

# Algorithmik kontinuierlicher Systeme

## *Volumen- und Flächenberechnung - Numerische Integration*

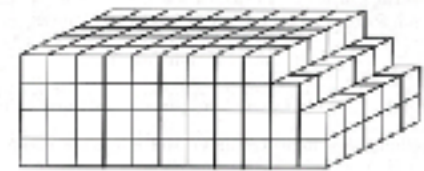
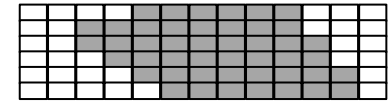


# Warum Berechnungen von Flächen bzw. Volumen, d.h. Berechnung von Integralen.

- Direkte Anwendungen
  - ▶ MT: Volumen der Herzkammern, Größe von Tumoren, Querschnitt von Gefäßen, Lungenvolumen, ...
  - ▶ Technik: Qualitätskontrolle: Einhaltung von Maßen,
- Indirekt: ein Schritt eines (komplexen) Lösungsverfahrens
  - ▶ Finite-Element- bzw. Finite-Volumen-Verfahren zur Lösung von partiellen Differentialgleichungen
  - ▶ Numerische Bestimmung von Integraltransformationen (Fourier-Transformation, Formfaktorberechnung (sh. *radiosity*), ...)

- Diskretisierung des Raumes (der Fläche) in lauter gleich grosse rechteckige (oder quadratische)

- Flächenelemente (Pixel = PictureElements)
  - Volumenelemente (Voxel = VolumeElements)

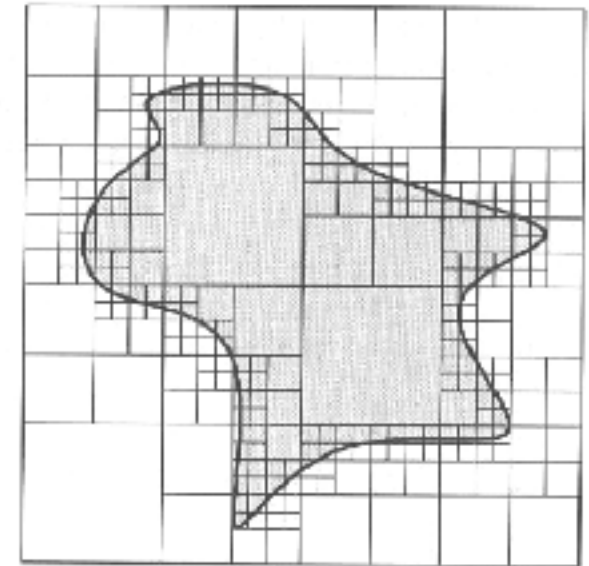


- Datenstrukturen und Alternativen:

- 3D (oder 2D) Bitarray: gehört zum Objekt ja/nein
  - 2D Array mit Grauwerten: Schwarz-Weiss-Bild
  - Array mit „Füllstandswert“  $\phi \in [0,1]$
  - Array mit „Materialnummern“  
(0= Luft, 1= Kopfhaut, 2= Knochen, 3= Gehirnflüssigkeit, ...)
  - Array mit „physikalischen Parametern“  
(Dichte, Temperatur, CO<sub>2</sub>-Konzentration, ....)
  - Varianten: Normzellen in anderen K.-systemen (Kugel, Zylinder, ...)



- Fragen der effizienten Darstellung des Baumes (Linearisierung)
- Bei 2D-Objekten mit „genügend glattem“ Rand, ist die (Speicher-) Komplexität eines Quadtree  $O(h^{-1})$  anstelle von  $O(h^{-2})$
- Bei 3D- Objekten mit „genügend glattem“ Rand, ist die (Speicher-) Komplexität eines Octrees (=Oktaalbaums)  $O(h^{-2})$  anstelle von  $O(h^{-3})$



- Im Normzellenschema:
  - Einfach zählen, multiplizieren mit Volumen der Normzelle
  - Verallgemeinerung: Wenn mit Füllstandsinformationen attribuiert (oder Dichte, etc.), dann entsprechende Gewichtung der Zellen.
- Zellzerlegungsschemata:  
Aufaddieren der Volumen der Grundobjekte
- Quadtree/Octree:
  - Zählen (Aufsummieren) unter Nutzen der Baumstruktur
- Andere Schemata:
  - Umwandeln in Normzellen/Octree-Darstellung
  - Nutzen von Integrationsformeln
    - ★ z.B. Verschiebungsgeometrie:  $V = \text{Höhe} * \text{Fläche}$  wenn die Verschiebungsrichtung orthogonal zur Fläche ist (Prinzip von Cavalieri)
  - **Numerische Integration** (s.u.)

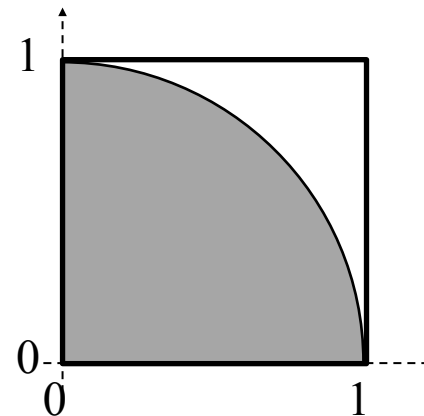
- **Monte Carlo Verfahren:**

- ▶ Skaliere auf Teilfläche/-volumen  $V_0$  des Einheitswürfels  $[0,1]^d$
- ▶ Erzeuge gleichverteilte Zufallszahlen  $x_i \in [0,1]^d$
- ▶ Zähle die Treffer  $x_i \in V_0$  und

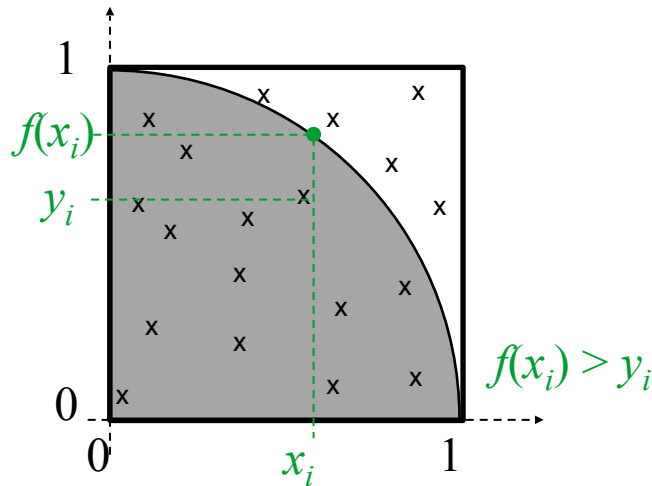
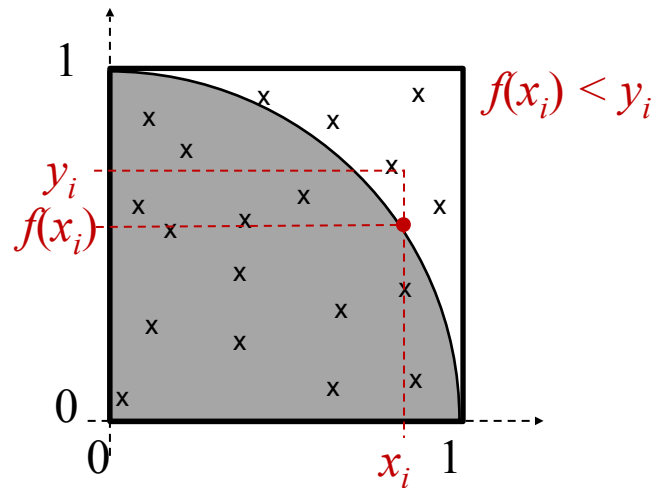
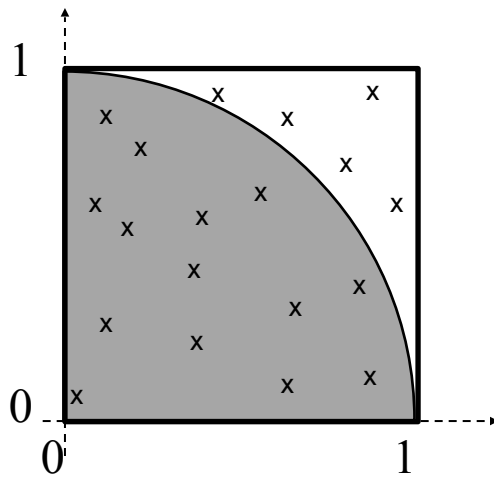
$$\text{Volumen} = \#(\text{hits}) / \#(\text{samples})$$

- ▶ Konvergiert -- aber seeeeeehr langsam  $O(\#(\text{samples})^{-1/2})$
- ▶ Wird interessant für hohe Dimensionen ( $d > 5$ )

- Beispiel:  
Fläche von (Viertel-)Kreis

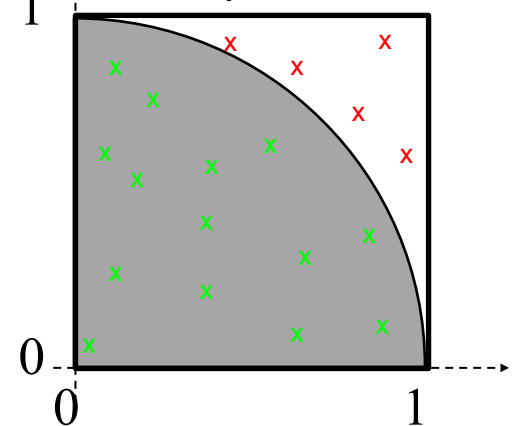


- Monte Carlo Verfahren:
  - Beispiel (Viertel-)Kreis



$$\text{Fläche} \approx \frac{14}{19} = 0.739\dots$$

$$\text{exact} = \frac{\pi}{4} = 0.7854\dots$$



	hits/N	Fehler
N		
100	0.820	0.034602
1,000	0.7888	0.003402
10,000	0.78986	0.004462
100,000	0.785516	0.000118
1,000,000	0.7855152	0.000117



- Berechnung der Fläche mittels Integral
- Aber: Wenige Funktionen kann man exakt integrieren
  - Falls  $f(x)$  eine Linearkombination von Polynomen (oder anderen Funktionen ist, deren Integrale bekannt sind), ist das Problem einfach zu lösen.

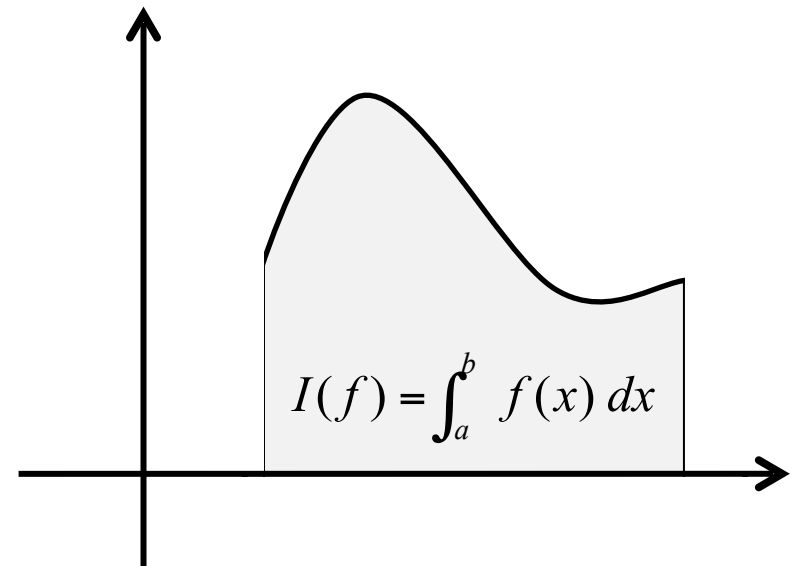
$$f(x) = \sum_{i=0}^n a_i p_i(x)$$

- Man berechnet vorab:

$$w_i = \int_a^b p_i(x) dx$$

- und dann

$$\int_a^b f(x) dx = \sum_{i=0}^n w_i a_i$$



- Polynome kann man exakt integrieren!
- Idee: Bestimme zu gegebener Funktion  $f(x)$  zunächst das Polynom  $p(x)$  welches die Funktionswerte an äquidistanten Stellen interpoliert und nimm als

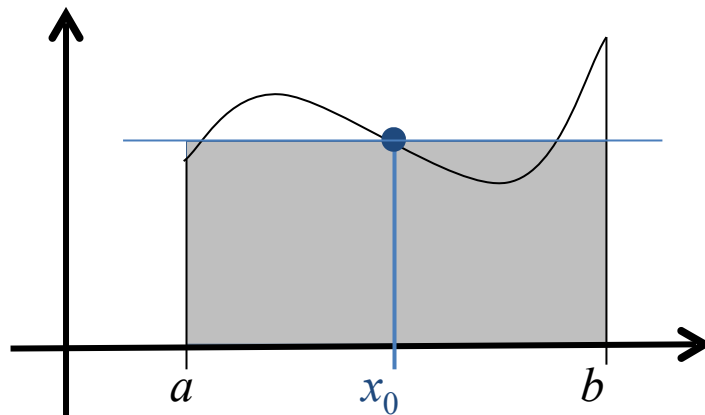
Näherungswert  $\int_a^b p(x) dx$  statt  $\int_a^b f(x) dx$

- Beachte: Zu  $n+1$  Sample-Punkten der Funktion  $f(x)$   
 $\{ (x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), \dots, (x_n, f(x_n)) \}$

gibt es genau eine interpolierendes Polynom vom Grad  $n$ .

Formel (z.B.)  $p(x) = \sum_{i=0}^n f(x_i) \cdot L_i(x)$  (Lagrange-Polynome)

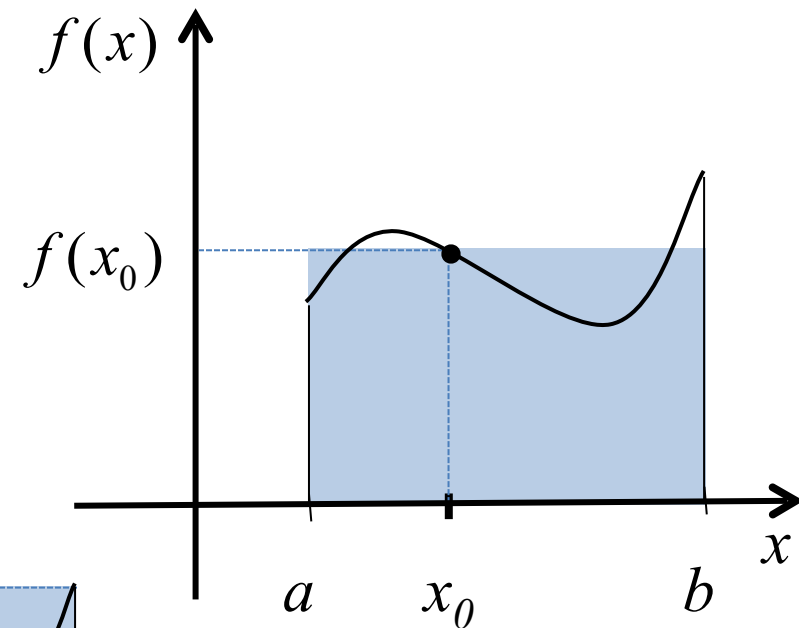
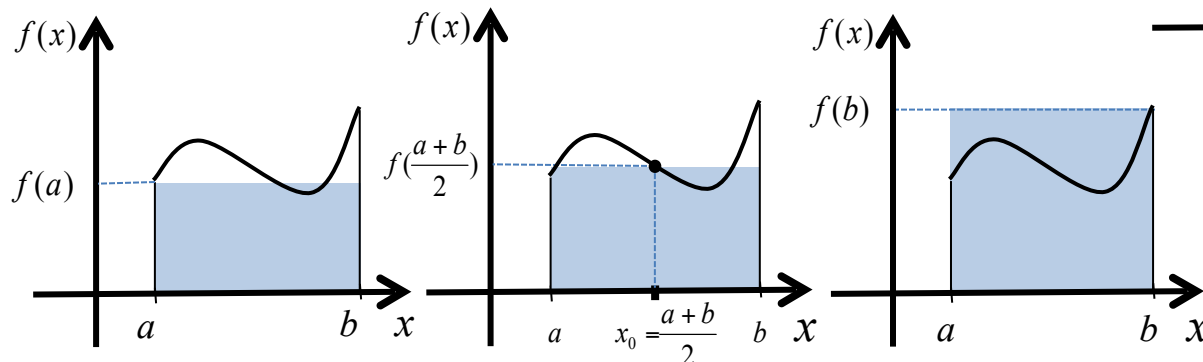
- Eine Sample-Punkt  $(x_0, f(x_0))$  mit  $x_0 = \frac{a+b}{2}$
- Interpolierendes Polynom Grad  $n=0$  :  $p_0(x) = f(x_0) = \text{const}$
- Näherungswert:  $\int_a^b f(x) dx \approx \int_a^b p_0(x) dx = (b-a)f(x_0)$



Mittelpunktsregel

- Interpolation mit konstanter Funktion  $(b - a) \cdot f(x_0)$
- Approximation des Integrals durch ein Rechteck :
- Man erhält sogenannte Rechtecksregeln.
- Je nach Position von  $x_0$  unterscheidet man

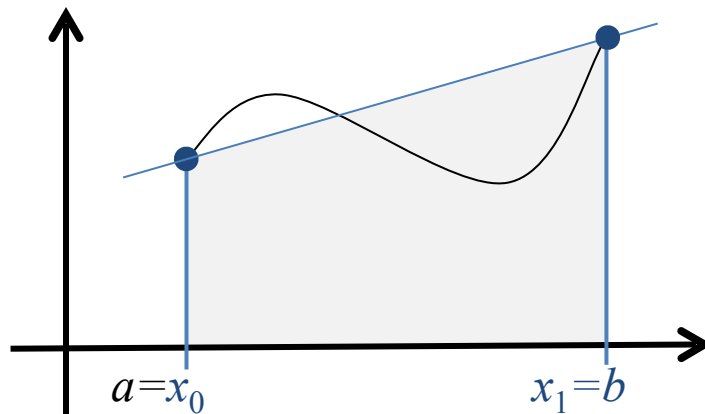
- Linke Rechtecksregel
- Mittelpunktsregel
- Rechte Rechtecksregel



- Zwei Sample-Punkte  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$  mit  $x_0 = a$ ,  $x_1 = b$
- Interpolierendes Polynom vom Grad  $n = 1$ :

$$p_1(x) = \frac{b-x}{b-a} f(x_0) + \frac{x-a}{b-a} f(x_1) \quad (\text{linearer Interpolant})$$

- Näherungswert:  $\int_a^b f(x) dx \approx \int_a^b p_1(x) dx = (b-a) \frac{f(a) + f(b)}{2}$



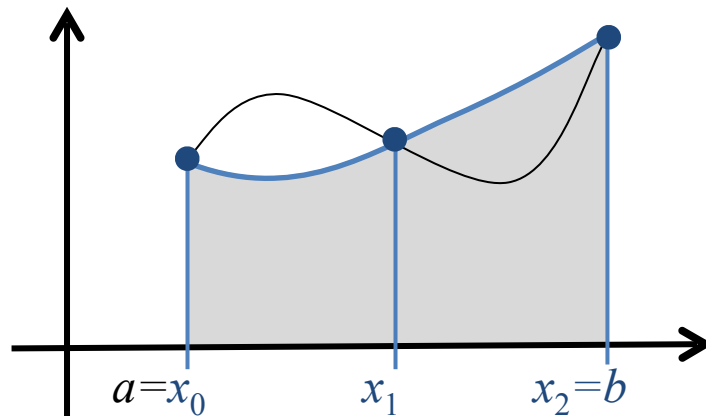
Trapezregel

- 3 Sample-Punkte  $(x_i, f(x_i))$ ,  $i=0,1,2$  mit  $x_0 = a$ ,  $x_1 = \frac{a+b}{2}$ ,  $x_2 = b$
- Interpolierendes Polynom vom Grad  $n = 2$  (Parabel) :

$$p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2)$$

- Näherungswert:

$$\int_a^b f(x) dx \approx \int_a^b p_2(x) dx = (b-a) \left( \frac{1}{6} f(x_0) + \frac{4}{6} f(x_1) + \frac{1}{6} f(x_2) \right)$$

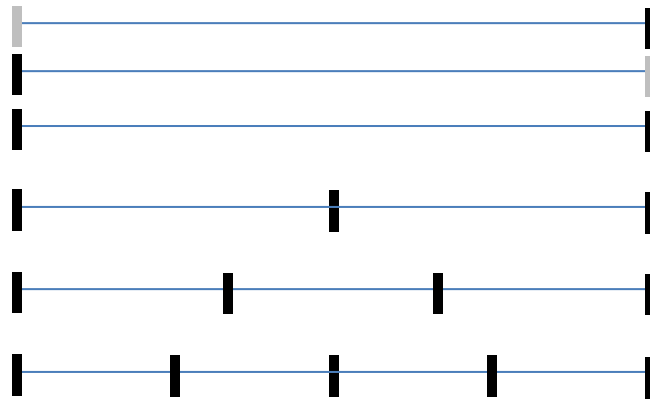


Simpsonregel

oder

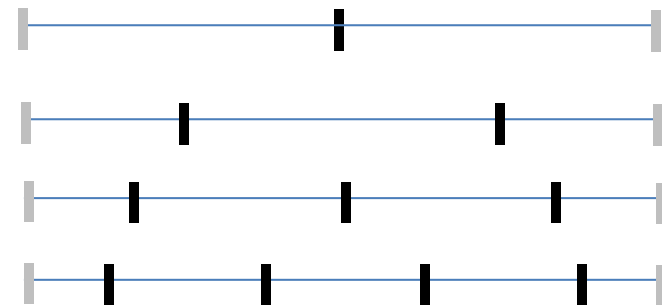
Keplersche Faßregel

- Sample die Funktion  $f(x)$  in  $n+1$  äquidistanten Punkten
- Bestimme das Interpolationspolynom und integriere dieses
  - Äquidistantes Sampling:  
geschlossen (inkl. Intervallenden) oder offen



$$x_i = a + i \cdot \frac{b-a}{n} \quad (i = 0, 1, \dots, n)$$

- speziell :  $a=0, b=1$ :  $x_i = \frac{i}{n}$



$$x_i = a + (2i+1) \cdot \frac{b-a}{2n+2} \quad (i = 0, 1, \dots, n)$$

$$x_i = \frac{2i+1}{2n+2}$$

- $n+1$  Samples  $(x_i, f(x_i))$ ,  $i=0,1,\dots,n$  mit  $x_i = a + i \cdot h$  wobei  $h = \frac{b-a}{n}$
- Interpolierendes Polynom vom Grad  $n$ :

$$p_n(x) = \sum_{i=0}^n f(x_i) L_i^{(n)}(x) \text{ wobei } L_i^{(n)}(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \text{ die Lagrange-Polynome sind.}$$

- Näherungswert:  $\int_a^b f(x) dx \approx \int_a^b p_n(x) dx = (b-a) \sum_{i=0}^n w_i^{(n)} f(x_i)$   
mit Gewichten  $w_i^{(n)} = \int_a^b L_i(x) dx$

n		$w_0^{(n)}$	$w_1^{(n)}$				
1	Trapez	1/2	1/2				
2	Simpson	1/6	4/6	1/6			
3	Newton 3/8	1/8	3/8	3/8	1/8		
4	Milne	7/90	32/90	12/90	32/90	7/90	



- **Simpson-Regel**

- ▶ Ergibt sich durch Interpolation mit quadratischen Funktionen bei drei äquidistanten Stützstellen ( $x_0=a$ ,  $x_1=(a+b)/2$ ,  $x_2=b$ ):

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

- Diese Formel ist auch als **Fassregel** bekannt, die ursprünglich auf Kepler zurückgeht

(J. Kepler: *Nova Stereometrica dolorium vinariorum*, Linz, 1615)  
(*Neue Inhaltsberechnung von Weinfässern*)

- **Johannes Kepler** (1571-1630): Naturphilosoph, Mathematiker, Astronom, Astrologe, Optiker und (evangelischer) Theologe
- Anlässlich der 2. Hochzeit im Jahr 1613 kaufte er für die Hochzeit einige Fässer Wein. Als der Wein eingekellert war, kam der Verkäufer mit einer Messrute und bestimmte den Inhalt für alle Fässer ohne Überlegung oder Rechnung nach der gleichen Methode. Die Messrute wurde mit ihrer metallenen Spitze durch das Spundloch quer bis zu den Rändern der beiden Böden eingeführt und die Marke am Spundloch ergab den Rauminhalt. Dieses Verfahren erschien ihm sehr ungenau.
- Kepler verfasste daraufhin die Schrift *Nova Stereometria doliorum vinariorum* 1615 („Neue Inhaltsberechnung von Weinfässern“), in der er nach überprüfbaren Methoden zur Inhaltsberechnung von Weinfässern suchte. Eine dieser Methoden bestand darin, die Krümmung des Fasses durch eine Parabel anzunähern, und das Volumen des Rotationskörpers zu ermitteln. Inhaltsberechnungen mit Hilfe von Parabeln können seit Archimedes exakt durchgeführt werden.



$$\int_a^b f(x) dx \approx (b-a) f\left(\frac{a+b}{2}\right)$$

$$\text{Error} \leq \frac{(b-a)^3}{24} \cdot \max_x \{|f''(x)|\}$$

- Trapezregel

$$\int_a^b f(x) dx \approx \frac{b-a}{2} (f(a) + f(b))$$

$$\text{Error} \leq \frac{(b-a)^3}{12} \cdot \max_x \{|f''(x)|\}$$

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left( f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

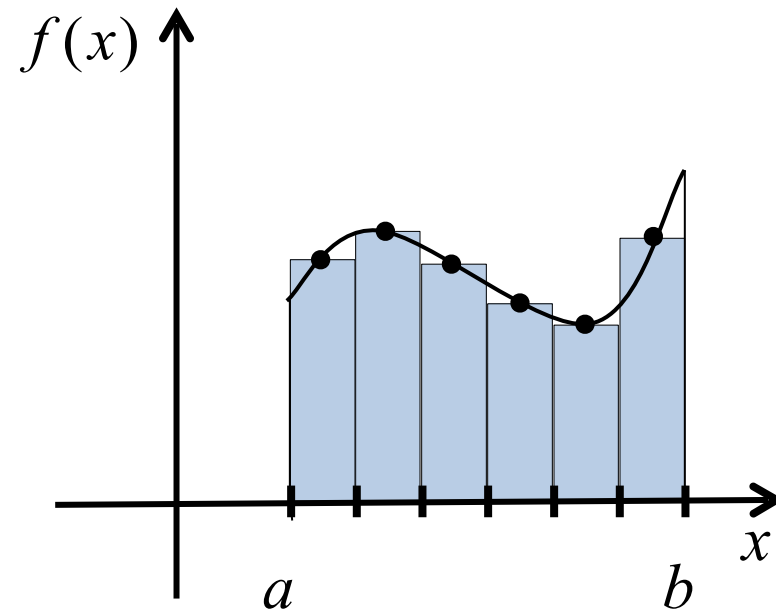
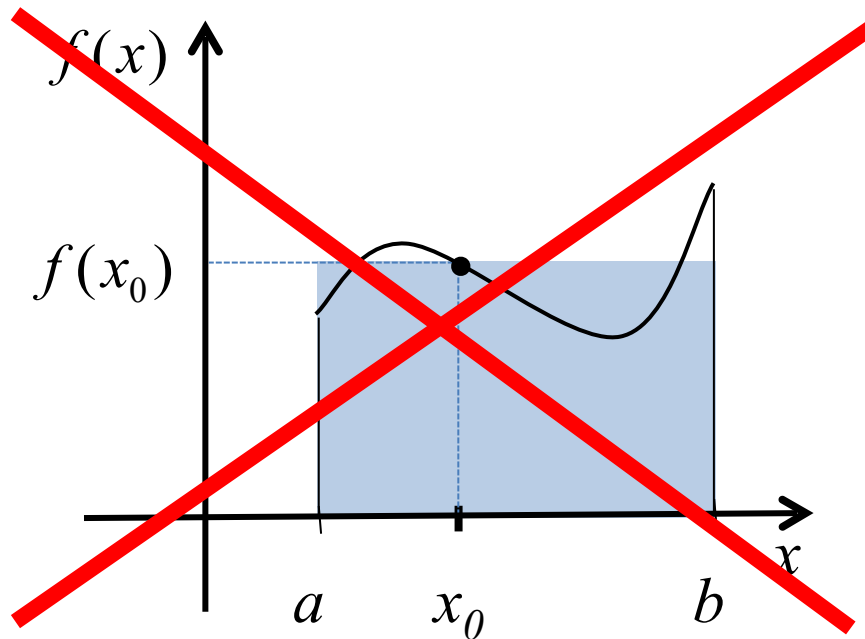
$$\text{Error} \leq \frac{(b-a)^5}{5760} \cdot \max_x \{|f^{(4)}(x)|\}$$

- Newtons 3/8 Regel

$$\int_a^b f(x) dx \approx \frac{b-a}{8} (f(a) + 3f(x_1) + 3f(x_2) + f(b))$$

$$\text{Error} \leq \frac{(b-a)^5}{6480} \cdot \max_x \{|f^{(4)}(x)|\}$$

- Diese Rechtecksregeln sind sehr ungenau, auch Trapez- und Simpson-Regel sind zu ungenau
- Sie werden so nicht verwendet sondern in **iterierter Form**, d.h.: *Man zerlegt das Intervall in (äquidistante) Teilintervalle und wendet auf jedes Teilintervall die Rechtecks- bzw. Trapezregel an*



- Newton-Cotes-Formeln von hoher Ordnung sind problematisch
  - „instabil“ (negative Gewichte)
  - Interpolations-Problematik
- Alternative: **Iterierte Newton-Cotes-Formel**  
 Zerlege Integrationsintervall in viele (gleich große) Teilintervalle der Länge  $h$

$$\int_a^b f(x) dx = \int_a^{a+h} f(x) dx + \int_{a+h}^{a+2h} f(x) dx + \dots + \int_{b-h}^b f(x) dx \quad \text{wobei} \quad \left( h = \frac{b-a}{n} \right)$$

und wende auf jedes Teilintervall eine NC-Formel niedrigen Grades an (z.B. Trapezregel oder Simpsonregel)

- **Iterierte Trapez-Regel (Trapezsumme)**

$$\int_a^b f(x) dx \approx T_f^{[a,b]}(h) = h \left( \frac{1}{2} f(a) + f(a+h) + f(a+2h) + \dots + f(a+(n-1)h) + \frac{1}{2} f(b) \right)$$

$$\text{Error} \leq \frac{h^2}{12} \cdot (b-a) \cdot \max_x \{ |f''(x)| \} = O(h^2) \quad [h = (b-a)/n]$$

- **Iterierte Simpsonregel (Simpsonsumme) für  $n$  gerade**

$$\int_a^b f(x) dx \approx S_f^{[a,b]}(h) = \frac{h}{3} (f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots \\ \dots + 2f(a+(n-2)h) + 4f(a+(n-1)h) + f(b))$$

$$\text{Error} \leq \frac{h^4}{180} \cdot (b-a) \cdot \max_x \{ |f^{(4)}(x)| \} = O(h^4) \quad [h = (b-a)/n]$$

- Beispiel

$$I_3 = \int_0^1 x^3 \exp(x/10) dx \quad \text{oder} \quad f(x) = x^3 \exp(x/10)$$

- Man kann sich überlegen, dass für  $x \in [0, 1]$ :

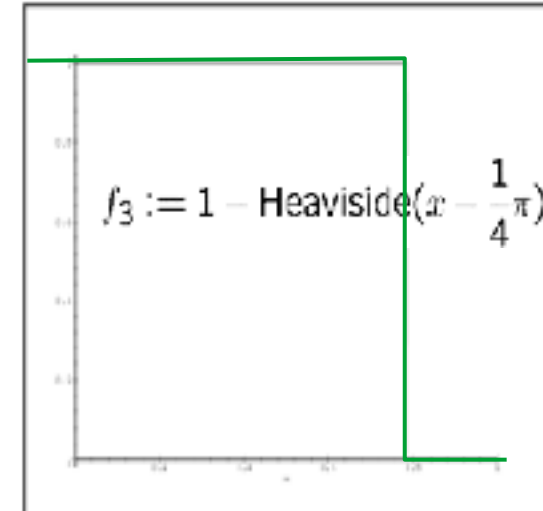
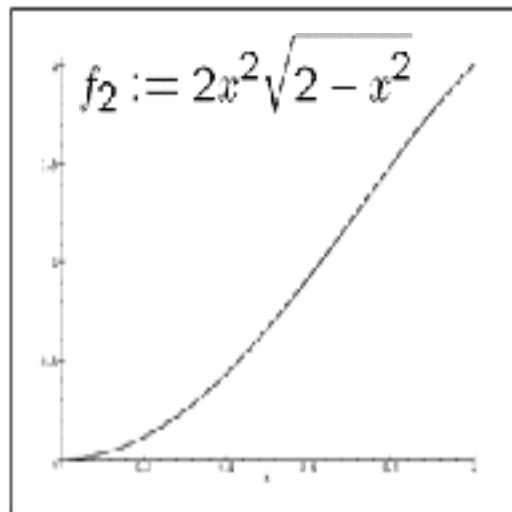
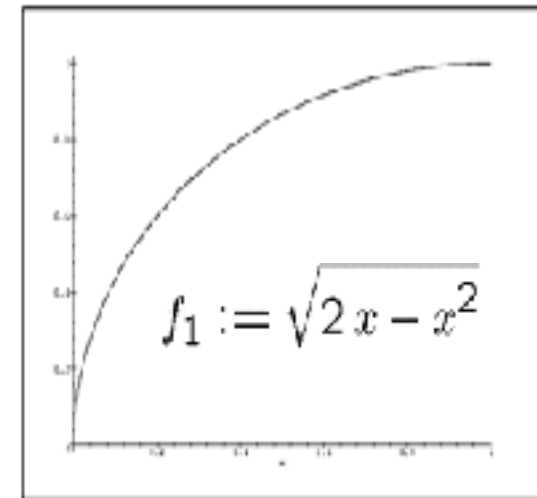
$$|f''(x)| \leq 10, \quad |f^{(4)}(x)| \leq 5.$$

- Wenn wir einen maximalen Fehler von  $10^{-6}$  (  $10^{-10}$  ) erreichen wollen, sind wir auf der sicheren Seite
  - ▶ für die Trapezsumme, wenn  $h \approx 10^{-3}$  (  $10^{-5}$  )
  - ▶ für die Simpsonsumme, wenn  $h \approx 7.5 \cdot 10^{-2}$  (  $7.5 \cdot 10^{-3}$  )

Beachte: Anzahl der Funktionsauswertungen bei gleichem  $h$  ist gleich!

Wir untersuchen drei Beispiele:

1. Viertelkreis
2. Glatte Funktion (entsteht aus #1 durch Variablensubstitution)
3. Unstetige Funktion





- Für alle drei Beispiele gilt:  $\int_0^1 f_i(x) dx = \frac{\pi}{4}$
- Deshalb können wir gut die Fehler studieren.
- Die mathematischen Fehlerschranken
  - ▶ sind im Fall 1 nutzlos, da die Ableitungen von  $f_1$  unbeschränkt sind;
  - ▶ liefern im Fall 2 Fehlerschranken;
  - ▶ sind im Fall 3 wiederum nutzlos, da diese Funktion nicht stetig ist.
- Beachte, dass man im Fall 3 natürlich die Fläche durch Approximation mit einer Treppenfunktion exakt berechnen könnte, wenn man die Sprungstelle genau kennt. Wir gehen davon aus, dass wir diese Stelle nicht kennen!

- Auswertung von Trapez- und Mittelpunktsregeln, sowie Simpson.
- Zahlenfriedhof
- Wie bringt man System hinein?

n	Links	Rechts	Mitte	Trapez	Simpson
1	0,0	1,0	0,866025404	0,5	
2	0,433012702	0,933012702	0,814841832	0,683012702	0,744016936
4	0,623927267	0,873927267	0,795982305	0,748927267	0,770898788
8	0,709954786	0,834954786	0,789171733	0,772454786	0,780297293
16	0,749563260	0,812063260	0,786737952	0,780813260	0,783599417
32	0,768150606	0,799400606	0,785872851	0,783775606	0,784763054
64	0,777011728	0,792636728	0,785566168	0,784824228	0,785173769
128	0,781288949	0,789101449	0,785457593	0,785195199	0,785318855
256	0,783373270	0,787279520	0,785419180	0,785326395	0,785370129
Exakt	0,785398164	0,785398164	0,785398164	0,785398164	0,785398164

- Die Trapez- und Mittelpunkt sind deutlich genauer als die rechte oder linke Rechtecksregel, Simpson noch besser.
- In allen Fällen nimmt der **Fehler** schön gleichmässig (aber relativ langsam) ab, wenn wir den Rechenaufwand erhöhen, d.h. die Maschenweite  $h$  fortgesetzt halbieren.

n	Links	Rechts	Mitte	Trapez	Simpson
1	0.785398164	0.214601837	0.080627241		
2	0.352385462	0.147614539	0.029443669	0.102385462	
4	0.161470897	0.088529104	0.010584142	0.036470897	0.005100871
8	0.075443378	0.049556623	0.003773570	0.012943378	0.001798747
16	0.035834904	0.026665096	0.001339789	0.004584904	0.000635109
32	0.017247558	0.014002442	0.000474687	0.001622558	0.000224394
64	0.008386435	0.007238565	0.000168005	0.000573935	7.93087E-05
128	0.004109215	0.003703285	5.94297E-05	0.000202965	2.80345E-05
256	0.002024893	0.001881357	2.10167E-05	7.17683E-05	9.912E-6
Quotient	≈ 2.03	≈ 1.97	≈ 2.83	≈ 2.83	≈ 2.83

- Die Trapez- und Mittelpunkt sind deutlich genauer als die rechte oder linke Rechtecksregel, Simpson deutlich besser
- In allen Fällen nimmt der **Fehler** schön gleichmäßig ab, Links  $\approx$  Rechts, besser ist Mitte  $\approx$  Trapez, noch besser Simpson (bei gleichen Aufwand ! )

n	Links	Rechts	Mitte	Trapez	Simpson
1	0.785398164	1.214601836	0.123960336	0.214601837	
2	0.45467925	0.545320751	0.023988037	0.045320750	0.011106278
4	0.239333644	0.260666357	0.005425847	0.010666357	8.8551E-4
8	0.122379745	0.127620255	0.001316144	0.002620255	6.178E-05
16	0.061847944	0.063152056	0.000326408	0.000652056	4.011E-06
32	0.031087176	0.031412824	8.14357E-05	0.000162824	2.534E-07
64	0.015584306	0.015665694	2.03481E-05	0.000040694	1.59E-08
128	0.007802328	0.007822672	5.0865E-06	1.01722E-05	1.2E-09
256	0.003903707	0.003908793	1.2719E-06	2.5427E-06	0
Quotient	$\approx 2.0$	$\approx 2.0$	$\approx .4.0$	$\approx 4.0$	$\approx 16$

- Mit der **gleichmäßigen Fehler-Reduktion** ist es jetzt nichts mehr, die Fehler springen hin und her.
- Trapez- und Simpsonregel sind jetzt auch *nicht* mehr eindeutig besser.

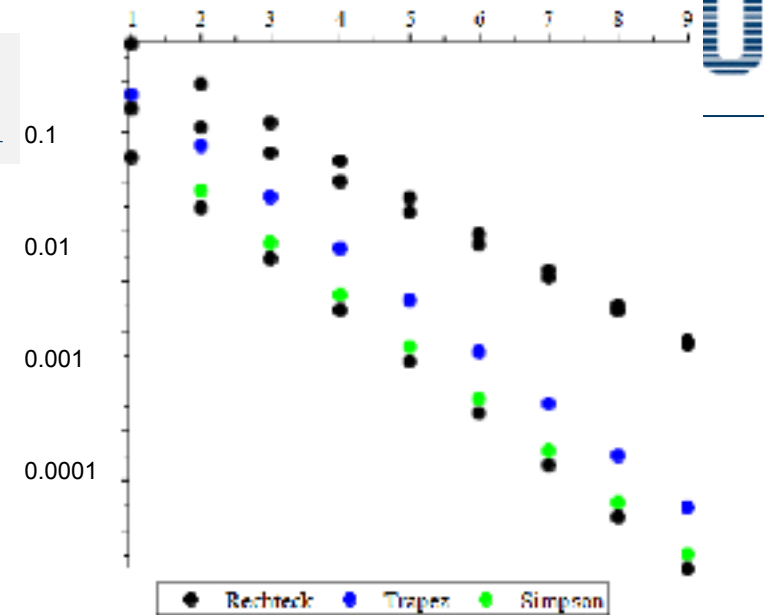
n	Links	Rechts	Mitte	Trapez	Simpson
1	1	0	1	0,5	
2	1	0,5	1	0,75	0,833333333
4	1	0,75	0,75	0,875	0,916666667
8	0,875	0,75	0,75	0,8125	0,791666667
16	0,8125	0,75	0,8125	0,78125	0,770833333
32	0,8125	0,78125	0,78125	0,796875	0,802083333
64	0,796875	0,78125	0,78125	0,7890625	0,786458333
128	0,7890625	0,78125	0,7890625	0,78515625	0,783854167
256	0,7890625	0,78515625	0,78515625	0,787109375	0,787760417
exakt	0,785398164	0,785398164	0,785398164	0,785398164	0,785398164

- Mit der **gleichmäßigen Fehler-Reduktion** ist es jetzt nichts mehr, die Fehler springen hin und her.
- Trapez- und Simpsonregel sind jetzt auch *nicht* mehr eindeutig besser.

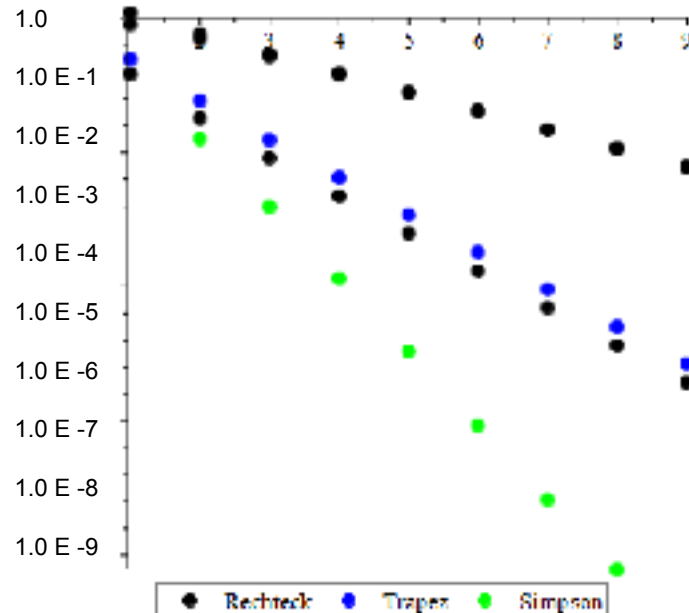
n	Links	Rechts	Mitte	Trapez	Simpson
1	0.214601837	0.785398164	0.214601837	0.285398164	
2	0.214601837	0.285398164	0.214601837	0.035398164	0.047935170
4	0.214601837	0.035398164	0.035398164	0.089601837	0.131268503
8	0.089601837	0.035398164	0.035398164	0.027101837	0.006268503
16	0.027101837	0.035398164	0.027101837	0.004148164	0.014564830
32	0.027101837	0.004148164	0.004148164	0.011476837	0.016685170
64	0.011476837	0.004148164	0.004148164	0.003664337	0.001060170
128	0.003664337	0.004148164	0.003664337	0.000241914	0.001543997
256	0.003664337	0.000241914	0.000241914	0.001711212	0.002362253
Quotient	???	???	???	???	???

$f_1$

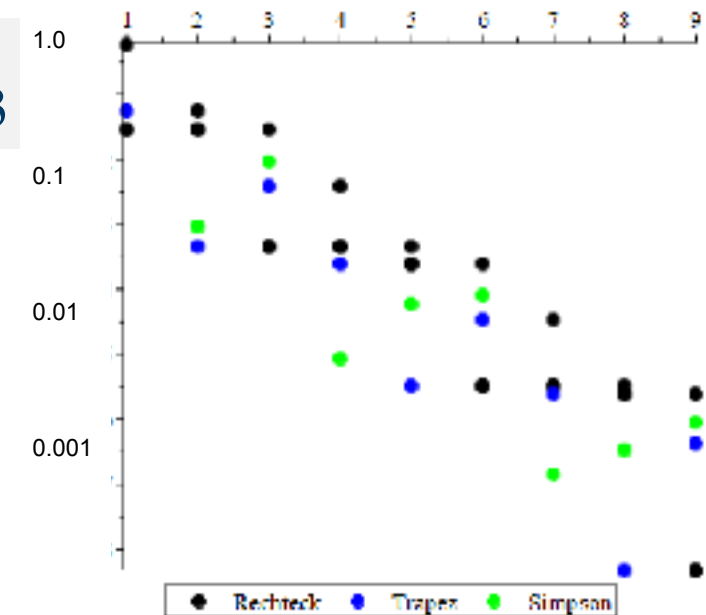
- jeweils Fehler (log-Skala) über Verfeinerungsstufe  $k$  : ( $2^{k-1}$  Teilintervalle)



$f_2$



$f_3$



- Bei den ersten beiden Beispielen verringert sich der Fehler gleichmäßig. Genauer:  
Bei Schrittweitenhalbierung jeweils um den etwa gleichen Faktor.
- Hintergrund: Es gibt eine theoretische Fehlerabschätzung (Euler-MacLaurin-Reihe)

$$I(h) - I_0 = c \cdot h^\tau + O(h^\sigma) \quad (\text{wobei } \sigma > \tau > 0)$$

dabei ist  $I_0$  der exakte Wert und  $I(h)$  der zur Schrittweite  $h$  berechnete Näherungswert.

- Dabei ist  $c$  meist unbekannt, oft auch  $\tau$  und  $\sigma$ . Es gilt:

$$\frac{I(2h) - I_0}{I(h) - I_0} = \frac{c \cdot (2h)^\tau + d_1 \cdot h^\sigma}{c \cdot h^\tau + d_2 \cdot h^\sigma} \rightarrow 2^\tau \quad (\text{für } h \rightarrow 0)$$



- Hintergrund: Es gibt eine theoretische Fehlerabschätzung

$$I(h) - I_0 = c \cdot h^\tau + O(h^\sigma) \quad (\text{wobei } \sigma > \tau > 0)$$

- Bei Kenntnis von  $\tau$  kann man durch einen einfachen Trick, das Ergebnis (stark) verbessern:

- Geschicktes kombinieren von

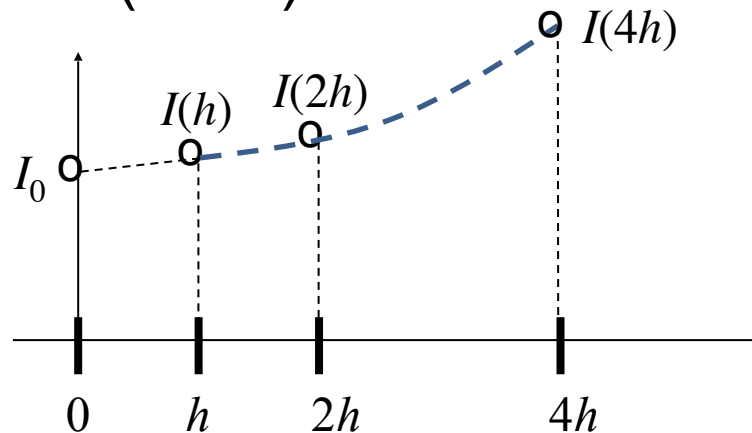
$$I(h) = I_0 + c \cdot h^\tau + O(h^\sigma) \quad \text{und} \quad I(2h) = I_0 + c \cdot (2h)^\tau + O(h^\sigma)$$

ergibt 
$$2^\tau \cdot I(h) - I(2h) = 2^\tau \cdot I_0 - I_0 + O(h^\sigma)$$

oder

$$I^{(1)}(h) = \frac{2^\tau \cdot I(h) - I(2h)}{2^\tau - 1} - I_0 = O(h^\sigma)$$

- Für  $f_1$  und  $f_2$  (aber nicht für  $f_3$ ) kann man die Regelmäßigkeit in der Konvergenz des Fehlers systematisch zur Verbesserung der Genauigkeit nützen.
- Dieses Prinzip heißt **Extrapolation** und geht auf Richardson (1927) zurück.



- Die Anwendung der Extrapolation bei der numerischen Integration hat Romberg 1955 entdeckt, deshalb heißt die systematische Anwendung auch **Romberg-Quadratur**.

- Für  $f_1$  Betrachtung der Fehler
  - ▶ Die Fehler verbessern sich bei jeder Halbierung des Gitterabstands um einen fast konstanten Faktor 2 ( $\tau = 1$ )
  - ▶ Dies kann durch Extrapolation genutzt werden:

$$I^{(1)}(h) = \frac{2 \cdot I(h) - I(2h)}{2 - 1} \approx I_0 \quad (\text{lineare Extrapolation})$$

- ▶ Die extrapolierten Werte  $I^{(1)}(h)$  zeigen die gleiche Gesetzmäßigkeit mit dem Faktor  $2.83 \dots \approx 2^{3/2}$

$$I^{(2)}(h) = \frac{2.83 \cdot I^{(1)}(h) - I^{(1)}(2h)}{2.83 - 1} \approx I_0$$

- ▶ Führt man immer weitere Extrapolationsschritte durch, erreicht man schnell Genauigkeiten besser als  $10^{-10}$

- Berechnung des Integrals von  $f_1$  mit linker Rechtecksregel
- R.-Extrapolation liefert gleiche(bessere) Approximation mit wesentlich weniger Aufwand (# Funktionsauswertungen)!

$$I^{(1)}(h) = \frac{2 \cdot I(h) - I(2h)}{2 - 1}$$

$$I^{(2)}(h) = \frac{2.83 \cdot I^{(1)}(h) - I^{(1)}(2h)}{2.83 - 1}$$

n	$I(h)$	Fehler	$I^{(1)}(h)$	Fehler	$I^{(2)}(h)$	Fehler
0	0,0	-0,785398164				
1	0,433012702	-0,352385462	0,866025404	0,080627241		
2	0,623927267	-0,161470897	0,814841832	0,029443669	0,786872666	0,001474503
4	0,709954786	-0,075443378	0,795982305	0,010584142	0,785676552	0,000278389
8	0,749563260	-0,035834904	0,789171733	0,003773570	0,785450109	5,19453E-05
16	0,768150606	-0,017247558	0,786737952	0,001339788	0,785408016	9,8524E-06
32	0,777011728	-0,008386435	0,78587285	0,000474687	0,785400117	1,9536E-06
64	0,781288949	-0,004109215	0,78556617	0,000168006	0,785398584	4,205E-07
128	0,783373270	-0,002024893	0,785457592	5,94286E-05	0,78539826	9,65E-08
Quotient		≈ 2.03		≈ 2.83		

- Berechnung des Integrals von  $f_2$  mit Trapezregel
- R.-Extrapolation liefert bessere Approximation mit wesentlich weniger Aufwand (# Funktionsauswertungen)!

n	$I(h)$	Fehler	$I^{(1)}(h)$	Fehler	$I^{(2)}(h)$	Fehler
0						
1	0,830718914	0,045320750				
2	0,796064520	0,010666357	0,784513055	-0,000885108	0,836813926	0,051415762
4	0,788018418	0,002620255	0,785336385	-6,17785E-05	0,785391274	-6,8899E-06
8	0,786050219	0,000652056	0,785394153	-4,0106E-06	0,785398004	-1,594E-07
16	0,785560987	0,000162824	0,785397910	-2,532E-07	0,785398161	-2,7E-09
32	0,785438858	4,06942E-05	0,785398148	-1,59E-08	0,785398163	
64	0,785408336	1,01723E-05	0,785398161	-2,1E-09		
128	0,785400706	2,5429E-06	0,785398163			
<b>Quotient</b> 0,785398163 $\approx 4.0$ $\approx 16.0$						

- Für glatte Funktionen (zB  $f_2$ ) ist die Richardson-Extrapolation besonders effizient, denn es gilt die Euler-Maclaurin'sche Summenformel:
- Theorem:** Ist  $f \in C^{2m+2}$  dann gilt für den Fehler der Trapezregel mit Schrittweite  $h$  folgendes:

$$T_f(h) - I_0 = c_1 \cdot h^2 + c_2 \cdot h^4 + \dots + c_m \cdot h^{2m} + O(h^{2m+2})$$

- Sukzessive Richardson-Extrapolation mit Exponenten

$$\tau = 2, 4, 6, \dots, 2m \text{ also}$$

$$2^\tau = 4, 16, 64, \dots, 4^m$$

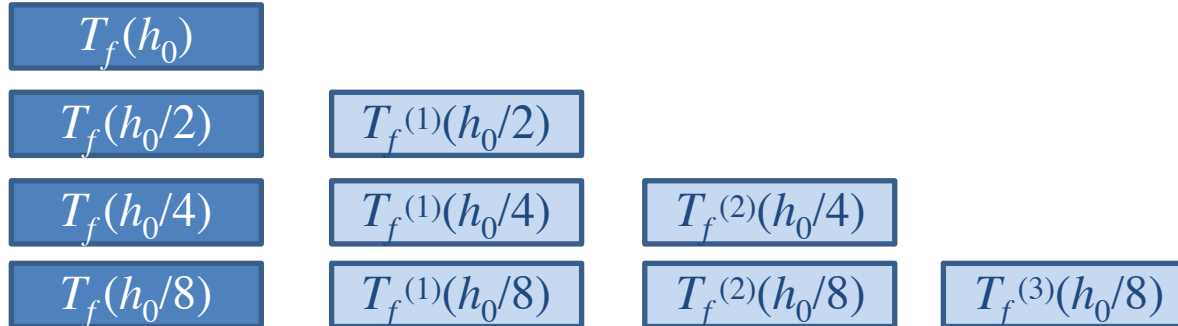
$$T_f^{(1)}(h) = \frac{4 \cdot T_f(h) - T_f(2h)}{4 - 1}$$

$$T_f^{(2)}(h) = \frac{4^2 \cdot T_f^{(1)}(h) - T_f^{(1)}(2h)}{4^2 - 1}$$

...

$$T_f^{(m)}(h) = \frac{4^m \cdot T_f^{(m-1)}(h) - T_f^{(m-1)}(2h)}{4^m - 1}$$

- Berechnungsschema und Pseudo-Code des Romberg-V.



```

▶  $h = (b-a)/n_0;$ 
▶  $T[1,1] = T_f(h);$ 
▶ for  $k = 1, 2, 3, \dots, m$ 
    ▶  $M = M_f(h);$ 
    ▶  $T[k+1,1] = (T[k,1] + M_f(h))/2;$            #  $= T_f(h/2);$ 
       $h = h/2;$ 
    ▶ for  $n = 1, 2, \dots, k$  do
        ★  $T[k+1,n+1] = (4^n \cdot T[k+1,n] - T[k,n]) / (4^n - 1);$ 

```

- Die Funktion  $f_2(x) = x^2 \sqrt{2 - x^2}$  ist im Intervall  $[0,1]$  beliebig oft differenzierbar, somit kann man das Integral  $\int_0^1 f_2(x) dx$  mit dem Romberg-Verfahren berechnen.
- Man erhält

$h$	$T_f(h)$	$T_f^{(1)}(h)$	$T_f^{(2)}(h)$	$T_f^{(3)}(h)$	$T_f^{(4)}(h)$
$2^{-1}$	0,83071891				
$2^{-2}$	0,79606452	0,78451306			
$2^{-3}$	0,78801842	0,78533638	0,785391273		
$2^{-4}$	0,78605022	0,78539415	0,785398004	0,785398111	
$2^{-5}$	0,78556099	0,78539791	0,785398161	0,785398163	0,785398163
exact	0,785398163	0,785398163	0,785398163	0,785398163	0,785398163



- Volumenberechnung durch zweidimensionale Integration

$$\iint_G g(x, y) \, dx dy$$

- bzw. Integrale trivariater Funktionen

$$\iiint_G h(x, y, z) \, dx dy dz$$



- Diese kann man bekanntlich auf 1D-Integrale zurückführen:

$$\iint_G g(x, y) \, dx dy = \int_*^* \left( \int_*^* g(x, y) \, dx \right) dy$$

$$\iiint_G h(x, y, z) \, dx dy dz = \int_*^* \left( \int_*^* \left( \int_*^* h(x, y, z) \, dx \right) dy \right) dz$$



$$\iint_G g(x, y) \, dx dy = \int_*^* \left( \int_*^* g(x, y) \, dx \right) dy$$

- Integrationsgebiet

- ▶ Rechteck  $G = [a, b] \times [c, d]$ , dann sind die Grenzen klar:

$$\iint_G g(x, y) \, dx dy = \int_c^d \left( \int_a^b g(x, y) \, dx \right) dy$$

- ▶ Beliebiges Gebiet:  
Erweitere Integrand auf umgebendes Rechteck

$$\tilde{g}(x, y) = \begin{cases} g(x, y) & \text{falls } (x, y) \in G \\ 0 & \text{sonst} \end{cases}$$

**Beachte: Diese Erweiterung ist unstetig !**

- Numerische Berechnung: Beide Integrale (inneres und äußeres) mittels Quadraturformeln bestimmen (Tensorproduktansatz)

$$\begin{aligned} \int_a^b \left( \int_c^d g(x, y) dy \right) dx &= \sum_{i=0}^n w_i \left( \int_c^d g(x_i, y) dy \right) = \\ &= \sum_{i=0}^n w_i \left( \sum_{j=1}^m v_j g(x_i, y_j) \right) = \sum_{i=0}^n \sum_{j=0}^m w_i v_j g(x_i, y_j) = \mathbf{w}^T \mathbf{S}_g \mathbf{v} \end{aligned}$$

- Speichere Sample-Punkte in Matrix  $\mathbf{S}_g = [g(a + i \cdot h, c + j \cdot k)]$  und multipliziere dies von links und rechts mit 1D-Gewichten  $\mathbf{w}^T = [w_0, w_1, \dots, w_n]$ ,  $\mathbf{v} = \dots$

- Beispiel: Tensorprodukt der (iterierten) Trapezregel:

$$\int_a^b \left( \int_c^d g(x, y) dy \right) dx \approx T_g^{[a,b] \times [c,d]}(h, k) = h \cdot k \cdot \mathbf{w}^T \mathbf{S}_g \mathbf{v}$$

wobei  $h = \frac{b-a}{n}, k = \frac{d-c}{m},$

$$\mathbf{w}^T = [0.5 \quad 1 \quad \dots \quad 1 \quad 0.5], \mathbf{v}^T = [0.5 \quad 1 \quad \dots \quad 1 \quad 0.5]$$

- Tensorprodukt der (iterierten) Simpsonregel

$$\int_a^b \left( \int_c^d g(x, y) dy \right) dx \approx S_g^{[a,b] \times [c,d]}(h, k) = \frac{h \cdot k}{9} \cdot \mathbf{w}^T \mathbf{S}_g \mathbf{v}$$

wobei  $\mathbf{w}^T = [1 \quad 4 \quad 2 \quad 4 \quad \dots \quad 4 \quad 1], \mathbf{v}^T = \dots$

- Fehlerabschätzungen:  
die von 1D-Fall bekannten Fehlerabschätzungen gelten auch hier, zB:

- iterierte Trapezregel in 2D:

$$\left| \int_a^b \left( \int_c^d g(x, y) dy \right) dx - T_g^{[a,b] \times [c,d]}(h, k) \right| = O(h^2 + k^2)$$

- Iterierte Simpsonregel in 2D:

$$\left| \int_a^b \left( \int_c^d g(x, y) dy \right) dx - S_g^{[a,b] \times [c,d]}(h, k) \right| = O(h^4 + k^4)$$

- aber nur bei entsprechender Differenzierbarkeit **und** rechteckigem Integrationsgebiet !
- In diesem Fall ist auch Romberg machbar!

- Normzellen-Verfahren (uniform, hierarchisch)
- Monte Carlo Verfahren
- Nutze Polynominterpolation (→ **Newton-Cotes Formeln**)
  - Rechtecksregeln, Trapez, Simpson
- **Iterierte** oder summierte **Newton-Cotes-Formeln**
- **Richardson-Extrapolation** als elegante Möglichkeit die Genauigkeit zu verbessern:
  - vor allem bei glatten Integranden → **Romberg-Verfahren**
  - aber mit Geschick auch im Fall singulärer Integranden anwendbar
- Numerische Integration in **2D/3D** Dimensionen mittels Tensor-Produkten,
- Es gibt (viele) weitere interessante Verfahren !!
  - Besonders wichtig: Gauss-Quadratur!