



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Elektrotechnik, Informatik und Mathematik
Im Fach Numerik

**Splitting-Verfahren und exponentielle Integratoren für
parabolische Probleme und effiziente Algorithmen durch
schnelle Fourier-Transformation**

Bachelorarbeit

im Rahmen des Studiengangs Mathematik
zur Erlangung des Grades

Bachelor of Science

von

DANIEL HEINS

vorgelegt bei:

Prof. Dr. Balázs Kovács

und

Prof. Dr. Sina Ober-Blöbaum

Paderborn, 13. Mai 2025

Inhaltsverzeichnis

1	Analytische Grundlagen und Definitionen	3
2	Wärmeleitungsgleichung	7
2.1	Exakte Lösung des Problems (HE)	7
2.2	Approximation der Lösung von (HE) mittels FFT	11
2.2.1	Fehlerabschätzung der numerischen Lösung	13
2.2.2	Numerische Experimente zu der Laufzeit	16
3	Splitting-Verfahren	17
3.1	Klasse von Splitting-Verfahren	18
3.2	Splitting-Verfahren für das Problem (NL)	19
3.2.1	Sequentielles Splitting-Verfahren	19
3.2.2	Marchuk-Strang Splitting-Verfahren	20
3.3	Numerische Experimente zum Splitting-Verfahren	22
4	Exponentielle Integratoren	25
4.1	Das exponentielle Runge-Kutta-Verfahren	26
4.1.1	Approximation durch Interpolation	26
4.1.2	Klasse der exponentiellen RKV	27
4.2	Exponentielle Euler-Verfahren für das Problem (NL)	28
4.3	Die exponentielle Quadraturregeln für das Problem (LE)	29
4.4	Beispiele einiger exponentiellen Quadraturregeln	31
4.4.1	Exponentielle Euler-Quadraturregel	31
4.4.2	Exponentielle Trapez-Quadraturregel	31
4.5	Numerische Experimente zu den exp. Quadraturregeln	32
5	weitere Anwendungen	35
5.1	Erstes Beispiel	35
5.2	Zweites Beispiel	36
6	Zusammenfassung	37
	Literaturverzeichnis	39

Motivation und Einleitung

Differentialgleichung jeglicher Art spielen heutzutage eine sehr wichtige Rolle, sowohl in der theoretischen, als auch in der praktischen Natur. Allerdings sind für diese in fast allen Fällen sehr schwer eine analytische Lösungen herzuleiten.

Diese Arbeit befasst sich mit Splitting-Verfahren und exponentiellen Integratoren zur numerischen Lösung parabolischer Differentialgleichungsprobleme. Ein besonderer Schwerpunkt liegt dabei auf dem Einsatz des Fast-Fourier-Transformations(FFT)-Algorithmus, der erstmals 1964 von Cooley und Tukey entwickelt wurde.

Vor diesem Hintergrund konzentrieren wir uns gezielt auf eine einzelne nichtlineare Differentialgleichung (NL), da, wie allzu oft, unterschiedliche Probleme unterschiedliche Ansätze erfordern. Wir betrachten hierzu die zugrunde liegenden Verfahren, deren Konsistenzordnung sowie entsprechende numerische Experimente.

Da die zugrundeliegende Fehler-Analyse recht komplex ist, werden wir in diesem Zusammenhang auf die Arbeiten von ISEM Team [BFCO12] bzw. von M. Hochbruck, A. Ostermann [HO10] verweisen.

Betrachte zu einem gegebenen $u^0 \in C^2[0, 1]$, $g \in C^2(\mathbb{R})$ mit $g(0) = 0$ und dem Differentialoperator $A := \frac{\partial^2}{\partial x^2}$ das eindimensionale Randwertproblem auf dem Intervall $[0, 1]$

$$\begin{cases} \frac{\partial}{\partial t} u(t, x) &= Au(t, x) + g(u(t, x)), & t > 0, \\ u(0, x) &= u_0(x) \end{cases} \quad (\text{NL})$$

mit den Dirichlet-Randbedingungen (D.-R.)

$$u(t, 0) = u(t, 1) = 0.$$

Sei hier die Annahme gegeben, dass für (NL) stets eine eindeutige Lösung existiert. Für eine numerische Approximation dieses Problems, betrachten wir sowohl eine Raumdiskretisierung im Raum $x \in [0, 1]$, als auch eine Zeitdiskretisierung in der Zeit $t \in [0, \infty)$.

Im ersten Kapitel werden wir uns mit der Wärmeleitungsgleichung auseinandersetzen. Im Hinblick auf einen effizienten Algorithmus werden wir uns an die Analysis der Fourierreihenentwicklung in $L^2(0, 1)$ orientieren. Die Lösung dieser Gleichung werden nämlich in den bereits oben erwähnten Verfahren auftauchen.

Das *Splitting-Verfahren*, bzw. die Varianten des Verfahrens, bieten eine Möglichkeit, numerische Lösungen zu approximieren. Die Grundidee besteht darin, das Problem (NL) in leichtere Teilprobleme zu zerlegen, die sich effizient lösen lassen. Die Lösung des ursprünglichen Problems wird dann aus den Lösungen der einzelnen Teilprobleme zusammengesetzt.

Im nächsten Kapitel erläutern wir das Verfahren der *exponentiellen Integratoren* und führen das Schema der *exponentielle Runge-Kutta-Verfahren (RKV)* ein. Für eine existierende Lösung u kann diese Lösung mithilfe der Variation der Konstanten Formel dargestellt werden. Ziel ist es, genau diese Formel zu approximieren. Anschließend befassen wir uns kurz mit den *exponentiellen Quadraturregeln*. Diese sind im Gegenzug numerische Verfahren zur Lösung der Differentialgleichung (LE).

Zum Abschluss wollen wir für beide Verfahren numerische Experimente anhand eines Beispiels durchführen. Anstelle des Randwertproblems (NL) untersuchen wir ein ähnliches Problem, bei dem der nichtlineare Term $g(u(t, x))$ durch $g(t, x)$ ersetzt wird.

Dazu betrachten wir nun zu einer gegebenen Funktion $g \in M$ das eindimensionale Randwertproblem mit den D.-R auf dem Intervall $(0, 1)$:

$$\begin{cases} \frac{\partial}{\partial t} u(t, x) &= Au(t, x) + g(t, x), & t > 0, \\ u(0, x) &= u_0(x). \end{cases} \quad (\text{LE})$$

Da beide Verfahren auch bei vielen weiteren Problemen Anwendung finden, möchten wir zum Abschluss einige Beispiele von Differentialgleichungen betrachten und diese mithilfe beider Verfahren numerisch lösen.

Die Codes für die programmierten Verfahren findet man unter folgendem GitHub Link wieder:

1 Analytische Grundlagen und Definitionen

In diesem Abschnitt werden die zentralen Definitionen und Sätze vorgestellt, die für diese Arbeit von grundlegender Bedeutung sind.

Zunächst definieren wir einige wichtige Funktionenräume. Anschließend werden zentrale Resultate aus der Analysis aufgeführt, die insbesondere im Kapitel über die Wärmeleitungsgleichung ihre Anwendung finden.

Um Aussagen zur Güte des Splitting-Verfahrens und der exponentiellen Runge-Kutta-Verfahren treffen zu können, führen wir zudem den Begriff des Einschrittverfahrens sowie der Konsistenzordnung ein.

Definition 1.1.

Sei $\Omega \subset \mathbb{R}$ offen, $n, p \in \mathbb{N}$. Für die Operatoren A, B definiert durch $(Aw)(x) := (\frac{\partial^2}{\partial x^2} w)(x)$, bzw. $(Bv)(x) := g(v(x))$ seien die Definitionsmengen $D(A)$, bzw. $D(B)$.

$$C^0(\Omega) := \left\{ f : \Omega \rightarrow \mathbb{R}^n \mid f \text{ ist stetig auf } \overline{\Omega} \right\}$$

$$C^p(\Omega) := \left\{ f : \Omega \rightarrow \mathbb{R}^n \mid f \in C^{p-1}(\Omega, \mathbb{R}^n), f \text{ stetig differenzierbar auf } \Omega \right\}$$

$$L^2(\Omega) := \left\{ f : \Omega \rightarrow \mathbb{R} \mid f \text{ messbar, } \int_{\Omega} |f(x)|^2 dx < \infty \right\}$$

$$D(A) := \left\{ f \in C^2(0, 1) \cap C^0[0, 1] \mid f(0) = f(1) = 0 \right\}$$

$$D(B) := \left\{ f \in C^2(\mathbb{R}) \mid f(0) = 0 \right\}$$

$$M := \left\{ f : [0, \infty) \times [0, 1] \rightarrow \mathbb{R} \mid \begin{array}{l} \forall t \geq 0 : f(t, \cdot) \in D(A), \\ \forall x \in [0, 1] : f(\cdot, x) \in C^1(0, \infty) \cap C^0[0, \infty) \end{array} \right\}$$

Bemerkung: Zur Vereinfachung der Notation werden wir hier und im Folgenden, falls nicht anders angegeben, das Skalarprodukt $\langle \cdot, \cdot \rangle$ und die Norm $\|\cdot\|$ im Sinne des $L^2(0, 1)$ -Raums verwenden.

Wir werden Lemma 1.2 verwenden um die zweite Bedingung im Satz 1.3 einfach festzustellen. Diesen benötigen wir wiederum um die Lösung der Wärmeleitungsgleichung herzuleiten.

Lemma 1.2 ([Kö04, Kapitel 15.2]) *Sei $I \subset \mathbb{R}$ ein Intervall und seien $f_n : I \rightarrow \mathbb{R}$ eine Folge von stetigen Funktionen auf dem Intervall I . Weiter sei $\sum_{n=0}^{\infty} a_n$ eine konvergente, reelle Folge mit $|f_n(x)| \leq a_n$ für alle $n \in \mathbb{N}$ und alle $x \in I$. Dann ist die Funktionenreihe $\sum_{n=0}^{\infty} f_n$ gleichmäßig konvergent auf I .*

Satz 1.3 ([Kö04, Kapitel 15.2]) *Sei $I \subset \mathbb{R}$ ein offenes Intervall und es sei $f_n : I \rightarrow \mathbb{R} \mid x \mapsto f_n(x)$ eine Folge von stetig differenzierbaren Funktionen auf dem Intervall I wie folgt:*

- 1) *Die Folge $(f_n)_{n \in \mathbb{N}}$ konvergiert punktweise auf I ,*
- 2) *Die Folge $(\frac{\partial}{\partial x} f_n)_{n \in \mathbb{N}}$ konvergiert gleichmäßig auf I .*

Dann ist die Grenzfunktion $f := \lim_{n \rightarrow \infty} f_n$ stetig differenzierbar auf I mit $\frac{\partial}{\partial x} f = \lim_{n \rightarrow \infty} \frac{\partial}{\partial x} f_n$.

Einen zentralen Satz aus der Funktionalanalysis bildet den Grundbaustein für unseren effizienten Algorithmus.

Satz 1.4 *Die Menge $S := \{f_n := \sqrt{2} \sin(\pi n x), n \in \mathbb{N}_0\}$ ist eine vollständige Orthonormalbasis auf $L^2(0, 1)$, d.h. der Grenzwert*

$$\sum_{n=0}^{\infty} \langle f_n, f \rangle f_n \tag{1.1}$$

existiert immer und ist gleich f für alle $f \in L^2(0, 1)$.

Außerdem gilt die Parseval'sche Gleichung:

$$\sum_{n=0}^{\infty} |\langle f_n, f \rangle|^2 = \|f\|_{L^2((0,1))}^2 \quad \forall f \in L^2(0, 1) \tag{1.2}$$

Beweis: Der analytische Beweis zu diesem Satz erfordert einiges an Vorarbeit. Daher verweise ich hier auf das Skript der Vorlesung „*Hilbertraummethode*“ von Prof. M. Winkler

□

Um eine Aussage über ein Verfahren treffen zu wollen, müssen wir diese zunächst einmal definieren.

Definition 1.5 (Numerisches Einschrittverfahren). Für eine Schrittweite $\tau > 0$ sei das Zeitgitter gegeben durch

$$t_0 = 0, \quad t_{k+1} := t_k + \tau \quad \text{für } k \in \mathbb{N}.$$

Ein numerisches Einschrittverfahren mit Schrittweite $\tau > 0$ ist eine Methode, die zu einer Differentialgleichung, wie zum Beispiel in (NL), für eine approximierte Lösung u^k an der Stelle t_k , die Lösung zum Zeitpunkt t_{k+1} näherungsweise bestimmt, also

$$u^{k+1} := \phi(t_k, u^k), \quad \text{für } k \in \mathbb{N}_0. \quad (1.3)$$

Zu dem vorhin definierten Einschrittverfahren stellt sich nun die Frage, wie groß der Fehler nach einem Zeitschritt ist. Somit betrachten wir zum Zeitpunkt t_k die exakte Lösung u einer gegebenen Differentialgleichung, wie z.B. in (NL) oder (LE) und wollen nun den Fehler als die Differenz der Funktion $\phi(t_k, u^k)$ und u^{k+1} betrachten. Diesen Fehler werden wir als den lokalen Diskretisierungsfehler definieren.

Definition 1.6 (Lokaler Diskretisierungsfehler). Betrachte ein numerisches Einschrittverfahren ϕ mit einer Schrittweite τ , wie in Definition 1.5. Dann sei der lokale Diskretisierungsfehler definiert durch:

$$\psi_{k+1} := \phi(t_k, u(t_k)) - u(t_{k+1}) \quad (1.4)$$

Der lokale Diskretisierungsfehler gibt den Fehler nach einem einzelnen Zeitschritt an. Es sei darauf hingewiesen, dass der Diskretisierungsfehler stets von k abhängt. Nachdem wir nun den Fehler nach einem Zeitschritt definiert haben, möchten wir eine Aussage darüber treffen, wie sich der Fehler verringert, wenn kleinere Schrittweiten gewählt werden. Dies wird durch die Konsistenzordnung beschrieben.

Definition 1.7 (Konsistenzordnung). Betrachte ein numerisches Einschrittverfahren ϕ mit einer Schrittweite $\tau > 0$ und sei der lokale Diskretisierungsfehler ψ_{k+1} wie in Definition 1.6. Das Verfahren ϕ hat eine Konsistenzordnung von $s \in \mathbb{N}$, falls

$$\|\psi_{k+1}\| = \mathcal{O}(\tau^{s+1}). \quad (1.5)$$

2 Wärmeleitungsgleichung

Die Wärmeleitungsgleichung, auch als Wärmediffusionsgleichung bekannt, ist wie der Name schon vermuten lässt, ein physikalisches Modell, der die zeitliche Änderung der Temperatur im Raum beschreibt. In diesem Abschnitt wollen wir ein numerisches Verfahren zur Lösung dieser Gleichung (HE) herleiten.

Grundlage dafür ist, wie bereits erwähnt, die in Satz 1.4 aufgeführte Orthonormalbasis von $L^2(0, 1)$. Damit lässt sich jede beliebige Funktion $u \in D(A)$ als ihre Fourierreihe darstellen: $u = \sum_{n=1}^{\infty} \langle u, f_n \rangle_{L^2(0,1)} f_n$.

Dafür wollen wir als ersten Schritt eine exakte Lösung der Wärmeleitungsgleichung herleiten und anschließend diese mittels dem FFT-Algorithmus approximieren. Danach leiten wir eine Fehlerabschätzung für die approximierte Lösung her und veranschaulichen die Effizienz des Algorithmus anhand eines Beispiels.

Falls die Leser:innen sich weitergehend für diese Gleichung interessieren, sei auf [Nar99] verwiesen. In der werden unter anderem geschichtliche Hintergründe und Entwicklung der Gleichung erläutert.

2.1 Exakte Lösung des Problems (HE)

Sei die eindimensionale Wärmeleitungsgleichung mit den D.-R. auf dem Intervall $[0, 1]$ gegeben durch:

$$\begin{cases} \frac{\partial}{\partial t} u(t, x) &= \frac{\partial^2}{\partial x^2} u(t, x), & t > 0, \\ u(0, x) &= u^0(x). \end{cases} \quad (\text{HE})$$

Angenommen u ist eine Funktion, die das Problem (HE) löst. Dann ist $u(t, \cdot) \in D(A)$ und somit für jedes $t \geq 0$ nach Satz 1.4 in eine Fourierreihe der Form

$$u(t, \cdot) = \sum_{n=0}^{\infty} \hat{u}_n(t) f_n \quad (2.1)$$

entwickelbar, wobei

$$\hat{u}_n(t) = \langle u(t, \cdot), f_n \rangle = \sqrt{2} \int_0^1 u(t, x) \sin(\pi n x). \quad (2.2)$$

Betrachten wir zunächst den einfacheren Fall, wenn der Anfangswert $u(0, \cdot) = u_0$ in eine endliche Fourierreihe entwickelbar ist.

Proposition 2.1 Sei $u^0 \in D(A)$, wobei ein $N \in \mathbb{N}$ existiert, sodass

$$u^0(x) = \sum_{n=0}^N \hat{u}_n^0 f_n. \quad (2.3)$$

Dann liegt die durch

$$u(t, x) := \sum_{n=0}^N e^{-t(\pi n)^2} \hat{u}_n^0 f_n(x) \quad (2.4)$$

definierte Funktion in M und erfüllt (HE).

Beweis. Da die Differentialoperatoren linear sind und die einzelnen Summanden in (2.4) die Gleichung

$$\frac{\partial}{\partial t} u(t, x) = \frac{\partial^2}{\partial x^2} u(t, x) \quad \text{für } t > 0 \quad (2.5)$$

erfüllen, muss nur noch überprüft werden, ob die Funktion u in (2.4) die D.-R. und $u(0, x) = u^0(x)$ für alle $x \in [0, 1]$ erfüllt. Dies ist aber leicht einzusehen. \square

Intuitiv könnte man annehmen, dass die Funktion $u(t, x)$ aus Prop. 2.1 die gleiche Form für eine beliebige Funktion $u^0 \in D(A)$ hat. In der Tat folgt dies nun aus den nächsten zwei Propositionen.

Proposition 2.2 Sei $g \in D(A)$ und $g = \sum_{n=0}^{\infty} \hat{g}_n f_n$ die Fourierreihe von g . Dann ist die zweite Ableitung gegeben durch:

$$\frac{\partial^2}{\partial x^2} g(x) = \sum_{n=0}^{\infty} -(\pi n)^2 \hat{g}_n f_n(x). \quad (2.6)$$

Außerdem gilt die Ungleichung:

$$\sum_{n=0}^{\infty} |\hat{g}_n| < \infty. \quad (2.7)$$

Beweis. Mit der Kurznotation $g_{xx} := \frac{\partial^2}{\partial x^2} g$, $g_x := \frac{\partial}{\partial x} g$ erhält man die Gleichung (2.6) durch zweifacher partieller Integration:

$$\begin{aligned} \frac{1}{\sqrt{2}} \widehat{g_{xx,n}} &= \frac{1}{\sqrt{2}} \langle g_{xx}, f_n \rangle = \int_0^1 g_{xx}(x) \sin(\pi n x) dx \\ &= g_x(x) \sin(\pi n x) \Big|_{x=0}^{x=1} - \pi n \int_0^1 g_x(x) \cos(\pi n x) dx \\ &= -\pi n \left[\int_0^1 g_x(x) \cos(\pi n x) dx \right] \\ &= -\pi n \left[g(x) \cos(\pi n x) \Big|_{x=0}^{x=1} + \pi n \int_0^1 g(x) \sin(\pi n x) dx \right] \\ &= \frac{1}{\sqrt{2}} (-(\pi n)^2 \hat{g}_n), \end{aligned} \quad (2.8)$$

wobei wir verwendet haben, dass $g(0) = g(1) = 0$ und $\sin(0) = \sin(\pi n) = 0$ ist. Die Ungleichung (2.7) erhält man mittels der Cauchy-Schwarz-Ungleichung:

$$\sum_{n=0}^{\infty} |\hat{g}_n| = \sum_{n=0}^{\infty} \left| \frac{n^2}{n^2} \hat{g}_n \right| \leq \left(\sum_{n=0}^{\infty} |n^2 \hat{g}_n|^2 \right)^{\frac{1}{2}} \left(\sum_{n=0}^{\infty} \left| \frac{1}{n^2} \right|^2 \right)^{\frac{1}{2}} \stackrel{\text{Satz 1.4}}{<} \infty. \quad (2.9)$$

□

Nun sind die Vorbereitungen abgeschlossen und wir können nun folgendes zeigen.

Proposition 2.3 Sei $u^0 \in D(A)$ und die Fourierreihe $u^0 = \sum_{n=0}^{\infty} \hat{u}_n^0 f_n$ gegeben. Dann liegt die durch

$$u(t, x) := \sum_{n=0}^{\infty} e^{-t(\pi n)^2} \hat{u}_n^0 f_n(x) \quad (t, x) \in [0, \infty) \times [0, 1] \quad (2.10)$$

definierte Funktion in M und erfüllt (HE). Somit löst u das Randwertproblem (HE).

Beweis. Wir zeigen nun, dass für beliebiges $(t_0, x_0) \in (0, \infty) \times (0, 1)$ die Funktion u in (t_0, x_0) nach dem ersten Argument stetig differenzierbar und nach dem zweiten Argument zweimal stetig differenzierbar ist und die beiden Ableitungen übereinstimmen. Für jedes $s > 0$ gilt zunächst:

$$|e^{-s}| \leq 1, \quad |se| \leq 2e^s, \quad (2.11)$$

also auch für $s = t(\pi n)^2$ die Ungleichungen:

$$|e^{-t(\pi n)^2}| \leq 1, \quad |(\pi n)e^{-t(\pi n)^2}| \leq |(\pi n)^2 e^{-t(\pi n)^2}| \leq \frac{2}{et}, \quad \forall t > 0, \forall n \in \mathbb{N}. \quad (2.12)$$

Für $s \in [\frac{t_0}{2}, \frac{3t_0}{2}]$ folgt die Abschätzung:

$$\begin{aligned} \sum_{n=0}^{\infty} |e^{-s(\pi n)^2} (\pi n)^2 \hat{u}_n^0 f_n(x_0)| &\stackrel{(2.12)}{\leq} \frac{4}{et_0} \sum_{n=0}^{\infty} |\hat{u}_n^0 f_n(x_0)| \\ &\stackrel{\text{Prop. 2.2}}{\leq} \frac{4\sqrt{2}}{et_0} \sum_{n=0}^{\infty} |\hat{u}_n^0| < \infty. \end{aligned} \quad (2.13)$$

Die Partialsumme

$$g_{x_0, N}(s) := \sum_{n=0}^N -(\pi n)^2 e^{-s(\pi n)^2} \hat{u}_n^0 f_n(x_0) \quad \text{für } s \in \left[\frac{t_0}{2}, \frac{3t_0}{2} \right] \quad (2.14)$$

ist wegen (2.13) und somit nach Lemma 1.2 gleichmäßig konvergent. Nach Satz 1.3 ist $u(s, x_0)$ nun auf $s \in (\frac{t_0}{2}, \frac{3t_0}{2})$ stetig differenzierbar mit $\frac{\partial}{\partial t} u(t_0, x_0) = \lim_{N \rightarrow \infty} g_{x_0, N}(t_0)$.

Nach dem selben Argument konvergieren die Partialsummen

$$h_{t_0,N}(x) := \sum_{n=0}^N (\pi n) e^{-t_0(\pi n)^2} \hat{u}_n^0 \cos(\pi n x), \quad (2.15)$$

$$g_{t_0,N}(x) := \sum_{n=0}^N -(\pi n)^2 e^{-t_0(\pi n)^2} \hat{u}_n^0 \sin(\pi n x) \quad (2.16)$$

gleichmäßig auf dem Intervall $[0, 1]$, sodass wieder nach Satz 1.3 die Funktion $u(t_0, x)$ auf $x \in (0, 1)$ stetig differenzierbar ist, mit $\frac{\partial^2}{\partial x^2} u(t_0, x_0) = \lim_{N \rightarrow \infty} g_{t_0,N}(x_0)$. Somit erhalten wir abschließend:

$$\begin{aligned} \frac{\partial}{\partial t} u(t_0, x_0) &\stackrel{\text{Satz 1.3}}{=} \lim_{N \rightarrow \infty} g_{t_0,N}(t_0) \\ &= \lim_{N \rightarrow \infty} g_{t_0,N}(x_0) \\ &\stackrel{\text{Satz 1.3}}{=} \frac{\partial^2}{\partial x^2} u(t_0, x_0) \quad \forall (t_0, x_0) \in (0, \infty) \times (0, 1). \end{aligned} \quad (2.17)$$

Es bleibt nur noch die Stetigkeit auf dem Rand zu zeigen. Aufgrund der Symmetrie der Sinusfunktion, reicht es aus die Stetigkeit in $x = 0$ zu zeigen. Wir zeigen dies hier nur für die Stetigkeit der Funktion $\frac{\partial}{\partial t} u$ und dies auch nur für den Rand bei $x = 0$. Die Begründung für u und der Rand auf $t = 0$ folgt analog.

Dazu nutzen wir die Parseval'sche Gleichung aus Satz 1.4. Die Funktion $u^0 \in D(A)$ erfüllt somit folgende Eigenschaft:

$$\sum_{n=n_0+1}^{\infty} (\pi n)^2 \hat{u}_n^0 f_n(x_0) \rightarrow 0 \text{ für } n_0 \rightarrow \infty. \quad (2.18)$$

wir zeigen die Stetigkeit auf dem Rand $x = 0$ für alle $t > 0$. Für ein beliebiges $s, \varepsilon > 0$ gilt dann:

$$\begin{aligned} \left| \frac{\partial}{\partial t} u(t, 0) - \frac{\partial}{\partial t} u(t, s) \right| &= \left| \sum_{n=0}^{\infty} -(\pi n)^2 e^{-t_0(\pi n)^2} \hat{u}_n^0 f_n(s) \right| \\ &\stackrel{\Delta\text{-Ung.}}{\leq} \sum_{n=0}^{n_0} \left| (\pi n)^2 \hat{u}_n^0 f_n(s) \right| + \left| \sum_{n=n_0+1}^{\infty} (\pi n)^2 e^{-t(\pi n)^2} \hat{u}_n^0 f_n(s) \right|. \end{aligned} \quad (2.19)$$

Den rechten Teil der Summe können wir mittels der Cauchy-Schwarz-Ungleichung wie folgt abschätzen:

$$\left| \sum_{n=n_0+1}^{\infty} (\pi n)^2 e^{-t(\pi n)^2} \hat{u}_n^0 f_n(s) \right| \leq \left(\sum_{n=n_0+1}^{\infty} |(\pi n)^2 \hat{u}_n^0|^2 \right)^{\frac{1}{2}} \left(\sum_{n=n_0+1}^{\infty} |e^{-t(\pi n)^2}|^2 \right)^{\frac{1}{2}}. \quad (2.20)$$

Den rechten Faktor aus (2.20) konvergiert absolut und der linke Faktor konvergiert nach der Parseval Identität gegen 0 für $n_0 \rightarrow \infty$. Somit wählen wir ein n_0 groß genug, sodass die rechte Reihe aus (2.19) kleiner als $\frac{\varepsilon}{2}$ wird und anschließend s so klein wählen, dass die linke Summe, aufgrund der Stetigkeit, kleiner als $\frac{\varepsilon}{2}$ ist. \square

Rechnerisch aufwändig ist es natürlich die Fouriertransformierten exakt auszurechnen. Wir wollen nun im nächsten Abschnitt eine approximative Lösung mittels dem bereits angesprochenen FFT-Algorithmus herleiten.

2.2 Approximation der Lösung von (HE) mittels FFT

Wir wissen nun aus der Prop. 2.3, dass

$$u(t, x) = \sum_{n=0}^{\infty} e^{-t(\pi n)^2} \hat{u}_n^0 f_n(x) \quad \text{für } (t, x) \in [0, \infty) \times [0, 1] \quad (2.21)$$

eine Lösung des Randwertproblems (HE) ist, wobei die Fourierkoeffizienten $\hat{u}(n) := \hat{u}_n^0$ gegeben sind durch:

$$\hat{u}(n) = \langle u^0, f_n \rangle = \int_0^1 u^0(x) f_n(x) dx. \quad (2.22)$$

In diesem Unterkapitel wollen wir Funktion $u(t, x)$ numerisch approximieren. Dafür führen wir eine Raumdiskretisierung mit der Gitterweite $h := \frac{1}{N}$ durch, wobei $N = 2^L$ für ein $L \in \mathbb{N}$ und erhalten $x_k = kh$ mit $k \in \{0, \dots, N\}$. Dadurch definieren wir die diskrete Funktion

$$u(t, x_k) \approx u_N(t, x_k) := \sqrt{2} \sum_{n=0}^{N-1} e^{-t(\pi n)^2} \hat{u}_N(n) \sin(\pi n k \frac{1}{N}) \quad (2.23)$$

und mit den durch die Trapezregel ermittelten diskreten Fourierkoeffizienten:

$$\begin{aligned} \hat{u}(n) \approx \hat{u}_N(n) &:= \frac{h}{2} u^0(0) f_n(0) + h \sum_{k=1}^{N-1} u^0(hk) f_k(kh) + \frac{h}{2} u^0(1) f_N(1) \\ &= \frac{\sqrt{2}}{N} \sum_{k=0}^{N-1} u^0(x_k) \sin(\pi n k \frac{1}{N}), \quad n \in \{0, \dots, N-1\}. \end{aligned} \quad (2.24)$$

Es ist nun leicht einzusehen, dass man in (2.23) und (2.24) eine lineare Abbildung der Dimension N erhält. Somit würde die Berechnung der diskreten Fourierkoeffizienten und somit auch der diskreten Funktion im Allgemeinen in $\mathcal{O}(N^2)$ durchgeführt werden. Um dies nun effizienter zu berechnen werden wir den FFT-Algorithmus einführen und diese verwenden, um die Laufzeit für die Berechnung der numerischen Lösung gravierend zu verbessern.

Proposition 2.4 (FFT-Algorithmus, Cooley und Tukey (1965) [DV90, Kapitel 4])
 Sei $N = 2^L$ für ein $L \in \mathbb{N}$ und $w_N := e^{i\frac{2\pi}{N}} \in \mathbb{C}$. Definiere die Funktion

$$\mathcal{F}_N : \mathbb{C}^N \rightarrow \mathbb{C}^N \quad \left| \quad [x]_{k=0, \dots, N-1} \mapsto \left[\sum_{n=0}^{N-1} x_n w_N^{nk} \right]_{k=0}^{N-1} \right. \quad (2.25)$$

Für die Vektoren $(u)_j := (x)_{2j}$ und $(v)_j := (x)_{2j+1}$, $j = 0, \dots, \frac{N}{2} - 1$ erfüllt \mathcal{F}_N für $k \in \{0, \dots, \frac{N}{2} - 1\}$ die Rekursionsformel:

1. $\left[\mathcal{F}_N(x) \right]_k = \left[\mathcal{F}_{\frac{N}{2}}(u) \right]_k + w_N^k \left[\mathcal{F}_{\frac{N}{2}}(v) \right]_k,$
2. $\left[\mathcal{F}_N(x) \right]_{k+\frac{N}{2}} = \left[\mathcal{F}_{\frac{N}{2}}(u) \right]_k - w_N^k \left[\mathcal{F}_{\frac{N}{2}}(v) \right]_k.$

Mittels Prop. 2.4 können wir die diskreten Fourierkoeffizienten in (2.24) äquivalent bestimmen. Sei w_N wie vorhin und $z(x) := [x, 0_N]$, also die Konkatination eines Vektors $x \in \mathbb{R}^N$ und dem Nullvektor $0_N \in \mathbb{R}^N$. Dann gilt wegen $e^{ix} = \cos(x) + i \sin(x)$:

$$\begin{aligned} \left[\hat{u}_N(n) \right]_{n=0}^{n=N-1} &= \frac{\sqrt{2}}{N} \left[\sum_{k=0}^{N-1} u^0(x_k) \sin(\pi n k \frac{1}{N}) \right]_{n=0}^{n=N-1} \\ &= \frac{\sqrt{2}}{N} \left[\sum_{k=0}^{2N-1} z(u^0(x))_k \operatorname{Imag} \left(w_{2N}^{nk} \right) \right]_{n=0}^{n=N-1} \\ &\stackrel{\text{Prop. 2.4}}{=} \frac{\sqrt{2}}{N} \operatorname{Imag} \left[\mathcal{F}_{2N} \left(\left[z(u^0(x))_k \right]_{k=0}^{k=2N-1} \right) \right]_{n=0}^{n=N-1} \end{aligned} \quad (2.26)$$

und analog für die diskrete Funktion in (2.23):

$$\begin{aligned} \left[u_N(t, x_k) \right]_{n=0}^{n=N-1} &= \sqrt{2} \left[\sum_{k=0}^{N-1} e^{-t(\pi n)^2} \hat{u}_N(n) \sin(\pi n k \frac{1}{N}) \right]_{n=0}^{n=N-1} \\ &\stackrel{\text{Prop. 2.4}}{=} \sqrt{2} \operatorname{Imag} \left[\mathcal{F}_{2N} \left(\left[e^{-t(\pi n)^2} z(\hat{u}_N)_k \right]_{k=0}^{k=2N-1} \right) \right]_{n=0}^{n=N-1}. \end{aligned} \quad (2.27)$$

Somit erhalten wir zu einem beliebigen Zeitpunkt $t \geq 0$ eine Approximation der Funktion $u_N(t, x)$ an den Stützstellen x_k , $k \in \{0, \dots, N-1\}$, die wir nun aufgrund der Rekursionsformel in Prop. 2.4 berechnen können. Um uns nun über die Effizienz des Algorithmus überzeugen zu lassen, wollen wir ein Theorem anbringen, in der wir eine Aussage über die Laufzeit des Algorithmus erhalten.

Theorem 2.5 (Master-Theorem, [OE11, Kapitel 1.2]) *Sei die Rekursionsgleichung einer Funktion $T : \mathbb{N} \rightarrow \mathbb{R}$ gegeben durch*

$$T(n) = aT\left(\frac{n}{b}\right) + d(n), \quad n \geq n_0, \quad (2.28)$$

für $a, b > 1$, $n_0 \in \mathbb{N}$ und $d : \mathbb{N} \rightarrow \mathbb{R}$. Falls d die asymptotische Laufzeit $d(n) = \Theta(n^{\log_b(a)})$ besitzt, dann gilt:

$$T(n) = \Theta\left(n^{\log_b(a)} \log n\right).$$

Da der FFT-Algorithmus nun der Rekursionsgleichung (mit $a, b = 2$ und $d(n) = \Theta(n)$) entspricht, erhalten wir nach dem Master-Theorem, dass (2.26) und (2.27) in $\mathcal{O}(N \log(N))$ berechnet werden.

Damit haben wir nun einen Kandidaten für einen effizienten Algorithmus gefunden, der eine approximative Lösung der Wärmeleitungsgleichung an den Diskretisierungspunkten liefert.

Zum Abschluss dieses Kapitels wollen wir eine Fehlerabschätzung sowohl für die diskreten Fourier-Koeffizienten als auch für die diskrete Funktion an den Knotenpunkten herleiten und anschließend ein kurzes Beispiel zur Veranschaulichung anführen.

2.2.1 Fehlerabschätzung der numerischen Lösung

Zunächst zeigen wir die Aliasing-Formel, die wir später für die Fehlerabschätzung der diskreten Fourierkoeffizienten und der diskreten Funktion benötigen.

Proposition 2.6 (Aliasing-Formel) *Sei $\hat{u}_N(n)$ wie in (2.24) definiert und $\hat{u}(n) := \hat{u}^0(n)$. Dann gilt für alle $n \in \{1, \dots, N-1\}$:*

$$\hat{u}_N(n) = \sum_{r=0}^{\infty} \hat{u}(2rN + n) - \sum_{r=1}^{\infty} \hat{u}(2rN - n). \quad (2.29)$$

Beweis. Wir nutzen nun die Fourierreihendarstellung von $u^0(x)$, also $u^0(x) = \sum_{s=1}^{\infty} \hat{u}(s) \sqrt{2} \sin(\pi s x)$. Mithilfe der Trigonometrischen Identitäten erhalten wir nun:

$$\begin{aligned} \hat{u}_N(n) &\stackrel{(2.24)}{=} \frac{\sqrt{2}}{N} \sum_{k=0}^{N-1} u^0(x_k) \sin(\pi n \frac{k}{N}) \\ &\stackrel{\text{Satz 1.4}}{=} \frac{\sqrt{2}}{N} \sum_{k=0}^{N-1} \sin(\pi n \frac{k}{N}) \sum_{s=1}^{\infty} \hat{u}(s) \sqrt{2} \sin(\pi s \frac{k}{N}) \\ &= \frac{2}{N} \sum_{s=1}^{\infty} \hat{u}(s) \sum_{k=1}^{N-1} \sin(\pi s \frac{k}{N}) \sin(\pi n \frac{k}{N}) \\ &\stackrel{[\text{HP84}]}{=} \frac{1}{N} \sum_{s=1}^{\infty} \hat{u}(s) \left[\sum_{k=1}^{N-1} \cos(\pi(s-n) \frac{k}{N}) - \sum_{k=1}^{N-1} \cos(\pi(s+n) \frac{k}{N}) \right] \end{aligned} \quad (2.30)$$

Die Cosinus Reihen sind nun die Dirichlet Kerne ausgewertet an den Stellen $x = (s-n)\frac{\pi}{N}$.

Nach [Vre03, Kapitel 4.2] ist auch leicht einzusehen, dass für den Fall: $m \neq 0 \bmod 2N$ die Gleichung

$$\begin{aligned} \sum_{k=0}^{N-1} \cos(k \frac{\pi m}{N}) &= \frac{1}{2} \left[1 + \frac{\sin(\pi m - \frac{\pi m}{2N})}{\sin(\frac{\pi m}{2N})} \right] - (-1)^m \\ &= \frac{1}{2} (1 + (-1)^m) - (-1)^m \\ &= \begin{cases} 0, & \text{für } m \text{ gerade} \\ 1, & \text{für } m \text{ ungerade} \end{cases} \end{aligned} \quad (2.31)$$

und für den Fall $m = 0 \bmod 2N$ die Gleichung

$$\sum_{k=0}^{N-1} \cos(k \frac{\pi m}{N}) = N \quad (2.32)$$

erfüllt ist. Falls $m_1 := s-n$ ungerade ist, so ist auch $m_2 := s+n$ ungerade und umgekehrt. Das Gleiche gilt ebenfalls für den geraden Fall. Somit ist

$$\sum_{k=1}^{N-1} \cos(\pi(s-n)\frac{k}{N}) - \sum_{k=1}^{N-1} \cos(\pi(s+n)\frac{k}{N}) = 0 \quad (2.33)$$

für $s-n \neq 0 \bmod 2N$ und $s+n \neq 0 \bmod 2N$. Es bleiben nur noch folgende Fälle übrig:

$$\begin{aligned} &s-n = 0 \bmod 2N \quad (\Rightarrow s+n \text{ gerade und } s+n \neq 0 \bmod 2N) \\ \text{oder } &s+n = 0 \bmod 2N \quad (\Rightarrow s-n \text{ gerade und } s-n \neq 0 \bmod 2N). \end{aligned}$$

Dadurch verkürzt sich die Summe in (2.30) wie folgt:

$$\begin{aligned} \hat{u}_N(n) &= \frac{1}{N} \sum_{r=0}^{\infty} \hat{u}(2rN+n)[N-0] - \frac{1}{N} \sum_{r=1}^{\infty} \hat{u}(2rN-n)[0-N] \\ &= \sum_{r=0}^{\infty} \hat{u}(2rN+n) - \sum_{r=1}^{\infty} \hat{u}(2rN-n). \end{aligned} \quad (2.34)$$

□

Proposition 2.7

Sei $n \in \{0, \dots, N-1\}$ und $\hat{u}_N(n), u_N(t, x_k)$ wie in (2.24), bzw. (2.23). Dann gelten die Abschätzungen:

1. $|\hat{u}_N(n) - \hat{u}^0(n)| \leq C_p \frac{1}{N^2},$
2. $|u_N(t, x) - u(t, x)| \leq 2\sqrt{2} \sum_{n=N}^{\infty} |\hat{u}(n)|,$

für eine gewisse Konstante $C_p \geq 0$ unabhängig von n, N .

Beweis. Für $n = 0$ ist dies trivial. Sei also $n \neq 0$ und definiere $u_{xx} := \frac{\partial^2}{\partial x^2} u$. Zunächst erhalten wir folgende Abschätzung mit $C_1 = \pi^{-2} \|u_{xx}\|_{L^1(0,1)}$:

$$|\hat{u}(n)| \stackrel{(2.8)}{=} \left| -(\pi n)^{-2} \widehat{u_{xx}}(n) \right| \leq (\pi n)^{-2} \|u_{xx}\|_{L^1(0,1)} = n^{-2} C_1. \quad (2.35)$$

Die erste Ungleichung erhalten wir mit der Aliasing Formel und $C_p := \frac{C_1 \pi^2}{6}$:

$$\begin{aligned} |\hat{u}_N(n) - \hat{u}(n)| &\stackrel{(2.29)}{\leq} \sum_{r=1}^{\infty} |\hat{u}(2rN + n)| + |\hat{u}(2rN - n)| \\ &\stackrel{(2.35)}{\leq} C_1 \sum_{r=1}^{\infty} \frac{1}{(2rN + n)^2} + \frac{1}{(2rN - n)^2} \\ &\leq C_1 \sum_{r=1}^{\infty} \frac{1}{(rN)^2} \\ &= \frac{1}{N^2} C_1 \sum_{r=1}^{\infty} \frac{1}{r^2} = \frac{1}{N^2} \frac{C_1 \pi^2}{6} \end{aligned} \quad (2.36)$$

Die zweite Ungleichung erhalten wir direkt:

$$\begin{aligned} |u_N(t, x) - u(t, x)| &= \sqrt{2} \left| \sum_{n=1}^{N-1} e^{-t(\pi n)^2} \hat{u}_N(n) \sin(\pi n x) - \sum_{n=1}^{\infty} e^{-t(\pi n)^2} \hat{u}(n) \sin(\pi n x) \right| \\ &\stackrel{\Delta\text{-Ung.}}{\leq} \sqrt{2} \sum_{n=1}^{N-1} |\hat{u}_N(n) - \hat{u}(n)| + \sqrt{2} \sum_{n=N}^{\infty} |\hat{u}(n)| \\ &\stackrel{\text{Prop. 2.6}}{\leq} \sqrt{2} \sum_{r=1}^{\infty} \sum_{n=1}^{N-1} |\hat{u}(2rN + n)| + |\hat{u}(2rN - n)| + \sqrt{2} \sum_{n=N}^{\infty} |\hat{u}(n)| \\ &\leq 2\sqrt{2} \sum_{n=N}^{\infty} |\hat{u}(n)|. \end{aligned} \quad (2.37)$$

□

2.2.2 Numerische Experimente zu der Laufzeit

Sei $u(0, x) = u^0(x) := x(x - 1)$. In der Abbildung 2.1 ist ein halblogarithmischer Plot dargestellt in der die Laufzeiten für die Berechnung der Fourierkoeffizienten sowohl mit der Brute-Force-Berechnung (in grün) als auch mit dem FFT-Algorithmus (in rot) gegenüber der Potenz L ($N = 2^L$ für ein $L \in \mathbb{N}$) mit 50-facher Iteration (für die durchschnittliche Laufzeit).

Die Abbildung zeigt, dass für größere Potenzen für L , der FFT-Algorithmus in $\mathcal{O}(N \log(N))$ und die Brute-Force Berechnung in $\mathcal{O}(N^2)$ liegt.

Um die Effizienz des FFT-Algorithmus zu verdeutlichen, sind in der Tabelle 2.1 die Faktoren, in der der FFT-Algorithmus um diesen schneller ist, als die Brute-Force Methode.

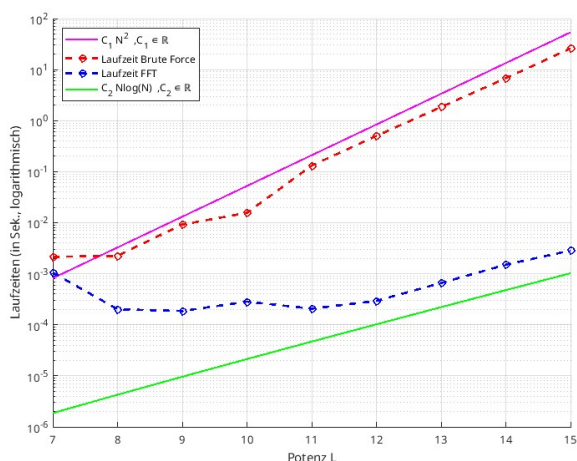


Abb. 2.1: Laufzeit Plot

Potenz L	7	8	9	10	11	12
Faktor	19	55	48	132	882	1766

Tabelle 2.1: Verdeutlichung der Effizienz vom FFT

Den Code für dieses Beispiel findet man unter dem folgenden GITHUB Link wieder:

3 Splitting-Verfahren

In diesem Kapitel befassen wir uns mit dem eigentlichen Problem der parabolischen Differentialgleichung (NL). In vielen Anwendungen werden kombinierte Effekte, z.B. in der Physik oder auch in der Chemie modelliert.

Sind diese in Form einer Differentialgleichung als Summe von Operatoren, so reduziert sich, mittels dem Splitting-Verfahren, das Problem auf einzelnen Teilprobleme. Die Lösung wird dann aus den Lösungen der Teilprobleme hergeleitet. Der Vorteil von dem Verfahren ist, dass die Teilprobleme meistens deutlich einfacher zu ermitteln sind oder sogar exakt bestimmt werden können.

Ebenso beschränkt sich die Grundidee hinter Splitting nicht nur auf eine Summe von nur zwei Operatoren, sondern auch mit beliebig vielen Summanden. Näheres in [BFCO12, Kapitel 10].

Wir werden hier und im nächsten Kapitel, um die Notation zu vereinfachen, Funktionen der Form $f : (0, 1) \rightarrow \mathbb{R}$ als Elemente in $L^2(0, 1)$ betrachten. Dadurch erhalten wir ein übersichtlicheres Problem, sodass wir uns nun auf eine Variable einschränken können.

Stellen wir diese unter den Annahmen noch einmal dar. Zu gegebenen $u^0 \in D(A)$, $g \in M$ seien die Operatoren $(Au)(t) := (\frac{\partial^2}{\partial x^2} u)(t)$ und $(Bu)(t) := g(u(t))$ gegeben. Betrachte die Differentialgleichung

$$\begin{cases} \frac{\partial}{\partial t} u(t) &= Au(t) + Bu(t), & t > 0, \\ u(0) &= u_0. \end{cases} \quad (\text{NL})$$

Um die Grundidee zu erläutern werden wir dies anhand eines Beispiels durchgehen. Sei dafür eine Schrittweite in der Zeit $\tau > 0$ gegeben. Im ersten Schritt wollen wir eine Lösung des Teilproblems $\frac{\partial}{\partial t} u = Au$ im Intervall $[0, \tau]$ mit dem Anfangswert $u(0, \cdot) = u^0$ approximieren. Das zweite Teilproblem $\frac{\partial}{\partial t} u = Bu$ wird im selben Intervall, dessen Anfangswert nun die Lösung des ersten Teilproblems zum Zeitpunkt τ ist, gelöst. Im nächsten Schritt werden die Teilprobleme auf $[\tau, 2\tau]$ betrachtet und nach dem selben Prinzip wieder gelöst (die Initialfunktion wird immer als Lösung des vorherigen Teilproblems genutzt). Im nächsten Unterkapitel werden wir dieses Prinzip mathematisch auffassen.

Bemerkung: Die Anfangswerte sind hier Funktionen aus $L^2(0, 1)$

3.1 Klasse von Splitting-Verfahren

Durch die Aufteilung in zwei kleinere Teilprobleme erhalten wir direkt ein Verfahren. Sei dazu ein Zeitgitter gegeben durch $t_k := k\tau$ mit $k = 0, \dots, p-1$ und einer Schrittweite $\tau > 0$ und $t \in [t_k, t_{k+1}]$. Sei die approximierte Lösung u^k zum Zeitpunkt t_k gegeben. Die jeweiligen Teilprobleme im Intervall $[t_k, t_{k+1}]$ erhält man durch:

$$\begin{aligned} 1) \quad & \frac{d}{dt} u_A(t, x) = \frac{d^2}{dx^2} u_A(t, x), & u_A(t_{k-1}, x) &= u^k(x), & t &\in [t_k, t_{k+1}], \\ 2) \quad & \frac{d}{dt} u_B(t, x) = g(u(t, x)), & u_B(t_{k-1}, x) &= u_A(t_k, x), & t &\in [t_k, t_{k+1}]. \end{aligned}$$

und setze $u^{k+1} := u_B(t_k)$. Eine ausführliche Darstellung zu diesem Ansatz ist im Abschnitt 3.2.1 angeführt. Zunächst wollen wir diesen rekursiven Aufruf kompakter darstellen. Dazu werden wir die Lösungen für die Teilproblem explizit definieren und führen, nach [EN00, Kapitel 3.28], den Begriff des Flusses ein.

Definition 3.1. Sei \mathcal{A} ein auf M definierter Operator und $u^{t_0} \in D(\mathcal{A})$. Betrachte die Differentialgleichungen $\frac{\partial}{\partial t} u(t, x) = \mathcal{A}u(t, x)$ mit Initialwert $u(t_0, x) = u^{t_0}$ für ein $t_0 \geq 0$. Dann definieren wir die Lösung zum Zeitpunkt $t_1 > t_0$ als:

$$\Phi_{\mathcal{A}}^{t_0, t_1} u^{t_0}.$$

Alternativ kann man mittels des Flusses die nächsten Iterierten definieren:

$$u^{k+1} = \Phi_B^{t_{k-1}, t_k} \Phi_A^{t_{k-1}, t_k} u^k.$$

Durch eine Verallgemeinerung der rekursiven Aufrufe erhalten wir eine generelle Klasse von Splitting Verfahren der Form

$$u^{k+1} = F(\tau, t_k) u^k.$$

Mit diesem allgemeinen Ansatz können wir nun die approximierte Lösung zur Zeit T definieren.

Definition 3.2. Sei $\tau > 0$ eine geeignete Schrittweite und $T = \tau p$ für ein $p \in \mathbb{N}$. Die approximierte Lösung zum Problem (NL) zum Zeitpunkt t_k ist gegeben durch:

$$u^{k+1} = F(\tau, t_k) \dots F(\tau, t_0) u^0 \tag{3.1}$$

für $k = 0, \dots, p-1$ und $u^0 \in D(\mathcal{A})$.

In dieser Arbeit werden wir uns mit zwei Splitting-Verfahren beschäftigen. Diese lauten:

1. **Sequentielle Splitting:** $F_{\text{Seq}}(\tau, t_k) = \Phi_B^{t_k, t_{k+1}} \Phi_A^{t_k, t_{k+1}},$
2. **Marchuk-Strang Splitting:** $F_{\text{Strang}}(\tau, t_k) = \Phi_A^{t_k, t_k + \frac{\tau}{2}} \Phi_B^{t_k, t_{k+1}} \Phi_A^{t_k + \frac{\tau}{2}, t_{k+1}}.$

3.2 Splitting-Verfahren für das Problem (NL)

Wir wollen nun beide Splitting Verfahren exakt zum Problem (NL) darstellen und miteinander vergleichen. Stellen wir das Problem übersichtshalber nochmal vor:

$$\begin{cases} \frac{\partial}{\partial t} u(t) = Au(t) + Bu(t), & t > 0, \\ u(0) = u_0. \end{cases} \quad (\text{NL})$$

Betrachten wir das Zeigitter $t_k = k\tau$ für $k = 0, \dots, p-1$ und eine geeignete Schrittweite τ , sodass $T = \tau p$ für ein $p \in \mathbb{N}$. Sei dazu u^k die jeweilige numerische Lösung zum Zeitpunkt t_k . Die nächste Iterierte u^{k+1} wird durch das jeweilige Verfahren zum Zeitpunkt t_{k+1} bestimmt. Zunächst betrachten wir das sequentielle Splitting-Verfahren und im nächsten Unterabschnitt das Marchuk-Strang Splitting-Verfahren.

3.2.1 Sequentielles Splitting-Verfahren

Im Folgenden ist eine kompakter Überblick, welche Probleme in welchem Intervall gelöst werden müssen. Daraufaufgehend ist eine kurze Erklärung gegeben, wie sich das Verfahren aufbaut. Die explizite Reihenfolgen der zu lösenden Teilprobleme mit A und B ist gegeben durch:

- 1) $\frac{d}{dt} u_A(t, x) = \frac{d^2}{dx^2} u_A(t, x), \quad u_A(t_k, x) = u^k(x), \quad , t \in [t_k, t_{k+1}],$
- 2) $\frac{d}{dt} u_B(t, x) = g(u(t, x)), \quad u_B(t_k, x) = u_A(t_{k+1}, x), \quad , t \in [t_k, t_{k+1}].$

Angenommen wir haben eine approximierte Lösung u^k für (NL) zum Zeitpunkt $t = t_k$ für ein $k = 0, \dots, p-1$.

Um eine approximierte Lösung zum Zeitpunkt t_{k+1} zu ermitteln, wird zunächst das Problem, wie in 1) dargestellt, im Intervall $[t_k, t_{k+1}]$ gelöst.

Die Lösung dieses Problems ist zum Zeitpunkt t_{k+1} durch $u_A(t_{k+1}, x)$ gegeben und dient gleichzeitig als Anfangswert für das Problem 2) im Intervall $[t_k, t_{k+1}]$.

Sei $u_B(t_{k+1}, x)$ die Lösung des Problems 2), dann ist $u_B(t_{k+1}, x)$ unsere approximierte Lösung des Problems (NL) zum Zeitpunkt t_{k+1} , also $u^{k+1} = u_B(t_{k+1}, x)$.

Mithilfe der Definition des Flusses erhalten wir die folgende Darstellung des Verfahrens.

Definition 3.3 (Sequentielle Splitting Verfahren [BFCO12, Kapitel 10]). Sei $T := \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$ und $u^0 \in D(A)$. Dann ist das sequentielle Splitting-Verfahren bis zur Zeit T definiert durch:

$$u^{k+1} := \phi_{\text{Sec}}(t_k, u^k) := \Phi_B^{t_k, t_{k+1}} \Phi_A^{t_k, t_{k+1}} u^k, \quad t_k = k\tau, \quad (3.2)$$

wobei $k = 0, \dots, p-1$.

Wir haben bereits in der Einleitung festgestellt, dass genau dieses Verfahren ziemlich intuitiv hergeleitet wurde, in dem wir die Probleme A und B hintereinander im selben Intervall lösen. Nichtsdestotrotz rechtfertigt sich dieses Verfahren durch die folgende Proposition.

Proposition 3.4 ([BFCO12, Kapitel 10.2]) Seien $T = p\tau$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$ und $g \in D(B)$. Für die Operatoren $(Au)(t) = (\frac{\partial^2}{\partial x^2} u)(t)$ und $(Bu)(t) = g(u(t))$ sei u die Lösung der Differentialgleichung mit den D.-R.

$$\begin{cases} \frac{\partial}{\partial t} u(t) = Au(t) + Bu(t), & 0 < t < T \\ u(0) = u_0 \end{cases} \quad (\text{NL})$$

Dann hat das sequentielle Splitting-Verfahren eine Konsistenzordnung von 1, also es gilt:

$$\|\phi_{\text{Sec}}(t_k, u(t_k)) - u(t_{k+1})\| \leq C_{\text{Seq}} h^2 \quad (3.3)$$

für eine gewisse Konstant $C_{\text{Seq}} > 0$.

3.2.2 Marchuk-Strang Splitting-Verfahren

Wir haben nach der letzten Proposition mit dem sequentiellen Splitting Verfahren somit ein Verfahren erster Ordnung erhalten. Wir wollen nun ein weiteres Verfahren anführen, in der wir sogar eine Konsistenzordnung von 2 erreichen, dem Marchuk-Strang-Splitting Verfahren. Erneut wird zunächst die kompakte Darstellung und anschließend die Erklärung zu diesem Verfahren gegeben.

- 1) $\frac{d}{dt} u_A(t, x) = \frac{d^2}{dx^2} u_A(t, x), \quad u_A(t_k, x) = u^k, \quad , t \in [t_k, t_k + \frac{\tau}{2}],$
- 2) $\frac{d}{dt} u_B(t, x) = g(u(t, x)), \quad u_B(t_k, x) = u_A(t_k + \frac{\tau}{2}, x), \quad , t \in [t_k, t_{k+1}],$
- 3) $\frac{d}{dt} u_A(t, x) = \frac{d^2}{dx^2} u_A(t, x), \quad u_A(t_k + \frac{\tau}{2}, x) = u_B(t_k, x), \quad , t \in [t_k + \frac{\tau}{2}, t_{k+1}].$

Bemerkung: Die Probleme in 1) und 3) sind unabhängig voneinander. Die Benennung der Funktion u_A soll nur illustrieren, dass die Probleme in 1) und 3) ausschließlich vom Operator A abhängen.

Angenommen wir haben eine approximierte Lösung u^k für (NL) zum Zeitpunkt $t = t_k$ für ein $k = 0, \dots, p-1$.

Um eine approximierte Lösung zum Zeitpunkt t_{k+1} zu ermitteln, wird zunächst das Problem, wie in 1) dargestellt, im Intervall $[t_k, t_k + \frac{\tau}{2}]$ gelöst.

Die Lösung dieses Problems ist durch $u_A(t_k + \frac{\tau}{2}, x)$ gegeben und dient gleichzeitig als Anfangswert für das Problem 2) im Intervall $[t_k, t_{k+1}]$.

Die Lösung dieses zweiten Problems ist zum Zeitpunkt t_{k+1} durch $u_B(t_{k+1}, x)$ gegeben, welche wiederum als Anfangswert für das Problem 3) im Intervall $[t_k + \frac{\tau}{2}, t_{k+1}]$ verwendet wird.

Sei $u_A(t_{k+1}, x)$ die Lösung des Problems 3), dann ist $u_A(t_{k+1}, x)$ unsere approximierte Lösung des Problems (NL) zum Zeitpunkt t_{k+1} , also $u^{k+1} = u_A(t_{k+1}, x)$.

Das daraus resultierende Verfahren beschränkt sich somit auf die Lösung von drei Teilproblemen. Aufgrund der Äquivalenz des ersten und dritten Teilproblems verkompliziert sich das Verfahren im Vergleich zum sequentiellen Splitting jedoch nicht.

Mithilfe der Definition des Flusses erhalten wir die folgende Darstellung des Verfahrens.

Definition 3.5 (Marchuk-Strang Splitting-Verfahren [BFCO12, Kapitel 10]). Sei $T := \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$ und $u^0 \in D(A)$.

Dann ist das Marchuk-Strang Splitting-Verfahren definiert durch:

$$u^{k+1} := \phi_{\text{Strang}}(t_k, u^k) := \Phi_A^{t_k, t_k + \frac{h}{2}} \Phi_B^{t_k, t_{k+1}} \Phi_A^{t_k + \frac{h}{2}, t_{k+1}} u^k, \quad t_k = k\tau, \quad (3.4)$$

wobei $k = 0, \dots, p-1$.

Eine Aussage über die Konsistenzordnung dieses Verfahrens gibt uns nun folgende Proposition:

Proposition 3.6 ([BFCO12, Kapitel 10.3])

Seien $T = p\tau$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$ und $g \in D(B)$. Für die Operatoren $(Au)(t) = (\frac{\partial^2}{\partial x^2} u)(t)$ und $(Bu)(t) = g(u(t))$ sei u die Lösung der Differentialgleichung mit den D.-R.

$$\begin{cases} \frac{\partial}{\partial t} u(t) = Au(t) + Bu(t), & 0 < t < T, \\ u(0) = u_0. \end{cases} \quad (\text{NL})$$

Dann hat das Marchuk-Strang Splitting-Verfahren eine Konsistenzordnung von 2, also es gilt:

$$\|\phi_{\text{Strang}}(t_k, u(t_k)) - u(t_{k+1})\| \leq C_{\text{Strang}} h^3 \quad (3.5)$$

für eine gewisse Konstante $C_{\text{Strang}} > 0$.

Nun können wir mittels einer weiteren Raumdiskretisierung mit der Schrittweite $h = \frac{1}{N}$ mit $N = 2^L$ für ein $L \in \mathbb{N}$ und mittels der in Kapitel 2 herausgearbeitetem FFT-Algorithmus eine approximative Lösung für die Wärmeleitungsgleichung angeben, mit der das Teilproblem mit A effizient berechnet werden kann.

3.3 Numerische Experimente zum Splitting-Verfahren

Um die Konsistenzordnung zu erhalten, müssen die jeweiligen Teilprobleme mit einem Verfahren der selben Konsistenzordnung gelöst werden. Für beide Verfahren nutzen wir, um das Teilproblem mit B zu lösen, somit die Trapezregel [DR22, Kapitel 10.1].

Wir wollen das Verfahren qualitativ anhand eines Beispiels untersuchen. Betrachten wir dazu die Funktionen $u(t, x) = 10e^t x^3(1 - x)^3$ und $g(t, x) = \frac{\partial}{\partial t} u(t, x) - \frac{\partial^2}{\partial x^2} u(t, x)$ bis zur Zeit $T = 1$. Die Funktion u ist somit die Lösung der Differentialgleichung (LE) und es gilt: $u, g \in M$. Damit sind die Voraussetzungen für die Anwendung des Splitting Verfahrens erfüllt.

Hinweis: In unserer Betrachtung haben wir den impliziten Ausdruck $g(u(t))$ aus dem Problem (NL) durch $g(t) \in M$ ersetzt. Die Konsistenzordnung für das sequentielle Splitting-Verfahren und für das Marchuk-Strang Splitting-Verfahren bleiben dennoch erhalten.

In den Abbildungen 3.1 und 3.2 ist ein loglog-Plot dargestellt, in der die Zeitschrittweite, immer verdoppelt, gegenüber dem globalen Fehler beider Verfahren geplottet ist. Der Vergleich mit den Referenzlinien zeigen die qualitative Übereinstimmung mit der im ausgearbeitetem Inhalt.

In den darunter stehenden Abbildungen 3.3 und 3.4 sind die Laufzeiten beider Verfahren einmal mit dem FFT- Algorithmus 2.4 und einmal mittels der Matrix-Vektor Multiplikation gegenüber der Potenz L dargestellt, wobei $N = 2^L$.

Bemerkung: Wir erinnern uns daran, dass N sowohl die Anzahl Knotenpunkte in $[0, 1]$, als auch die Größe der abgeschnittenen Fourierreihe wiedergibt. Im Abschnitt 2.2.1 haben wir gezeigt, dass die Fehlerabschätzung für das Teilproblem A unabhängig von der gewählten Schrittweite ist. Daher mussten wir ein genügend großes N wählen, sodass der durch das Teilproblem A verursachte Fehler asymptotisch vernachlässigbar ist.

Die Codes für dieses Beispiel findet man unter dem folgenden GitHub Link wieder:

3.3 Numerische Experimente zum Splitting-Verfahren

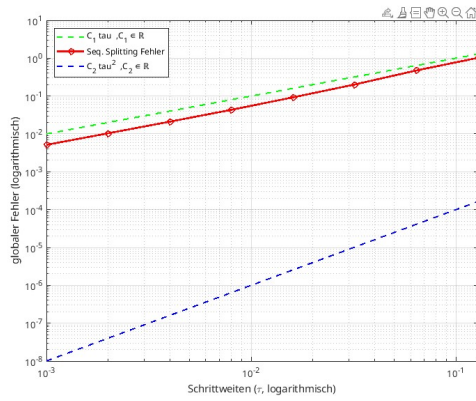


Abb. 3.1: Fehler des seq. Splitting-Verfahrens mit $N = 256$, ($L = 9$).

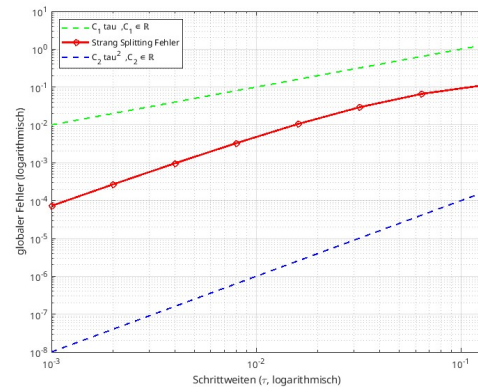


Abb. 3.2: Fehler des M.-Strang Splitting-Verfahrens mit $N = 256$, ($L = 9$).

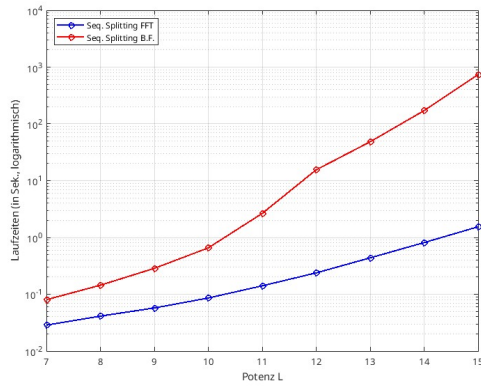


Abb. 3.3: Laufzeiten des seq. Splitting-Verfahrens mit $\tau = 0.001$, ($N = 2^L$).

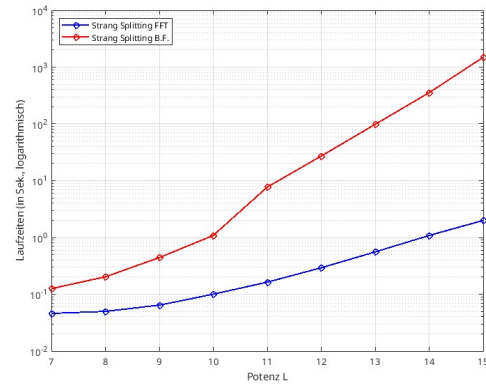


Abb. 3.4: Laufzeiten des M.-Strang Splitting-Verfahrens mit $\tau = 0.001$, ($N = 2^L$).

4 Exponentielle Integratoren

Nun wollen wir eine weitere Methode kennenlernen, in der wir das Problem (NL) lösen wollen. Wir werden uns hier wieder an die Fourierreihenentwicklung und das in Kapitel 2 ausgearbeiteten Verfahren, um die Wärmeleitungsgleichung (HE) zu lösen, heranziehen.

Wir wollen das Problem noch einmal darstellen. Zu gegebenen $u^0 \in D(A)$, $g \in M$ seien die Operatoren $(Au)(t) := (\frac{\partial^2}{\partial x^2} u)(t)$ und $(Bu)(t) := g(u(t))$ gegeben. Betrachte die Differentialgleichung mit den D.-R.:

$$\begin{cases} u_t(t, x) = Au(t, x) + g(u(t, x)), & t > 0, \\ u(0, x) = u^0(x). \end{cases} \quad (\text{NL})$$

Im Splitting Verfahren konnten wir das eigentliche Problem in zwei kleinere Teilprobleme zerlegen, die wir einfacher lösen konnten. Im Gegensatz dazu wollen wir für eine existierende Lösung u , diese mithilfe der Variation der Konstanten Formel approximieren. Allerdings werden wir zuvor, um den Einstieg zu schaffen, zunächst die Lösung der Wärmeleitungsgleichung (HE) zum Zeitpunkt $t > 0$ kompakt darstellen. Dieser Ausdruck wird nämlich in der bereits genannten Formel auftauchen und wird wie folgt definiert.

Definition 4.1 ([HO10, Kapitel 2.1]). Sei $v \in D(A)$ und $v^0 = \sum_{n=1}^{\infty} \hat{v}_n f_n$ die entsprechende Fourierreihe. Dann bezeichne die Funktion

$$e^{-tA} : D(A) \rightarrow L^2(0, 1) \left| \sum_{n=1}^{\infty} \hat{v}_n f_n \rightarrow \sum_{n=1}^{\infty} e^{-(\pi n)^2 t} \hat{v}_n f_n \right. \quad (4.1)$$

als den *Exponentialoperator*.

Bemerkung: Betrachten wir die Differentialgleichung

$$\begin{cases} \frac{\partial}{\partial t} u(t) = Au(t), & t > 0, \\ u(0) = u^0. \end{cases} \quad (4.2)$$

Die Lösung dieser Gleichung ist, wie in Kapitel 2 ausgearbeitet, gegeben durch $u(t) := e^{-tA} u^0$. Mit dieser entsprechenden Kurznotation können wir die implizierte Lösung durch die Variation der Konstanten Formel formulieren.

Falls die Funktion u die DGL. (NL) löst, so kann man u mittels der Variation der Konstanten Formel ausdrücken als

$$u(t) = e^{-tA}u^0 + \int_0^t e^{-(t-\tau)A}(g \circ u)(\tau) \, d\tau. \quad (4.3)$$

Den linke Teil der Summe kennen wir bereits aus Kapitel 2. Der rechte Teil ist aufgrund der unbekannten Funktion $u(\tau)$ sehr schwierig zu ermitteln. Wir wollen nun in diesem Kapitel das Integral rechts, mittels Interpolation der Funktion $(g \circ u)(\tau)$, approximieren.

Falls dieses Integral für den Leser:innen schwer zu handhaben ist, möchte ich darauf hinweisen, dass man sich in diesem Fall immer auf die Fourierreihenentwicklung und somit auf die Variation der Konstanten Formel im reellen Fall orientieren kann:

$$u(t_k, x) = \sum_{n=1}^{\infty} \hat{u}_n(t_k) f_n(x), \quad (4.4)$$

wobei aufgrund der Eindeutigkeit die Identität nach den Integrationsregeln gilt:

$$\hat{u}_n(t_{k+1}) = e^{-\tau(\pi n)^2} \hat{u}_n(t_k) + \int_0^{\tau} e^{-(\tau-v)(\pi n)^2} \widehat{(g \circ u)_n}(t_k + v) \, dv. \quad (4.5)$$

4.1 Das exponentielle Runge-Kutta-Verfahren

In diesem Abschnitt wollen wir kurz die Idee in der Herleitung des exponentiellen Runge-Kutta-Verfahrens wiedergeben. Zunächst erhalten wir mittels dem Zeitgitter

$$t_k = \tau k, \quad \text{für } k = 0, \dots, p-1, \quad (4.6)$$

folgende Vereinfachung des Integrals (4.3)

$$\begin{aligned} u(t_{k+1}) &= e^{-t_{k+1}A}u^0 + \int_0^{t_{k+1}} e^{-(t_{k+1}-v)A}(g \circ u)(v) \, dv \\ &= e^{-\tau A}u(t_k) + \int_0^{\tau} e^{-(\tau-v)A}(g \circ u)(t_k + v) \, dv. \end{aligned} \quad (4.7)$$

Um das Integral zu approximieren werden wir dafür die Funktion $(g \circ u)$ an jeweils paarweise verschiedenen Interpolationsknoten c_1, \dots, c_s in $[0, 1]$ für eine $s \in \mathbb{N}$ interpolieren. Die Vorgehensweise wird im nächsten Abschnitt erläutert.

4.1.1 Approximation durch Interpolation

Seien die Interpolationsknoten $c_1, \dots, c_s \in [0, 1]$ gegeben. Dann definieren wir die nächsten Iterierten als

$$u^{k+1} := e^{-\tau A}u^k + h \sum_{i=1}^s b_i(-\tau A)(g \circ u)(t_k + \tau c_i) \quad (4.8)$$

mit entsprechenden Gewichten

$$b_i(z) = \int_0^1 e^{(1-\theta)z} \ell_i(\theta) \, d\theta. \quad (4.9)$$

Die Funktionen $\ell_i(\theta)$ entsprechen den Interpolationspolynomen

$$\ell_i(\theta) = \prod_{\substack{m=1 \\ m \neq i}}^s \frac{\theta - c_m}{c_i - c_m}. \quad (4.10)$$

Da ℓ_i nun Polynome vom Grad $s-1$ sind, sind die Gewichte $b_k(z)$ eine Linearkombination der Funktionen

$$\varphi_k(z) = \int_0^1 e^{z(1-\theta)} \frac{\theta^{k-1}}{(k-1)!} \, d\theta, \quad k = 1, \dots, s. \quad (4.11)$$

Diese erfüllen die Rekurrenzrelation

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_0(z) = e^z. \quad (4.12)$$

Anstelle die Gewichte $b_i(-\tau A)$ exakt auszurechnen, würde somit nur die Bestimmung der Koeffizienten der Polynome ℓ_i und die Rekurrenzrelation genügen.

4.1.2 Klasse der exponentiellen RKV

Seien wieder die Interpolationsknoten $c_1, \dots, c_s \in [0, 1]$ gegeben. In [HO10, Kapitel 2.3] wird eine generelle Klasse von exponentiellen RKV der Form

$$\begin{aligned} u_{k+1} &= \chi(-\tau A)u_k + \tau \sum_{i=1}^s b_i(-\tau A)g(U_{ki}), \\ U_{ki} &= \chi_i(-\tau A)u_k + \tau \sum_{j=1}^s a_{ij}(-\tau A)g(U_{kj}), \end{aligned} \quad (4.13)$$

angebracht, wobei die Koeffizienten $\chi, \chi_i, b_i, a_{ij}$ so konstruiert werden, dass das Verfahren eine gewisse Konvergenzordnung besitzt. Um eine bessere Übersicht dieses Schemata zu erhalten, werden dafür Butcher-Tablues 4.1 verwendet. Unter der Deklaration

$$\chi(z) := e^z, \quad \chi_i(z) := e^{c_i z}, \quad i = 1, \dots, s, \quad (4.14)$$

bauen sich die verschiedensten explizite RKV auf (siehe [HO10, Kapitel 2.3], insbesondere die im weiteren aufgeführten Verfahren.

c_1				
c_2	$a_{21}(-\tau A)$			
\vdots	\vdots	\vdots	\ddots	
c_s	$a_{s1}(-\tau A)$	\cdots	$a_{s,s-1}(-\tau A)$	
<hr/>				
	$b_1(-\tau A)$	\cdots	$b_{s-1}(-\tau A)$	$b_s(-\tau A)$

Tabelle 4.1: Butcher-Tableau für exponentiellen Integratoren

Bemerkung: Das in Tabelle 4.1 illustrierte Butcher-Tableau soll die expliziten Verfahren einmal hervorheben. Für implizite Verfahren verkompliziert sich entsprechend das Verfahren aufgrund der impliziten Aufrufe der unbekannten Funktion u natürlich enorm.

4.2 Exponentielle Euler-Verfahren für das Problem (NL)

Das wohl einfachste Verfahren für (4.8) ist die Wahl nur eines Interpolationspunktes $c_1 = 0$. Dadurch erhalten wir den Ausdruck:

$$u^{k+1} = e^{-hA}u^k + h\varphi_1(-hA)g(u(t_k)). \quad (4.15)$$

Daraus ergibt sich das *exponentielle Euler-Verfahren*.

Definition 4.2 (Exponentielle Euler-Verfahren [HO10, Kapitel 2.3]). Seien $T = \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$ und $g \in D(B)$.

Dann ist das exponentielle Euler-Verfahren bis zur Zeit T definiert durch:

$$u^{k+1} = \phi_{\text{Exp.Euler}}(t_k, u^k) := e^{-\tau A}u^k + \tau\varphi_1(-\tau A)g(u(t_k)), \quad (4.16)$$

für $k = 0, \dots, p-1$.

Das dazugehörige Butcher-Tableau zu diesem Verfahren lässt sich lesen als

0	
	φ_1

Tabelle 4.2: Butcher-Tableau exp. Euler-Verfahren

Wir erhalten hier ebenfalls eine Aussage über die Konsistenzordnung, falls wir eine numerische Lösung bis zur Zeit $T > 0$ suchen.

Proposition 4.3 ([HO10, Kapitel 2.3]) *Seien $T = p\tau$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$ und $g \in D(B)$. Für den Operator $(Au)(t) = (\frac{\partial^2}{\partial x^2}u)(t)$ sei u die Lösung der Differentialgleichung mit den D.-R.*

$$\begin{cases} \frac{\partial}{\partial t}u(t) = Au(t) + g(u(t)), & 0 < t < T, \\ u(0) = u_0. \end{cases} \quad (\text{NL})$$

Dann hat das exponentielle Euler-Verfahren $\phi_{\text{Exp.Euler}}(t_k, u^k)$ eine Konsistenzordnung von 1, also es gilt:

$$\left\| \phi_{\text{Exp.Euler}}(t_k, u^k) - u(t_{k+1}) \right\| = \mathcal{O}(\tau^2), \quad t_k = k\tau, \quad (4.17)$$

für $k = 0, \dots, p-1$.

4.3 Die exponentielle Quadraturregeln für das Problem (LE)

In diesem Abschnitt betrachten wir ein ähnliches Problem wie zu (NL). Dazu ändern wir den impliziten Aufruf von $g(u(t, x))$ zu $g(t, x)$.

Der Grund für die Untersuchung dieses Problems liegt darin, dass für die numerischen Experimente die unbekannte Lösung nicht benötigt wird. Im Abschnitt 4.5 zu dem sehen, wie wir aus einer gegebenen Funktion $u \in M$, genau die Lösung einer Differentialgleichung der Form (LE) für ein geeignetes $g \in M$ erhält. Sei also $T > 0$, $u^0 \in D(A)$ und $g \in M$. Wir betrachten die Differentialgleichung mit den D.R.:

$$\begin{cases} u_t(t, x) &= Au(t, x) + g(t, x), & t > 0, \\ u(0, x) &= u^0(x). \end{cases} \quad (\text{LE})$$

Der Unterschied zu (NL) besteht hier lediglich darin, dass g nicht mehr von der Funktion u abhängig ist. Wie bereits in Abschnitt 4.1 gezeigt, erhalten wir die exakte Lösung zu (LE), mittels der Variation der Konstanten Formel:

$$u(t_{k+1}) = e^{-\tau A}u(t_k) + \int_0^\tau e^{-(\tau-v)A}g(t_k + v) \, dv. \quad (4.18)$$

Dieses Problem reduziert sich erneut auf die Fourierkoeffizienten, analog zu der Gleichung (4.5):

$$\hat{u}_n(t_{k+1}) = e^{-\tau(\pi n)^2}\hat{u}_n(t_k) + \int_0^\tau e^{-(\tau-v)(\pi n)^2}\hat{g}_n(t_k + v) \, dv. \quad (4.19)$$

Den Beweis dazu, dass u in (4.18) tatsächlich in $D(M)$ liegt und die Differentialgleichung (LE) erfüllt überlassen wir an dieser Stelle dem Leser. Mit dem selben Ansatz wie in Unterabschnitt 4.1.1 erhalten wir durch Interpolation an den paarweise verschiedenen

Interpolationspunkten c_1, \dots, c_s in $[0, 1]$ die nächsten Iterierten durch:

$$u^{k+1} = e^{-\tau A} u_n^k + \tau \sum_{i=1}^s b_i(-\tau A) g_n(t_k + \tau c_i). \quad (4.20)$$

Daraus erhalten wir eine Klasse von numerischen Verfahren, nämlich die *exponentiellen Quadraturregeln*.

Definition 4.4 (exponentielle Quadraturregel [HO10, Kapitel 2.2]). Seien $T = \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$, $g \in M$ und $c_1, \dots, c_s \in [0, 1]$ jeweils paarweise verschieden für ein $s \in \mathbb{N}$.

Dann ist die exponentielle Quadraturregel bis zur Zeit T definiert durch:

$$u^{k+1} = \phi_{\text{Quad.}}(t_k, u^k) := e^{-\tau A} u^k + \tau \sum_{i=1}^s b_i(-\tau A) g(t_k + \tau c_i), \quad (4.21)$$

wobei $k = 0, \dots, p-1$.

Durch dieses Verfahren erhalten wir, angewendet auf das Problem (LE), eine Aussage über die Konsistenzordnung.

Lemma 4.5 ([HO10, Kapitel 2.2]) Seien $T = \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$, $u^0 \in D(A)$, $g \in D(B)$ und $c_1, \dots, c_s \in [0, 1]$ jeweils paarweise verschieden für ein $s \in \mathbb{N}$. Für den Operator $(Au)(t) = (\frac{\partial^2}{\partial x^2} u)(t)$ sei u die Lösung der Differentialgleichung

$$\begin{cases} \frac{\partial}{\partial t} u(t) &= Au(t) + g(t), & 0 < t < T, \\ u(0) &= u^0. \end{cases} \quad (\text{LE})$$

Dann hat das numerische Verfahren ϕ_{Quad} bis zur Zeit T gemäß Definition 4.4 eine Konsistenzordnung von s , also es gilt:

$$\left\| \phi_{\text{Quad.}}(t_k, u^k) - u(t_{k+1}) \right\| = \mathcal{O}(h^{s+1}), \quad t_k = k\tau, \quad (4.22)$$

für $k = 0, \dots, p-1$.

4.4 Beispiele einiger exponentiellen Quadraturregeln

Wir betrachten in diesem Abschnitt die einfachsten Quadraturregeln und analysieren deren qualitative Fehlerentwicklung anhand eines Beispiels. Dazu führen wir zwei Verfahren der Quadraturregeln ein: Zuerst die *exponentielle Euler-Quadraturregel*, bei der nur ein Interpolationsknoten verwendet wird, gefolgt von der *exponentielle Trapez-Quadraturregel* mit zwei Interpolationsknoten.

Aufgrund des Lemmas 4.5 geht hervor, dass die Euler-Quadraturregel eine Konsistenzordnung von 1 hat und die Trapez-Quadraturregel eine Konsistenzordnung von 2 besitzt.

4.4.1 Exponentielle Euler-Quadraturregel

Durch die Wahl eines Interpolationspunktes mit $c_1 = 0$ erhalten wir die exponentielle Euler-Quadraturregel:

$$u^{k+1} = e^{-hA}u^k + h\varphi_1(-hA)g(t_k) \quad (4.23)$$

Das daraus resultierende Verfahren erhalten wir durch:

Definition 4.6 (Exponentielle Euler-Quadraturregel [HO10, Kapitel 2.2]). Sei $T = \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$. Dann ist die exponentielle Euler-Quadraturregel bis zur Zeit T definiert durch:

$$u^{k+1} = \phi_{\text{Quad.Eul}}(t_k, u^k) := e^{-hA}u^k + h\varphi_1(-hA)g(t_k) \quad (4.24)$$

für $k = 0, \dots, p-1$.

4.4.2 Exponentielle Trapez-Quadraturregel

Durch die Wahl zweier Interpolationspunkte mit $c_1 = 0, c_2 = 1$ erhalten wir aus den durch die Rekurrenzrelation ermittelten Gewichten

$$\begin{aligned} b_1(z) &= \varphi_1(z) - \varphi_2(z), \\ b_2(z) &= \varphi_2(z), \end{aligned} \quad (4.25)$$

die exponentielle Trapez-Quadraturregel

$$u^{k+1} = e^{-\tau A}u^k + h(b_1(-\tau A)g(t_k) + b_2(-\tau A)g(t_{k+1})). \quad (4.26)$$

Das daraus resultierende Verfahren ergibt sich nun durch

Definition 4.7 (Exponentielle Trapez-Quadraturregel [HO10, Kapitel 2.2]). Sei $T = \tau p$ für ein $p \in \mathbb{N}$ und einer Schrittweite $\tau > 0$. Dann ist die exponentielle Trapez-Quadraturregel bis zur Zeit T definiert durch:

$$u^{k+1} = \phi_{\text{Quad.Trapez}}(t_k, u^k) := e^{-\tau A}u^k + \tau(b_1(-\tau A)g(t_k) + b_2(-\tau A)g(t_{k+1})) \quad (4.27)$$

für $k = 0, \dots, p-1$.

4.5 Numerische Experimente zu den exp. Quadraturregeln

Die in den Abbildungen 4.1 und 4.2 illustrierten Graphen wurden durch den Zugang der Fourierkoeffizienten programmiert. Betrachten wir beispielsweise den Term $\tau\varphi_1(-\tau A)g(t_k)$ der exponentiellen Euler-Quadraturregel. Unter der Annahme, dass $g(t_k)$ sich in eine Fourierreihe entwickeln lässt, vereinfacht sich daher der Ausdruck (4.23) direkt mittels der Rekurrenzrelation zu:

$$\begin{aligned}\tau\varphi_1(-\tau A)g(t_k) &= \sum_{n=1}^{\infty} \tau \int_0^1 e^{-\tau(1-\tau)(\pi n)^2} d\tau \hat{g}_n(t_k) \\ &= \sum_{n=1}^{\infty} \frac{1 - e^{-\tau(\pi n)^2}}{(\pi n)^2} \hat{g}_n(t_k).\end{aligned}\tag{4.28}$$

Analog erhält man einen Ausdruck für den in der exponentiellen Trapez-Quadraturregel auftauchenden $\varphi_2(-\tau A)$.

Wir wollen das Verfahren erneut qualitativ anhand des bereits eingeführten Beispiels untersuchen. Betrachten wir dazu die Funktionen $u(t, x) = 10e^t x^3(1-x)^3$ und $g(t, x) = \frac{\partial}{\partial t} u(t, x) - \frac{\partial^2}{\partial x^2} u(t, x)$ bis zur Zeit $T = 1$. Die Funktion u ist somit die Lösung der Differentialgleichung (LE) und es gilt: $u, g \in M$. Damit sind die Voraussetzungen für die Konsistenzordnung der exponentiellen Quadraturregeln erfüllt.

In den beiden Abbildungen 4.1 und 4.2 ist ein loglog-Plot dargestellt, in der die Zeitschrittweite, immer verdoppelt, gegenüber dem globalen Fehler beider Verfahren geplottet ist. Der Vergleich mit den Referenzlinien zeigen die qualitative Übereinstimmung mit der im vorherigen ausgearbeiteten Inhalt.

In den darunter stehenden Abbildungen 4.3 und 4.4 sind die Laufzeiten beider Verfahren einmal mit dem FFT-Algorithmus 2.4 und einmal mittels der Matrix-Vektor Multiplikation (Brute-Force) gegenüber der Potenz L dargestellt, wobei $N = 2^L$

Bemerkung: Wir erinnern uns daran, dass N sowohl die Anzahl Knotenpunkte in $[0, 1]$, als auch die Größe der abgeschnittenen Fourierreihe wiedergibt. Im Abschnitt 2.2.1 haben wir gezeigt, dass die Fehlerabschätzung für die approximierte Lösung des Exponentialoperators nicht von der Schrittweite, sondern von N abhängt. Daher mussten wir ein genügend großes N wählen, sodass der dadurch verursachte Fehler asymptotisch vernachlässigbar ist.

Die Codes für dieses Beispiel findet man unter dem folgenden GitHub Link wieder:

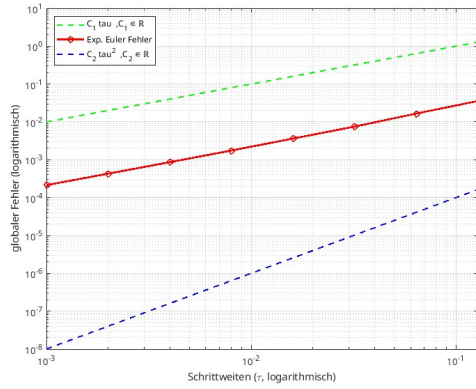


Abb. 4.1: Fehler der exp. Euler-Quadraturregel mit $N = 256$, ($L = 9$).

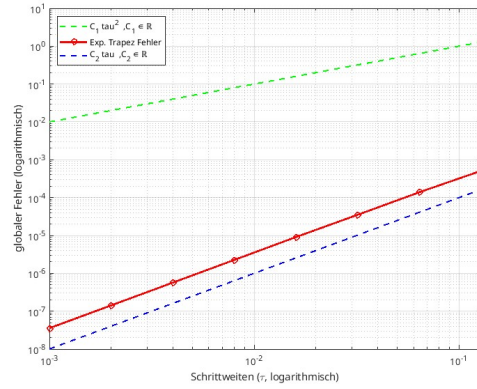


Abb. 4.2: Fehler der exp. Trapez-Quadraturregel mit $N = 256$, ($L = 9$).

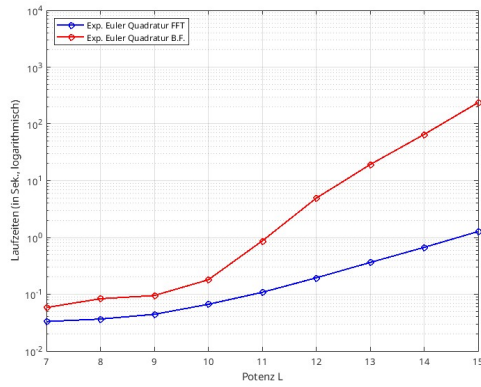


Abb. 4.3: Laufzeiten der exp. Euler-Quadraturregel mit $\tau = 0.001$, ($N = 2^L$).

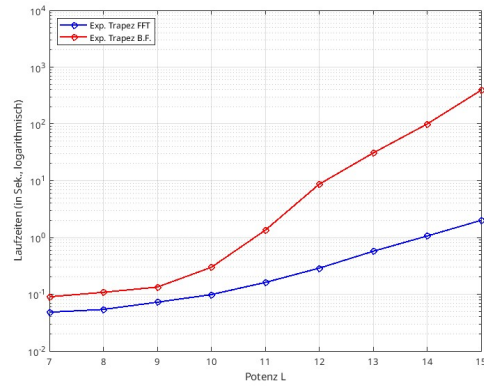


Abb. 4.4: Laufzeiten der exp. Trapez-Quadraturregel mit $\tau = 0.001$, ($N = 2^L$).

5 weitere Anwendungen

Bisher haben wir uns nur auf eine Sorte von Differentialgleichungen fokussiert und an derer effiziente Verfahren mittels dem FFT-Algorithmus hergeleitet. Sowohl das Splitting Verfahren als auch das Verfahren der Exponentiellen Runge-Kutta-Verfahren haben allerdings einen deutlich breiteren Spektrum an Anwendungsmöglichkeiten.

In diesem Kapitel wollen wir verschiedene anwendungsnahe Differentialgleichungen betrachten, die wir mittels beider Verfahren eine numerische Lösung approximieren wollen.

5.1 Erstes Beispiel

Wir können die behandelten Verfahren nicht nur auf Funktionen mit zwei Parameter, sondern auch auf ein-parametrigen Differentialgleichungen anwenden. Betrachte die folgende Gleichung

$$\begin{cases} \frac{\partial}{\partial t} x(t) &= -2x(t) + \cos(t) \quad , t > 0 \\ x(0) &= x_0 \end{cases} \quad (5.1)$$

mit einem beliebigen Startwert $x_0 \in \mathbb{R}$. Die Lösung dieser Differentialgleichung, kann man relativ leicht durch die Variation der Konstanten Formel ermitteln. Mittels beider Verfahren können wir diese sehr gut approximieren. Die Abbildungen 5.3 und 5.4 zeigen die Graphen der numerischen Lösung einmal mittels dem exponentiellen Euler-Verfahren und einmal dem sequentiellen Splitting-Verfahren für die Schrittweite $\tau = 0.1$.

Man erkennt, dass selbst für große Schrittweiten, beide Verfahren die Lösung sehr gut approximieren.

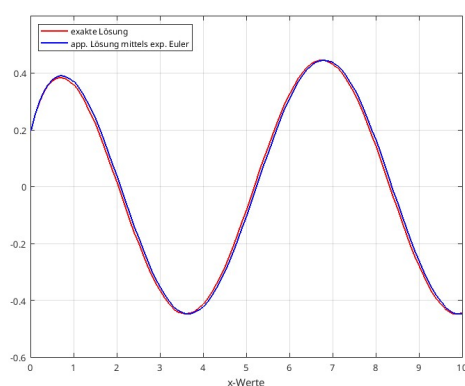


Abb. 5.1: Exp. Euler-Verfahren
mit Schrittweite $\tau = 0.1$

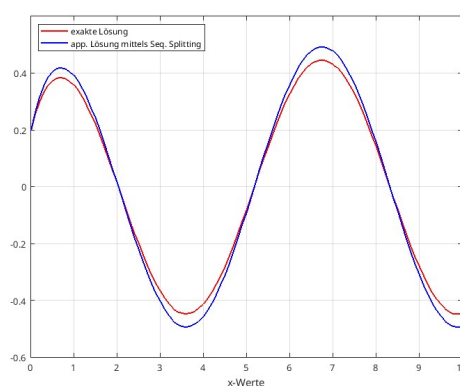


Abb. 5.2: Seq. Splitting-Verfahren
mit Schrittweite $\tau = 0.1$

5.2 Zweites Beispiel

Betrachte nun das Problem (NL) mit den Matrizen $A, B \in \text{Mat}_{n \times n}(\mathbb{R})$, $u^0 \in \mathbb{R}^n$ die nun ein lineares System gewöhnlicher Differentialgleichungen ergeben. Die Lösungen zu diesem Problem gegeben durch die Exponentialmatrizen, also:

1. $\Phi_A^{t_k, t_{k+1}} u(t_k) = e^{\tau A} u(t_k),$
2. $\Phi_B^{t_k, t_{k+1}} u(t_k) = e^{\tau B} u(t_k),$
3. $\Phi_{A+B}^{t_k, t_{k+1}} u(t_k) = e^{\tau(A+B)} u(t_k),$

wobei hier die Exponentialmatrizen auftauchen. In den Abbildungen 5.3 und 5.4 ist ein loglog-Plot, in der die Schrittweiten gegenüber dem globalen Fehler einmal mittels dem exponentiellen Euler-Verfahren und dem sequentiellen Splitting-Verfahren mit dem folgenden Beispiel dargestellt:

$$A = \begin{pmatrix} -2 & 0 \\ 0 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 2 \\ -4 & 0 \end{pmatrix} \quad \text{und} \quad u^0 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \quad (5.2)$$

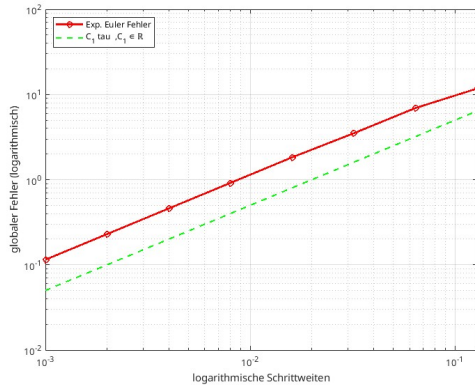


Abb. 5.3: Exp. Euler-Verfahren
mit Schrittweite $\tau = 0.01$

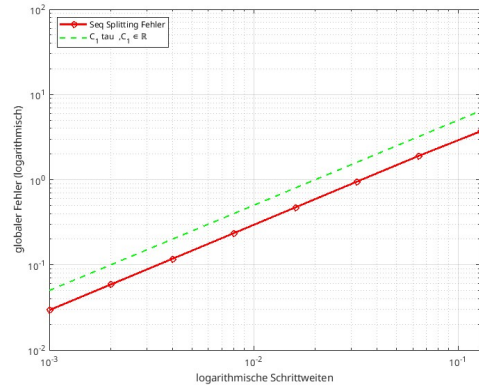


Abb. 5.4: Seq. Splitting-Verfahren
mit Schrittweite $\tau = 0.01$

6 Zusammenfassung

Wir haben uns insgesamt mit zwei verschiedene Verfahren, dem Splitting-Verfahren und dem Verfahren des exponentiellen Integratoren, für die Lösung der Differentialgleichung (NL) mit den D.-R.

$$\begin{cases} \frac{\partial}{\partial t} u(t, x) &= Au(t, x) + g(u(t, x)), & t > 0, \\ u(0, x) &= u_0(x) \end{cases} \quad (\text{NL})$$

beschäftigt. Im Hinblick auf einen effizienten Algorithmus haben wir dazu den FFT-Algorithmus hinzugenommen und uns sowohl über die Effizienz, als auch über die Konsistenzordnung anhand eines Beispiel überzeugen lassen.

Zum Abschluss vergleichen wir die beiden Verfahren anhand der numerischen Experimente. Dabei ist deutlich zu erkennen, dass der durch das Splitting Verfahren verursachte Fehler erheblich größer ist als derjenige der exponentiellen Integratoren. Das exponentielle Euler-Verfahren erzielt eine etwa zehnfach höhere Genauigkeit im Vergleich zum sequentiellen Splitting-Verfahren, während das exponentielle Trapez-Verfahren sogar um den Faktor 3000 genauer ist als das Marchuk-Strang-Splitting Verfahren.

Diese Unterschiede in der Genauigkeit werden in den Abbildungen zur Fehlerentwicklung verdeutlicht: Abbildung 3.1 zeigt den Fehler des sequentiellen Splitting-Verfahrens, während Abbildung 3.2 jenen des Marchuk-Strang-Verfahrens darstellt. Im Vergleich dazu veranschaulichen Abbildung 4.1 und Abbildung 4.2 die deutlich geringeren Fehler der exponentiellen Verfahren.

Trotz ihrer geringeren Genauigkeit, ist die Implementierung des Splitting-Verfahrens viel einfacher. Die Laufzeitanalysen in den Abbildungen 3.3 und 3.4 zeigen im Vergleich zu den exponentiellen Integratoren in 4.3 und 4.4 keinen nennenswerten Unterschied.

Insgesamt zeigen die numerischen Experimente, dass exponentielle Verfahren zwar komplexer in der Umsetzung, aber in der Genauigkeit deutlich besser sind.

Die Codes findet man unter dem folgenden GitHub Link wieder:

Literaturverzeichnis

- [BFCO12] A. Batkai, B. Farkas, P. Csomos, and A. Ostermann. *Operator Semigroups for Numerical Analysis*. Uni Wuppertal/ 15th Internet Seminar on Evolution Equations, 2011/2012.
- [DR22] W. Dahmen and A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer Berlin, Heidelberg, 2nd edition, 2022.
- [DV90] P. Duhamel and M. Vetterli. Fast fourier transforms: A tutorial review and a state of the art. *Signal Processing*, 19(4):259–299, 1990.
- [EN00] K.-J. Engel and R. Nagel. *One-Parameter Semigroups for Linear Evolution Equations*. Springer New York, 2000.
- [HO10] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.
- [HP84] J. E. Hebborn and C. Plumpton. *Trigonometric Identities*. Palgrave, London, 1984.
- [Kö04] K. Königsberger. *Analysis 1*. Springer Verlag, 2004.
- [Nar99] T. N. Narasimhan. Fourier’s Heat Conduction Equation: History, Influence, and Connections. *Reviews of Geophysics*, 37(1):151–172, 1999.
- [OE11] M. Ogihara and J. Tarui (Eds.). *A Real Elementary Approach to the Master Recurrence and Generalizations*. Springer Verlag, 2011.
- [Vre03] Anders Vretblat. *Fourier Analysis and Its Applications*, volume 1. Springer Verlag, 2003.