + f(x_j) \right)$$

$$= \frac{h}{6} \left[ f(a) + 2 \cdot \sum_{j=1}^{N-1} f(x_j) + 4 \cdot \sum_{j=1}^N f\left(\frac{x_{j-1} + x_j}{2}\right) + f(b) \right]$$

Wie verhält sich der QF von SQR<sub>n</sub>?

Der QF einer SQR ist (offensichtlich) die Summe der gemachten Fehler auf jedem Teil-Intervall:

$$\begin{aligned}
 E^N[f] &= |I[f] - Q_n^N[f]| \\
 &= \left| \sum_{j=1}^N I[f \text{ auf } I_j] - Q_n[f \text{ auf } I_j] \right| \\
 &\leq \sum_{j=1}^N \underbrace{|I[f \text{ auf } I_j] - Q_n[f \text{ auf } I_j]|}_{E[f \text{ auf } I_j]}
 \end{aligned}$$

$\Delta$ -Ungleichung  
 (Dreieck)  
 $|a+b| \leq |a|+|b|$

GG von  $Q_n$ !

$$\leq \sum_{j=1}^N \frac{\max_{x \in I_j} |f^{(q+1)}(x)|}{(q+1)!} \underbrace{|x_j - x_{j-1}|}_{h}^{q+2}$$

$\infty$ -Norm über  $I=[a,b]$ !

$$\leq \frac{\|f^{(q+1)}\|_{\infty}}{(q+1)!} h^{q+1} \underbrace{\sum_{j=1}^N h}_{N \cdot h = b-a}$$

Zusammengefasst

$$E^N[f] \leq \frac{\|f^{(q+1)}\|_{\infty}}{(q+1)!} \cdot (b-a) \cdot h^{q+1} = \frac{\|f^{(s)}\|_{\infty}}{s!} \cdot (b-a) \cdot h^s$$

GG
Ordnung

Solche Abschätzungen sind typisch und dazu verwendet man das Landau-Symbol. Man schreibt:

$$e = \mathcal{O}(h^p)$$

Falls

$$|e| \leq C \cdot h^p$$

für positive Konstanten  $C$  und  $p$  gilt

für alle  $h > 0$  klein genug.

Für den SQR Fehler gilt also

$$E^N[f] = \mathcal{O}(h^{q+1}) = \mathcal{O}(h^s).$$

28.02.22

Bsp.: (13) → Slides

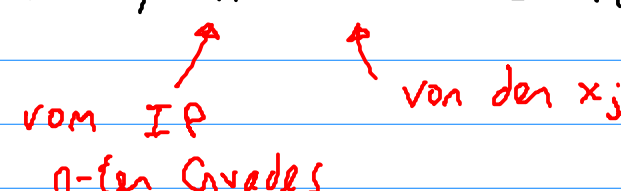
Bem.: (i) Die Ordnung  $s$  kann man sehr einfach in einem log-log plot ablesen

(ii) Um die volle Ordnung zu erhalten muss die zu integrierende Funktion genügend glatt sein

## I.6 Gauss-Quadratur

Bei den NC QR<sub>n</sub> wählt man  $n+1$  äquidistant verteilte Knoten, legt das IP vom Grad  $n$  durch und erhält damit eine QR mit  $G_G$  mindestens  $n$  ( $n/n+1$  falls  $n$  ungerade/gerade).

Idee: wähle die  $n+1$  Knoten so, dass der grösstmögliche  $G_G$  erreicht wird (Hoffnung:  $G_G$  mit  $g \approx n + n + 1 = 2n + 1$ )


  
vom IP von den  $x_j$ 's  
n-ten Grades



Frage: Was ist der grösstmögliche GGr den man mit  $n+1$  Knoten erreichen könnte?

Betrachte folgendes Polynom vom Grad  $2n+2$  auf dem RI:

$$p(x) = \prod_{i=0}^n (x - x_i)^2 \in \mathbb{P}_{2n+2}$$

↑ Knoten

Klar:  $I[p] = \int_{-1}^1 p(x) dx > 0$

Aber mit Quadratur

$$Q[p] = \sum_{j=0}^n \omega_j \cdot \underbrace{p(x_j)}_0 = 0$$

↗ ≠ ↘

Also der grösstmögliche GGr den man erreichen könnte ist  $g = 2n+1$  !

Diesen GGr kann man auch erreichen und man stösst dabei auf einen wichtigen Begriff der linearen Algebra:

Orthogonalität

Betrachten wir hierzu den Interpolationsfehler

$$e(x) = f(x) - p[f | x_0, \dots, x_n] = K(x) \cdot \prod_{i=0}^n (x - x_i)$$

Sei nun  $f(x) = x^m$  ein Monom mit  $m \geq 0$  ganzzahlig. Dann ist

$$e(x) = x^m - p[x^m | x_0, \dots, x_n] = K(x) \cdot \prod_{i=0}^n (x - x_i)$$

↑  
Polynom  
Grad  $m$ 
↑  
Polynom  
Grad  $n$ 
↑  
Polynom  
Grad  $n+1$

Polynom  
Grad  $\max\{m-n-1, 0\}$

mit

$$K(x) = \begin{cases} 0 & \text{für } m \leq n \\ r(x) \in \mathbb{P}_{m-n-1} & \text{für } m > n \end{cases}$$

Integrieren wir nun  $e(x)$  über das RI:

$$\int_{-1}^1 e(x) dx = \int_{-1}^1 x^m dx - \int_{-1}^1 p[x^m | x_0, \dots, x_n] dx$$

$$= I[x^m] - Q[x^m]$$

$$= \int_{-1}^1 K(x) \cdot \prod_{i=0}^n (x - x_i) dx$$

$$= \begin{cases} 0 & \text{für } m \leq n \\ \int_{-1}^1 r(x) \cdot \prod_{i=0}^n (x - x_i) dx & \text{für } m > n \end{cases}$$

Dies bestätigt uns noch einmal, dass eine QR mit  $n+1$  Knoten  $G_n$  von  $n$  hat.

Aber viel mehr noch: Wenn wir  $n+1$  Knoten mit

$$\int_{-1}^1 r(x) \cdot \prod_{i=0}^n (x - x_i) dx \stackrel{!}{=} 0$$

$\underbrace{\int_{-1}^1}_{\in \mathbb{P}_n} \cdot \underbrace{\prod_{i=0}^n (x - x_i)}_{\in \mathbb{P}_{n+1}}$

für  $n < m < 2n+2$  bestimmen können, so erhalten wir ein QR mit grösstmöglichen  $G_n$ !

Aus der linearen Algebra ist bekannt,

dass

$$\langle f, g \rangle = \int_{-1}^1 f(x) g(x) dx$$

ein Skalarprodukt in  $C[-1, 1]$  definiert.

Wenn  $\langle f, g \rangle = 0$ , so sind  $f$  und  $g$  orthogonal zueinander.

Also

Polynome!

$$\langle v(x), \prod_{i=0}^n (x - x_i) \rangle = 0$$

sagt uns wir suchen Orthogonalpolynome!

Dies führt uns zu den Legendre-Polynomen welche durch folgende Rekursionsformel gegeben sind

$$P_0(x) = 1, \quad P_1(x) = x$$

$$P_{j+1}(x) = \frac{2j+1}{j+1} x \cdot P_j(x) - \frac{j}{j+1} P_{j-1}(x), \quad j \geq 1$$

$(j=0,1,\dots,n)$

29

Die  $P_j(x)$  bilden eine orthogonale Basis von  $\mathbb{P}_n$ :

$$\langle P_i(x), P_j(x) \rangle = \int_{-1}^1 P_i(x) \cdot P_j(x) dx = 0, \quad i \neq j$$

Um den maximalen GCN zu erhalten wählen wir die  $n+1$  Knoten so, dass

$$\prod_{i=0}^n (x-x_i) \sim P_{n+1}(x)$$

ein (skalares) vielfaches vom  $(n+1)$ -ten Legendre-Polynom ist

$\leadsto$  Wähle die Knoten  $x_i$  als die Nullstellen von  $P_{n+1}(x)$ !

## Gauss (Legendre) Quadratur

Die  $(n+1)$ -Punkte Gauss (Legendre) Quadratur (GLQ) auf dem RI  $[-1, 1]$  ist gegeben durch:

$$G_n[f] = \sum_{j=0}^n w_j \cdot f(x_j)$$

wobei die Gauss-Punkte  $x_j$  die Nullstellen des  $(n+1)$ -ten Legendre-Polynoms  $P_{n+1}(x)$  und die Gewichte

$$w_j = \frac{2(1-x_j^2)}{((n+1)P_n(x_j))^2}, \quad j=0, 1, \dots, n$$

sind.

Sie hat den grösstmöglichen GG  $q=2n+1$  und damit Ordnung  $s=2n+2$ .

Bsp.: (14) 2-Punkte GLQ ( $n=1$ )

(1) Berechne  $P_{n+1}(x) = P_2(x)$  mit Rekursionsformel

$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2} = \frac{1}{2}(3x^2 - 1)$$

(2) Berechne Nullstellen von  $P_2(x)$

$$\frac{1}{2}(3x^2 - 1) \stackrel{!}{=} 0$$

$$x_{0,1} = \pm \frac{1}{\sqrt{3}}$$

(3) Berechne Gewichte

$$\begin{aligned} \omega_0 &= \frac{2(1 - x_0^2)}{(2 \cdot P_1(x_0))^2} = \frac{2(1 - 1/3)}{(2 \cdot (-1/\sqrt{3}))^2} \\ &= \frac{\cancel{2} \cdot \overbrace{(1 - 1/3)}^{2/3}}{\cancel{2} \cdot \frac{1}{3}} = 1 \end{aligned}$$

$$\omega_1 = 1$$

$$\text{Also } G_1[f] = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

Bem.: (i) Die Gewichte bei GLQ sind stets positiv

(ii) Für  $n$  "nicht zu groß" sind die GLQ tabelliert

Für grosse  $n$  werden die GLQ numerisch bestimmt (z.B. wie in Übung S03, no gau leg. m)

(iii) Es gilt stets: hohe Ordnung bedeutet nicht zwingend hohe Genauigkeit !

Die gilt nur wenn  $f$  glatt genug ist.

(iv) Allgemeiner betrachte man

$$I[f] = \int_a^b w(x) \cdot f(x) dx$$

wobei  $w(x)$  eine nichtnegative Gewichtsfunktion ist:

-  $w(x) = 1$   $\rightsquigarrow$  Gauss-Legendre

-  $w(x) = \frac{1}{\sqrt{1-x^2}}$   $\rightsquigarrow$  Gauss-Tschebyscheff

-  $w(x) = e^{-x^2}$   $\rightsquigarrow$  Gauss-Hermite

...  $w(x) = 1/\sqrt{x}$   $\rightsquigarrow$  Serie 3



(v) Manchmal beginnt der Summationsindex bei 1

$$G[f] = \sum_{j=1}^n \omega_j \cdot f(x_j)$$

Dann ist  $G_n \quad q = 2n-1$  und die Ordnung  $s = 2n$

## I.7 Adaptive Quadratur

Ziel: Berechne

$$I[f] = \int_a^b f(x) dx$$

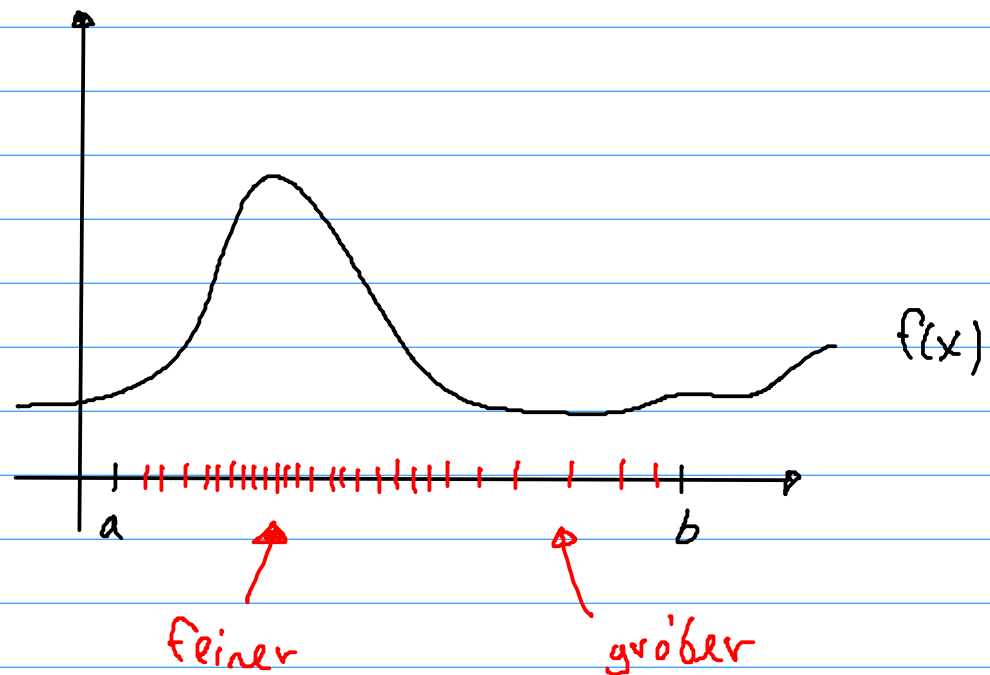
bis auf eine vorgegebene Toleranz

$$|Q[f] - I[f]| < \epsilon_{01}$$

so effizient wie möglich

d.h. mit so wenig Funktionsauswertungen wie möglich

Idee: Anstatt das Intervall  $I = [a, b]$  in gleich grosse Teil-Intervalle aufzuteilen, verwende feinere oder gröbere Teil-Intervalle je nachdem wie stark die Funktion  $f(x)$  variiert, oder äquivalent, wo der QF gross ist:



Dazu benötigen wir eine Schätzung des Fehlers.

Idee: Vergleiche das Resultat einer Methode mit dem Resultat einer genaueren Methode

07.03.22

Bsp.: (15)  $I[e^x] = \int_0^1 e^x dx = e - 1 = 1.71828\dots$

$$TR: Q_1[e^x] = \frac{1}{2} (e^0 + e^1) = 1.85914\dots$$

$$SR: Q_2[e^x] = \frac{1}{6} (e^0 + 4 \cdot e^{1/2} + e^1) = 1.71886\dots$$

$$STR: Q_1^2[e^x] = \frac{1/2}{2} (e^0 + 2 \cdot e^{1/2} + e^1) = 1.75393\dots$$

$$\text{Exakter Fehler: } |Q_1[e^x] - I[e^x]| = 0.14085\dots$$

$$\text{Schätzung 1: } |Q_1[e^x] - Q_2[e^x]| = 0.14027\dots$$

$$\text{Schätzung 2: } |Q_1[e^x] - Q_1^2[e^x]| = 0.10520\dots$$

Bsp. (15) zeigt, dass obige einfache Idee durchaus brauchbare Fehler-Schätzer liefert.

Untersuchen wir theoretisch die Schätzung 2, d.h. der Vergleich einer QR mit derselben QR aber mit halbiertem Teil-Intervall Länge, sog. Intervall-halbierungs Fehler-Schätzer.

Betrachten wir hierzu ein Intervall  $I = [a, b]$  und den QF einer QR der Ordnung  $s$  (QFA)

$$E[F] = |Q[F] - I[F]| \leq \frac{\|f^{(s)}\|_{\infty}}{s!} (b-a)^{s+1} = K \cdot (b-a)^{s+1}$$

Dies nennt man einen a priori Fehler-Schätzer. Dieser ist natürlich nur brauchbar, wenn  $\|f^{(s)}\|_{\infty}$  bzw.  $K$  bekannt ist (was i.A. natürlich nicht der Fall ist!).

Für den QF gilt also:

$$E^1[F] = |Q^1[F] - I[F]| \approx K \cdot (b-a)^{s+1}$$

$$E^2[F] = |Q^2[F] - I[F]| \approx K \cdot \left(\frac{b-a}{2}\right)^{s+1} + K \cdot \left(\frac{b-a}{2}\right)^{s+1}$$

fehler auf halben Teil-Int.

$$= \frac{K}{2^s} (b-a)^{s+1} = \frac{E^1[F]}{2^s}$$

Nun

$$\begin{aligned}
 E^1[f] &= | \overbrace{Q^1[f] - Q^2[f] + Q^2[f] - I[f]}^{+ 0 \text{ Trick!}} | \\
 &\stackrel{\Delta\text{-UG}}{\leq} | Q^1[f] - Q^2[f] | + \underbrace{| Q^2[f] - I[f] |}_{\approx \frac{E^1[f]}{2^s}}
 \end{aligned}$$

$$\Rightarrow E^1[f] \approx \frac{2^s}{2^s - 1} | Q^1[f] - Q^2[f] |$$

und

$$E^2[f] \approx \frac{1}{2^s - 1} | Q^1[f] - Q^2[f] |$$

Dies sind sog. a posteriori Fehler-Schätzer.

Bsp.: (16) TR hat Ordnung  $s=2$  und es ergibt sich für Zahlen aus Bsp. (15) (Schätzung 2):

$$\begin{aligned}
 |Q_2[e^x] - I[e^x]| &= 0.14085\dots \\
 &\approx \frac{2^2}{2^2 - 1} \cdot 0.10520\dots \\
 &= 0.1402\dots \quad \checkmark
 \end{aligned}$$

$$|Q_n^2[e^x] - I[e^x]| = 0.0356\dots$$

$$\approx \frac{1}{2^2-1} \cdot 0.10520\dots$$

$$= 0.0350\dots \quad \checkmark$$

Nun haben wir alles zur Hand um einen adaptiven Quadratur Algorithmus zu bauen!

Adaptive Quadratur (~ Pseudo-MATLAB-Code)

function  $Q = \text{adapt\_quad}(f, a, b, \text{tol})$  Toleranz

if  $E < \text{tol}$  fehler-Schätzer

$Q = \text{quad}(f, a, b)$

else zugrundeliegende QR

REKURSIV!  $Q_1 = \text{adapt\_quad}(f, a, \frac{a+b}{2}, \frac{\text{tol}}{2})$

$Q_2 = \text{adapt\_quad}(f, \frac{a+b}{2}, b, \frac{\text{tol}}{2})$

$Q = Q_1 + Q_2$

end

↑  
halbieren  
fehler-Toleranz, da nun  
halbiertes Int.-Intervall!

Bsp.: (17) Adaptive Simpson Quadratur

→ Slides

Bem.: (i) Adaptive Quadratur kann oft sehr effizient Integrale approximieren

(ii) Adaptive Quadratur kann auch schief gehen !

(iii) In MATLAB gibt es die Befehle `quad` und `integral`

## I.8 Zwei-dimensionale Quadratur

Bis jetzt haben wir nur ein-dimensionale Quadratur behandelt. Diese simplen Methoden kann man relativ einfach auf zwei oder drei Dimensionen verallgemeinern (im Prinzip auch mehr, aber dann wird es sehr schnell sehr Teuer!).

In zwei Dimensionen:

$$I[F] = \int_a^b \int_c^d f(x,y) dx dy$$

$$\approx \sum_{i=0}^n w_i^x \cdot \int_c^d f(x_i, y) dy$$

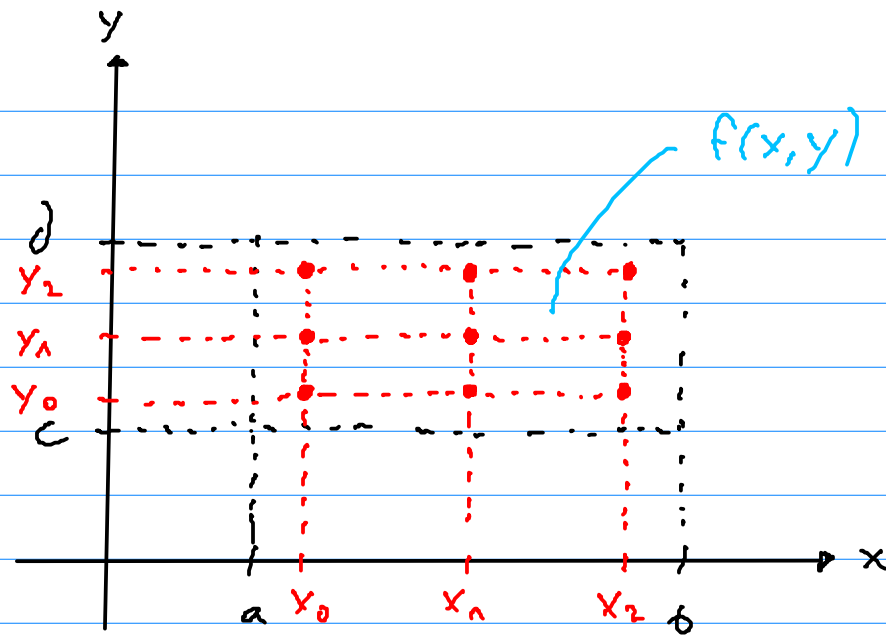
Gewichte & Knoten in x-Koordinate

$$\approx \sum_{i=0}^n w_i^x \sum_{j=0}^m w_j^y \cdot f(x_i, y_j)$$

Gewichte & Knoten in y-Koord.

$$= \sum_{i=0}^n \sum_{j=0}^m w_i^x \cdot w_j^y \cdot f(x_i, y_j)$$





Dies kann man auch einfach auf den Fall variabler Grenzen verallgemeinern:

$$I[f] = \int_a^b \int_{\phi_1(x)}^{\phi_2(x)} f(x,y) dx dy$$