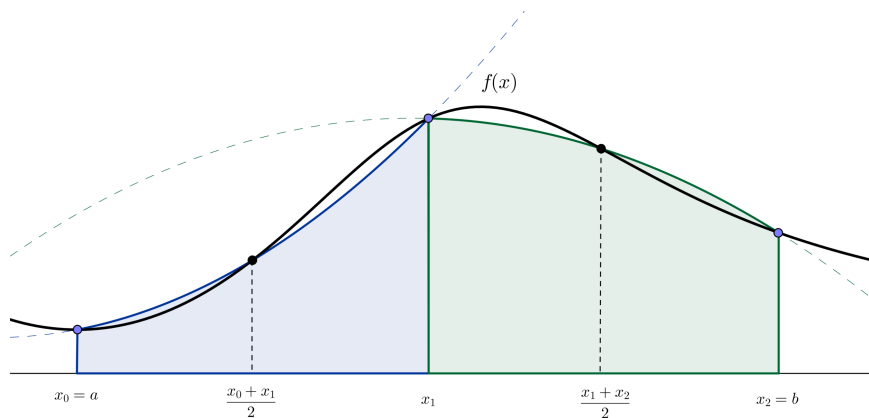


Numerische Integration

Sebastian Hirnschall
Rafael Dorigo

Betreut von:
Markus Wess Dipl.-Ing.



Abstract

Vordergründiges Ziel dieser Arbeit ist es, exponentiell abfallende Funktionen auf dem unbeschränkten Intervall $[0, \infty)$ numerisch zu integrieren und verschiedene Vorgehensweisen zu vergleichen. Zunächst werden zwei unterschiedliche Möglichkeiten um auch das Integral über unbeschränkte Intervalle zu approximieren vorgestellt und untersucht wie sich verschiedener Eigenschaften der zu integrierenden Funktion auf das Konvergenzverhalten des näherungsverfahrens auswirken. Anschließend werden die beiden gewählten Methoden miteinander verglichen. Das letzte Kapitel befasst sich mit der Frage, inwiefern sich das Konvergenzverhalten durch die Wahl unterschiedlicher Gewichtsfunktionen ändert. Dabei wird verglichen wie schnell der Fehler der zuvor gewählten Vorgehensweisen für zwei häufig verwendete Gewichtsfunktionen (e^{-x} und e^{-x^2}) konvergieren.

Inhaltsverzeichnis

1	Einleitung	4
2	Summierte Trapezformel auf einem endlichen Intervall	5
2.1	Von T abhängiger Fehler	6
2.2	Von n abhängiger Fehler	8
3	Gauß-Laguerre Quadratur	13
4	Vergleich der Methoden aus Abschnitt 2 und 3	19

1 Einleitung

Um exponentiell abfallende Funktionen auf dem unbeschränkten Intervall $[0, \infty)$ zu integrieren, bieten sich zwei Möglichkeiten an. Das Abschneiden des unbeschränkten Intervalls und Anwendung einer Quadraturformel für ein beschränktes Intervall $[0, T]$ für $T > 0$ und die Anwendung einer Gaußquadratur für die Gewichtsfunktion $w(x) = e^{-x}$ (Gauss–Laguerre Quadratur).

2 Summierte Trapezformel auf einem endlichen Intervall

LEMMA 2.0.1

Sei $f : [0, \infty) \rightarrow \mathbb{R}$ eine beschränkte Funktion. Des weiteren definieren wir für eine Menge $M \subset [0, \infty)$ das gewichtete Integral von f mit

$$Q_M(f) := \int_M f(x) \exp(-x) dx$$

Das Integral $Q_{[0, \infty)}(f)$ kann man approximieren, indem man für ein $T > 0$ das Integral auf dem beschränkten Intervall $Q_{[0, T]}(f)$ durch die summierte Trapezformel $Q_{h, T}(f)$ berechnet. Es folgt eine Fehlerabschätzung der Form

$$|Q_{[0, \infty)}(f) - Q_{h, T}(f)| \leq C_1 \varepsilon_T + C_2 T \varepsilon_h \quad (2.1)$$

wobei die Terme $\varepsilon_T, \varepsilon_h$ lediglich von T bzw. h abhängen und C_1, C_2 jeweils von T, h unabhängig sind. Des weiteren wir vorausgesetzt, dass f hinreichend glatt ist und die Supremumsnorm der zweiten Ableitung von f beschränkt ist.

BEWEIS

Mit (vgl. Nannen 2019, S. 40–41):

und $C_1 := \|f\|_\infty$, $\varepsilon_T := e^{-T}$, $C_2 := \|f''\|_\infty$, $\varepsilon_h := \frac{h^2}{12}$ folgt unmittelbar

$$\begin{aligned} |Q_{[0, \infty)}(f) - Q_{h, T}(f)| &\leq |Q_{[0, \infty)}(f) - Q_{[0, T]}(f)| + |Q_{[0, T]}(f) - Q_{h, T}(f)| \\ &= |Q_{(T, \infty)}(f)| + |Q_{[0, T]}(f) - Q_{h, T}(f)| \\ &\leq \|f\|_\infty \int_T^\infty e^{-x} dx + \left| -\frac{T}{12} h^2 f''(\xi) \right| \\ &= C_1 e^{-T} + T \frac{h^2}{12} |f''(\xi)| \\ &\leq C_1 \varepsilon_T + \|f''\|_\infty T \frac{h^2}{12} \\ &= C_1 \varepsilon_T + C_2 T \varepsilon_h \end{aligned}$$

□

BEMERKUNG. Damit der Fehler minimal ist, soll $\varepsilon_T \approx T \varepsilon_h$ gelten. Anders angeschrieben soll also $h \approx \sqrt{\frac{12e^{-T}}{T}}$ bzw. $n \approx \sqrt{\frac{T^3}{12e^{-T}}}$ erfüllt sein.

Die Summierte Trapezregel kann zum Beispiel wie folgt implementiert werden:

```
1 double sumtrapez(double t, double n, double fkt(double)){  
    double h = t/n;  
3    double sum = 0;  
    for(double i = 1; i <= n; ++i){  
5        sum += (fkt(i*h)+fkt((i-1)*h))/2*h;  
    }  
7    return sum;  
}
```

Listing 1: Summierte Trapezregel

2.1 Von T abhängiger Fehler

Da für Funktionen f, g , die auf dem Intervall $[a, b]$ Riemann-integrierbar sind gilt (vgl. Kaltenböck 2014, S. 271):

$$f(x) \leq g(x) : x \in [a, b] \Rightarrow \int_a^b f(x)dx \leq \int_a^b g(x)dx,$$

ergibt sich für den Fehler $C_1 \varepsilon_T$ aus Gleichung (2.1) für andere Gewichtsfunktionen $\omega : f(x)\omega(x) \leq f(x)e^{-x}, x \in [T, \infty)$

$$\begin{aligned} \left| \int_0^\infty f(x)e^{-x}dx - \int_0^T f(x)e^{-x}dx \right| &= \left| \int_T^\infty f(x)e^{-x}dx \right| \geq \\ \left| \int_T^\infty f(x)\omega(x)dx \right| &= \left| \int_0^\infty f(x)\omega(x)dx - \int_0^T f(x)\omega(x)dx \right| \end{aligned}$$

Der bei Abschnitt 2 durch das Abschneiden des Intervalls $[0, \infty)$ entstehende Fehler ist also stark von der Wahl der Gewichtsfunktion abhängig. (Abb. 1b) Durch die Wahl einer geeigneten Gewichtsfunktion kann die Konvergenz des Integrationsverfahrens also beschleunigt werden.

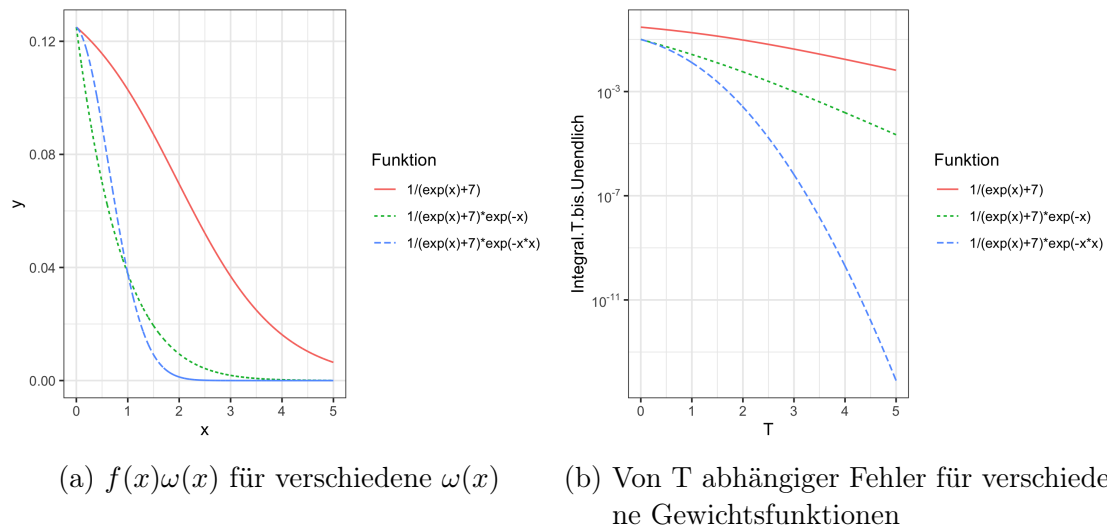


Abbildung 1: Vergleich verschiedener Gewichtsfunktionen

Für eine Zerlegung $Z = \{\xi_j : j = 0, \dots, n(Z) : \xi_j = jh + \xi_0\}, h \in \mathbb{R}$ von $[0, T]$ bleibt für $T \rightarrow \infty$ nur der von h abhängige Fehler.

```
def f(x):
    return 1/(np.exp(x)+7)*np.exp(-x)

def sumtrapez(t,h):
    xs=np.arange(0,t,h)
    fs=f(xs)
    return h/2*(fs[0]+2*sum(fs[1:-1])+fs[-1])

h=0.001
errors=[]
T = 2*np.arange(12)
for t in T:
    errors.append(abs(1/49*(7-np.log(8))-sumtrapez(t,h)))
plt.loglog(T,errors)
plt.grid(which='major', linewidth=0.5)
plt.xlabel('T', size='large')
plt.ylabel('Fehler', size='large')
```

Listing 2: Implementierung des Fehlers mit fixem h

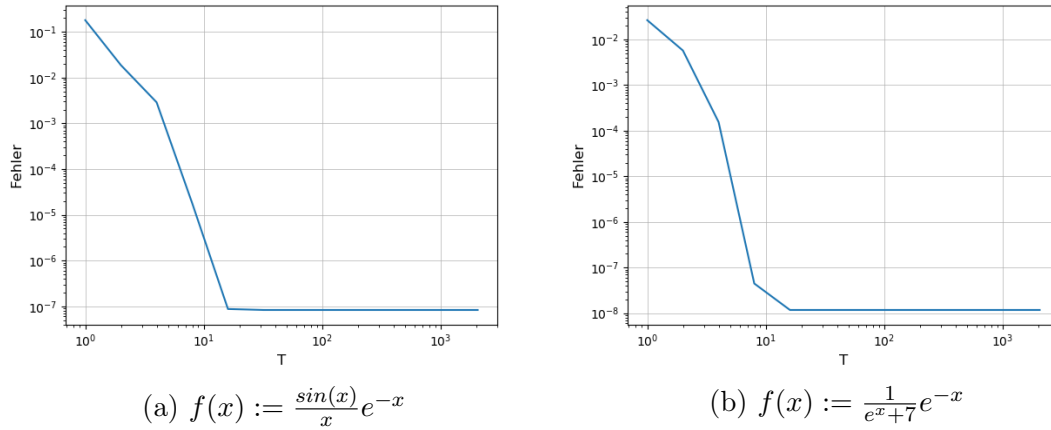


Abbildung 2: Fehler abhängig von T.

Man bemerkt in Abb. 2, dass bei verschiedenen Funktionen sich die Konvergenzgeschwindigkeit des Integrationsfehler ändert.

2.2 Von n abhängiger Fehler

SATZ 2.2.1

Für auf einem Intervall $[a, b] : 0 \leq a < b$ Riemann-integrierbare Funktionen $f, g \in C^2([a, b])$ und eine Zerlegung $Z = \{\xi_i : i = 0, \dots, n(Z)\}$ von $[a, b]$ ist der von n abhängige Fehler für f durch

$$\tau : (C^2([a, b]), M) \rightarrow \mathbb{R}, \tau(f, Z) = \left| \sum_{i=1}^{n(Z)} \left(\int_{\xi_{i-1}}^{\xi_i} f(x) dx - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) + f(\xi_i)}{2} \right) \right) \right|$$

gegeben, wobei $d(\xi_{i-1}, \xi_i)$ der Abstand zwischen ξ_{i-1} und ξ_i ist und M die Menge der Zerlegungen von $[a, b]$ ist und es gilt:

$$\forall x \in [a, b] : \left| \frac{d^2}{dx^2} f(x) \right| \geq \left| \frac{d^2}{dx^2} g(x) \right| \geq 0 \Rightarrow \tau(f, Z) \geq \tau(g, Z).$$

BEWEIS

Dass $\tau(f, Z)$ der Fehler für eine der in Satz 2.2.1 beschriebenen Funktionen f und eine Zerlegung Z ist, folgt direkt aus der Definition der summierten Trapezformel.

Da $\forall x \in [a, b] : |\frac{d^2}{dx^2}f(x)| \geq |\frac{d^2}{dx^2}g(x)|$, genügt es ein einzelnes Intervall (ξ_{i-1}, ξ_i) zu betrachten. Für $c \in \mathbb{R}$ gilt

$$\begin{aligned} \tau(f, \{\xi_{i-1}, \xi_i\}) &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) dx - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) + f(\xi_i)}{2} \right) \right| = \\ &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) dx - \int_{\xi_{i-1}}^{\xi_i} c dx + d(\xi_{i-1}, \xi_i) c - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) + f(\xi_i)}{2} \right) \right| = \\ &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) - c dx - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) - c + f(\xi_i) - c}{2} \right) \right| = \tau(f - c, \{\xi_{i-1}, \xi_i\}). \end{aligned}$$

Es kann also $\min(|f(\xi_{i-1})|, |f(\xi_i)|) = 0$ vorausgesetzt werden. Für die Verbindungsgerade $s(x) := f(\xi_{i-1}) + \frac{f(\xi_i) - f(\xi_{i-1})}{\xi_i - \xi_{i-1}}x$ der Punkte $(\xi_{i-1}, f(\xi_{i-1}))$ und $(\xi_i, f(\xi_i))$ gilt somit $\min(|s(\xi_{i-1})|, |s(\xi_i)|) = 0$ und:

$$\begin{aligned} \tau(f, \{\xi_{i-1}, \xi_i\}) &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) dx - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) + f(\xi_i)}{2} \right) \right| = \\ &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) dx - \int_{\xi_{i-1}}^{\xi_i} s(x) dx + d(\xi_{i-1}, \xi_i) \frac{s(\xi_{i-1}) + s(\xi_i)}{2} - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) + f(\xi_i)}{2} \right) \right| = \\ &= \left| \int_{\xi_{i-1}}^{\xi_i} f(x) - s(x) dx - \left(d(\xi_{i-1}, \xi_i) \frac{f(\xi_{i-1}) - s(\xi_{i-1}) + f(\xi_i) - s(\xi_i)}{2} \right) \right| = \tau(f - s, \{\xi_{i-1}, \xi_i\}) \end{aligned}$$

Es existiert also für jedes f ein $\hat{f}$ der Form $\hat{f}(x) = (f(x) - s(x) - c)$, das wie oben gezeigt $\hat{f}(\xi_{i-1}) = \hat{f}(\xi_i) = 0$, $\tau(f, Z) = \tau(\hat{f}, Z)$ und somit

$$\tau(f, \{\xi_{i-1}, \xi_i\}) = \left| \int_{\xi_{i-1}}^{\xi_i} \hat{f}(x) dx - \left(d(\xi_{i-1}, \xi_i) \frac{\hat{f}(\xi_{i-1}) + \hat{f}(\xi_i)}{2} \right) \right| = \left| \int_{\xi_{i-1}}^{\xi_i} \hat{f}(x) dx \right|$$

erfüllt. Weiters gilt

$$\frac{d^2}{dx^2} \hat{f}(x) = \frac{d^2}{dx^2} (f(x) - s(x) - c) = \frac{d^2}{dx^2} f(x) - \frac{d^2}{dx^2} s(x) - \frac{d^2}{dx^2} c = \frac{d^2}{dx^2} f(x)$$

und weil außerdem $f, g \in C^2([a, b])$ und $\forall x \in [a, b] : |\frac{d^2}{dx^2}f(x)| \geq |\frac{d^2}{dx^2}g(x)| \geq 0$, kann für den Fehler wegen $\left| \int_{\xi_{i-1}}^{\xi_i} \hat{f}(x) dx \right| = \left| \int_{\xi_{i-1}}^{\xi_i} -\hat{f}(x) dx \right|$ vorausgesetzt werden, dass $\forall x \in [a, b] : \frac{d^2}{dx^2}f(x) \leq \frac{d^2}{dx^2}g(x) \leq 0$ gilt.

Nun folgt aus dem Satz von Rolle die Existenz von $x_{1,2}$ für die gilt, dass $\hat{f}'(x_1) = \hat{g}'(x_2) = 0$ ist. Weil $\hat{f}'$ und $\hat{g}'$ stetig und streng monoton sind und $[a, b]$ kompakt ist, ist $\hat{f}'([a, b]) \cap \hat{g}'([a, b]) = [u, v]$ ein nicht leeres, abgeschlossenes Intervall.

Sei nun $h(x) := f(x) - g(x)$, $h'(x) = f'(x) - g'(x)$, dann folgt aus der Surjektivität von $\hat{f}', \hat{g}'$ auf $[u, v]$ die Existenz von ζ_1, ζ_2 , für die folgendes gilt:

$$\begin{aligned}\hat{f}'(\zeta_1) &= u \wedge \hat{g}'(\zeta_1) \geq u \\ \hat{f}'(\zeta_2) &= v \wedge \hat{g}'(\zeta_2) \leq v\end{aligned}$$

Es ist also entweder $h'(\zeta_1) \leq 0 \leq h'(\zeta_2)$ oder $h'(\zeta_1) \geq 0 \geq h'(\zeta_2)$. Aus dem Zwischenwertsatz folgt die Existenz eines $\zeta \in [u, v]$ für das $h'(\zeta) = 0$ ist, woraus $\hat{f}'(\zeta) = \hat{g}'(\zeta)$ folgt.

Das Intervall $[a, b]$ kann nun in zwei Intervalle $[a, \zeta], [\zeta, b]$ aufgeteilt werden. Für $h''(x) = f''(x) - g''(x)$ folgt dann mit dem Mittelwertsatz für $\tilde{a} \in [a, \zeta]$ und $\tilde{b} \in [\zeta, b]$:

$$\begin{aligned}0 \geq h''(x_0) &= \frac{\hat{f}'(\zeta) - \hat{g}'(\zeta) - \hat{f}'(\tilde{a}) + \hat{g}'(\tilde{a})}{\zeta - \tilde{a}} = \frac{\hat{g}'(\tilde{a}) - \hat{f}'(\tilde{a})}{\zeta - \tilde{a}} \\ &\quad \hat{f}'(\tilde{a}) \leq \hat{g}'(\tilde{a}) \\ 0 \geq h''(x_0) &= \frac{\hat{f}'(\tilde{b}) - \hat{g}'(\tilde{b}) - \hat{f}'(\zeta) + \hat{g}'(\zeta)}{\tilde{b} - \zeta} = \frac{\hat{f}'(\tilde{b}) - \hat{g}'(\tilde{b})}{\zeta - \tilde{b}} \\ &\quad \hat{g}'(\tilde{b}) \geq \hat{f}'(\tilde{b})\end{aligned}$$

Durch erneutes anwenden des Mittelwertsatzes folgt nun:

$$\begin{aligned}0 \leq h'(x_0) &= \frac{\hat{f}(\tilde{a}) - \hat{g}(\tilde{a}) - \hat{f}(a) + \hat{g}(a)}{\tilde{a} - a} = \frac{\hat{g}(\tilde{a}) - \hat{f}(\tilde{a})}{\tilde{a} - a} \\ &\quad \hat{f}(\tilde{a}) \geq \hat{g}(\tilde{a}) \\ 0 \geq h'(x_0) &= \frac{\hat{f}(b) - \hat{g}(b) - \hat{f}(\tilde{b}) + \hat{g}(\tilde{b})}{b - \tilde{b}} = \frac{\hat{g}(\tilde{b}) - \hat{f}(\tilde{b})}{b - \tilde{b}} \\ &\quad \hat{f}(\tilde{b}) \geq \hat{g}(\tilde{b})\end{aligned}$$

und somit $\forall x \in [a, b] : \hat{f}(x) \geq \hat{g}(x)$. Es gilt also

$$\tau(f, \{\xi_{i-1}, \xi_i\}) = \int_{\xi_{i-1}}^{\xi_i} \hat{f}(x) dx \geq \int_{\xi_{i-1}}^{\xi_i} \hat{g}(x) dx = \tau(g, \{\xi_{i-1}, \xi_i\})$$

und

$$\tau(f, Z) \geq \tau(g, Z).$$

□

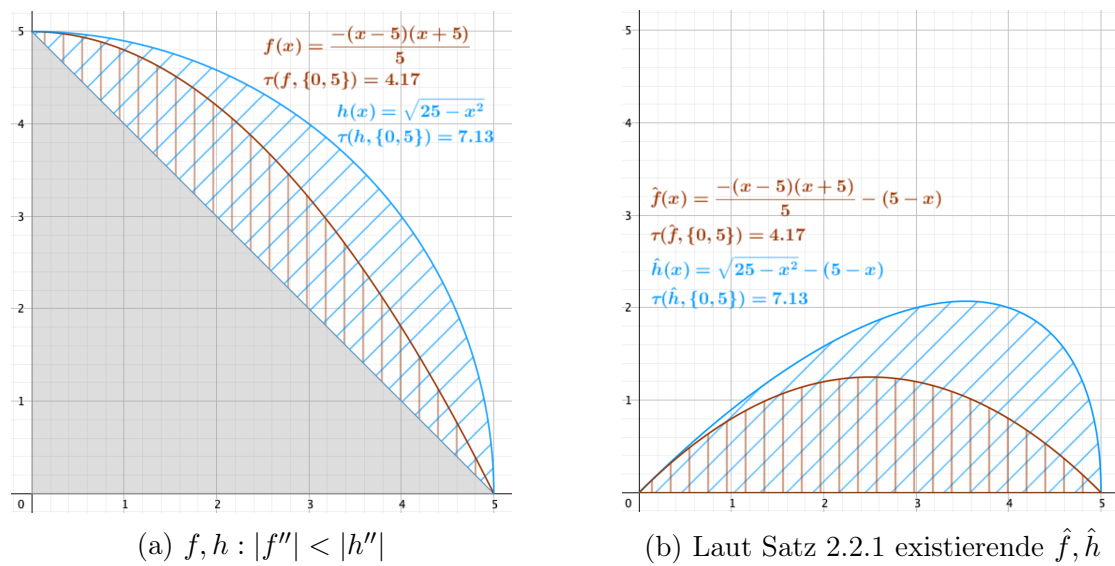


Abbildung 3: Veranschaulichung von Satz 2.2.1

Wird nun T fix gehalten und h variiert, bleibt für $h \rightarrow 0$ der von T abhängige Fehler über.

```
def f(x):
2     return 1/(np.exp(x)+7)*np.exp(-x)

4 def sumtrapez(t,h):
    xs=np.arange(0,t,h)
6     fs=f(xs)
    return h/2*(1+2*sum(fs[1:-1])+fs[-1]) # 1=f(0) stetig fortgesetzt

8
T=10
10 errors=[]
    hs=1/2**np.arange(15)
12 for h in hs:
        errors.append(abs(0.10042-sumtrapez(T,h)))
14 plt.loglog(hs,errors)
    plt.loglog(hs,hs**2)
16 plt.grid(which='major',linewidth=0.5)
    plt.xlabel('h',size='large')
```

Listing 3: Implementierung des Fehlers mit fixem T

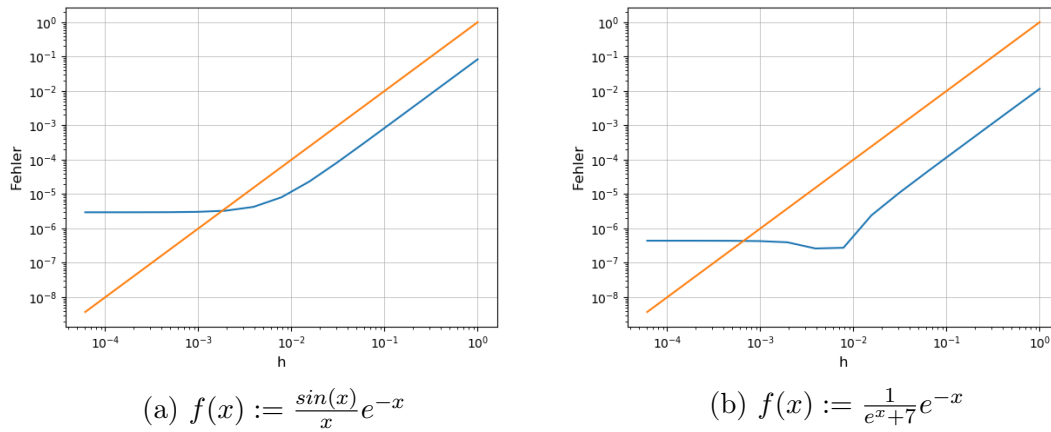


Abbildung 4: Fehlerplot abhängig von h .

Man sieht nun aus Abb. 4, wobei die gerade Funktion als Vergleich dient wie sich der Fehler verhalten sollte, dass bei kleiner werdenden h sich der Fehler ab einem gewissen Wert nicht mehr ändert. D.h. man ist zum von T abhängigen Fehler gelangt.

3 Gauß-Laguerre Quadratur

Zunächst sei bemerkt, dass (Nannen 2019, S. 45):

KOROLLAR 3.0.1

Es existiert eine eindeutig bestimmte Folge (q_n) von Polynomen der Form

$$\begin{aligned} q_0(x) &:= 1 \\ q_n(x) &:= x^n + r_{n-1}(x), \quad x \in [a, b] \end{aligned}$$

mit $r_{n-1} \in \Pi_{n-1}$, die die Orthogonalitätsbeziehungen $(q_n, q_m)_w = 0$, $n \neq m$ erfüllen.

und

LEMMA 3.0.2

Jedes der orthogonalen Polynome $q_n \in \Pi_n$, aus Korollar 3.0.1 besitzt n einfache Nullstellen in (a, b) .

Woraus sich dann folgender Satz ergibt:

SATZ 3.0.3

Die Orthogonalpolynome $q_n \in \Pi_n$ aus Korollar 3.0.1 sind eindeutig durch die folgenden 3 Terme gegeben

$$\begin{aligned} q_0(x) &:= 1 \\ q_1(x) &:= (x - \beta_0)q_0 = x - \beta_0 \\ q_{n+1}(x) &:= (x - \beta_n)q_n(x) - \gamma_n^2 q_{n-1}(x) \quad \text{für } n \geq 1 \end{aligned}$$

mit

$$\beta_n := \frac{(xq_n, q_n)_w}{\|q_n\|_w^2} \quad \text{und} \quad \gamma_n := \frac{\|q_n\|_w}{\|q_{n-1}\|_w}$$

Weiter sind die Eigenwerte der Matrix

$$T := \begin{pmatrix} \beta_0 & \gamma_1 & & & \\ \gamma_1 & \beta_1 & \gamma_2 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & \beta_{n-1} & \gamma_n \\ & & & \gamma_n & \beta_n \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$$

die Nullstellen $x_0, \dots, x_n$ des $(n+1)$ -ten Orthogonalpolynoms q_{n+1} und die zugehörigen Gewichte der Gauß-Quadratur sind gegeben durch

$$\alpha_j := \left(\frac{(v_j)_1}{\|v_j\|_2} \right)^2 \int_a^b w(x) dx, \quad j = 0, \dots, n$$

wobei $(v_j)_1$ die 1.Komponente eines Eigenvektors $v_j \in \mathbb{R}^{n+1} \setminus \{0\}$ zum Eigenwert x_j ist.

BEWEIS

Durch die Gewichtsfunktion $w(x) = \exp(-x)$ haben die Polynome und die Matrix folgende Form:

$$\begin{aligned} p_0(x) &:= 1 \\ p_1(x) &:= (x - 1)p_0 = x - 1 \\ p_{n+1}(x) &:= (x - 2n - 1)p_n(x) - n^2 p_{n-1}(x) \quad \text{für } n \geq 1 \end{aligned}$$

und

$$T_n := \begin{pmatrix} 1 & -1 & & & \\ -1 & 3 & & & \\ & & \ddots & \ddots & \\ & & \ddots & & -n+1 \\ & & & -n+1 & 2n-1 \end{pmatrix}$$

Mit vollständiger Induktion wird gezeigt, dass $T_n v_{k,n} = x_k v_{k,n}$ gilt, mit dem Eigenvektor $v_{k,n} := (\tau_0, \tau_1 p_1(x_k), \dots, \tau_{n-1} p_{n-1}(x_k))^T$ und $\tau_j = \frac{(-1)^j}{j!}$ zum Eigenwert x_k .

I.V. Es gilt $T_{n-1} v_{k,n-1} = x_k v_{k,n-1}$

I.A. Mit $n = 1$ folgt unmittelbar

$$T_1 v_{k,1} = (1)(\tau_0) = \tau_0 \stackrel{!}{=} x_k \tau_0 \Leftrightarrow T_1 v_{k,1} - x_k \tau_0 = \tau_0 - x_k \tau_0 = 1 - x_k = 0$$

Also ist der Eigenwert $x_k = 1$ was bei genauerer Betrachtung auch die Nullstelle des Orthogonalpolynoms $p_1(x_k) = x_k - 1 = 0$ ist.

I.S. $n \rightarrow n + 1$

$$\begin{aligned}
T_n v_{k,n} &= \begin{pmatrix} 1 & -1 & & & \\ -1 & 3 & & & \\ & & \ddots & \ddots & \\ & & \ddots & & -n+1 \\ & & & -n+1 & 2n-1 \end{pmatrix} \begin{pmatrix} \tau_0 \\ \tau_1 p_1(x_k) \\ \vdots \\ \tau_{n-1} p_{n-1}(x_k) \end{pmatrix} \\
&= \begin{pmatrix} \tau_0 - \tau_1 p_1(x_k) \\ -\tau_0 + 3\tau_1 p_1(x_k) - 2\tau_2 p_1(x_k) \\ \vdots \\ (-n+1)\tau_{n-2} p_{n-2}(x_k) + (2n-1)\tau_{n-1} p_{n-1}(x_k) \end{pmatrix} \\
&\stackrel{!}{=} x_k \begin{pmatrix} \tau_0 \\ \tau_1 p_1(x_k) \\ \vdots \\ \tau_{n-1} p_{n-1}(x_k) \end{pmatrix}
\end{aligned}$$

Durch einsetzen der Rekursionsformel für die Orthogonalpolynome p_n auf der linken Seite für die ersten $n - 1$ Einträge, folgt direkt die rechte Seite. Für den letzten Eintrag gilt nun

$$\begin{aligned}
x_k \tau_{n-1} p_{n-1}(x_k) &= (-n+1)\tau_{n-2} p_{n-2}(x_k) + (2n-1)\tau_{n-1} p_{n-1}(x_k) \\
\Leftrightarrow x_k \tau_{n-1} p_{n-1}(x_k) - (-n+1)\tau_{n-2} p_{n-2}(x_k) - (2n-1)\tau_{n-1} p_{n-1}(x_k) &= \\
\tau_{n-1} p_{n-1}(x_k)(x_k - 2n+1) + (n-1)\tau_{n-2} p_{n-2}(x_k) &= 0
\end{aligned}$$

Sei nun o.B.d.A n ungerade und mit Erweiterung des rechten Terms und einsetzen von τ_j folgt

$$\begin{aligned}
\frac{p_{n-1}(x_k)(x_k - 2n+1)}{(n-1)!} - \frac{(n-1)^2 p_{n-2}(x_k)}{(n-1)!} &= 0 \\
\Leftrightarrow p_{n-1}(x_k)(x_k - 2(n-1) - 1) - (n-1)^2 p_{n-2}(x_k) &= p_n(x_k) = 0
\end{aligned}$$

x_k sind also die Nullstellen des Polynoms p_n und sind zugleich die Eigenwerte von T_n , wegen $p_n(x) = \det(T_n(x))$. (*Numerische Mathematik Übung 2019*, vgl. Bsp. 40)

Was noch zu zeigen bleibt, ist dass die Gewichte folgende Form haben

$$\begin{aligned}
\alpha_j &= \left(\frac{(v_j)_1}{\|v_j\|_2} \right)^2 \int_a^b w(x) dx = \frac{((v_j)_1)^2}{(\|v_j\|_2)^2} \int_0^\infty e^{-x} dx \\
&= \frac{1}{\|v_j\|_2^2}
\end{aligned}$$

Bei Betrachtung des Ausdrucks

$$\sum_{j=0}^{n-1} \sum_{l=0}^{j+1} w_{l,n} \tau_j^2 p_j(x_k) p_j(x_l)$$

folgt nun

$$\begin{aligned} \sum_{j=0}^{n-1} \sum_{l=0}^{j+1} w_{l,n} \tau_j^2 p_j(x_k) p_j(x_l) &= \sum_{j=0}^{n-1} \tau_j^2 p_j(x_k) \int_0^\infty p_j(x) e^{-x} dx \\ &= \tau_0^2 p_0(x_k) \int_0^\infty p_0(x) e^{-x} dx + \sum_{j=1}^{n-1} \tau_j^2 p_j(x_k) \int_0^\infty p_j(x) e^{-x} dx \\ &= 1 + \sum_{j=1}^{n-1} \tau_j^2 p_j(x_k) \cdot 0 \\ &= 1 \end{aligned}$$

und andererseits

$$\begin{aligned} \sum_{j=0}^{n-1} \sum_{l=0}^{j+1} w_{l,n} \tau_j^2 p_j(x_k) p_j(x_l) &= \sum_{j=0}^{n-1} \sum_{l=0}^{j+1} w_{l,n} \tau_j^2 p_j(x_k)^2 \delta_{l,k} = \\ &= \sum_{j=0}^{n-1} w_{k,n} \tau_j^2 p_j(x_k)^2 = w_{k,n} \sum_{j=0}^{n-1} \tau_j^2 p_j(x_k)^2 = w_{k,n} \|v_{k,n}\|_2^2 \end{aligned}$$

Also

$$w_{k,n} \|v_{k,n}\|_2^2 = 1 \Rightarrow w_{k,n} = \frac{1}{\|v_{k,n}\|_2^2}$$

□

Somit ist also die in Listing 4 und 5 zu sehende Vorgehensweise gerechtfertigt.

```
1 int main() {  
2     Gauss Q(N);  
3     printf("Ergebnis: %.16e\n",Q.result(f));  
4     return 0;  
5 }
```

Listing 4: Mögliche Implementierung von Satz 3.0.3 - main.cpp

```
1     T = MatrixXcd::Zero(n+1,n+1);  
2  
3     T(0,0)=1;  
4     for(int i = 1; i < n+1; ++i){  
5         T(i-1,i)=-i;  
6         T(i,i-1)=-i;  
7         T(i,i)=2*(i+1)-1;  
8     }  
9     ces = ComplexEigenSolver<MatrixXcd>(T);  
10    eigenvectors = ces.eigenvectors();  
11    nodes = T.eigenvalues();  
12 }  
13  
14 double Gauss::norm(int i){  
15     double sum = 0;  
16     for(int j = 0; j<eigenvectors.rows();++j) {  
17         sum += std::abs(eigenvectors(i,j)) * std::abs(  
18             eigenvectors(i,j));  
19     }  
20     return std::sqrt(sum);  
21 }  
22  
23 std::complex<double> Gauss::weigth(int j) {  
24     return eigenvectors(0,j) * eigenvectors(0,j);  
25 }  
26  
27 std::complex<double> Gauss::node(int i) {  
28     return nodes(i);  
29 }  
30  
31 double Gauss::result(double f(double)){  
32     std::complex<double> sum = 0;  
33     for(int i = 0; i < n+1;++i){  
34         sum += weigth(i)*f(node(i).real());  
35     }  
36     return sum.real();  
37 }
```

Listing 5: gauss.cpp

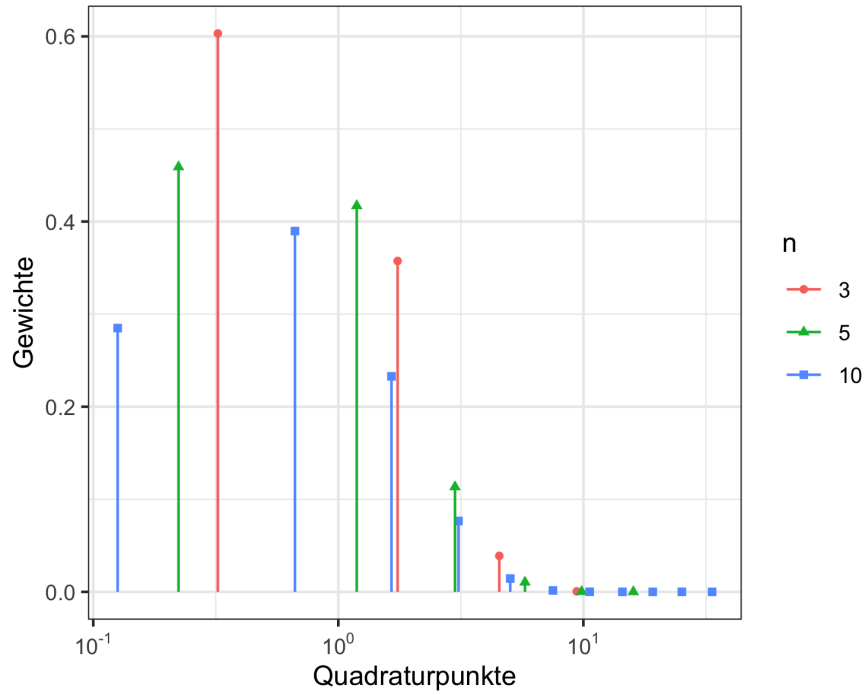


Abbildung 5: Quadraturpunkte und zugehörige Gewichte

BEMERKUNG. Um, wie in Abb. 1b auch Integrale der Form $\int_a^\infty f(x)\omega(x)dx$: $a > 0$ zu berechnen kann für Funktionen $f, \omega : \int_0^\infty f(x)\omega(x)dx < \infty$ wegen (vgl. Kaltenbäck 2014, S. 278)

$$\int_a^\infty f(x)\omega(x)dx = \int_0^\infty f(x+a)\omega(x+a)dx = \int_0^\infty f(x+a)\omega(x+a)\omega^{-1}(x)\omega(x)dx$$

eine neue Funktion g mit $g(x) := f(x+a)\omega(x+a)\omega^{-1}(x)$ definiert werden für die gilt:

$$\int_a^\infty f(x)\omega(x)dx = \int_0^\infty g(x)\omega(x)dx \approx \sum_{i=1}^n w_i g(x_i).$$

Weiters können wegen $\int_a^b f(x)\omega(x)dx = \int_a^c f(x)\omega(x)dx - \int_b^c f(x)\omega(x)dx$ für $0 \leq a \leq b \leq c$ auch Integrale der Form $\int_a^b f(x)\omega(x)dx$ berechnet werden.

4 Vergleich der Methoden aus Abschnitt 2 und 3

Wie bereits gezeigt, ist das Konvergenzverhalten der Summierten Trapezregel stark von der zu integrierenden Funktion abhängig. Um die Konvergenz zu beschleunigen kann entweder eine stärker fallende Gewichtsfunktion gewählt werden, um den von T abhängigen Fehler zu verkleinern oder eine schwächer gekrümmte Funktion, um den von n abhängigen Fehler zu minimieren.

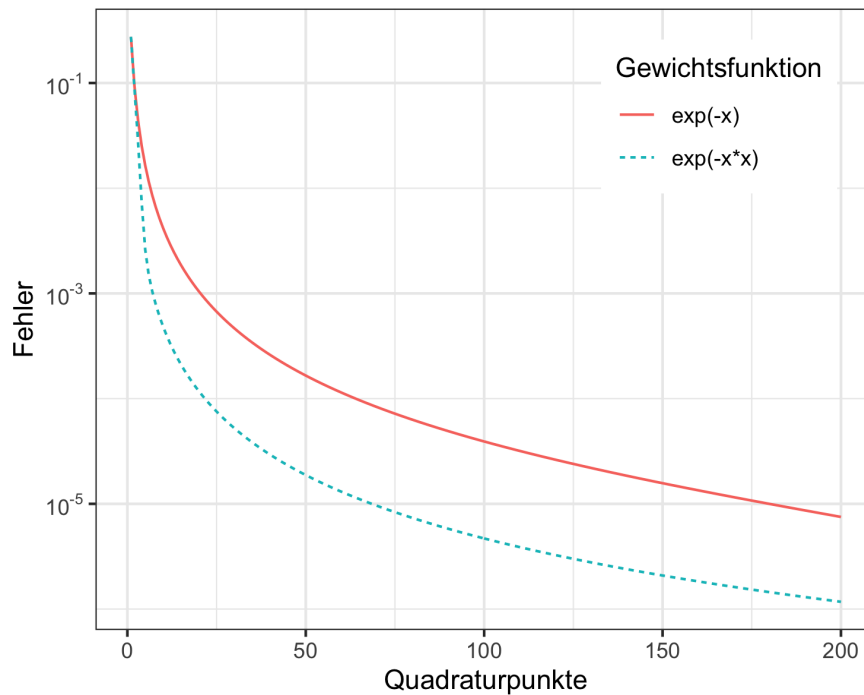


Abbildung 6: Fehler der Summierten Trapezregel für unterschiedlich stark fallende Gewichtsfunktionen und $f(x) = \frac{1}{e^x + 7}$

Wie in Abb. 6 zu sehen ist, geht der Fehler für e^{-x^2} wesentlich schneller gegen Null als bei e^{-x} . Trotzdem bleibt die Konvergenz langsamer als bei der in Abschnitt 3 beschriebenen Gauß-Quadratur (Abb. 7).

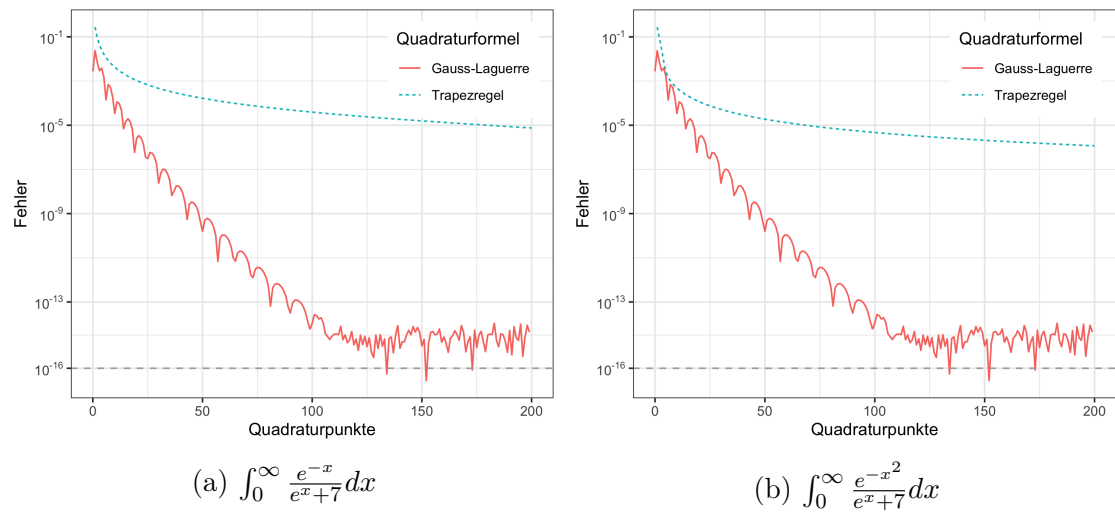


Abbildung 7: Vergleich der Integrationsverfahren aus Abschnitt 2 und 3 für verschiedene Gewichtsfunktionen

Literatur

Kaltenbäck, Michael (2014). *Fundament Analysis*. Berlin: H. Begehr und R. Gorenflo (siehe S. 6, 18).

Nannen, Lothar (2019). *Numerische Mathematik A*. URL: <https://tiss.tuwien.ac.at/education/course/documents.xhtml?dswid=3351&dsrid=39&courseNr=101313&semester=2019W#> (besucht am 03.12.2019) (siehe S. 5, 13).

Numerische Mathematik Übung (2019). URL: <https://tuwel.tuwien.ac.at/course/view.php?id=19681> (besucht am 23.12.2019) (siehe S. 15).

Abbildungsverzeichnis

1	Vergleich verschiedener Gewichtsfunktionen	7
2	Fehler abhängig von T.	8
3	Veranschaulichung von Satz 2.2.1	11
4	Fehlerplot abhängig von h.	12
5	Quadraturpunkte und zugehörige Gewichte	18
6	Fehler der Summierten Trapezregel für unterschiedlich stark fallende Gewichtsfunktionen und $f(x) = \frac{1}{e^x+7}$	19
7	Vergleich der Integrationsverfahren aus Abschnitt 2 und 3 für ver- schiedene Gewichtsfunktionen	20