# Numerische Methoden für gewöhnliche Differentialgleichungen (Numerische Mathematik II)

Lars Grüne
Mathematisches Institut
Fakultät für Mathematik und Physik
Universität Bayreuth
95440 Bayreuth
lars.gruene@uni-bayreuth.de
www.math.uni-bayreuth.de/∼lgruene/

Vorlesungsskript

3. Auflage

Sommersemester 2008

# Vorwort

Dieses Skript ist im Rahmen einer gleichnamigen Vorlesung entstanden, die ich im Sommersemester 2008 an der Universität Bayreuth gehalten habe. Es ist die dritte Auflage eines Skriptes, das zuerst im Sommersemester 2003 erstellt wurde. Ich möchte mich an dieser Stelle bei all den StudentInnen bedanken, die mit zum Teil sehr ausführlichen Fehlerkorrekturen zur Verbesserung dieser dritten Auflage beigetragen haben. Neben der Verbesserungen von Fehlern wurde gegenüber der zweiten Auflage ein Abschnitt über Randwertprobleme ergänzt und der Beweis und die Diskussion von Satz 2.4 ausführlicher formuliert.

Über das Hauptthema der Vorlesung — die Numerik für gewöhnliche Differentialgleichungen — hinaus geben zwei einführende Kapitel über stochastische gewöhnliche Differentialgleichungen und die Finite Elemente Methode für partielle Differentialgleichungen erste Einblicke in weitere Themen.

Die einzelnen Kapitel des Skriptes wurden auf Basis verschiedener Lehrbücher und Monographien erstellt. Im Hauptabschnitt über gewöhnliche Differentialgleichungen wurde insbesondere das Buch von Deuflhard und Bornemann [2] verwendet, allerdings wurden sowohl in Aufbau und Notation als auch bei einer Reihe von Beweisen Änderungen vorgenommen. Der Abschnitt über stochastische Differentialgleichungen wurde auf Basis der Bücher von Kloeden und Platen [4] sowie Kloeden, Platen und Schurz [5] erstellt und im Abschnitt über die Wärmeleitungsgleichung wurden einige Passagen aus Stoffel [6] benutzt.

Eine elektronische Version dieses Skripts sowie die zu dieser Vorlesung gehörigen Übungsaufgaben finden sich im WWW unter dem Link "Lehrveranstaltungen" auf der Seite `http://www.math.uni-bayreuth.de/~lgruene/`.

Bayreuth, August 2008 Lars Grüne

# Inhaltsverzeichnis

# Kapitel 1

# Übersicht

In dieser Vorlesung werden wir uns mit numerischen Methoden zur Lösung von Differentialgleichungen beschäftigen. Dieses Teilgebiet der numerischen Mathematik ist so groß, dass wir uns nur mit einer Auswahl von Themen befassen können; speziell werden wir drei Bereiche behandeln:

Besonders ausführlich werden wir uns mit Anfangswertproblemen gewöhnlicher Differentialgleichungen beschäftigen. Dies hat mehrere Gründe: Gewöhnliche Differentialgleichungen gehören seit Jahrhunderten zum Standardwerkzeug der mathematischen Modellierung und ihre numerische Behandlung ist ein Gebiet, das zum Basiswissen eines Mathematikstudiums gehören sollte. Die Theorie der numerischen Verfahren ist dabei so ausgereift, dass sie eine umfassende und geschlossene Darstellung ermöglicht. Zudem können an diesem Gebiet grundlegende Prinzipien der numerischen Behandlung von Differentialgleichungen erklärt und eingeübt werden, die sich auch bei anderen Typen von Gleichungen wieder finden. Wir werden hier die gebräuchlichsten Verfahren (Einschrittverfahren, Mehrschrittverfahren, Extrapolationsverfahren, Schrittweitensteuerung. . . ) betrachten und auch einen ersten Einblick in das Gebiet der numerischen Dynamik geben, das im kommenden Wintersemester Thema einer eigenen Vorlesung sein wird.

Als weitere Bereiche werden wir numerische Verfahren für stochastische (gewöhnliche) Differentialgleichungen und für partielle Differentialgleichungen betrachten. Erstere bilden eine Erweiterung der im ersten Teil ausführlich behandelten gewöhnlichen Differentialgleichungen, die in vielen Anwendungen — z.B. in der Finanzmathematik — wichtig sind. Dementsprechend sind auch die Verfahren denen aus dem ersten Teil ähnlich; es gibt allerdings einige fundamentale Unterschiede, auf die wir speziell eingehen werden. Partielle Differentialgleichungen spielen in praktisch allen Anwendungsgebieten der Mathematik eine Rolle und ihre Numerik ist seit mehr als drei Jahrzehnten ein aktuelles Forschungsthema. Beide Themen werden wir nur anreißen können, ohne tief in die Details einzusteigen; in erster Linie sollen diese Kapitel Grundkenntnisse vermitteln und als Einführungen in weiterführende Seminare und Spezialvorlesungen dienen.

# Kapitel 2

# Gewöhnliche Differentialgleichungen

Im Rahmen unserer numerischen Betrachtungen werden wir die benötigten theoretischen Resultate dort einführen, wo wir sie verwenden. Bevor wir mit der Numerik beginnen können, benötigen wir aber zumindest ein theoretisches Grundgerüst mit einigen Basisdefinitionen und Resultaten zu den gewöhnlichen Differentialgleichungen, das der nun folgende Abschnitt bereit stellt.

## 2.1 Grundlagen

In diesem Abschnitt werden wir die grundlegenden Gleichungen definieren, mit denen wir uns im ersten Teil dieser Vorlesung beschäftigen wollen und einige ihrer Eigenschaften betrachten. Zudem werden wir zwei verschiedene grafische Darstellungsmöglichkeiten für die Lösungen kennen lernen. Für weitergehende Informationen über gewöhnliche Differentialgleichungen kann z.B. das einführende Lehrbuch [1] empfohlen werden.

### 2.1.1 Definition

Eine gewöhnliche Differentialgleichung setzt die Ableitung einer Funktion $x : \mathbb{R} \to \mathbb{R}^n$ nach ihrem (eindimensionalen) Argument mit der Funktion selbst in Beziehung. Formal beschreibt dies die folgende Definition.

**Definition 2.1** Ein *gewöhnliche Differentialgleichung* (DGL) im $\mathbb{R}^n$, $n \in \mathbb{N}$, ist gegeben durch die Gleichung

$$\frac{d}{dt}x(t) = f(t, x(t)), \tag{2.1}$$

wobei $f : D \to \mathbb{R}^n$ eine stetige Funktion ist und *Vektorfeld* genannt wird, deren Definitionsbereich $D$ eine offene Teilmenge von $\mathbb{R} \times \mathbb{R}^n$ ist.

Eine *Lösung* von (2.1) ist eine stetig differenzierbare Funktion $x : \mathbb{R} \to \mathbb{R}^n$, die (2.1) erfüllt. □

Einige Anmerkungen zur Notation bzw. Sprechweise:

- Die unabhängige Variable $t$ werden wir üblicherweise als Zeit interpretieren, obwohl (abhängig vom modellierten Sachverhalt) gelegentlich auch andere Interpretationen möglich sind.
- Statt $\frac{d}{dt}x(t)$ schreiben wir oft kurz $\dot{x}(t)$.
- Die Lösungsfunktion $x(t)$ nennen wir auch *Lösungskurve* oder *(Lösungs–)Trajektorie*.
- Falls das Vektorfeld $f$ nicht von $t$ abhängt, also $\dot{x}(t) = f(x(t))$ ist, nennen wir die Differentialgleichung *autonom*.

### 2.1.2 Anfangswertprobleme

Eine gewöhnliche Differentialgleichung besitzt im Allgemeinen unendlich viele Lösungen. Als Beispiel betrachte die (sehr einfache) eindimensionale DGL mit $f(x,t) = x$, also

$$\dot{x}(t) = x(t)$$

mit $x(t) \in \mathbb{R}$. Betrachte die Funktion $x(t) = Ce^t$ mit beliebigem $C \in \mathbb{R}$. Dann gilt

$$\dot{x}(t) = \frac{d}{dt}Ce^t = Ce^t = x(t).$$

Für jedes feste $C$ löst $Ce^t$ die obige DGL, es gibt also unendlich viele Lösungen.

Um *eindeutige* Lösungen zu erhalten, müssen wir eine weitere Bedingung festlegen. Dies geschieht in der folgenden Definition.

**Definition 2.2** Ein *Anfangswertproblem* für die gewöhnliche Differentialgleichung (2.1) besteht darin, zu gegebenem $t_0 \in \mathbb{R}$ und $x_0 \in \mathbb{R}^n$ eine Lösungsfunktion $x(t)$ zu finden, die (2.1) erfüllt und für die darüberhinaus die Gleichung

$$x(t_0) = x_0 \tag{2.2}$$

gilt. □

Notation und Sprechweisen:

- Für die Lösung $x(t)$, die (2.1) und (2.2) erfüllt, schreiben wir $x(t; t_0, x_0)$. Im Spezialfall $t_0 = 0$ werden wir oft kurz $x(t; x_0)$ schreiben.
- Die Zeit $t_0 \in \mathbb{R}$ bezeichnen wir als *Anfangszeit*, den Wert $x_0 \in \mathbb{R}^n$ als *Anfangswert*. Das Paar $(t_0, x_0)$ bezeichnen wir als *Anfangsbedingung*, ebenso nennen wir die Gleichung (2.2) *Anfangsbedingung*.

**Bemerkung 2.3** Eine stetig differenzierbare Funktion $x : I \to \mathbb{R}^n$ löst das Anfangswertproblem (2.1), (2.2) für ein $t_0 \in I$ und ein $x_0 \in \mathbb{R}^n$ genau dann, wenn sie für alle $t \in I$ die *Integralgleichung*

$$x(t) = x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau \tag{2.3}$$

erfüllt. Dies folgt sofort durch Integrieren von (2.1) bzgl. $t$ bzw. durch Differenzieren von (2.3) nach $t$ unter Verwendung des Hauptsatzes der Differential– und Integralrechnung. Beachte dabei, dass eine stetige Funktion $x$, die (2.3) erfüllt, „automatisch“ stetig differenzierbar ist, da aus der Stetigkeit von $x$ sofort die stetige Differenzierbarkeit der rechten Seite in (2.3) und damit wegen der Gleichhet auch für $x$ selbst folgt. □

### 2.1.3 Ein Existenz– und Eindeutigkeitssatz

Unter geeigneten Bedingungen an $f$ können wir einen Existenz– und Eindeutigkeitssatz für Anfangswertprobleme der Form (2.1), (2.2) erhalten.

**Satz 2.4** Betrachte die gewöhnliche Differentialgleichung (2.1) für ein $f : D \to \mathbb{R}^n$ mit $D \subseteq \mathbb{R} \times \mathbb{R}^n$ offen. Das Vektorfeld $f$ sei stetig, darüberhinaus sei $f$ Lipschitz–stetig im zweiten Argument im folgenden Sinne: Für jede kompakte Teilmenge $K \subset D$ existiere eine Konstante $L > 0$, so dass die Ungleichung

$$\|f(t, x) - f(t, y)\| \leq L\|x - y\|$$

gilt für alle $t \in \mathbb{R}$ und $x, y \in \mathbb{R}^n$ mit $(t, x)$, $(t, y) \in K$.

Dann gibt es für jede Anfangsbedingung $(t_0, x_0) \in D$ genau eine Lösung $x(t; t_0, x_0)$ des Anfangswertproblems (2.1), (2.2). Diese ist definiert für alle $t$ aus einem offenen *maximalen Existenzintervall* $I_{t_0,x_0} \subseteq \mathbb{R}$ mit $t_0 \in I_{t_0,x_0}$.

**Beweis: Teil 1:** Wir zeigen zunächst, dass es für jede Anfangsbedingung $(t_0, x_0) \in D$ ein abgeschlossenes Intervall $J$ um $T_0$ gibt, auf dem die Lösung existiert und eindeutig ist.

Dazu wählen wir ein beschränktes abgeschlossenes Intervall $I$ um $t_0$ und ein $\varepsilon > 0$, so dass die kompakte Umgebung $U = I \times \overline{B}_\varepsilon(x_0)$ von $(t_0, x_0)$ in $D$ liegt (dies ist möglich, da $D$ eine offene Menge ist). Da $f$ stetig ist und $U$ kompakt ist, existiert eine Konstante $M$, so dass $\|f(t, x)\| \leq M$ für alle $(t, x) \in U$ gilt. Wir wählen nun $J = [t_0 - \delta, t_0 + \delta]$ wobei $\delta > 0$ so gewählt ist, dass $J \subseteq I$ gilt und $L\delta < 1$ sowie $M\delta < \varepsilon$ erfüllt ist, wobei $L$ die Lipschitz-Konstante von $f$ für $K = U$ ist. Alle somit konstruierten Mengen sind in Abbildung 2.1 dargestellt.

Nun verwenden wir zum Beweis der Existenz und Eindeutigkeit der Lösung auf $J$ den Banachschen Fixpunktsatz auf dem Banachraum $C(J, \mathbb{R}^d)$ mit der Norm

$$\|x\|_\infty := \sup_{t \in J} \|x(t)\|.$$

Auf $C(J, \mathbb{R}^d)$ definieren wir die Abbildung

$$T : C(J, \mathbb{R}^d) \to C(J, \mathbb{R}^d), \quad T(x)(t) := x_0 + \int_{t_0}^{t} f(\tau, x(\tau))d\tau.$$

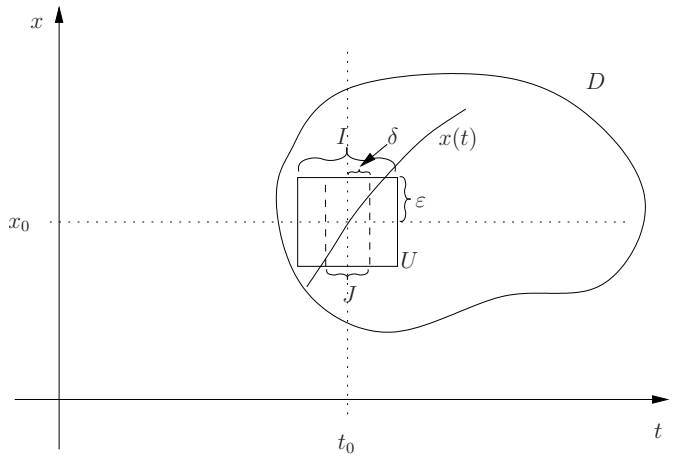



Abbildung 2.1: Mengen im Beweis von Teil 1

Beachte, dass für jedes $t \in J$ und jedes $x \in B := C(J, \overline{B}_\varepsilon(x_0))$ die Ungleichung

$$\begin{aligned} \|T(x)(t) - x_0\| &= \left\| \int_{t_0}^t f(\tau, x(\tau)) d\tau \right\| \leq \left| \int_{t_0}^t \underbrace{\|f(\tau, x(\tau))\|}_{\leq M, \text{ weil } (\tau, x(\tau)) \in \overline{U}} d\tau \right| \\ &\leq \delta M \leq \varepsilon \end{aligned}$$

gilt, weswegen $T$ die Menge $B$ in sich selbst abbildet.

Um den Banachschen Fixpunktsatz auf dieser Menge anzuwenden, müssen wir zeigen, dass $T : B \to B$ eine Kontraktion ist, also dass

$$\|T(x) - T(y)\|_\infty \leq k \|x - y\|_\infty$$

gilt für alle $x, y \in B$ und ein $k < 1$. Diese Eigenschaft folgt für $k = L\delta < 1$ aus

$$\begin{aligned} \|T(x) - T(y)\|_\infty &= \sup_{t \in J} \left\| \int_{t_0}^t f(\tau, x(\tau)) d\tau - \int_{t_0}^t f(\tau, y(\tau)) d\tau \right\| \\ &\leq \sup_{t \in J} \left| \int_{t_0}^t \underbrace{\|f(\tau, x(\tau)) - f(\tau, y(\tau))\|}_{\leq L\|x(\tau) - y(\tau)\| \leq L\|x - y\|_\infty} d\tau \right| \\ &\leq \sup_{t \in J} |t - t_0| L \|x - y\|_\infty = \delta L \|x - y\|_\infty. \end{aligned}$$

Also sind die Voraussetzungen des Banachschen Fixpunktsatzes erfüllt, weswegen $T$ einen eindeutigen Fixpunkt $x \in B$, also eine „Fixpunktfunktion“, besitzt. Da diese Fixpunktfunktion $x$ nach Konstruktion von $T$ die Integralgleichung (2.3) erfüllt, ist sie nach Bemerkung 2.3 eine stetig differenzierbare Lösung des Anfangswertproblems.

Es bleibt zu zeigen, dass diese eindeutig ist, dass also kein weiterer Fixpunkt $y \in C(J, \mathbb{R}^d)$ existiert. Aus dem Banachschen Fixpunktsatz folgt bereits, dass in $B = C(J, \overline{B}_\varepsilon(x_0))$ kein

weiterer Fixpunkt von $T$ liegt. Zum Beweis der Eindeutigkeit reicht es also zu zeigen, dass außerhalb von $B$ kein Fixpunkt $y$ liegen kann. Wir beweisen dies per Widerspruch: Angenommen, es existiert eine Fixpunktfunktion $y \notin B$ von $T$, d.h. es gilt $\|y(t) - x_0\| > \varepsilon$ für ein $t \in J$, für das wir o.B.d.A. $t > t_0$ annehmen. Dann existiert aus Stetigkeitsgründen ein $t^* \in J$ mit $\|y(t^*) - x_0\| = \varepsilon$ und $y(s) \in \overline{B}_\varepsilon(x_0)$ für $s \in [t_0, t^*]$. Damit folgt

$$\begin{aligned} \varepsilon &= \|y(t^*) - x_0\| = \left\| \int_{t_0}^{t^*} f(s, y(s)) ds \right\| \leq \int_{t_0}^{t^*} \|f(s, y(s))\| ds \\ &\leq (t^* - t_0) M < \delta M, \end{aligned}$$

was wegen $\delta M \leq \varepsilon$ ein Widerspruch ist. Daher liegt jeder mögliche Fixpunkt $y \in C(J, \mathbb{R}^d)$ von $T$ bereits in $B$, womit die Eindeutigkeit folgt.

Zusammenfassend liefert uns Teil 1 des Beweises also, dass *lokal* – also auf einem kleinen Intervall $J$ um $t_0$ – eine eindeutige Lösung $x(t) = x(t; t_0, x_0)$ existiert. Dies ist die Aussage des *Satzes von Picard-Lindelöf*[1], der in vielen Büchern als eigenständiger Satz formuliert ist.

**Teil 2:** Wir zeigen als nächstes die Eindeutigkeit der Lösung auf beliebig großen Intervallen $I$. Seien dazu $x$ und $y$ zwei auf einem Intervall $I$ definierte Lösungen des Anfangswertproblems. Wir beweisen $x(t) = y(t)$ für alle $t \in I$ per Widerspruch und nehmen dazu an, dass ein $t \in I$ existiert, in dem die beiden Lösungen nicht übereinstimmen, also $x(t) \neq y(t)$. O.b.d.A. sei $t > t_0$. Da beide Lösungen nach Teil 1 auf $J$ übereinstimmen und stetig sind, existieren $t_2 > t_1 > t_0$, so dass

$$x(t_1) = y(t_1) \quad \text{und} \quad x(t) \neq y(t) \text{ für alle } t \in (t_1, t_2) \tag{2.4}$$

gilt. Offenbar lösen beide Funktionen das Anfangswertproblem mit Anfangsbedingung $(t_1, x(t_1)) \in D$. Aus Teil 1 des Beweises folgt die Eindeutigkeit der Lösungen dieses Problems auf einem Intervall $\widetilde{J}$ um $t_1$, also

$$x(t) = y(t) \text{ für alle } t \in \widetilde{J}.$$

Da $\widetilde{J}$ als Intervall um $t_1$ einen Punkt $t$ mit $t_1 < t < t_2$ enthält, widerspricht dies (2.4), weswegen $x$ und $y$ für alle $t \in I$ übereinstimmen müssen.

**Teil 3:** Schließlich zeigen wir die Existenz des maximalen Existenzintervalls. Für $J$ aus Teil 1 definieren wir dazu

$$t^+ := \sup\{s > t_0 \,|\, \text{es existiert eine Lösung auf } J \cup [t_0, s)\}$$

sowie

$$t^- := \inf\{s < t_0 \,|\, \text{es existiert eine Lösung auf } J \cup (s, t_0]\}$$

und setzen $I_{t_0,x_0} = (t^-, t^+)$. Sowohl $t^-$ als auch $t^+$ existieren, da die Mengen, über die das Supremum bzw. Infimum genommen wird, nichtleer sind, da sie alle $s \in J$ enthalten. Per Definition von $t^+$ bzw. $t^-$ kann es keine Lösung auf einem größeren Intervall $I \supset I_{t_0,x_0}$ geben, also ist dies das maximale Existenzintervall.

□

[1] Charles Picard, französischer Mathematiker, 1856–1941
Ernst Lindelöf, finnischer Mathematiker, 1870–1946

Am Rand des maximalen Existenzintervalls $I_{t_0,x_0} = (t^-, t^+)$ hört die Lösung auf zu existieren. Ist das Intervall in einer Zeitrichtung beschränkt, so kann dies nur zwei verschiedene Ursachen haben: Entweder die Lösung divergiert, oder sie konvergiert gegen einen Randpunkt von $D$. Formal ausgedrückt:

Falls $t^+ < \infty$ ist und die Lösung $x(t; t_0, x_0)$ für $t \nearrow t^+$ gegen ein $x^+ \in \mathbb{R}^d$ konvergiert, so muss $(t^+, x^+) \notin D$ gelten. Analog gilt die Aussage für $t \searrow t^-$. Hierbei steht $t \nearrow t^+$ kurz für $t \to t^+$ und $t < t^+$ und $t \searrow t^-$ für $t \to t^-$ und $t > t^-$.

Anschaulich sind die zwei Möglichkeiten in Abbildung 2.2 dargestellt.

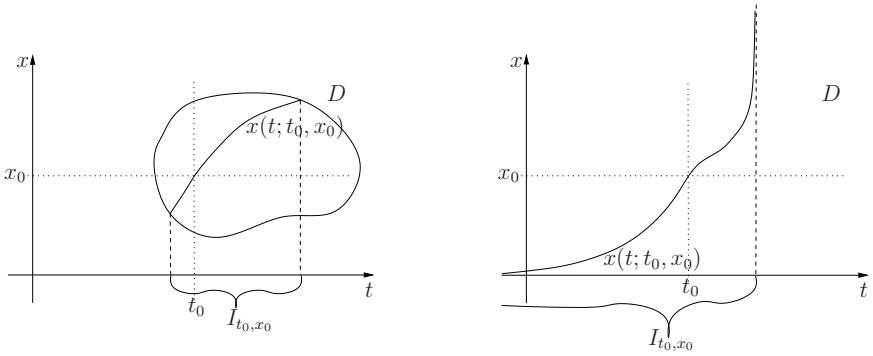



Abbildung 2.2: Lösungsverhalten am Rand des Existenzintervalls für eine beschränkte (links) und eine unbeschränkte Definitionsmenge $D$ (rechts)

Die Begründung für dieses Verhalten ist wie folgt:

Wenn $x(t; t_0, x_0)$ für $t \nearrow t^+$, gegen $x^+ \in \mathbb{R}^d$ mit $(t^+, x^+) \in D$ konvergiert, so existiert eine Lösung $x(t; t^+, x^+)$ auf einem offenen Intervall $I_{t^+,x^+}$ um $t^+$. Dann ist die zusammengesetzte Lösung

$$y(t) = \begin{cases} x(t; t_0, x_0), & t \in I_{t_0,x_0} \\ x(t; t^+, x^+), & t \in I_{t^+,x^+} \setminus I_{t_0,x_0} \end{cases}$$

stetig und erfüllt für alle $t \in I_{t_0,x_0} \cup I_{t^+,x^+}$ die Integralgleichung (2.3), damit nach Bemerkung 2.3 auch das Anfangswertproblem und ist folglich eine Lösung, die über $t^+$ hinaus definiert ist: ein Widerspruch zur Definition von $t^+$.

Im Fall $D = \mathbb{R} \times \mathbb{R}^d$ gilt daher für $t^+ < \infty$ bzw. $t^- > -\infty$ insbesondere, dass die Lösung $x(t; t_0, x_0)$ für $t \nearrow t^+$ bzw. $t \searrow t^-$ divergieren muss, da eine Konvergenz gegen $(t^+, x^+) \notin D$ bzw. $(t^-, x^-) \notin D$ nicht möglich ist. Beachte, dass dieser Fall tatsächlich auftreten kann: eine unbeschränkte Definitionsmenge $D$ von $f$ bedeutet nicht, dass auch die Lösungen auf einem unbeschränkten Intervall $I_{t_0,x_0} = \mathbb{R}$ existieren, wie das letzte der drei folgenden Beispiele zeigt.

Wir werden im Folgenden immer annehmen, dass die Annahmen von Satz 2.4 erfüllt sind, auch ohne dies explizit zu erwähnen. Auch werden wir oft Mengen der Form $[t_1, t_2] \times K$ mit $K \subset \mathbb{R}^n$ betrachten, bei denen wir — ebenfalls ohne dies immer explizit zu erwähnen — annehmen, dass alle Lösungen $x(t; t_0, x_0)$ mit $x_0 \in K$ für alle $t_0, t \in [t_1, t_2]$ existieren.

Eine einfache Konsequenz aus Satz 2.4 ist die sogenannte *Kozykluseigenschaft* der Lösungen, die für $(t_0, x_0) \in D$ und zwei Zeiten $t_1, t \in \mathbb{R}$ gegeben ist durch

$$x(t; t_0, x_0) = x(t; t_1, x(t_1; t_0, x_0)), \tag{2.5}$$

vorausgesetzt natürlich, dass alle hier auftretenden Lösungen zu den angegebenen Zeiten auch existieren. Zum Beweis rechnet man nach, dass der linke Ausdruck in (2.5) das Anfangswertproblem (2.1), (2.2) zur Anfangsbedingung $(t_1, x(t_1; t_0, x_0))$ löst. Da der rechte dies ebenfalls tut, müssen beide übereinstimmen.

Unter den Voraussetzungen von Satz 2.4 ist die Lösungsabbildung $x(t; t_0, x_0)$ zudem stetig in all ihren Variablen, also in $t$, $t_0$ und $x_0$.

### 2.1.4 Grafische Darstellung der Lösungen

Zur grafischen Darstellung von Lösungen verwenden wir zwei verschiedene Methoden, die wir hier an der zweidimensionalen DGL

$$\dot{x}(t) = \begin{pmatrix} -0.1 & 1 \\ -1 & -0.1 \end{pmatrix} x(t)$$

mit $x(t) = (x_1(t), x_2(t))^T$ und Anfangsbedingung $x(0) = (1, 1)^T$ illustrieren wollen. Da jede Lösung einer Differentialgleichung eine Funktion von $\mathbb{R}$ nach $\mathbb{R}^n$ darstellt, kann man die Graphen der einzelnen Komponenten $x_i(t)$ der Lösung in Abhängigkeit von $t$ darstellen. Für die obige DGL ist dies in Abbildung 2.3 dargestellt. Die durchgezogene Linie zeigt $x_1(t)$ während die gestrichelte Linie $x_2(t)$ darstellt.

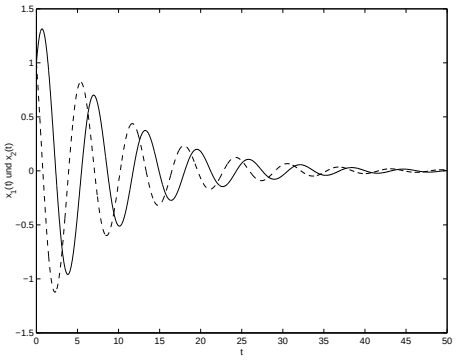



Abbildung 2.3: Darstellung von $x(t)$ mittels Graphen ($x_1(t)$ durchgezogen, $x_2(t)$ gestrichelt)

Eine alternative Darstellung, die speziell für zwei– und dreidimensionale Differentialgleichungen geeignet ist, ergibt sich, wenn man statt der Funktionsgraphen der Komponenten $x_i$ die Kurve $\{x(t) \,|\, t \in [0, T]\} \subset \mathbb{R}^n$ darstellt. Hier geht in der Grafik die Information über die Zeit (sowohl über die Anfangszeit $t_0$ als auch über die laufende Zeit $t$) verloren. Letzteres kann zumindest teilweise durch das Anbringen von Pfeilen, die die Zeitrichtung symbolisieren, ausgeglichen werden. Ein Beispiel für diese Darstellung zeigt Abbildung 2.4.

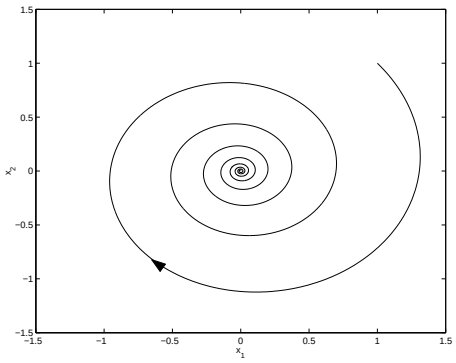



Abbildung 2.4: Darstellung von $x(t)$ als Kurve

Am Computer kann man die Darstellung als Kurve mit einer Animation verbinden, so dass man die Information über den zeitlichen Ablauf der Lösung über die Animation wieder zurück erhält. Ein MATLAB M–File, das sowohl die Abbildungen 2.3 und 2.4 sowie eine animierte Version von Abbildung 2.4 erstellt, findet sich auf der Vorlesungs–Homepage[2] unter dem Namen "darstellung.m".

Für autonome Differentialgleichungen ist der Verlust der Anfangszeit in der Grafik nicht weiter schlimm, da die Lösungen nicht wirklich von der Anfangszeit abhängen: man rechnet leicht nach, dass hier für die Anfangszeiten $t_0$ und $t_0 + t_1$ die Beziehung

$$x(t; t_0 + t_1, x_0) = x(t - t_1; t_0, x_0) \tag{2.6}$$

gilt. Die Lösung verschiebt sich also auf der $t$–Achse, verändert sich aber ansonsten nicht. Insbesondere ist die in Abbildung 2.4 dargestellte Kurve für autonome DGL für alle Anfangszeiten gleich.

[2]http://www.uni-bayreuth.de/departments/math/∼lgruene/numerik05/

## 2.2 Allgemeine Theorie der Einschrittverfahren

In diesem Abschnitt werden wir eine wichtige Klasse von Verfahren zur Lösung gewöhnlicher Differentialgleichungen einführen und analysieren, die *Einschrittverfahren*.

### 2.2.1 Diskrete Approximationen

In der Numerik gewöhnlicher Differentialgleichungen wollen wir eine Approximation an die Lösungsfunktion $x(t;t_0,x_0)$ für $t \in [t_0,T]$ berechnen (wir nehmen hier immer an, dass die Lösungen auf den angegebenen Intervallen existieren). In der folgenden Definition definieren wir die Art von Approximationen, die wir betrachten wollen und einen Begriff der Konvergenzordnung.

**Definition 2.5** (i) Eine Menge $\mathcal{T} = \{t_0, t_1, \ldots, t_N\}$ von Zeiten mit $t_0 < t_1 < \ldots < t_N = T$ heißt *Gitter* auf dem Intervall $[t_0,T]$. Die Werte

$$h_i = t_{i+1} - t_i$$

heißen *Schrittweiten*, der Wert

$$\bar{h} = \max_{i=0,\ldots,N-1} h_i$$

heißt *maximale Schrittweite*. Im Fall *äquidistanter Schrittweiten* $h_0 = h_1 = \ldots = h_{N-1}$ schreiben wir zumeist $h$ statt $h_i$.

(ii) Eine Funktion $\tilde{x} : \mathcal{T} \to \mathbb{R}^n$ heißt *Gitterfunktion*.

(iii) Eine Familie von Gitterfunktionen $\tilde{x}_j$, $j \in \mathbb{N}$, auf Gittern $\mathcal{T}_j$ auf dem Intervall $[t_0,T] \subset I_{t_0,x_0}$ mit maximalen Schrittweiten $\bar{h}_j$ heißt *(diskrete) Approximation* der Lösung $x(t;t_0,x_0)$ von (2.1), falls

$$\max_{t_i \in \mathcal{T}_j} \|\tilde{x}_j(t_i) - x(t_i;t_0,x_0)\| \to 0$$

für $\bar{h}_j \to 0$. Eine von der Anfangsbedingung $(t_0,x_0)$ abhängige Familie von Gitterfunktionen $\tilde{x}_j(\cdot;t_0,x_0)$, $j \in \mathbb{N}$, hat die *Konvergenzordnung* $p > 0$, falls für jede kompakte Menge $K \subset D$ und alle $T > 0$ mit $[t_0,T] \subset I_{t_0,x_0}$ für alle $(t_0,x_0) \in K$ ein $C > 0$ existiert, so dass

$$\max_{t_i \in \mathcal{T}_j} \|\tilde{x}_j(t_i;t_0,x_0) - x(t_i;t_0,x_0)\| \le C\bar{h}_j^p$$

gilt für alle $(t_0,x_0) \in K$ und alle hinreichend feinen Gitter $\mathcal{T}_j$ auf $[t_0,T]$. In diesem Fall schreiben wir auch $\tilde{x}_j(t_i : t_0,x_0) = x(t_i;t_0,x_0) + O(\bar{h}_j^p)$. □

**Bemerkung 2.6** Wir haben in der Numerik I verschiedene Methoden kennen gelernt, mit denen man Funktionen numerisch darstellen kann, z.B. Polynom– oder Splineinterpolation. Jede Gitterfunktion gemäß Definition 2.5 kann natürlich mit diesen Methoden zu einer "echten" Funktion erweitert werden. □

Ein Einschrittverfahren ist nun gegeben durch eine numerisch auswertbare Funktion $\Phi$, mittels derer wir eine Gitterfunktion zu einem gegebenen Gitter berechnen können. Formal ist dies wie folgt definiert.

**Definition 2.7** Ein *Einschrittverfahren* ist gegeben durch eine stetige Abbildung

$$\Phi : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n,$$

mit der zu jedem Gitter $\mathcal{T}$ und jedem Anfangswert $x_0$ mittels

$$\tilde{x}(t_0) = x_0, \;\; \tilde{x}(t_{i+1}) = \Phi(t_i, \tilde{x}(t_i), h_i) \text{ für } i = 0, 1, \ldots, N-1$$

rekursiv eine Gitterfunktion definiert werden kann.

Wenn die so erzeugten Gitterfunktionen die Bedingung aus Definition 2.5 (iii) erfüllen, so nennen wir das Einschrittverfahren *konvergent* bzw. *konvergent mit Konvergenzordnung p*.
□

Der Name *Einschrittverfahren* ergibt sich dabei aus der Tatsache, dass der Wert $\tilde{x}(t_{i+1})$ nur aus dem direkten Vorgängerwert $\tilde{x}(t_i)$ berechnet wird. Wir werden später auch *Mehrschrittverfahren* kennen lernen, bei denen $\tilde{x}(t_{i+1})$ aus $\tilde{x}(t_{i-k}), \tilde{x}(t_{i-k+1}), \ldots, \tilde{x}(t_i)$ berechnet wird.

### 2.2.2 Erste einfache Einschrittverfahren

Bevor wir in die Konvergenztheorie einsteigen und mathematisch untersuchen, welche Bedingungen $\Phi$ erfüllen muss, damit die erzeugte Gitterfunktion eine Approximation darstellt, wollen wir in diesem Abschnitt zwei Einschrittverfahren heuristisch betrachten.

Die Idee der Verfahren erschließt sich am einfachsten über die Integralgleichung (2.3). Die exakte Lösung erfüllt ja gerade

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(\tau, x(\tau)) d\tau.$$

Die Idee ist nun, das Integral durch einen Ausdruck zu ersetzen, der numerisch berechenbar ist, wenn wir $x(\tau)$ für $\tau > t_i$ nicht kennen. Die einfachste Approximation ist die Rechteck–Regel (oder Newton–Cotes Formel mit $n = 0$, die wir in der Numerik I wegen ihrer Einfachheit gar nicht betrachtet haben)

$$\int_{t_i}^{t_{i+1}} f(\tau, x(\tau)) d\tau \approx (t_{i+1} - t_i) f(t_i, x(t_i)) = h_i f(t_i, x(t_i)). \tag{2.7}$$

Setzen wir also

$$\Phi(t, x, \tau) = x + h f(t, x), \tag{2.8}$$

so gilt

$$\tilde{x}(t_{i+1}) = \Phi(t_i, \tilde{x}(t_i), h_i) = \tilde{x}(t_i) + h_i f(t_i, \tilde{x}(t_i))$$

und wenn wir $\tilde{x}(t_i) \approx x(t_i)$ annehmen, so können wir fortfahren

$$\ldots \approx x(t_i) + h_i f(t_i, x(t_i)) \approx x(t_i) + \int_{t_i}^{t_{i+1}} f(\tau, x(\tau)) d\tau.$$

Da $\tilde{x}(t_0) = x_0 = x(t_0)$ ist, kann man damit rekursiv zeigen, dass $\tilde{x}(t_{i+1})$ eine Approximation von $x(t_{i+1})$ ist. Wir werden dies im nächsten Abschnitt mathematisch präzisieren.

Das durch (2.8) gegebene Verfahren ist das einfachste Einschrittverfahren und heißt *Euler'sche Polygonzugmethode* oder einfach *Euler–Verfahren*. Es hat eine einfache geometrische Interpretation: In jedem Punkt $\tilde{x}(t_i)$ berechnen wir die Steigung der exakten Lösung durch diesen Punkt (das ist gerade $f(t_i, \tilde{x}(t_i))$) und folgen der dadurch definierten Geraden bis zum nächsten Zeitschritt. Das Prinzip ist in Abbildung 2.5 grafisch dargestellt.

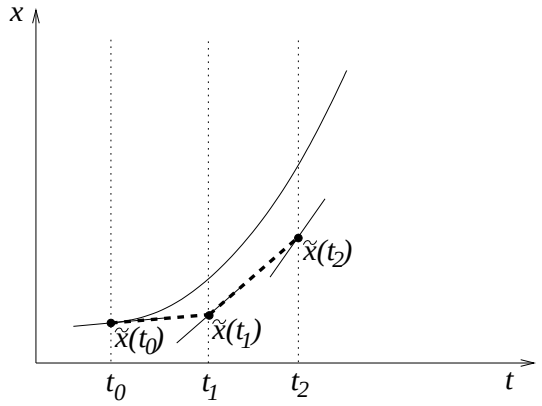



Abbildung 2.5: Grafische Veranschaulichung des Euler–Verfahrens

Das Euler–Verfahren liefert nur eine recht grobe Approximation der Lösung. Bessere Verfahren kann man erhalten, wenn man statt (2.7) eine genauere Approximation verwendet. Eine bessere Möglichkeit ist z.B.

$$\int_{t_i}^{t_{i+1}} f(\tau, x(\tau))d\tau \approx \frac{h_i}{2}\left(f(t_i, x(t_i)) + f\Big(t_{i+1}, x(t_i) + h_i f(t_i, x(t_i))\Big)\right). \tag{2.9}$$

Dies ist nichts anderes als die Trapez–Regel (oder Newton–Cotes Formel mit $n = 1$), bei der wir den unbekannten Wert $x(t_{i+1})$ durch die Euler–Approximation $x(t_{i+1}) \approx x(t_i) + h_i f(t_i, x(t_i))$ ersetzen. Das daraus resultierende Verfahren ist gegeben durch

$$\Phi(t, x, h) = x + \frac{h}{2}\left(f(t, x) + f\Big(t + h, x + hf(t, x)\Big)\right)$$

und heißt *Heun–Verfahren*. Es ist tatsächlich schon deutlich besser als das Euler–Verfahren.

Man kann sich leicht vorstellen, dass weitere bessere Verfahren sehr komplizierte Formeln benötigen. Wir werden deshalb später einen Formalismus kennen lernen, mit dem man auch sehr komplizierte Verfahren einfach aufschreiben und implementieren kann.

Ein Grundalgorithmus zur Approximation einer Lösung $x(t; t_0, x_0)$ auf $[t_0, T]$ mittels eines Einschrittverfahrens $\Phi$ lässt sich nun leicht angeben. Wir beschränken uns hierbei zunächst auf Gitter mit konstanter Schrittweite, also $h_i = h$ für alle $i = 0, 1, 2, \ldots, N$, wobei wir $N$ als Parameter vorgeben.

**Algorithmus 2.8 (Lösung eines Anfangswertproblems mit Einschrittverfahren)**

**Eingabe:** Anfangsbedingung $(t_0, x_0)$, Endzeit $T$, Schrittzahl $N$, Einschrittverfahren $\Phi$

(1) Setze $h := (T - t_0)/N$, $\tilde{x}_0 = x_0$

(2) Berechne $t_{i+1} = t_i + h$, $\tilde{x}_{i+1} := \Phi(t_i, \tilde{x}_i, h)$ für $i = 0, \ldots, N-1$.

**Ausgabe:** Werte der Gitterfunktion $\tilde{x}(t_i) = \tilde{x}_i$ in $t_0, \ldots, t_N$ □

### 2.2.3 Konvergenztheorie

Die Grundidee der Konvergenztheorie für numerische Methoden für Differentialgleichungen liegt in einem geschickten Trick, mit dem verschiedene Fehlerquellen separiert werden können. Wir schreiben hier kurz $x(t) = x(t; t_0, x_0)$. Um nun den Fehler

$$\|\tilde{x}(t_i) - x(t_i)\| = \|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), h_{i-1}) - x(t_i)\|$$

abzuschätzen, schieben wir mittels der Dreiecksungleichung die Hilfsgröße

$$\Phi(t_{i-1}, x(t_{i-1}), h_{i-1})$$

ein. Wir erhalten so mit (2.5) die Abschätzung

$$\begin{aligned} \|\tilde{x}(t_i) - x(t_i)\| &\leq \|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), h_{i-1}) - \Phi(t_{i-1}, x(t_{i-1}), h_{i-1})\| \\ &\quad + \|\Phi(t_{i-1}, x(t_{i-1}), h_{i-1}) - x(t_i)\| \\ &= \|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), h_{i-1}) - \Phi(t_{i-1}, x(t_{i-1}), h_{i-1})\| \\ &\quad + \|\Phi(t_{i-1}, x(t_{i-1}), h_{i-1}) - x(t_i; t_{i-1}, x_{i-1})\| \end{aligned}$$

Statt also direkt den Fehler zur Zeit $t_i$ abzuschätzen, betrachten wir getrennt die zwei Terme

(a) $\|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), h_{i-1}) - \Phi(t_{i-1}, x(h_{i-1}), h_{i-1})\|$, also die Auswirkung des Fehlers bis zur Zeit $t_{i-1}$ in $\Phi$

(b) $\|\Phi(t_{i-1}, x(t_{i-1}), h_{i-1}) - x(t_i; t_{i-1}, x_{i-1})\|$, also den lokalen Fehler beim Schritt von $x(t_{i-1})$ nach $x(t_i)$

Die folgende Definition gibt die benötigten Eigenschaften an $\Phi$ an, mit denen diese Fehler abgeschätzt werden können.

**Definition 2.9** (i) Ein Einschrittverfahren erfüllt die *Lipschitzbedingung* (oder *Stabilitätsbedingung*), falls für jede kompakte Menge $K \subset D$ des Definitionsbereiches der Differentialgleichung ein $L > 0$ existiert, so dass für alle Paare $(t_0, x_1), (t_0, x_2) \in K$ und alle hinreichend kleinen $h > 0$ die Abschätzung

$$\|\Phi(t_0, x_1, h) - \Phi(t_0, x_2, h)\| \leq (1 + Lh)\|x_1 - x_2\| \tag{2.10}$$

gilt.

(ii) Ein Einschrittverfahren $\Phi$ heißt *konsistent*, falls für jede kompakte Menge $K \subset D$ des Definitionsbereiches der Differentialgleichung eine Funktion $\varepsilon(h)$ mit $\lim_{h\to 0} \varepsilon(h) = 0$ existiert, so dass für alle $(t_0, x_0) \in K$ und alle hinreichend kleinen $h > 0$ die Ungleichung

$$\|\Phi(t_0, x_0, h) - x(t_0 + h; t_0, x_0)\| \leq h\varepsilon(h) \tag{2.11}$$

gilt. O.B.d.A. nehmen wir dabei an, dass $\varepsilon(h)$ monoton ist, ansonsten können wir $\varepsilon(h)$ durch $\sup_{h\in[0,h]} \varepsilon(h)$ ersetzen.
Das Verfahren hat die *Konsistenzordnung* $p > 0$, falls für jede kompakte Menge $K \subset D$ ein $E > 0$ existiert, so dass $\varepsilon(h) = Eh^p$ gewählt werden kann. In diesem Fall schreiben wir auch $\Phi(t_0, x_0, h) = x(t_0 + h; t_0, x_0) + O(h^{p+1})$. □

Offenbar garantiert (2.10), dass der Fehlerterm (a) nicht zu groß wird, während (2.11) dazu dient, den Term (b) abzuschätzen. Der formale Beweis folgt in Satz 2.11. Bevor wir diesen formulieren, wollen wir uns noch überlegen, ob die im vorherigen Abschnitt definierten Verfahren diese Bedingungen erfüllen.

Man rechnet leicht nach, dass das Euler– und das Heun–Verfahren die Lipschitzbedingung erfüllen. Die Konsistenzbedingung (2.11) ist allerdings nicht so leicht nachzuprüfen, da sie mit Hilfe der (unbekannten) Lösungen $x(t; t_0, x_0)$ formuliert ist. Das folgende Lemma stellt eine alternative und leichter nachprüfbare Formulierung der Bedingung vor.

**Lemma 2.10** Gegeben sei ein Einschrittverfahren $\Phi$ der Form

$$\Phi(t, x, h) = x + h\varphi(t, x, h)$$

mit einer stetigen Funktion $\varphi : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$. Dann ist das Verfahren genau dann konsistent, falls für alle $(t, x) \in D$ die Bedingung

$$\varphi(t, x, 0) = f(t, x) \tag{2.12}$$

gilt.

**Beweis:** Wir schreiben wieder kurz $x(t) = x(t; t_0, x_0)$. Es gilt

$$\begin{aligned}
&\frac{\Phi(t_0, x_0, h) - x(t_0 + h)}{h} \\
&= \frac{1}{h}\left(\Phi(t_0, x_0, h) - x_0 - \int_{t_0}^{t_0+h} f(\tau, x(\tau))d\tau\right) \\
&= \frac{1}{h}\left(\Phi(t_0, x_0, h) - x_0 - \int_{t_0}^{t_0+h} f(t_0, x_0)d\tau + \int_{t_0}^{t_0+h} f(t_0, x_0)d\tau - \int_{t_0}^{t_0+h} f(\tau, x(\tau))d\tau\right) \\
&= \frac{1}{h}\left(h\varphi(t_0, x_0, h) - \int_{t_0}^{t_0+h} f(t_0, x_0)d\tau\right) + \frac{1}{h}\left(\int_{t_0}^{t_0+h} f(t_0, x_0) - f(\tau, x(\tau))d\tau\right) \\
&= \varphi(t_0, x_0, h) - f(t_0, x_0) + \frac{1}{h}\left(\int_{t_0}^{t_0+h} f(t_0, x_0) - f(\tau, x(\tau))d\tau\right)
\end{aligned}$$

Sei nun $K \subset D$ gegeben. Die Funktion $f(t_0 + s, x(t_0 + s; t_0, x_0))$ ist stetig in $s$, $t_0$ und $x_0$, also gleichmäßig stetig für $(s, t_0, x_0) \in [0, h] \times K$ für hinreichend kleines $h > 0$ (so klein, dass die Lösungen $x(t_0 + s; t_0, x_0))$ für $s \in [0, h]$ existieren), da diese Menge kompakt ist. Also existiert eine Funktion $\varepsilon_1(h) \to 0$ mit

$$\|f(\tau, x(\tau)) - f(t_0, x(t_0))\| \leq \varepsilon_1(h)$$

für $\tau = t_0 + s \in [t_0, t_0 + h]$ und damit

$$\frac{1}{h} \left\| \int_{t_0}^{t_0+h} f(t_0, x_0) - f(\tau, x(\tau)) d\tau \right\| \leq \frac{1}{h} \int_{t_0}^{t_0+h} \|f(t_0, x_0) - f(\tau, x(\tau))\| d\tau \leq \varepsilon_1(h). \quad (2.13)$$

Wir nehmen nun an, dass (2.12) gilt. Ebenfalls wegen gleichmäßiger Stetigkeit und wegen (2.12) existiert eine Funktion $\varepsilon_2(h) \to 0$ mit

$$\|\varphi(t_0, x_0, h) - f(t_0, x_0)\| \leq \varepsilon_2(h).$$

Damit folgt

$$\frac{\|\Phi(t_0, x_0, h) - x(t_0 + h)\|}{h} \leq \varepsilon_2(h) + \varepsilon_1(h),$$

also (2.11) mit $\varepsilon(h) = \varepsilon_1(h) + \varepsilon_2(h)$.

Gelte umgekehrt (2.11). Sei $(x, t) \in D$ gegeben und sei $[t_1, t_2]$ und $K$ so gewählt, dass $(x, t) \in [t_1, t_2] \times K$ gilt. Wiederum mit (2.13) folgt

$$\|\varphi(t_0, x_0, h) - f(t_0, x_0)\| \leq \varepsilon(h) + \varepsilon_1(h),$$

also

$$\lim_{h \to 0} \|\varphi(t_0, x_0, h) - f(t_0, x_0)\| = 0$$

und damit (2.12) wegen der Stetigkeit von $\varphi$. □

Mit Hilfe der Bedingung (2.12) prüft man leicht nach, dass das Euler– und das Heun–Verfahren konsistent sind. Die Konsistenzordnung kann man aus (2.12) allerdings nicht ableiten, da die Abschätzung von $\varepsilon(h)$ mittels $\varepsilon_1(h)$ und $\varepsilon_2(h)$ dafür zu grob ist, denn falls $f \not\equiv 0$ ist, gilt $\varepsilon_1(h) \geq O(h)$, so dass man maximal die Konsistenzordnung $p = 1$ nachweisen könnte. Wir werden später sehen, wie man die Konsistenzordnung berechnen kann.

Wir kommen nun zu unserem ersten wichtigen Satz, der besagt, dass Lipschitzbedingung und Konsistenz tatsächlich ausreichend für die Konvergenz sind.

**Satz 2.11** Betrachte ein Einschrittverfahren $\Phi$, das die Lipschitzbedingung erfüllt und konsistent ist. Dann ist das Verfahren konvergent. Falls das Verfahren dabei die Konsistenzordnung $p$ besitzt, so besitzt es auch die Konvergenzordnung $p$.

**Beweis:** Wir müssen die Eigenschaft aus Definition 2.5(iii) nachprüfen. Sei dazu eine kompakte Menge $K \subset D$ und ein $T > 0$ mit $[t_0, T] \subset I_{t_0,x_0}$ für alle $(t_0, x_0) \in K$ gegeben. Die Menge

$$K_1 := \{(t, x(t; t_0, x_0)) \,|\, (t_0, x_0) \in K, t \in [t_0, T]\}$$

ist dann ebenfalls kompakt, da $x$ stetig in allen Variablen ist und Bilder kompakter Mengen unter stetigen Funktionen wieder kompakt sind. Wir wählen ein $\delta > 0$ und betrachten die kompakte Menge

$$K_2 := \bigcup_{(t,x)\in K_1} \{t\} \times \overline{B}_\delta(x).$$

Die Menge $K_2$ ist also genau die Menge aller Punkte $(t, x)$, deren $x$–Komponente einen Abstand $\leq \delta$ von einer Lösung $x(t; t_0, x_0)$ mit $x_0 \in K$ hat. Für hinreichend kleines $\delta > 0$ ist $K_2$ Teilmenge des Definitionsbereiches $D$ von $f$, da $D$ offen ist und $K_1 \subset D$ gilt. Das betrachtete Einschrittverfahren ist deswegen konsistent auf $K_2$ mit einer Funktion $\varepsilon(h)$, wobei $\varepsilon(h) = Eh^p$ im Falle der Konsistenzordnung $p$ ist. Ebenfalls erfüllt $\Phi$ auf $K_2$ die Lipschitzbedingung mit einer Konstanten $L > 0$.

Wir beweisen die Konvergenz nun zunächst unter der folgenden Annahme, deren Gültigkeit wir später beweisen werden:

> Für alle hinreichend feinen Gitter $\mathcal{T}$ und alle Anfangswerte $x_0 \in K$
> gilt für die gemäß Definition 2.7 erzeugte Gitterfunktion $\tilde{x}$ die Beziehung (2.14)
> $(t_i, \tilde{x}(t_i)) \in K_2$ für alle $t_i \in \mathcal{T}$.

Zum Beweis der Konvergenz wählen wir einen Anfangswert $x_0 \in K$ und schreiben wieder kurz $x(t) = x(t; t_0, x_0)$. Mit $\tilde{x}$ bezeichnen wir die zugehörige numerisch approximierende Gitterfunktion und mit

$$e(t_i) := \|\tilde{x}(t_i) - x(t_i)\|$$

bezeichnen wir den Fehler zur Zeit $t_i \in \mathcal{T}$. Dann gilt nach den Vorüberlegungen am Anfang dieses Abschnitts

$$\begin{aligned}
e(t_i) \;=\; \|\tilde{x}(t_i) - x(t_i)\| \;&\leq\; \|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), \tau_{i-1}) - \Phi(t_{i-1}, x(t_{i-1}), \tau_{i-1})\| \\
&\quad +\; \|\Phi(t_{i-1}, x(t_{i-1}), h_{i-1}) - x(t_i)\| \\
&=\; \|\Phi(t_{i-1}, \tilde{x}(t_{i-1}), h_{i-1}) - \Phi(t_{i-1}, x(t_{i-1}), h_{i-1})\| \\
&\quad +\; \|\Phi(t_{i-1}, x(t_{i-1}), h_{i-1}) - x(t_i; t_{i-1}, x(t_{i-1}))\| \\
&\leq\; (1 + Lh_{i-1})\|\tilde{x}(t_{i-1}) - x(t_{i-1})\| \;+\; h_{i-1}\varepsilon(h_{i-1}) \\
&=\; (1 + Lh_{i-1})e(t_{i-1}) + h_{i-1}\varepsilon(h_{i-1})
\end{aligned}$$

wobei wir im vorletzten Schritt die Lipschitzbedingung und die Konsistenz sowie die Tatsache, dass $(t_{i-1}, \tilde{x}(t_{i-1})) \in K_2$ liegt, ausgenutzt haben. Wir erhalten also für den Fehler $e(t_i)$ die rekursive Gleichung

$$e(t_i) \leq (1 + Lh_{i-1})e(t_{i-1}) + h_{i-1}\varepsilon(h_{i-1})$$

gemeinsam mit der "Anfangsbedingung" $e(t_0) = 0$, da $\tilde{x}(t_0) = x_0 = x(t_0)$ ist.

Mittels Induktion zeigen wir nun, dass daraus die Abschätzung

$$e(t_i) \leq \varepsilon(\bar{h})\frac{1}{L}(\exp(L(t_i - t_0)) - 1)$$

folgt. Für $i = 0$ ist die Abschätzung klar. Für $i - 1 \to i$ verwenden wir

$$\exp(Lh_i) = 1 + Lh_i + \frac{L^2 h_i^2}{2} + \ldots \geq 1 + Lh_i$$

und erhalten damit mit der Induktionsannahme

$$\begin{aligned}
e(t_i) &\leq (1 + Lh_{i-1})e(t_{i-1}) + h_{i-1}\varepsilon(h_{i-1}) \\
&\leq (1 + Lh_{i-1})\varepsilon(\bar{h})\frac{1}{L}(\exp(L(t_{i-1} - t_0)) - 1) + h_{i-1}\underbrace{\varepsilon(h_{i-1})}_{\leq \varepsilon(h)} \\
&= \varepsilon(\bar{h})\frac{1}{L}\Big(h_{i-1}L + (1 + Lh_{i-1})(\exp(L(t_{i-1} - t_0)) - 1)\Big) \\
&= \varepsilon(\bar{h})\frac{1}{L}\Big(h_{i-1}L + (1 + Lh_{i-1})\exp(L(t_{i-1} - t_0)) - 1 - Lh_{i-1}\Big) \\
&= \varepsilon(\bar{h})\frac{1}{L}\Big((1 + Lh_{i-1})\exp(L(t_{i-1} - t_0)) - 1\Big) \\
&\leq \varepsilon(\bar{h})\frac{1}{L}\Big(\exp(Lh_{i-1})\exp(L(t_{i-1} - t_0)) - 1\Big) \\
&= \varepsilon(\bar{h})\frac{1}{L}(\exp(L(t_i - t_0)) - 1).
\end{aligned}$$

Damit folgt die Konvergenz und im Falle von $\varepsilon(\bar{h}) \leq E\bar{h}^p$ auch die Konvergenzordnung mit $C = E(\exp(L(T - t_0)) - 1)/L$.

Es bleibt zu zeigen, dass unsere oben gemachte Annahme (2.14) tatsächlich erfüllt ist. Wir zeigen, dass (2.14) für alle Gitter $\mathcal{T}$ gilt, deren maximale Schrittweite $\bar{h}$ die Ungleichung

$$\varepsilon(\bar{h}) \leq \frac{\delta L}{\exp(L(T - t_0)) - 1}$$

erfüllt. Wir betrachten dazu eine Lösung $\tilde{x}$ mit Anfangswert $x_0 \in K$ und beweisen die Annahme per Induktion. Für $\tilde{x}(t_0)$ ist wegen $\tilde{x}(t_0) = x_0$ nichts zu zeigen. Für den Induktionsschritt $i - 1 \to i$ sei $(t_k, \tilde{x}(t_k)) \in K_2$ für $k = 0, 1, \ldots, i - 1$. Wir müssen zeigen, dass $(t_i, \tilde{x}(t_i)) \in K_2$ liegt. Beachte, dass die oben gezeigte Abschätzung

$$e(t_i) \leq \varepsilon(\bar{h})\frac{1}{L}(\exp(L(T - t_0)) - 1)$$

bereits gilt, falls $(t_k, \tilde{x}(t_k)) \in K_2$ liegt für $k = 0, 1, \ldots, i - 1$. Mit der Wahl von $h$ folgt damit $e(t_i) \leq \delta$, also

$$\|\tilde{x}(t_i) - x(t_i)\| \leq \delta.$$

Da $(t_i, x(t_i)) \in K_1$ liegt, folgt $(t_i, \tilde{x}(t_i)) \in \{t_i\} \times \overline{B}_\delta(x(t_i)) \subset K_2$, also die gewünschte Beziehung. □

**Bemerkung 2.12** (i) Schematisch dargestellt besagt Satz 2.11 das Folgende:

| | | |
|---|---|---|
| Lipschitzbedingung + Konsistenz | $\Rightarrow$ | Konvergenz |
| Lipschitzbedingung + Konsistenzordnung $p$ | $\Rightarrow$ | Konvergenzordnung $p$ |

(ii) Die Schranke für $e(T)$ wächst — sogar sehr schnell — wenn die Intervallgröße $T - t_0$ wächst. Insbesondere lassen sich mit dieser Abschätzung keinerlei Aussagen über das Langzeitverhalten numerischer Lösungen machen, z.B. über Grenzwerte $\tilde{x}(t_i)$ für $t_i \to \infty$. Tatsächlich kann es passieren, dass der "numerische Grenzwert" von $\tilde{x}(t_i)$ für $t_i \to \infty$ für beliebig feine Gitter $\mathcal{T}$ weit von dem tatsächlichen Grenzwert der exakten Lösung $x(t)$ entfernt ist. Wir werden später genauer auf dieses Problem eingehen.

(iii) Der Konsistenzfehler $\varepsilon(h)h$ wird auch als *lokaler Fehler* bezeichnet, während der im Beweis abgeschätzte Fehler $e(t)$ als *globaler Fehler* bezeichnet wird. Im Falle der Konsistenzordnung $p$ gilt $\varepsilon(h)h = O(h^{p+1})$ und $e(t) = O(h^p)$. Man "verliert" also eine Ordnung beim Übergang vom lokalen zum globalen Fehler. Dies lässt sich anschaulich wie folgt erklären: Bis zur Zeit $t$ muss man (bei äquidistantem Gitter) gerade ca. $N(t) = (t - t_0)/h$ Schritte machen, weswegen sich $N(t)$ lokale Fehler aufsummieren, was zu dem globalen Fehler $O(h^{p+1})N(t) = O(h^{p+1})/h = O(h^p)$ führt. □

### 2.2.4 Kondition

Wie bei allen numerischen Problemen sollte auch hier die Kondition des Problems "Berechne eine Lösung des Anfangswertproblems (2.1), (2.2)" betrachtet werden. Eine detaillierte Darstellung der hierfür nötigen Theorie würde den Rahmen dieser Vorlesung leider sprengen. Wir werden hier nur kurz (ohne Beweise) beschreiben, wie sich die Kondition bzgl. Störungen $\Delta x_0$ im Anfangswert $x_0$ berechnen lässt, d.h., wir wollen eine Abschätzung für den Ausdruck

$$\kappa := \max_{\Delta x_0 \in \mathbb{R}^n, \|\Delta x_0\|=1} \left\| \frac{\partial}{\partial x_0} x(t; t_0, x_0) \Delta x_0 \right\|$$

berechnen. Dazu betrachtet man das Anfangswertproblem

$$\dot{y}(t) = f_x(t, x(t; t_0, x_0))y(t), \quad y(t_0) = \Delta x_0, \tag{2.15}$$

wobei $f_x(t, x) = \frac{\partial}{\partial x} f(t, x) \in \mathbb{R}^{n \times n}$ und $x(t; t_0, x_0)$ die Lösung von (2.1), (2.2) ist. Die Lösung von (2.15) lässt sich in der Form

$$y(t; t_0, \Delta x_0) = W(t; t_0)\Delta x_0$$

mit einer Matrix $W(t; t_0) \in \mathbb{R}^{n \times n}$ schreiben. Dieses $W$ ist dann gerade gleich der obigen Ableitung $\frac{\partial}{\partial x_0} x(t; t_0, x_0)$, die Matrix–Norm $\|W(t; t_0)\|$ gibt also gerade die Kondition $\kappa$ an.

Als Beispiel betrachte die eindimensionale DGL

$$\dot{x}(t) = \lambda x(t)$$

für $\lambda \in \mathbb{R}$. Für diese Gleichung ist $f(t, x) = \lambda x$, also $f_x(t, x) = \lambda$, weswegen (2.15) die Form

$$\dot{y}(t) = \lambda y(t)$$

hat. Die Lösungen sind durch $y(t; t_0, \Delta x_0) = e^{\lambda(t-t_0)}\Delta x_0$ gegeben, es gilt also $W(t; t_0) = e^{\lambda(t-t_0)}$. Die Matrixnorm dieser $1 \times 1$–Matrix ist gerade der Betrag, da $e^{\lambda(t-t_0)}$ positiv ist, gilt also

$$\kappa = e^{\lambda(t-t_0)}.$$

Für $t >> t_0$ und $\lambda > 0$ ist das Problem also schlecht konditioniert ($\kappa$ wird sehr groß), während das Problem für $t >> t_0$ und $\lambda < 0$ sehr gut konditioniert ist, da $\kappa \approx 0$ ist.

Eine ausführliche Diskussion der Kondition für gewöhnliche Differentialgleichungen findet sich im Kapitel 3 des Buches [2].

## 2.3 Taylor–Verfahren

Wir werden in diesem Abschnitt eine spezielle Klasse von Einschrittverfahren einführen, die in der numerischen Praxis zwar eher selten verwendet werden (wir werden später sehen, wieso), für das Verständnis der weiteren Einschrittverfahren aber sehr nützlich sind.

Die Taylor–Verfahren haben ihren Namen von der zu Grunde liegenden Taylor–Formel und gehen in direkter Weise aus diesen hervor. Allerdings wird die Taylor–Formel in zunächst etwas ungewohnt erscheinender Weise angewendet: Wir verwenden den Differentialoperator $L_f^i$, $i \in \mathbb{N}$, der für (hinreichend oft differenzierbare) Funktionen $f$, $g : D \to \mathbb{R}^n$ mit $D \subseteq \mathbb{R} \times \mathbb{R}^n$ mittels

$$L_f^0 g(t,x) := g(t,x), \quad L_f^1 g(t,x) := \frac{\partial g}{\partial t}(t,x) + \frac{\partial g}{\partial x}(t,x) f(t,x), \quad L_f^{i+1} g(t,x) = L_f^1 L_f^i g(t,x)$$

definiert ist. Beachte, dass $L_f^i g$ wieder eine Funktion von $D$ nach $\mathbb{R}^n$ ist. Der folgende Satz stellt die hier benötigte Version der Taylor–Formel vor.

**Satz 2.13** Gegeben sei eine Differentialgleichung (2.1) mit $p$–mal stetig differenzierbarem Vektorfeld $f$. Sei $x(t) = x(t; t_0, x_0)$ eine Lösung dieser Differentialgleichung. Dann gilt

$$x(t) = x_0 + \sum_{i=1}^{p} \frac{(t-t_0)^i}{i!} L_f^{i-1} f(t_0, x_0) + O((t-t_0)^{p+1}),$$

wobei das $O$–Symbol im Sinne von Definition 2.5(iii) verwendet wird.

**Beweis:** Aus der Theorie der gewöhnlichen Differentialgleichungen ist bekannt, dass die Lösung $x(t)$ unter der vorausgesetzten Differenzierbarkeitsbedingung an $f$ $p+1$–mal stetig differenzierbar nach $t$ ist. Nach der aus der Analysis bekannten Taylor–Formel für Funktionen von $\mathbb{R}$ nach $\mathbb{R}^n$ gilt demnach

$$x(t) = x_0 + \sum_{i=1}^{p} \frac{(t-t_0)^i}{i!} \frac{d^i x}{dt^i}(t_0) + O((t-t_0)^{p+1}).$$

Zum Beweis des Satzes werden wir nun nachweisen, dass

$$\frac{d^i x}{dt^i}(t) = L_f^{i-1} f(t, x(t)) \tag{2.16}$$

ist, denn dann folgt die Behauptung aus

$$\frac{d^i x}{dt^i}(t_0) = L_f^{i-1} f(t_0, x(t_0)) = L_f^{i-1} f(t_0, x_0).$$

Wir zeigen (2.16) per Induktion über $i$. Für $i = 1$ gilt

$$\frac{dx}{dt}(t_0) = f(t_0, x(t_0)) = f(t_0, x_0) = L_f^0 f(t_0, x_0).$$

Für $i \to i+1$ beachte, dass für je zwei differenzierbare Funktionen $g : D \to \mathbb{R}^n$ und $x : \mathbb{R} \to \mathbb{R}^n$ die Gleichung

$$\frac{d}{dt} g(t, x(t)) = \frac{\partial g}{\partial t}(t, x(t)) + \frac{\partial g}{\partial x}(t, x(t)) \frac{d}{dt} x(t)$$

gilt (man nennt dies auch die *totale Ableitung* von $g$ entlang der Funktion $x(t)$). Mit $g(t, x) = L_f^{i-1} f(t, x)$ gilt damit

$$\begin{aligned} \frac{d^{i+1}x}{d^{i+1}t}(t) &= \frac{d}{dt}\frac{d^i x}{d^i t}(t) = \frac{d}{dt} L_f^{i-1} f(t, x(t)) = \frac{d}{dt} g(t, x(t)) \\ &= \frac{\partial g}{\partial t}(t, x(t)) + \frac{\partial g}{\partial x}(t, x(t)) \frac{d}{dt} x(t) \\ &= \frac{\partial g}{\partial t}(t, x(t)) + \frac{\partial g}{\partial x}(t, x(t)) f(t, x(t)) \\ &= L_f^1 g(t, x(t)) = L_f^1 L_f^{i-1} f(t, x(t)) = L_f^i f(t, x(t)), \end{aligned}$$

also gerade (2.16). □

Die Idee der Taylor–Verfahren ist nun denkbar einfach: Wir verwenden die Taylor–Formel und lassen den Restterm weg.

**Definition 2.14** Das *Taylor–Verfahren der Ordnung* $p \in \mathbb{N}$ ist gegeben durch

$$\Phi(t, x, h) = x + \sum_{i=1}^{p} \frac{h^i}{i!} L_f^{i-1} f(t, x).$$

□

Der folgende Satz gibt die wesentlichen Eigenschaften der Taylor–Verfahren an.

**Satz 2.15** Gegeben sei eine Differentialgleichung mit $p$–mal stetig differenzierbarem Vektorfeld $f : D \to \mathbb{R}^n$. Dann erfüllt das Taylor–Verfahren der Ordnung $p$ die Lipschitzbedingung und ist konsistent mit Konsistenzordnung $p$.

**Beweis:** Wir zeigen zunächst die Lipschitzbedingung. Beachte, dass in der Formulierung der Taylor–Verfahren partielle Ableitungen von $f$ bis zur Ordnung $p-1$ auftreten. Jede der auftretenden Funktionen $L_f^{i-1} f$ ist also ein weiteres mal stetig differenzierbar, woraus (mit dem Mittelwertsatz der Differentialrechnung) folgt, dass für jede kompakte Menge $K \subset D$ Lipschitz–Konstanten $L_i > 0$ existieren, so dass $L_f^{i-1} f$ Lipschitz in $x$ mit dieser Konstante ist. Für die Funktion $\Phi$ gilt also für alle $h \leq 1$ die Abschätzung

$$\begin{aligned} \|\Phi(t, x_1, h) - \Phi(t, x_2, h)\| &\leq \|x_1 - x_2\| + \sum_{i=1}^{p} \frac{h^i}{i!} L_i \|x_1 - x_2\| \\ &\leq \|x_1 - x_2\| + \sum_{i=1}^{p} h L_i \|x_1 - x_2\| = (1 + Lh)\|x_1 - x_2\| \end{aligned}$$

mit

$$L = \sum_{i=1}^{p} L_i.$$

Dies ist gerade die gewünschte Lipschitz–Bedingung.

Die Konsistenz sowie die behauptete Konsistenzordnung folgt direkt aus Satz 2.13. □

**Bemerkung 2.16** Wenn alle auftretenden Ableitungen auf ganz $D$ beschränkt sind, so sind auch die Konstanten in den Lipschitz– und Konsistenzabschätzungen unabhängig von $K$ gültig, man erhält also globale Fehlerabschätzungen. □

Beachte, dass das Taylor–Verfahren der Ordnung $p = 1$ durch

$$\Phi(t, x, h) = x + hL_f^0 f(t, x) = x + hf(t, x).$$

gegeben ist, also gerade das Euler–Verfahren ist. Dies führt sofort zu dem folgenden Korollar.

**Korollar 2.17** Falls $f$ einmal stetig differenzierbar ist, so ist das Euler–Verfahren konsistent mit Konsistenzordnung $p = 1$.

**Beweis:** Das Taylor–Verfahren der Ordnung $p = 1$ ist gerade das Euler–Verfahren, das also nach Satz 2.15 die Konsistenzordnung $p = 1$ besitzt. □

**Bemerkung 2.18** Mit einem direkten Beweis kann man die Konsistenzordnung $p = 1$ für das Euler–Verfahren auch beweisen, wenn $f$ nur Lipschitz–stetig (in $x$ *und* $t$) ist. Die Beweisidee geht wie folgt: Zunächst zeigt man, dass $\|x(t+h)-x(t)\| \leq C_1|h|$ für ein $C_1 > 0$ und alle hinreichend kleinen $h$ ist; dies verwendet man dann, um

$$\int_t^{t+h} \|f(\tau, x(\tau)) - f(t, x(t))\| d\tau \leq C_2 h^2$$

für ein $C_2 > 0$ zu beweisen. Damit kann man schließlich die Konsistenzordnung zeigen. □

Das Euler–Verfahren ist das einzige Taylor–Verfahren, bei dem keine Ableitungen des Vektorfeldes $f$ auftreten. Das Auftreten der Ableitungen ist tatsächlich der Hauptgrund dafür, dass Taylor–Verfahren in der Praxis eher selten verwendet werden, da man dort Verfahren bevorzugt, die ohne explizite Verwendung der Ableitung funktionieren (auch wenn symbolische Mathematikprogramme wie z.B. MAPLE heutzutage zur automatischen Berechnung der benötigten Ableitungen verwendet werden können). Trotzdem gibt es Spezialanwendungen, in denen Taylor–Verfahren verwendet werden: Für hochgenaue Numerik, bei der Verfahren sehr hoher Ordnung ($p \geq 15$) benötigt werden, sind Taylor–Verfahren nützlich, da sie systematisch für beliebige Konsistenzordnungen hergeleitet werden können und die auftretenden Konstanten (in der Lipschitzbedingung und der Konsistenzabschätzung) durch genaue Analyse der Ableitungen und Restterme exakt abgeschätzt werden können.

Eine der Hauptanwendungen der Taylor–Verfahren bzw. der Taylor–Entwicklung aus Satz 2.13 ist die Konsistenzanalyse beliebiger Einschrittverfahren. Hier gilt der folgende Satz.

**Satz 2.19** Sei $f : D \to \mathbb{R}^n$ $p$–mal stetig differenzierbar. Gegeben sei ein Einschrittverfahren $\Phi : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}$, das $p+1$–mal stetig differenzierbar ist. Dann besitzt $\Phi$ genau dann die Konsistenzordnung $p \in \mathbb{N}$, wenn die Bedingungen

$$\Phi(t,x,0) = x \quad \text{und} \quad \frac{\partial^i \Phi}{\partial h^i}(t,x,0) = L_f^{i-1} f(t,x) \text{ für } i = 1,\dots,p \tag{2.17}$$

für alle $(t,x) \in D$ gelten.

**Beweis:** Es bezeichne $\Phi_{T,p}$ das Taylor-Verfahren der Ordnung $p$. Die Taylor–Entwicklung von $\Phi$ nach der Variablen $h$ in $h = 0$ ist gegeben durch

$$\Phi(t,x,h) = \Phi(t,x,0) + \sum_{i=1}^{p} \frac{h^i}{i!} \frac{\partial^i \Phi}{\partial h^i}(t,x,0) + O(h^{p+1}).$$

Sei nun (2.17) erfüllt. Dann liefert der Koeffizientenvergleich mit $\Phi_{T,p}$

$$\Phi(t,x,h) = \Phi_{T,p}(t,x,h) + O(h^{p+1})$$

Aus Satz (2.15) folgt daher

$$x(t+h;t,x) = \Phi_{T,p}(t,x,h) + O(h^{p+1}) = \Phi(t,x,h) + O(h^{p+1}),$$

was die Konsistenz zeigt.

Falls (2.17) nicht erfüllt ist, so gibt es $(t,x) \in D$, so dass entweder $\Phi(t,x,0) \neq x$ gilt (in diesem Fall setzen wir $i^* = 0$) oder

$$\frac{\partial^{i^*} \Phi}{\partial h^{i^*}}(t,x,0) \neq L_g^{i^*-1} f(t,x)$$

für ein $i^* \in \{1,\dots,p\}$ gilt. Wenn wir $i^*$ minimal mit dieser Eigenschaft wählen, so folgt aus dem Koeffizientenvergleich mit $\Phi_{T,p}$, dass ein $C > 0$ existiert, so dass für alle hinreichend kleinen $h > 0$ die Ungleichung

$$\|\Phi(t,x,h) - \Phi_{T,p}(t,x,h)\| > C h^{i^*}$$

gilt. Mit Satz (2.15) erhalten wir daher

$$\|x(t+h,t,x) - \Phi(t,x,h)\| > C h^{i^*} - O(h^{p+1}) > \widetilde{C} h^{i^*}$$

für geeignetes $0 < \widetilde{C} < C$ und alle hinreichend kleinen $h > 0$, was der Konsistenz widerspricht. Also folgt die behauptete Äquivalenz. □

Mit diesem Satz können wir die Konsistenzordnung beliebiger Einschrittverfahren überprüfen. Beachte, dass die Aussage über die Ordnung nur stimmt, wenn das Vektorfeld $f$ hinreichend oft differenzierbar ist. Verfahren mit hoher Konsistenzordnung verlieren diese typischerweise, wenn das Vektorfeld der zu lösendenden DGL nicht die nötige Differenzierbarkeit besitzt!

Ein wesentlicher Nachteil dieses Satzes ist, dass die Ausdrücke $L_f^i f(t,x)$ für große $i$ sehr umfangreich und kompliziert werden. Hier können — wie bereits erwähnt — symbolische Mathematikprogramme wie MAPLE bei den Rechnungen helfen. Das folgende MAPLE Programm berechnet die Ableitungen $L_f^i f(t,x)$ für $i = 0,\dots,p$. (Vor der Ausführung muss der Variablen `p` natürlich ein Wert zugewiesen werden.)

```
> L[0]:=f(t,x);
> for i from 1 to p do
>   L[i] := simplify(diff(L[i-1],t) + diff(L[i-1],x)*f(t,x));
> od;
```

Die Ausgabe für `p:=3` ist

$$
\begin{aligned}
L_0 &:= \mathrm{f}(t,\,x)\\
L_1 &:= (\frac{\partial}{\partial t}\,\mathrm{f}(t,\,x)) + (\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)\\
L_2 &:= (\frac{\partial^2}{\partial t^2}\,\mathrm{f}(t,\,x)) + 2\,(\frac{\partial^2}{\partial x\,\partial t}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x) + (\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))\,(\frac{\partial}{\partial t}\,\mathrm{f}(t,\,x))\\
&\quad + (\frac{\partial^2}{\partial x^2}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)^2 + \mathrm{f}(t,\,x)\,(\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))^2\\
L_3 &:= (\frac{\partial^3}{\partial t^3}\,\mathrm{f}(t,\,x)) + 3\,(\frac{\partial^3}{\partial x\,\partial t^2}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x) + 3\,(\frac{\partial^2}{\partial x\,\partial t}\,\mathrm{f}(t,\,x))\,(\frac{\partial}{\partial t}\,\mathrm{f}(t,\,x))\\
&\quad + (\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))\,(\frac{\partial^2}{\partial t^2}\,\mathrm{f}(t,\,x)) + 3\,(\frac{\partial^3}{\partial x^2\,\partial t}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)^2\\
&\quad + 3\,(\frac{\partial^2}{\partial x^2}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)\,(\frac{\partial}{\partial t}\,\mathrm{f}(t,\,x)) + (\frac{\partial}{\partial t}\,\mathrm{f}(t,\,x))\,(\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))^2\\
&\quad + 5\,\mathrm{f}(t,\,x)\,(\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))\,(\frac{\partial^2}{\partial x\,\partial t}\,\mathrm{f}(t,\,x)) + (\frac{\partial^3}{\partial x^3}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)^3\\
&\quad + 4\,(\frac{\partial^2}{\partial x^2}\,\mathrm{f}(t,\,x))\,\mathrm{f}(t,\,x)^2\,(\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x)) + \mathrm{f}(t,\,x)\,(\frac{\partial}{\partial x}\,\mathrm{f}(t,\,x))^3
\end{aligned}
$$

Diese Ausdrücke gelten für den skalaren Fall $x \in \mathbb{R}$, für höhere Dimensionen muss das MAPLE–Programm erweitert werden.

**Bemerkung 2.20** Man sieht, dass die Ausdrücke tatsächlich sehr unübersichtlich werden; ebenso ist das natürlich bei den entsprechenden Termen der Einschrittverfahren. Eine Hilfe hierfür bietet ein Formalismus, der von dem neuseeländischen Mathematiker J.C. Butcher in den 1960er Jahren entwickelt wurde, und bei dem die auftretenden Ableitungen mittels einer grafischen Repräsentierung in einer Baumstruktur übersichtlich strukturiert werden.

□

## 2.4 Explizite Runge–Kutta–Verfahren

In diesem Abschnitt kommen wir zu einer der wichtigsten Klassen von Einschrittverfahren, zu denen z.B. das Euler– und das Heun–Verfahren gehören.

Bei der Konstruktion des Heun–Verfahrens haben wir das Euler–Verfahren verwendet, um einen Schätzwert für den unbekannten Wert $x(t_{i+1})$ zu erhalten. Es liegt nun nahe, diese Methode systematisch rekursiv anzuwenden, um zu Verfahren höherer Konsistenzordnung zu gelangen. Genau dies ist die Grundidee der Runge–Kutta–Verfahren.

Um die dabei entstehenden Verfahren übersichtlich zu schreiben, benötigen wir einen geeigneten Formalismus. Wir erläutern diesen am Beispiel des Heun–Verfahrens

$$\Phi(t,x,h) = x + \frac{h}{2}\Big(f(t,x) + f\Big(t+h, x+hf(t,x)\Big)\Big).$$

Wir schreiben dieses nun als

$$\begin{aligned} k_1 &= f(t,x) \\ k_2 &= f(t+h, x+hk_1) \\ \Phi(t,x,h) &= x + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right) \end{aligned}$$

Was zunächst vielleicht komplizierter als die geschlossene Formel aussieht, erweist sich als sehr günstige Schreibweise, wenn man weitere $k_i$–Terme hinzufügen will. Dies ist gerade die Schreibweise der expliziten Runge–Kutta–Verfahren.

**Definition 2.21** Ein *$s$–stufiges explizites Runge–Kutta–Verfahren* ist gegeben durch

$$\begin{aligned} k_i &= f\left(t + c_i h,\, x + h\sum_{j=1}^{i-1} a_{ij}k_j\right) \quad \text{für } i = 1,\dots,s \\ \Phi(t,x,h) &= x + h\sum_{i=1}^{s} b_i k_i. \end{aligned}$$

Den Wert $k_i = k_i(t,x,h)$ bezeichnen wir dabei als $i$–te Stufe des Verfahrens. □

Die Koeffizienten eines Runge–Kutta–Verfahrens können wir mittels

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_s \end{pmatrix} \in \mathbb{R}^s, \quad c = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_s \end{pmatrix} \in \mathbb{R}^s, \quad \mathcal{A} = \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{s1} & \cdots & \cdots & a_{s,s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s\times s}$$

kompakt schreiben. Konkrete Verfahren werden meist in Form des Butcher–Tableaus (oder Butcher–Schemas)

$$\begin{array}{c|cccccc} c_1 & & & & & & \\ c_2 & a_{2\,1} & & & & & \\ c_3 & a_{3\,1} & a_{3\,2} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ c_s & a_{s\,1} & a_{s\,2} & \cdots & a_{s\,s-1} & & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

geschrieben, das wiederum auf J.C. Butcher zurückgeht.

Einfache Beispiele solcher Verfahren sind das Euler–Verfahren ($s = 1$), das Heun–Verfahren ($s = 2$) und das sogenannte *klassische Runge–Kutta–Verfahren* ($s = 4$), das von C. Runge[3] und M. Kutta[4] entwickelt wurde, und dem die ganze Verfahrensklasse ihren Namen verdankt. Diese Verfahren sind (von links nach rechts) gegeben durch die Butcher–Tableaus

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

Beachte, dass das Euler–Verfahren sowohl das einfachste Runge–Kutta–Verfahren als auch das einfachste Taylor–Verfahren ist; es ist das einzige Verfahren, das in beiden Klassen liegt, da alle Runge–Kutta–Verfahren per Definition ohne Ableitungen von $f$ auskommen, was gegenüber den Taylor–Verfahren einen großen Vorteil darstellt.

Es ist in diesem Zusammenhang interessant, den Aufwand des Heun–Verfahrens und des Taylor–Verfahrens der Ordnung 2 z.B. für $x \in \mathbb{R}$ zu vergleichen, die ja die gleiche Konsistenzordnung besitzen. Beim Taylor–Verfahren der Ordnung 2 müssen in jedem Schritt $L_f^0 f(t,x) = f(t,x)$ und

$$L_f^1 f(t,x) = \frac{\partial f}{\partial t}(t,x) + \frac{\partial f}{\partial x}(t,x) f(t,x)$$

ausgewertet werden, also 3 Funktionsauswertungen; beim Heun Verfahren müssen $k_1 = f(t,x)$ und $f(t+h, x+hk_1)$, also 2 Funktionen ausgewertet werden. Der Aufwand ist folglich nur 2/3 so groß. Dieser geringere Aufwand, der bei höherer Konsistenzordnung noch deutlicher ausfällt, ist typisch für Runge–Kutta–Verfahren, ein weiterer Vorteil gegenüber den Taylor–Verfahren.

Beachte, dass Runge–Kutta–Verfahren immer die Lipschitz–Bedingung erfüllen, wenn das Vektorfeld $f$ Lipschitz–stetig im Sinne des Eindeutigkeitssatzes 2.4 ist: Mittels Induktion

[3] deutscher Mathematiker, 1856–1927
[4] deutscher Mathematiker und Ingenieur, 1867–1944

sieht man leicht, dass jede Stufe $k_i$ Lipschitz–stetig ist. Damit gilt dies auch für ihre Summe, weswegen $\Phi$ die gewünschte Bedingung erfüllt.

Wir wollen nun untersuchen, wie sich die Konsistenzeigenschaften der Runge–Kutta–Verfahren über ihre Koeffizienten auszudrücken lassen. Das erste wichtige Resultat ist das folgende Lemma.

**Lemma 2.22** Ein explizites Runge–Kutta–Verfahren ist genau dann konsistent, wenn die Bedingung

$$\sum_{i=1}^{s} b_i = 1$$

erfüllt ist.

**Beweis:** Beachte, dass ein Runge–Kutta–Verfahren von der Form

$$\Phi(t, x, h) = x + h\varphi(t, x, h)$$

mit

$$\varphi(t, x, h) = \sum_{i=1}^{s} b_i k_i(t, x, h)$$

ist. Nach Lemma 2.10 ist das Verfahren also genau dann konsistent, wenn

$$\varphi(t, x, 0) = \sum_{i=1}^{s} b_i k_i(t, x, 0) = f(t, x)$$

ist. Aus Definition 2.21 folgt sofort, dass $k_i(t, x, 0) = f(t, x)$, also ist das Verfahren genau dann konsistent, falls $\sum_{i=1}^{s} b_i f(t, x) = f(t, x)$, was für beliebige $f$ dann und nur dann der Fall ist, wenn $\sum_{i=1}^{s} b_i = 1$ ist. □

Etwas schwieriger wird die Sache, wenn wir Aussagen über die Konsistenzordnung machen wollen. Zunächst wollen wir eine obere Schranke für die Konsistenz beweisen.

**Lemma 2.23** Für ein $s$–stufiges explizites Runge–Kutta–Verfahren $\Phi$ mit Konsistenzordnung $p$ gilt die Ungleichung $p \leq s$, d.h. die Konsistenzordnung ist maximal so groß wie die Stufenzahl.

**Beweis:** Wir wenden das Verfahren auf das Anfangswertproblem

$$\dot{x}(t) = x(t), \;\; x(0) = 1$$

an. Für die exakte Lösung gilt hier

$$x(h; 0, 1) = e^h = 1 + h + \frac{h^2}{2!} + \cdots + \frac{h^s}{s!} + \frac{h^{s+1}}{(s+1)!} + O(h^{s+2}).$$

Andererseits sieht man durch Induktion über $i$, dass $k_i(0, 1, \cdot) \in \mathcal{P}_{i-1}$ ist, also ein Polynom vom Grad $\leq i - 1$ in $h$ ist. Also ist $\Phi(0, 1, \cdot) \in \mathcal{P}_s$, weswegen in $\Phi(0, 1, h)$ kein Term der

Form $ah^{s+1}$ auftreten kann. Daher gilt für alle hinreichend kleinen $h$ und eine geeignete Konstante $C > 0$ die Abschätzung

$$\|x(h;0,1) - \Phi(0,1,h)\| \geq \frac{h^{s+1}}{(s+1)!} - O(h^{s+2}) \geq Ch^{s+1},$$

weswegen die Konsistenzordnung maximal $s$ sein kann, also $p \leq s$ gilt. □

Um nun genauere Aussagen über die Konsistenzordnung zu machen, empfiehlt es sich, die zu betrachtenden Differentialgleichungen etwas zu vereinfachen: Wir wollen uns auf autonome DGL einschränken. Damit wir trotzdem Aussagen für allgemeine Probleme erhalten können, überlegen wir uns zuerst, dass dies keine echte Einschränkung ist. Tatsächlich kann man aus jeder Differentialgleichung

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0 \tag{2.18}$$

mittels

$$y = \begin{pmatrix} x \\ s \end{pmatrix}, \quad \hat{f}(y) = \begin{pmatrix} f(s,x) \\ 1 \end{pmatrix}$$

(mit $s \in \mathbb{R}$) eine *autonome* Differentialgleichung

$$\dot{y}(t) = \hat{f}(y(t)), \quad y(t_0) = y_0 = \begin{pmatrix} x_0 \\ t_0 \end{pmatrix} \tag{2.19}$$

machen, für deren Lösungen die Beziehung

$$y(t; t_0, y_0) = \begin{pmatrix} x(t; t_0, x_0) \\ t \end{pmatrix} \tag{2.20}$$

gilt. Die ursprüngliche Lösung $x(t; t_0, x_0)$ von (2.18) findet sich also gerade in den ersten $n$ Komponenten der $n+1$–dimensionalen Lösung $y(t; t_0, y_0)$ der autonomen Gleichung (2.19) wieder. Mit anderen Worten kann jede DGL im $\mathbb{R}^n$ in eine autonome DGL im $\mathbb{R}^{n+1}$ umgewandelt werden, dieses Verfahren nennt man *Autonomisierung*. Beachte, dass die neue DGL die Bedingungen des Eindeutigkeitssatzes nur dann erfüllt, wenn $f$ Lipschitz–stetig bezüglich $x$ *und* $t$ ist, was eine stärkere Forderung als die Lipschitz–Stetigkeit bzgl. $x$ ist. Da wir diese Bedingung für unsere numerischen Aussagen aber sowieso immer benötigen (meist nehmen wir ja sogar Differenzierbarkeit von $f$ bzgl. $x$ und $t$ an), stellt diese Annahme für unsere numerischen Untersuchungen keine Einschränkung dar.

Wir betrachten nun die von einem Runge–Kutta–Verfahren $\Phi$ erzeugten approximativen Lösungen $\tilde{x}(t_i)$ und $\tilde{y}(t_i)$ der Gleichungen (2.18) und (2.19). Unser Ziel ist es, uns bei der folgenden Konsistenzordnungsanalyse auf autonome Gleichungen einzuschränken. Damit wir dabei trotzdem Resultate für allgemeine nichtautonome Gleichungen erhalten können, also die für (2.19) gültigen Resultate auf (2.18) übertragen können, muss hier die zu (2.20) analoge Beziehung

$$\tilde{y}(t_i) = \begin{pmatrix} \tilde{x}(t_i) \\ t_i \end{pmatrix} \tag{2.21}$$

gelten. Ein Runge–Kutta–Verfahren, das (2.21) erfüllt, wird *invariant unter Autonomisierung* genannt. Nicht jedes Runge–Kutta–Verfahren ist aber invariant unter Autonomisierung. Das folgende Lemma gibt die dafür notwendige und hinreichende Bedingung an.

**Lemma 2.24** Ein explizites Runge–Kutta–Verfahren ist genau dann invarant unter Autonomisierung, wenn es konsistent ist und die Bedingung

$$c_i = \sum_{j=1}^{i-1} a_{ij}$$

für $i = 1, \ldots, s$ erfüllt ist.

**Beweis:** Wir bezeichnen das Verfahren für (2.18) mit $\Phi$ und das Verfahren für (2.19) mit $\widehat{\Phi}$, die zugehörigen Stufen bezeichnen wir mit $k_i$ und $\hat{K}_i = (\hat{k}_i, \theta_i)^T$. Das Verfahren ist genau dann invariant unter Autonomisierung, wenn

$$\widehat{\Phi}(t, x, h) = \begin{pmatrix} \Phi(t, x, h) \\ t + h \end{pmatrix} \tag{2.22}$$

gilt, da sich (2.21) dann mittels Induktion über $i$ ergibt. Wegen

$$\widehat{\Phi}(t, x, h) = \begin{pmatrix} x + h \sum_{i=1}^{s} b_i \hat{k}_i \\ t + h \sum_{i=1}^{s} b_i \theta_i \end{pmatrix} \quad \text{und} \quad \Phi(t, x, h) = x + h \sum_{i=1}^{s} b_i k_i$$

gilt (2.22) genau dann, wenn

$$\hat{k}_i = k_i \quad \text{und} \quad t + h \sum_{i=1}^{s} b_i \theta_i = t + h \tag{2.23}$$

erfüllt ist. Für $\hat{k}_i$ und $\theta_i$ gilt gerade

$$\begin{pmatrix} \hat{k}_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} f\left(t + h \sum_{j=1}^{i-1} a_{ij} \theta_j,\, x + h \sum_{j=1}^{i-1} a_{ij} \hat{k}_j\right) \\ 1 \end{pmatrix}.$$

Wegen $k_i = f\left(t + c_i h,\, x + h \sum_{j=1}^{i-1} a_{ij} k_j\right)$ und $\theta_j = 1$ ergibt sich, dass die erste Gleichung in (2.23) genau dann gilt, wenn $c_i = \sum_{j=1}^{i-1} a_{ij}$ gilt. Wegen $\theta_i = 1$ gilt die zweite Gleichung in (2.23) genau dann, wenn $t + h \sum_{i=1}^{s} b_i = t + h$ erfüllt ist, also wenn $\sum_{i=1}^{s} b_i = 1$ ist, was gerade äquivalent zur Konsistenz ist. □

Auf Basis dieses Lemmas können wir uns also im Folgenden auf autonome DGL einschränken, wenn wir Verfahren betrachten, die die Bedingung von Lemma 2.24 erfüllen. Dies hat den Vorteil, dass sich der Differentialoperator $L_f^1$ zu

$$L_f^1 g(x) := \left(\frac{d}{dx} g(x)\right) f(x)$$

vereinfacht, was die Taylorentwicklung deutlich übersichtlicher macht. Dies wird im folgenden Satz ausgenutzt.

**Satz 2.25** Betrachte ein Runge–Kutta–Verfahren, das die Bedingung aus Lemma 2.24 erfüllt. Dann gilt:

(i) Das Verfahren besitzt genau dann die Konsistenzordnung $p = 1$, wenn die Gleichung

$$\sum_i b_i = 1$$

gilt.

(ii) Es besitzt genau dann die Konsistenzordnung $p = 2$, wenn zusätzlich zu (i) die Gleichung

$$\sum_i b_i c_i = 1/2$$

gilt.

(iii) Es besitzt genau dann die Konsistenzordnung $p = 3$, wenn zusätzlich zu (i), (ii) die Gleichungen

$$\sum_i b_i c_i^2 = 1/3, \quad \sum_{ij} b_i a_{ij} c_j = 1/6$$

gelten.

(iv) Es besitzt genau dann die Konsistenzordnung $p = 4$, wenn zusätzlich zu (i)–(iii) die Gleichungen

$$\sum_i b_i c_i^3 = 1/4, \qquad \sum_{ij} b_i a_{ij} c_i c_j = 1/8$$

$$\sum_{ij} b_i a_{ij} c_j^2 = 1/12, \qquad \sum_{ijk} b_i a_{ij} a_{jk} c_k = 1/24$$

gelten.

Hierbei laufen die Summations–Indizes in den Grenzen $i = 1, \ldots, s$, $j = 1, \ldots, i-1$ und $k = 1, \ldots, j-1$.

**Beweis:** Die Gleichung ergeben sich aus der Bedingung (2.17), wobei die für $p \in \mathbb{N}$ angegebenen Gleichungen gerade äquivalent zu der Bedingung

$$\frac{\partial^p \Phi}{\partial h^p}(x, 0) = L_f^{p-1} f(x) \tag{2.24}$$

aus (2.17) sind. Für $p = 1$ ergibt sich die angegebene Gleichung dabei aus den gleichen Rechnungen wie im Beweis von Lemma 2.22.

Wir zeigen die Behauptung hier exemplarisch für $p = 2$, die höheren Ordnungen folgen mit der gleichen Beweistechnik, allerdings mit aufwändigeren Rechnungen.

Wir zeigen also, dass die in (ii) angegebene Gleichung äquivalent zu (2.24) für $p = 2$ ist. Die zweite Ableitung von $\Phi = x + h\varphi$ nach $h$ ist gerade

$$\begin{aligned} \frac{\partial^2 \Phi}{\partial h^2} &= \frac{\partial}{\partial h}\frac{\partial}{\partial h}(x + h\varphi) = \frac{\partial}{\partial h}\left(\varphi + h\frac{\partial}{\partial h}\varphi\right) \\ &= \frac{\partial}{\partial h}\varphi + \frac{\partial}{\partial h}\varphi + h\frac{\partial^2}{\partial h^2}\varphi = 2\frac{\partial}{\partial h}\varphi + h\frac{\partial^2}{\partial h^2}\varphi \end{aligned}$$

In $h = 0$ ergibt sich damit

$$\frac{\partial^2 \Phi}{\partial h^2}(x,0) = 2\frac{\partial}{\partial h}\varphi(x,0) = 2\sum_{i=1}^{s} b_i \sum_{j=1}^{i-1} a_{ij} \left(\frac{d}{dx} f(x)\right) f(x).$$

Andererseits ist die Ableitung $L_f^1 f(x)$ gerade durch

$$L_f^1 f(x) = \left(\frac{d}{dx} f(x)\right) f(x)$$

gegeben ist. Damit diese Ausdrücke für alle $f(x)$ übereinstimmen, muss also gerade

$$2\sum_{i=1}^{s} b_i \sum_{j=1}^{i-1} a_{ij} = 1$$

gelten, was wegen der angenommen Autonomieinvarianzbedingung

$$c_i = \sum_{j=1}^{i-1} a_{ij}$$

genau dann der Fall ist, wenn die Gleichung aus (ii) erfüllt ist. □

Diese Gleichungen an die Koeffizienten werden *Bedingungsgleichungen* genannt. Wie komplex das Problem des Aufstellens der Bedingungsgleichungen für große $p$ wird, zeigt die folgende Tabelle, die die Anzahl der Gleichungen für gegebenes $p$ angibt.

| Konsistenzordnung $p$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Anzahl Bedingungsgl'en | 1 | 2 | 4 | 8 | 17 | 37 | 85 | 200 | 486 | 1205 | 20247374 |

Nicht nur das Aufstellen, auch das Lösen dieser (nichtlinearen!) Gleichungssysteme wird ziemlich komplex. Hier kommt wieder das in Bemerkung 2.20 bereits erwähnte grafische Verfahren von Butcher ins Spiel. Mit diesem Verfahren können die einzelnen Terme der $L_f^i f$–Ableitungen ebenso wie die Terme der Ableitungen von $\Phi$ mittels einer Baumstruktur grafisch dargestellt werden. Dieses Verfahren erlaubt eine Einsicht in die Struktur dieser riesigen nichtlinearen Gleichungssysteme, womit es gelungen ist, die Gleichungen bis $p = 10$ (ohne Computerhilfe) zu lösen. Eine wichtige Rolle spielt dabei natürlich die Stufenzahl $s$ der betrachtetem Verfahren. Insbesondere ist hierbei wichtig, wie viele Stufen $s$ man zur Realisierung einer gegebenen Konsistenzordnung $p$ benötigt. Die folgende Tabelle gibt die ebenfalls durch Butcher (in den Jahren 1964–1985) berechneten bekannten minimalen Schranken an.

| Konsistenzordnung $p$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | $\geq 9$ |
|---|---|---|---|---|---|---|---|---|---|
| minimale Stufenzahl $s$ | 1 | 2 | 3 | 4 | 6 | 7 | 9 | 11 | $\geq p+3$ |

Der Eintrag für $p \geq 9$ bedeutet nicht, dass für jedes $p \geq 9$ ein Verfahren mit $s = p + 3$ Stufen bekannt ist, sondern dass es kein Verfahren mit weniger Stufen geben kann. Für $p = 10$ wurde 1978 von E. Hairer ein Verfahren mit $s = 17$ Stufen angegeben, das sich im Guinness–Buch der Rekorde findet. Möglichst wenig Stufen zu verwenden ist allerdings nicht das einzige Qualitätsmerkmal für Runge–Kutta–Verfahren, oftmals spielen andere Kriterien eine wichtigere Rolle. Wir kommen später darauf zurück.

## 2.5 Implizite Runge–Kutta–Verfahren

Bisher haben wir Runge–Kutta–Verfahren betrachtet, bei denen die Koeffizientenmatrix die Form

$$A = \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ a_{31} & a_{32} & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{s1} & \cdots & \cdots & a_{s,s-1} & 0 \end{pmatrix} \in \mathbb{R}^{s\times s}$$

hatte. Es stellt sich nun die Frage, was passiert, wenn wir hier "volle" Matrizen der Form

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix} \in \mathbb{R}^{s\times s}$$

zulassen. Zunächst einmal können wir auch mit solchen Koeffizienten ganz formal durch Erweiterung von Definition 2.21 wieder Runge–Kutta–Verfahren definieren.

**Definition 2.26** Ein *s–stufiges implizites Runge–Kutta–Verfahren* ist gegeben durch

$$k_i = f\left(t + c_i h,\, x + h\sum_{j=1}^{s} a_{ij}k_j\right) \quad \text{für } i = 1,\dots,s$$

$$\Phi(t,x,h) = x + h\sum_{i=1}^{s} b_i k_i.$$

Den Wert $k_i = k_i(t,x,h)$ bezeichnen wir dabei als $i$–te Stufe des Verfahrens. □

Der Grund für den Namen *implizites Verfahren* liegt darin, dass die Definition der $k_i$ nun keine "Zuweisung" mehr ist, sondern ein $s$–dimensionales nichtlineares Gleichungssystem bildet, dessen Lösung gerade der Vektor $k^T = (k_1^T,\dots,k_s^T) \in \mathbb{R}^{s\cdot n}$ ist. Die Werte $k_i \in \mathbb{R}^n$ sind also *implizit* definiert.

Das einfachste Verfahren dieser Klasse ist durch das Butcher–Tableau

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

gegeben. Ausgeschrieben lautet es

$$k_1 = f(t+h, x+hk_1), \quad \Phi(t,x,h) = x + hk_1,$$

die dadurch erzeugt Gitterfunktion ist rekursiv gegeben durch

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_i) + h_i f(t_{i+1}, \tilde{x}(t_{i+1})).$$

Dieses Verfahren heißt *implizites Euler–Verfahren* und besitzt genau wie sein explizites Gegenstück die Konsistenzordnung $p = 1$. Beachte, dass hier tatsächlich in jedem Schritt ein nichtlineares Gleichungssystem gelöst werden muss. Implizite Runge–Kutta–Verfahren mit Konsistenzordnung $p = 2$ sind z.B. die implizite Mittelpunktregel oder die implizite Trapezregel, die durch

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} \quad \text{bzw.} \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

gegeben sind.

Wir werden später sehen, dass implizite Verfahren für manche Differentialgleichungen gegenüber den expliziten Verfahren deutliche Vorteile besitzen. Zunächst wollen wir uns aber Gedanken darüber machen, wie solch ein implizites Verfahren implementiert werden kann, d.h., wie wir das nichtlineare Gleichungssystem zur Berechnung der $k_i$ lösen können.

Zunächst einmal gibt es manchmal die Möglichkeit, die entstehenden Gleichungen per Hand in explizite Form zu bringen. Betrachten wir z.B. das implizite Euler–Verfahren angewendet auf die eindimensionale DGL

$$\dot{x}(t) = \lambda x(t),$$

so erhalten wir

$$k_1 = f(t+h, x+hk_1) = \lambda(x+hk_1) = \lambda x + h\lambda k_1,$$

woraus für hinreichend kleine $h$ die Gleichung

$$k_1 = \frac{\lambda x}{1-h\lambda}$$

folgt.

Oft kommt man mit dieser Strategie aber nicht weiter, wir müssen das entstehende Gleichungssystem

$$k = F(k)$$

mit

$$k = \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix} \in \mathbb{R}^{s\cdot n} \quad \text{und} \quad F(k) = \begin{pmatrix} f\left(t + c_1 h,\, x + h\sum_{j=1}^{s} a_{1j}k_j\right) \\ \vdots \\ f\left(t + c_s h,\, x + h\sum_{j=1}^{s} a_{sj}k_j\right) \end{pmatrix}$$

also numerisch lösen.

Eine einfache Möglichkeit hierzu beruht auf der Tatsache, dass $f$ nach Voraussetzung Lipschitz–stetig mit Konstante $L$ ist. Hieraus folgt sofort, dass auch die Abbildung $F$ Lipschitz–stetig mit Konstante $hL$ ist. Falls $hL =: K < 1$ ist, folgt damit

$$\|F(k^1) - F(k^2)\| \le K\|k^1 - k^2\|,$$

so dass $F$ eine Kontraktion ist, weswegen der Vektor $k$ mittels der aus der Numerik 1 bekannten Fixpunktiteration

$$k^{(j+1)} = F(k^{(j)}) \tag{2.25}$$

berechnet werden kann. Als Startwert für diese Iteration empfiehlt es sich, im ersten Schritt $k_i^{(0)} = f(t+c_i h, x)$ und in den folgenden Schritten den Wert von $k$ aus dem vorhergehenden

Schritt zu verwenden. Ein geeignetes Abbruchkriterium ergibt sich wie in der Numerik 1 diskutiert aus dem Banach'schen Fixpunktsatz: Die Iteration wird so lange durchgeführt, bis

$$\|k^{(j+1)} - k^{(j)}\| \leq \varepsilon$$

für eine vorgegebene Toleranz $\varepsilon$ ist, damit ist dann die Genauigkeit

$$\|k^{(j+1)} - k^*\| \leq \frac{hL}{1-hL}\varepsilon$$

garantiert, wobei $k^*$ die exakte Lösung bezeichnet. Als Letztes müssen wir uns noch überlegen, wie $\varepsilon$ gewählt werden sollte. Damit das Verfahren

$$\Phi(t, x, h) = x + h\sum_{i=1}^{s} b_i k_i$$

den Konsistenzfehler $O(h^{p+1})$ einhält, sollte $\|k^{(j+1)} - k^*\| \leq \varepsilon_0 h^p$ für ein $\varepsilon_0 > 0$ gelten. Damit diese Schranke eingehalten wird, muss

$$\frac{hL}{1-hL}\varepsilon \leq \varepsilon_0 h^p$$

gelten, was für kleine $h$ gerade durch die Wahl $\varepsilon \approx \varepsilon_0 h^{p-1}$ garantiert wird. Das Abbruchkriterium hängt für $p \geq 2$ also von der Schrittweite $h$ ab.

Die Iteration (2.25) wird auch *Gesamtschrittiteration* genannt. Eine einfache Modifikation dieser Iteration ist die *Einzelschrittiteration*, die durch die Vorschrift

$$k_i^{(j+1)} = f\left(t + c_i h,\, x + h\sum_{l=1}^{i-1} a_{il} k_l^{(j+1)} + h\sum_{l=i}^{s} a_{il} k_l^{(j)}\right), \quad i = 1, \ldots, s \tag{2.26}$$

gegeben ist. Dies ist ein ähnlicher Trick, wie wir ihn in der Numerik 1 beim Übergang vom Jacobi– zum Gauß–Seidel–Verfahren angewendet haben: Wir verwenden die bereits bekannten Werte $k_1^{j+1}, \ldots, k_{i-1}^{j+1}$ der $j+1$–ten Iteration bei der Berechnung von $k_i^{j+1}$. Im Allgemeinen konvergiert die Einzelschrittiteration (2.26) etwas schneller als die Gesamtschrittiteration (2.25).

Falls die Lipschitz–Konstante $L$ des Vektorfeldes groß ist, werden bei diesen Fixpunktiterationen sehr kleine Zeitschrittweiten $h > 0$ benötigt, um die Kontraktionsbedingung $K = hL < 1$ sicher zu stellen. In diesem Falle können andere Verfahren vorteilhaft sein. So kann man das Problem $k = F(k)$ in ein geeignetes Nullstellenproblem umwandeln, z.B. mittels $0 = G(k) := k - F(k)$ (es gibt weitere, u.U. numerisch günstigere äquivalente Nullstellenprobleme, vgl. [2], Abschnitt 6.2.2). Wenn man nun die Ableitung $DG$ ausrechnen kann, die sich aus der Ableitung $\partial/\partial x f(x)$ ergibt, so ist das Newton–Verfahren sehr gut geeignet, da man mit $k$ aus dem vorhergehenden Schritt bzw. mit $k_i = f(t + c_i, x)$ einen guten Startwert für das (ja nur lokal konvergente) Newton–Verfahren besitzt.

Zusammenfassend führt dies auf den folgenden Algorithmus.

**Algorithmus 2.27 (Lösung eines Anfangswertproblems mit implizitem Runge–Kutta–Verfahren)**

**Eingabe:** Anfangsbedingung $(t_0, x_0)$, Endzeit $T$, Schrittzahl $N$, Einschrittverfahren $\Phi$

(1) Setze $h := (T - t_0)/N$, $\tilde{x}_0 = x_0$

(2) Für $i = 0, \ldots, N-1$:

(2a) Berechne $t_{i+1} = t_i + h$ und löse das nichtlineare Gleichungssystem $k = F(k)$

(2b) Berechne $\tilde{x}_{i+1} := \Phi(t_i, \tilde{x}_i, h) = \tilde{x}_i + h \sum_{j=1}^{s} b_j k_j$

**Ausgabe:** Werte der Gitterfunktion $\tilde{x}(t_i) = \tilde{x}_i$ in $t_0, \ldots, t_N$ □

Die Analyse impliziter Runge–Kutta–Verfahren ist im Vergleich zu den expliziten Verfahren komplizierter, da die Ableitungen von $\Phi$ (mit denen man sowohl die Konsistenz gemäß Satz 2.19 als auch die Lipschitz–Bedingung über die Ableitung nach $x$ überprüfen kann) mit Hilfe des Satzes über implizite Funktionen berechnet werden müssen. Die Grundideen der Beweise sind aber völlig identisch, weswegen wir die technischen Details hier nicht vertiefen wollen.

**Bemerkung 2.28** Für explizite Runge–Kutta–Verfahren haben wir in Lemma 2.23 gesehen, dass die Stufenanzahl $s$ eine obere Schranke für die Konsistenzordnung $p$ bildet, also immer $p \leq s$ gilt. Für implizite Verfahren ist die Schranke nicht ganz so strikt: Für ein $s$–stufiges implizites Runge–Kutta–Verfahren $\Phi$ mit Konsistenzordnung $p$ gilt die Ungleichung $p \leq 2s$, d.h. die Konsistenzordnung ist maximal zwei mal so groß wie die Stufenzahl. Zum Beweis dieser Aussage wenden wir das Verfahren wieder auf das Anfangswertproblem

$$\dot{x}(t) = x(t), \;\; x(0) = 1$$

mit exakter Lösung $e^t$ an. Man kann nun zeigen, dass die numerische Lösung von der Form

$$\Phi(0, 1, h) = P(h)/Q(h)$$

für zwei Polynome $P, Q \in \mathcal{P}_s$ mit $Q \not\equiv 0$ ist. Falls nun $\Phi(0, 1, h) - e^h = O(h^{2s+2})$ gilt, so folgt auch $P(h) - Q(h)e^h = O(h^{2s+2})$. Mittels Induktion über $s$ zeigt man dann, dass dies nur für $P \equiv Q \equiv 0$ gelten kann, was ein Widerspruch zu $Q \not\equiv 0$ ist. Also kann $\Phi(0, 1, h) - e^h = O(h^{2s+2})$ nicht gelten, weswegen im besten Fall $\Phi(0, 1, h) - e^h = O(h^{2s+1})$ sein kann, also $p \leq 2s$. □

Während es bei expliziten Runge–Kutta–Verfahren sehr schwierig ist, Verfahren für große $p$ zu konstruieren, lässt sich die maximale Konsistenzordnung $p = 2s$ bei impliziten Verfahren relativ leicht realisieren. Wiederum auf Butcher geht nämlich die Familie der *Gauß–Verfahren* zurück, bei denen sich die die Koeffizienten durch Nullstellen der Legendre–Polynome (ähnlich wie bei der Gauß–Quadratur) ermitteln lassen und die eine Familie von impliziten Verfahren mit $p = 2s$ bildet.

## 2.6 Steife Differentialgleichungen

Steife Differentialgleichungen sind eine Klasse von Differentialgleichungen, die mit expliziten Verfahren nur schwer zu lösen sind. Sie bilden die Hauptmotivation dafür, implizite Verfahren zu betrachten und zu verwenden. Leider ist es nicht ganz leicht, einer Differentialgleichung anzusehen, ob sie "steif" ist; es ist nicht einmal leicht, diese Eigenschaft formal zu definieren. Vielleicht ist die informelle Beschreibung "mit expliziten Verfahren schwer zu lösen" bereits die beste mögliche Definition. Wir wollen aber trotzdem versuchen, diese Eigenschaft etwas zu formalisieren und gewisse Kriterien herausarbeiten, an denen man erkennen kann, ob man es mit einer steifen DGL zu tun hat.

Wir wollen dazu zunächst den Begriff "schwer zu lösen" etwas genauer fassen. Aus Satz 2.11 wissen wir, dass für allgemeine Einschrittverfahren mit Konvergenzordnung $p > 0$ die Abschätzung der Form

$$\|\tilde{x}(t_i) - x(t_i)\| \leq CEh^p$$

für alle hinreichend kleinen $h > 0$ gilt, wobei $E > 0$ aus der Konsistenzbedingung stammt und

$$C = \frac{1}{L}(\exp(L(t_i - t_0)) - 1)$$

von der Konstanten $L$ der Lipschitzbedingung sowie von der Größe des Zeitintervalls $T - t_0$ abhängt. Eine Differentialgleichung ist nun schwer zu lösen, wenn $CE$ eine sehr große Konstante ist oder wenn die Abschätzung für $\|\tilde{x}(t_i) - x(t_i)\|$ nur für sehr kleine $h > 0$ gilt. Was "sehr groß" bzw. "sehr klein" in diesem Zusammenhang bedeutet, hängt im Wesentlichen davon ab, wieviel Zeit man in die Berechnung der Lösung investieren möchte und wie kleine Zeitschritte man noch zulassen möchte. Eine genaue Schranke kann man — ähnlich wie bei der Frage "wann ist ein Problem schlecht konditioniert?" — nicht angeben.

Sicherlich muss man damit rechnen, dass eine Differentialgleichung schwer zu lösen ist, wenn sie schlecht konditioniert ist. Steife Differentialgleichungen zeichnen sich nun dadurch aus, dass sie mit expliziten Verfahren schwer zu lösen sind, *obwohl* sie gut konditioniert sind. Dass dies tatsächlich passieren kann, wollen wir an einem bereits bekannten Beispiel illustrieren: Wir betrachten wieder die 1d DGL

$$\dot{x}(t) = \lambda x(t)$$

mit $\lambda \in \mathbb{R}$. Für diese Gleichung hatten wir gesehen, dass sie die Kondition

$$\kappa = e^{\lambda(t-t_0)}.$$

besitzt und deswegen für $t >> t_0$ und $\lambda < 0$ sehr gut konditioniert ist, da $\kappa \approx 0$ ist. Wir wollen die exakte Lösung $x(t; x_0) = e^{\lambda t}x_0$ dieser Gleichung für $\lambda << 0$ mit der numerischen Approximation durch das Euler–Verfahren vergleichen. Diese Approximation ist gegeben durch

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_i) + h\lambda\tilde{x}(t_i) = (1 + h\lambda)\tilde{x}(t_i).$$

Durch Induktion sieht man leicht, dass die Euler–Lösung für $t_i = hi$ damit gerade durch

$$\tilde{x}(t_i) = (1 + h\lambda)^i x_0$$

gegeben ist. Für kleine $\lambda < 0$ konvergiert die exakte Lösung z.B. mit Anfangswert $x_0 = 1$ sehr schnell gegen 0. Damit die Euler–Lösung eine vernünftige Approximation darstellt, sollte diese also auch gegen Null streben. Damit dies passiert, muss $|1 + h\lambda| < 1$ sein, was für negative $\lambda$ genau dann der Fall ist, wenn $|h\lambda| < 2$, also

$$h < 2/|\lambda|$$

ist. Z.B. für $\lambda = -10000$ müssen wir den Zeitschritt $h < 1/5000$ wählen, um überhaupt eine halbwegs sinnvolle Approximation zu erhalten und das, obwohl die Gleichung sehr gut konditioniert ist.

Zum Vergleich betrachten wir nun das implizite Euler–Verfahren, das durch

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_i) + h\lambda\tilde{x}(t_{i+1}) \quad \Leftrightarrow \quad \tilde{x}(t_{i+1}) = \frac{\tilde{x}(t_i)}{1 - h\lambda}$$

gegeben ist. Die approximierte Lösung ist also

$$\tilde{x}(t_i) = \frac{1}{(1 - h\lambda)^i} x_0.$$

Hier strebt die Lösung genau dann gegen Null, wenn $|1/(1 - h\lambda)| < 1$ ist, also wenn $|1 - h\lambda| > 1$ ist. Da $\lambda < 0$ ist, ist diese Bedingung für sämtliche Zeitschritte $h > 0$ erfüllt, die Lösung konvergiert also für alle Zeitschritte gegen Null und stellt damit eine sinnvolle Approximation dar. Abbildung 2.6 zeigt die exakte Lösung sowie die numerischen Approximationen für $\lambda = -100$ für verschiedene Zeitschritte.

### 2.6.1 Stabilität

Für die 1d–Gleichung $\dot{x}(t) = \lambda x(t)$ können wir also sagen, dass sie steif ist, wenn $\lambda < 0$ und $|\lambda|$ groß ist. Wir wollen dieses Kriterium auf eine größere Klasse von Differentialgleichungen verallgemeinern.

Wir betrachten dazu die Klasse der *linearen zeitinvarianten* DGL, die gegeben ist durch

$$\dot{x}(t) = Ax(t), \tag{2.27}$$

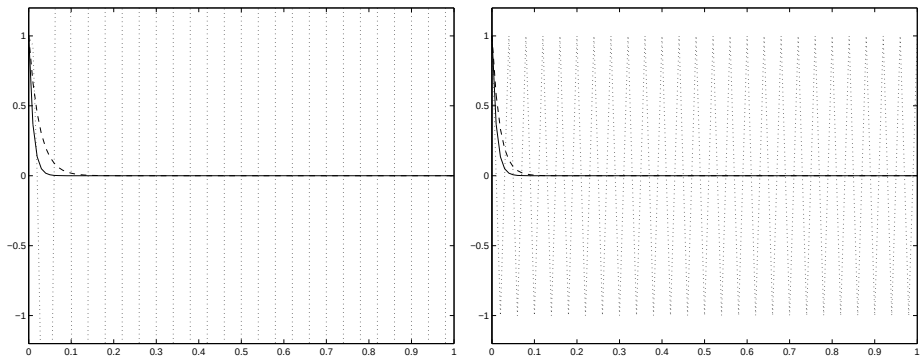
wobei $x(t) \in \mathbb{R}^n$ und $A \in \mathbb{R}^{n\times n}$ ist. Die Idee, diese Klasse von Differentialgleichungen zu betrachten, geht auf Germund Dahlquist[5] zurück. Für solche Gleichungen sind die durch ein Runge–Kutta–Verfahren erzeugten approximativen Lösungen stets von der Form

$$\tilde{x}(t_{i+1}) = \widetilde{A}\tilde{x}(t_i) \tag{2.28}$$

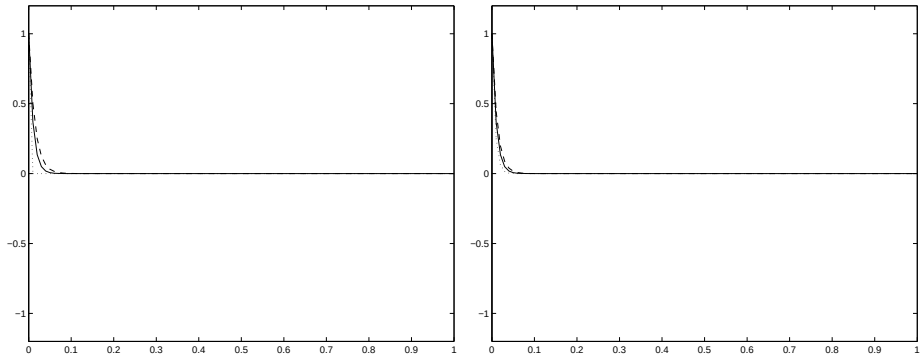
für ein $\widetilde{A} \in \mathbb{R}^{n\times n}$. Wir beschränken uns in diesem Abschnitt auf den Fall äquidistanter Zeitschritte $h_i = h$ und $t_0 = 0$, woraus sich $t_i = hi$ ergibt. Eine Gleichung der Form (2.28) wird *lineare zeitinvariante Differenzengleichung* genannt. Wir bezeichnen die Lösungen von (2.28) mit $\tilde{x}(0) = x_0$ mit $\tilde{x}(t; x_0)$, wobei $t \in \mathbb{R}$ ein Vielfaches von $h$ ist. Offenbar gilt gerade $\tilde{x}(hi; x_0) = \widetilde{A}^i x_0$.

Für das explizite Euler–Verfahren gilt z.B. $\widetilde{A} = \mathrm{Id} + hA$, während für das implizite Euler–Verfahren $\widetilde{A} = (\mathrm{Id} - hA)^{-1}$ gilt, wobei $\mathrm{Id} \in \mathbb{R}^{n\times n}$ die Einheitsmatrix bezeichnet. Genauer beschreibt das folgende Lemma, wie $A$ und $\widetilde{A}$ zusammenhängen.

[5] schwedischer Mathematiker, 1925-2005



$h = 0.04$ $h = 0.02$



$h = 0.01$ $h = 0.005$

Abbildung 2.6: Exakte Lösung (durchgezogen), explizite Euler–Lösung (gepunktet) und implizite Euler–Lösung (gestrichelt) für $\dot{x}(t) = \lambda x(t)$, $x(0) = 1$, $\lambda = -100$

**Lemma 2.29** Für jedes $s$–stufige Runge–Kutta–Verfahren lässt sich die Matrix $\widetilde{A}$ in (2.28) als

$$\widetilde{A} = R(hA)$$

schreiben, wobei $R$ eine von $h$ unabhängige Funktion ist. Für explizite Runge–Kutta–Verfahren ist $R$ ein Polynom vom Grad $\leq s$, für implizite Verfahren ist $R$ eine rationale Funktion, d.h. eine Funktion der Form $R(z) = P(z)Q(z)^{-1}$, wobei $P$ und $Q$ wieder Polynome vom Grad $\leq s$ sind.

**Beweis:** Es seien $a_{ij}$ und $b_i$ die Koeffizienten des Verfahrens. Dann gilt für die Stufen $k_i$ bei Anwendung auf (2.27) die Beziehung

$$hk_i = hAx + \sum_{j=1}^{s} a_{ij} hAhk_j$$

wobei wir beim expliziten Verfahren die Konvention $a_{ij} = 0$ für $j \geq i$ machen. Im expliziten Fall folgt per Induktion, dass jedes $hk_i$ ein Polynom in $hA$ vom Grad $\leq i$ ist und linear in $x$ ist. Damit ist $\Phi(t, x, h) = x + \sum b_i hk_i$ ein Polynom vom Grad $\leq s$ in $hA$ und linear in $x$, also gerade von der behaupteten Form.

Im impliziten Fall erhalten wir

$$\left( hk_i - \sum_{j=1}^{s} a_{ij} hAhk_j \right) = hAx$$

für $i = 1, \ldots, s$. Der $n \cdot s$–dimensionale Vektor $k = (k_1^T, \ldots, k_s^T)^T$ ist also gerade die Lösung eines $n \cdot s$–dimensionalen linearen Gleichungssystems, dessen Matrix affin linear von $A$ und dessen rechte Seite linear von $A$ und $x$ abhängt. Durch Auflösen dieses Gleichungssystems sieht man (nach länglicher Rechnung, die wir hier nicht durchführen wollen), dass sich die $k_i$ als

$$hk_i = \hat{P}_i(hA)Q(hA)^{-1}x$$

schreiben lassen, wobei die $\hat{P}_i$ und $Q$ Polynome vom Grad $\leq s$ sind. Damit ist auch $\Phi$ wegen

$$\begin{aligned} \Phi(t,x,h) &= x + \sum b_i hk_i \\ &= x + \sum b_i \hat{P}_i(hA)Q(hA)^{-1}x \\ &= \left( Q(hA) + \sum b_i \hat{P}_i(hA) \right) Q(hA)^{-1}x \\ &= P(hA)Q(hA)^{-1}x \end{aligned}$$

von der behaupteten Form. □

**Bemerkung 2.30** Aus dem Gleichungssystem des Beweises kann man eine explizite Formel für $R(hA)$ berechnen, die aber recht kompliziert ist. Wir werden allerdings später sehen, dass es für unsere Zwecke ausreicht, die Funktion $R$ für komplexwertige Argumente $z \in \mathbb{C}$ explizit zu kennen. Wenn wir die Koeffizienten des Verfahrens mit $\mathcal{A} = (a_{ij})_{i,j=1,\ldots,s}$ und $b = (b_1, \ldots, b_s)^T$ bezeichnen, so kann man hierfür den expliziten Ausdruck

$$R(z) = 1 + zb^T(\mathrm{Id} - z\mathcal{A})^{-1}\mathbf{e}$$

mit $\mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^s$ berechnen. Für komplexe Argumente wird die Funktion $R$ als *Stabilitätsfunktion* des Verfahrens bezeichnet.

Z.B. ergeben sich für das explizite Euler–Verfahren $R(z) = 1 + z$, für das implizite Euler–Verfahren $R(z) = (1 - z)^{-1}$ und für die implizite Trapezregel aus Abschnitt (2.5) $R(z) = (1 + z/2)/(1 - z/2)$. □

Wie im obigen eindimensionalen Fall wollen wir speziell Lösungen betrachten, die gegen Null streben und untersuchen, für welche Zeitschritte die numerische Approximation dieses Verhalten widerspiegelt. Dazu verwenden wir die folgende Definition.

**Definition 2.31** Eine Differentialgleichung (2.27) bzw. eine Differenzengleichung (2.28) heißt *(global) exponentiell stabil*, falls Konstanten $c, \sigma > 0$ existieren, so dass für alle Anfangswerte $x_0 \in \mathbb{R}^n$ die Ungleichung

$$\|x(t; x_0)\| \leq ce^{-\sigma t}\|x_0\| \text{ für alle } t \geq 0$$

bzw.

$$\|\tilde{x}(t;x_0)\| \leq ce^{-\sigma t}\|x_0\| \text{ für alle } t = ih \geq 0$$

gilt. □

Für die obigen Gleichungstypen (2.27) und (2.28) kann man zeigen, dass sie genau dann exponentiell stabil sind, wenn alle Lösungen gegen Null konvergieren. Die spezielle exponentielle Abschätzung ergibt sich dann aus der Linearität der Gleichungen.

In Analogie zum eindimensionalen Fall nennen wir eine exponentiell stabile Differentialgleichung der Form (2.27) steif, wenn für explizite Verfahren ein sehr kleiner Zeitschritt nötig ist, damit die durch das Verfahren erzeugte Differenzengleichung (2.28) ebenfalls exponentiell stabil ist.

Um nun zu sehen, wie man anhand der Matrix $A$ die Steifheit erkennen kann und zu verstehen, warum implizite Verfahren hier Vorteile haben, brauchen wir ein geeignetes Kriterium für exponentielle Stabilität. Glücklicherweise muss man nicht alle Lösungen kennen, um zu entscheiden, ob exponentielle Stabilität vorliegt; man kann diese Eigenschaft anhand der Matrizen $A$ bzw. $\widetilde{A}$ erkennen, wie der folgende Satz zeigt. Hierbei bezeichnet $\Re(z) = a$ den Realteil und $|z| = \sqrt{a^2 + b^2}$ den Betrag einer komplexen Zahl $z = a + ib \in \mathbb{C}$.

**Satz 2.32** (i) Die Differentialgleichung (2.27) ist genau dann exponentiell stabil, wenn für alle Eigenwerte $\lambda_i$ von $A$ die Ungleichung $\Re(\lambda_i) < 0$ gilt.

(ii) Die Differenzengleichung (2.28) ist genau dann exponentiell stabil, wenn für alle Eigenwerte $\tilde{\lambda}_i$ von $\widetilde{A}$ die Ungleichung $|\tilde{\lambda}_i| < 1$ gilt.

**Beweisskizze:** Wir beweisen Teil (ii) unter der Annahme, dass $\widetilde{A}$ diagonalisierbar ist (der Beweis von (ii) im nicht–diagonalisierbaren Fall funktioniert genauso, ist aber technischer; der Beweis von (i) ist ähnlich, verlangt aber weitere Kenntnisse über die Lösungsstruktur von (2.27), auf die wir hier nicht eingehen können).

Falls $\widetilde{A}$ diagonalisierbar ist, so existiert eine Koordinatentransformationsmatrix $T \in \mathbb{R}^{n\times n}$, so dass

$$T^{-1}\widetilde{A}T = \widetilde{\Lambda} = \begin{pmatrix} \tilde{\lambda}_1 & & & \\ & \tilde{\lambda}_2 & & \\ & & \ddots & \\ & & & \tilde{\lambda}_n \end{pmatrix}$$

ist, wobei die $\tilde{\lambda}_i$ gerade die Eigenwerte von $\widetilde{A}$ sind. Für die Lösung $\tilde{x}(hi;x_0)$ gilt dann gerade

$$\begin{aligned} \tilde{x}(ih;x_0) &= \widetilde{A}^i x_0 \\ &= (T\widetilde{\Lambda}T^{-1})^i x_0 \\ &= T\widetilde{\Lambda}^i T^{-1} x_0. \end{aligned}$$

Sei nun $\alpha = \max_i |\tilde{\lambda}_i| < 1$. Wenn wir $y = (y_1, \ldots, y_n)^T = T^{-1}x_0$ setzen, so folgt

$$\widetilde{\Lambda}^i y = \begin{pmatrix} \tilde{\lambda}_1^i y_1 \\ \vdots \\ \tilde{\lambda}_n^i y_n \end{pmatrix}$$

und damit $\|\widetilde{\Lambda}^i y\| \leq \alpha^i \|y\|$. Mit $\sigma = -\ln(\alpha)/h > 0$ und $t = hi$ folgt

$$\|\widetilde{\Lambda}^i y\| \leq e^{-\sigma t}\|y\|$$

und damit

$$\|\tilde{x}(t;x_0)\| \leq \|T\|e^{-\sigma t}\|T^{-1}x_0\| \leq e^{-\sigma t}\|T\|\|T^{-1}\|\|x_0\| = ce^{-\sigma t}\|x_0\|$$

mit $c = \|T\|\|T^{-1}\|$.

Sei umgekehrt $|\tilde{\lambda}_j| \geq 1$ für ein $j$ und sei $x_0$ ein zugehöriger Eigenvektor. Dann gilt

$$\|\tilde{x}(t;x_0)\| = \|\widetilde{A}^i x_0\| = |\tilde{\lambda}_j^i|\|x_0\| \geq \|x_0\|$$

für alle $t = ih > 0$, weswegen (2.28) nicht exponentiell stabil ist. □

Wir bezeichnen mit

$$\Sigma(A) = \{\lambda_i \,|\, \lambda_i \text{ ist Eigenwert von } A\}$$

die Menge aller Eigenwerte, das sogenannte *Spektrum* von $A$.

Für die Differentialgleichung muss damit gerade

$$\Sigma(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re(z) < 0\}$$

gelten, damit exponentielle Stabilität vorliegt. Ein Eigenwert $\lambda_i \in \mathbb{C}^-$ wird dabei als *stabiler Eigenwert* bezeichnet. Analog muss für die numerische Approximation (2.28)

$$\Sigma(\widetilde{A}) \subset B_1(0) := \{z \in \mathbb{C} \,|\, |z| < 1\}$$

gelten, damit exponentielle Stabilität vorliegt.

Zu klären bleibt die Frage, welche Bedingung $A$ aus (2.27) erfüllen muss, damit (2.28) für die Matrix $\widetilde{A} = R(hA)$ exponentiell stabil ist. Sicherlich hängt dies vom verwendeten Verfahren und vom Zeitschritt ab. Hierzu verwenden wir die folgende Definition. Beachte dabei, dass

$$\Sigma(A) \subset \mathbb{C}^- \;\Leftrightarrow\; \Sigma(hA) \subset \mathbb{C}^- \text{ für alle } h > 0$$

gilt, da $\lambda_i$ genau dann ein Eigenwert von $A$ ist, wenn $h\lambda_i$ ein Eigenwert von $hA$ ist.

**Definition 2.33** (i) Das *Stabilitätsgebiet* $\mathcal{S} \subset \mathbb{C}$ eines Runge–Kutta–Verfahrens mit Stabilitätsfunktion $R$ ist definiert als die maximale Teilmenge der komplexen Zahlen, für die die Folgerung

$$\Sigma(hA) \subset \mathcal{S} \;\Rightarrow\; \Sigma(R(hA)) \subset B_1(0)$$

gilt. Mit anderen Worten ist $\mathcal{S}$ gerade die Menge von Eigenwerten $\lambda_i$, die $hA$ aus (2.27) annehmen darf, damit (2.28) mit $\widetilde{A} = R(hA)$ exponentiell stabil ist.

(ii) Ein Runge–Kutta–Verfahren heißt *$A$–stabil*, falls

$$\mathbb{C}^- \subseteq \mathcal{S}$$

gilt bzw., äquivalent dazu, falls die Folgerung

$$\Sigma(hA) \subset \mathbb{C}^- \;\Rightarrow\; \Sigma(R(hA)) \subset B_1(0)$$

gilt. □

Die Interpretation von (i) ist wie folgt: Zur korrekten numerischen Approximation einer exponentiell stabilen Gleichung muss die Schrittweite $h > 0$ so gewählt werden, dass die Eigenwerte von $hA$ in $\mathcal{S}$ liegen. Je besser $\mathcal{S}$ die Menge $\mathbb{C}^-$ ausschöpft, desto geringer sind die Anforderungen an die Schrittweite; im Falle der A–Stabilität gibt es überhaupt keine Einschränkungen der Schrittweite, die exponentielle Stabilität von (2.27) wird für alle Zeitschritte $h > 0$ von (2.28) "geerbt".

Das folgende Lemma zeigt, wie der Stabilitätsbereich $\mathcal{S}$ berechnet werden kann.

**Lemma 2.34** Gegeben sei ein Runge–Kutta–Verfahren mit Stabilitätsfunktion $R$ aus Bemerkung 2.30. Dann ist der Stabilitätsbereich gegeben durch

$$\mathcal{S} = \{z \in \mathbb{C} \,|\, |R(z)| < 1\}.$$

**Beweis:** Sei $B = hA$ mit beliebigem $A \in \mathbb{R}^{n\times n}$ und $h > 0$. Zum Beweis der Behauptung zeigen wir, dass $\lambda_i \in \mathbb{C}$ genau dann ein Eigenwert von $B$ ist, wenn $R(\lambda_i)$ ein Eigenwert von $R(B)$ ist.

Sei $C \in \mathbb{R}^{n\times n}$ eine beliebige Matrix mit Eigenwerten $\lambda_i$, $i = 1, \ldots, p \leq n$. Für ein Polynom

$$P(C) = \alpha_0 \mathrm{Id} + \alpha_1 C + \ldots + \ldots \alpha_s C^s$$

sind die Eigenwerte von $P(C)$ gerade die Eigenwerte $P(\lambda_i)$ von $C$, was man am einfachsten sieht, indem man $P$ auf die Jordan–Normalform $J$ von $C$ anwendet. Ein Jordanblock $J_i$ zum Eigenwert $\lambda_i$ wird dabei auf eine obere Dreiecksmatrix mit $P(\lambda_i)$ in der Diagonalen abgebildet, die genau den einzigen Eigenwert $P(\lambda_i)$ besitzt (lediglich die Vielfachheiten können sich u.U. ändern). Hierbei sind Eigenvektoren von $C$ wieder Eigenvektoren von $P(C)$.

Für die Inverse $C^{-1}$ ist sind die Eigenwerte gerade $1/\lambda_i$ und für ein Produkt zweier Matrizen mit gleichen Eigenvektoren sind die Eigenwerte gerade die Produkte der Eigenwerte.

Also folgt, dass die Eigenwerte von $R(B) = P(B)Q(B)^{-1}$ gerade die Produkte der Eigenwerte $P(\lambda_i)$ und $Q(\lambda_i)^{-1}$, also $P(\lambda_i)Q(\lambda_i)^{-1}$ sind. □

Mit Hilfe dieses Satzes können wir die Stabilitätsbereiche nun bestimmen. Für das explizite Euler–Verfahren mit $R(z) = 1 + z$ gilt

$$|R(z)| < 1 \quad \Leftrightarrow \quad |1 + z| < 1$$

also ist $\mathcal{S} = \{z \in \mathbb{C} || 1 + z| < 1\} = B_1(-1)$, also gerade der offene Ball mit Radius 1 um $-1$. Der Zeitschritt muss also so klein gewählt werden, dass für alle Eigenwerte die Bedingung $h\lambda_i \in B_1(-1)$ erfüllt ist.

Abbildung 2.7 zeigt die Stabilitätsbereiche einiger expliziter Runge–Kutta–Verfahren mit den Ordnungen $p = 1, \ldots, 4$. Man sieht, dass der Stabilitätsbereich $\mathcal{S}$ für wachsende Konsistenz größer wird, allerdings die Menge $\mathbb{C}^-$ bei weitem nicht ausschöpft. Im Falle betragsmäßig großer Eigenwerte $\lambda_i$ erhält man für all diese Verfahren starke Einschränkungen bei der Wahl der Zeitschritte.

Beachte, dass bei mehrdimensionalen Problemen nicht unbedingt der *Realteil* eines Eigenwertes betragsmäßig groß werden muss, damit der Betrag des Eigenwertes groß wird. Das folgende Beispiel illustriert dies.

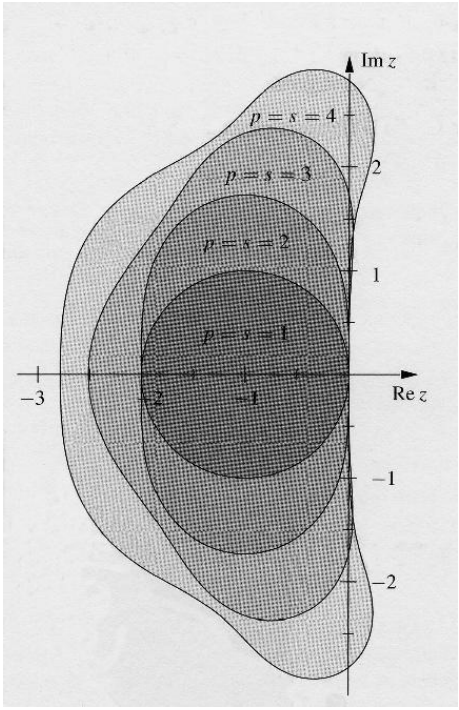



Abbildung 2.7: Stabilitätsbereiche expliziter Runge–Kutta–Verfahren, entnommen aus [2]

Betrachte die zweidimensionale lineare DGL

$$\dot{x}(t) = \begin{pmatrix} -1 & \alpha \\ -\alpha & -1 \end{pmatrix} x(t). \tag{2.29}$$

Die zugehörige Matrix besitzt die Eigenwerte $\lambda_{1/2} = -1 \pm i\alpha$. Hier haben Realteil und Imaginärteil eine geometrische Bedeutung für die Lösung: Der Realteil gibt an, wie schnell die Lösung gegen Null konvergiert (diese Größe ist hier konstant gleich $-1$), während der Imaginärteil angibt, wie schnell die Lösung sich dabei dreht. Abbildung (2.8) zeigt die exakten Lösungen für $\alpha = 0,\ 1,\ 10$ sowie die zugehörigen Euler–Lösungen mit $h = 0.02$. Man sieht: Wenn der Eigenwert betragsmäßig größer wird, weil der Imaginärteil wächst, dann wird die Euler–Lösung instabil.

Wie verhalten sich nun implizite Verfahren? Für das implizite Euler–Verfahren z.B. berechnet man

$$|R(z)| < 1 \ \Leftrightarrow\ 1/|1-z| < 1 \ \Leftrightarrow\ |1-z| > 1 \ \Leftarrow\ \Re(z) < 0.$$

Folglich gilt $\mathbb{C}^- \subset \mathcal{S}$, das Verfahren ist also A–stabil.

Viele implizite Verfahren sind A–stabil, und von denjenigen, die es nicht sind, besitzen viele einen Stabilitätsbereich, der deutlich größer ist als bei expliziten Verfahren. Eine Übersicht über die Stabilitätsbereiche einiger impliziter Verfahren findet sich z.B. im Abschnitt IV.3 des Buchs [3].

Eine lineare DGL (2.27) kann auch dann steif sein, wenn sie nicht exponentiell stabil ist, aber zumindest einige stabile Eigenwerte besitzt, also solche mit negativem Realteil.

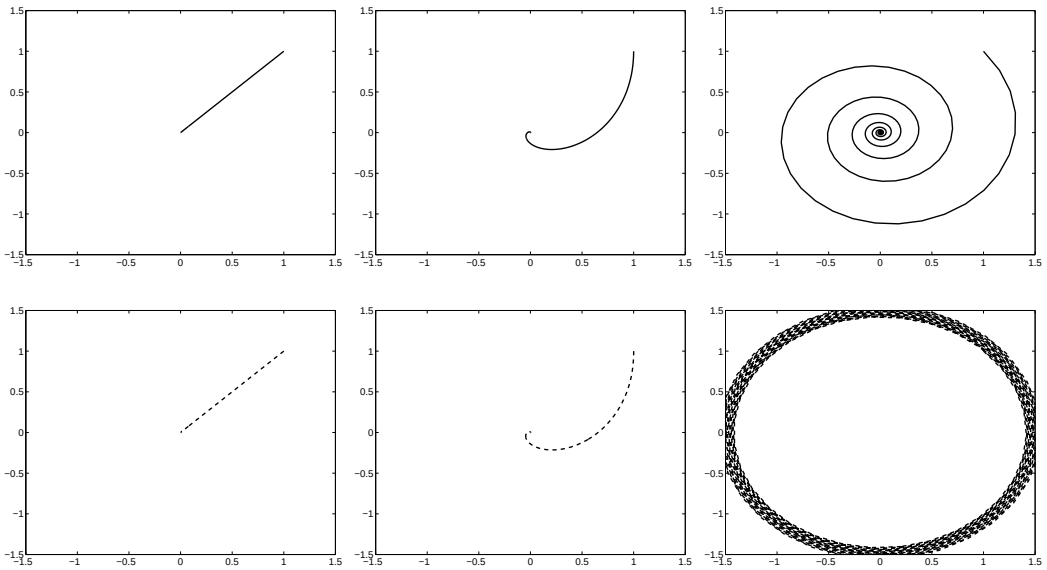


Abbildung 2.8: Exakte und Euler–Lösungen von (2.29) mit $\alpha = 0,\ 1,\ 10$, $h = 0.02$

Die Lösungskomponenten in den zugehörigen Eigenräumen (man nennt deren Vereinigung *stabilen Unterraum*) verhalten sich dann wie bei einer exponentiell stabilen Gleichung. Folglich treten bei betragsmäßig großen stabilen Eigenwerten exakt die gleichen Probleme auf, auch wenn die Gleichung insgesamt nicht exponentiell stabil ist. Dies führt uns auf die folgende Charakterisierung.

**Bemerkung 2.35** Eine lineare zeitinvariante Differentialgleichung ist steif, falls die zugehörige Matrix $A$ betragsmäßig große stabile Eigenwerte besitzt. □

Für nichtlineare DGL $\dot{x}(t) = f(t, x(t))$ gibt es viele weitere Phänomene, die zur Steifheit führen; meistens kann man diese nicht so einfach am Vektorfeld $f$ ablesen. Im einfachsten Fall ist $f$ autonom und besitzt ein Gleichgewicht $x^*$, in dem $f$ stetig differenzierbar ist. In diesem Fall kann man $A = Df(x^*)$ betrachten; wenn diese Matrix betragsmäßig große stabile Eigenwerte besitzt, so wird auch die nichtlineare DGL typischerweise steif sein. Steifheit kann aber auch auftreten, wenn kein Gleichgewicht vorliegt, z.B. wenn die DGL eine exponentiell stabile periodische Lösung besitzt (also eine periodische Lösung, gegen die alle Lösungen exponentiell konvergieren, zumindest für nahe liegende Anfangswerte). In diesem Fall kann die Gleichung steif sein, wenn die anderen Lösungen sehr schnell gegen die periodische Lösung streben (dies entspricht betragsmäßig großen negativen Realteilen im linearen Fall) oder wenn sich die periodische Lösung sehr schnell bewegt (dies entspricht den großen Imaginärteilen.)

**Bemerkung 2.36** Der Begriff der A–Stabilität wurde von G. Dahlquist in den 1960er Jahren eingeführt. A–Stabilität ist für sich genommen weder eine positive noch eine negative Eigenschaft: Zwar ist es zur numerischen Lösung steifer DGL vorteilhaft, wenn die exponentielle Stabilität von der numerischen Approximation geerbt wird. Allerdings kann

es andererseits auch passieren, dass die numerische Approximation exponentiell stabil ist, obwohl die exakte Gleichung diese Eigenschaft *nicht* besitzt, was zu falschen Rückschlüssen auf das Verhalten der exakten Lösungen führen kann. Eine stärkere Eigenschaft ist die *Erhaltung der Isometrie*, die verlangt, dass $\mathcal{S} = \mathbb{C}^-$ ist, d.h. die numerische Approximation ist exponentiell stabil *genau dann*, wenn die exakte Gleichung exponentiell stabil ist. Diese Eigenschaft besitzen nur sehr wenige Verfahren, z.B. die bereits erwähnten Gauß–Verfahren.

Auch dies bietet jedoch keine endgültige Lösung des Problems: Zum einen ist es u.U. sinnvoll, Verfahren zu verwenden, die andere Kenngrößen der DGL erhalten (vgl. Übungsblatt 5, Aufgabe 17) und die die Erhaltung der Isometrie u.U. ausschließen. Zum anderen gibt es bei nichtlinearen DGL viele andere Langzeitphänomene, die auch von isometrieerhaltenden Verfahren i.A. nicht korrekt erfasst werden. Mit diesen Problemen beschäftigt sich die *Numerik dynamischer Systeme*. Eine Vorlesung zu diesem Thema wird im kommenden Wintersemester angeboten. □

## 2.7 Schrittweitensteuerung

Nach den eher theoretischen Überlegungen des letzten Abschnittes wollen wir uns jetzt wieder algorithmischen Aspekten widmen. Bisher sind wir davon ausgegangen, dass die Schrittweiten $h_i$ gegeben sind, meistens haben wir sie als konstant $h_i \equiv h$ angenommen. In diesem Abschnitt wollen wir uns überlegen, wie man die Schrittweiten automatisch so steuern kann, so dass dort, wo es nötig ist, kleine Schrittweiten gewählt werden, damit eine gewünschte Genauigkeit eingehalten wird und dort, wo es ohne Genauigkeitsverlust möglich ist, große Schrittweiten erlaubt werden, die eine schnellere Rechnung ermöglichen. Wir nehmen dabei durchgehend an, dass das Vektorfeld der betrachteten DGL hinreichend oft differenzierbar ist, so dass die Konsistenzordnungen der betrachteten Verfahren tatsächlich realisiert werden.

### 2.7.1 Fehlerschätzung

Zur Entscheidung darüber, ob die Schrittweite groß oder klein gewählt werden soll, ist es nötig, den Fehler zu kennen, den wir im aktuellen Schritt machen. Wir wollen uns zuerst überlegen, welcher Fehler hierfür wichtig ist. Hierbei müssen wir zunächst überlegen, wie wir die Schrittweite steuern wollen. Wie in der numerischen Praxis üblich wollen wir uns hier darauf beschränken, zur Zeit $t_i$ eine gute Schrittweite $h_i$ für den Schritt von $t_i$ nach $t_{i+1} = t_i + h_i$ zu bestimmen und dabei auch einen "Schrittweitenvorschlag" $h_{i+1}$ für den nächsten Schritt zu machen. Wir wollen aber nicht zum Zeitpunkt $t_i$ die Schrittweiten in vorhergehenden Schritten $t_j$ für $j < i$ nachträglich korrigieren, da die dadurch anfallenden Neuberechnungen algorithmisch sehr ineffizient wären.

Um ein gutes $h_i$ zu bestimmen, müssen wir den Fehleranteil kennen, der durch den Schritt von $t_i$ nach $t_{i+1}$ hervorgerufen wird. Dieser Fehleranteil wird *lokaler Fehler* genannt. Wir haben in der Konvergenzanalyse in Abschnitt 2.2.3 verwendet, dass sich der Fehler zur Zeit $t_{i+1}$ mittels

$$\begin{aligned} \|\tilde{x}(t_{i+1}) - x(t_{i+1})\| &\leq \|\Phi(t_i, \tilde{x}(t_i), h_i) - \Phi(t_i, x(t_i), h_i)\| \\ &\quad + \|\Phi(t_i, x(t_i), h_i) - x(t_{i+1}; t_i, x(t_i))\| \end{aligned}$$

zerlegen lässt. Diese Zerlegung war für unsere theoretischen Überlegungen nützlich, hier ist sie nicht so günstig, da wir den in diesem Schritt hinzukommeden Fehleranteil

$$\|\Phi(t_i, x(t_i), h_i) - x(t_{i+1}; t_i, x(t_i))\|$$

nicht berechnen können, da wir $x(t_i)$ nicht kennen. Statt also in der Dreiecksungleichung den Term $\Phi(t_i, x(t_i), h_i)$ einzuschieben, schieben wir den Term $x(t_{i+1}; t_i, \tilde{x}(t_i))$ und erhalten so

$$\begin{aligned} \|\tilde{x}(t_{i+1}) - x(t_{i+1})\| &\leq \|\Phi(t_i, \tilde{x}(t_i), h_i) - x(t_{i+1}, t_i, \tilde{x}(t_i))\| \\ &\quad + \|x(t_{i+1}, t_i, \tilde{x}(t_i)) - x(t_{i+1}, t_i, x(t_i))\| \end{aligned}$$

Der zweite Fehlerterm hängt hierbei im Wesentlichen von dem bis zum Zeitpunkt $t_i$ gemachten Fehler ab, den wir nur durch Änderung der Zeitschritte $h_j$ für $j < i$ beeinflussen

können, was wir gerade nicht machen wollen. Der Fehlerterm, den wir mit der Wahl von $h_i$ wirklich beeinflussen können, ist der erste.

Die Idee der Schrittweitensteuerung (man sagt auch "adaptive Wahl der Schrittweite") liegt nun darin, $h_i$ so groß zu wählen, dass die Fehlerbedingung

$$\|\Phi(t_i, \tilde{x}(t_i), h_i) - x(t_{i+1}, t_i, \tilde{x}(t_i))\| \leq tol$$

für eine vorgegebene Größe $tol > 0$ gerade eingehalten wird. Dies ist natürlich so nicht möglich, da wir dafür die exakte Lösung $x(t_{i+1}, \tilde{x}(t_i), t_i)$ kennen müssten. Um dieses Problem zu lösen, verwendet man einen sogenannten *Fehlerschätzer*, den wir bereits in der Numerik I bei der adaptiven Integration kennen gelernt haben. Wir wiederholen die Definition.

**Definition 2.37** Eine numerisch berechenbare Größe $\bar{\varepsilon}$ heißt *Fehlerschätzer* für den tatsächlichen Fehler $\varepsilon$ eines numerischen Verfahrens, falls von $\bar{\varepsilon}$ und $\varepsilon$ unabhängige Konstanten $\kappa_1, \kappa_2 > 0$ existieren, so dass die Abschätzung

$$\kappa_1 \varepsilon \leq \bar{\varepsilon} \leq \kappa_2 \varepsilon$$

gilt. □

Wie können wir nun für unsere Einschrittverfahren einen solchen Fehlerschätzer bekommen? Die Idee besteht darin, den Schritt von $t$ nach $t_{i+1} = t_i + h_i$ mit zwei Verfahren $\widehat{\Phi}$ und $\Phi$ verschiedener Konsistenzordnung $\hat{p}$ und $p$ zu berechnen. Für

$$\hat{\eta}_i := \widehat{\Phi}(t_i, \tilde{x}(t_i), h_i) - x(t_{i+1}, t_i, \tilde{x}(t_i)) \quad \text{und} \quad \eta_i := \Phi(t_i, \tilde{x}(t_i), h_i) - x(t_{i+1}, t_i, \tilde{x}(t_i))$$

gilt damit

$$\hat{\varepsilon}_i := \|\hat{\eta}_i\| \leq \widehat{E} h_i^{\hat{p}+1} \quad \text{und} \quad \varepsilon_i := \|\eta_i\| \leq E h_i^{p+1}. \tag{2.30}$$

Wir nehmen hierbei an, dass $p \geq \hat{p} + 1$ gilt und dass $\hat{p}$ die maximale (oder echte) Konsistenzordnung von $\widehat{\Phi}$ ist. Damit ist $\Phi$ das genauere Verfahren, weswegen für alle hinreichend kleinen $h_i > 0$ die Ungleichung $\varepsilon_i < \hat{\varepsilon}_i$ bzw.

$$\theta = \frac{\varepsilon_i}{\hat{\varepsilon}_i} < 1 \tag{2.31}$$

gilt, da $\theta \to 0$ strebt, wenn $h_i \to 0$ geht.

Wir definieren den Fehlerschätzer nun als

$$\bar{\varepsilon} := \|\bar{\eta}\| \quad \text{mit} \quad \bar{\eta} = \widehat{\Phi}(t_i, \tilde{x}(t_i), h_i) - \Phi(t_i, \tilde{x}(t_i), h_i). \tag{2.32}$$

Der folgende Satz zeigt, dass diese Größe tatsächlich ein Fehlerschätzer im Sinne von Definition 2.37 ist.

**Satz 2.38** Betrachte zwei Einschrittverfahren $\widehat{\Phi}$ und $\Phi$ mit Konsistenzordnungen $\hat{p}$ und $p$ mit $p \geq \hat{p} + 1$. Dann ist die Größe $\bar{\varepsilon}$ aus (2.32) für alle hinreichend kleinen Schrittweiten $h_i > 0$ ein Fehlerschätzer für $\hat{\varepsilon}_i$ aus (2.30).

**Beweis:** Wir wählen $h_i$ so klein, dass die Abschätzung (2.31) gilt und $\theta < \theta_0 < 1$ ist. Aus der Definition von $\bar{\eta}$ folgt $\bar{\eta} = \hat{\eta}_i - \eta_i$, also

$$\frac{\|\hat{\eta}_i - \bar{\eta}\|}{\|\hat{\eta}_i\|} = \frac{\|\eta_i\|}{\|\hat{\eta}_i\|} = \frac{\varepsilon_i}{\hat{\varepsilon}_i} = \theta.$$

Damit ergibt sich

$$(1-\theta)\hat{\varepsilon}_i = (1-\theta)\|\hat{\eta}_i\| = \left(1 - \frac{\|\hat{\eta}_i - \bar{\eta}\|}{\|\hat{\eta}_i\|}\right)\|\hat{\eta}_i\| = \|\hat{\eta}_i\| - \underbrace{\|\hat{\eta}_i - \bar{\eta}\|}_{\geq \|\hat{\eta}_i\| - \|\bar{\eta}\|} \leq \|\bar{\eta}\| = \bar{\varepsilon},$$

also die untere Abschätzung mit $\kappa_1 = 1 - \theta_0$ und

$$\bar{\varepsilon} = \|\bar{\eta}\| \leq \|\hat{\eta}_i\| + \|\hat{\eta}_i - \bar{\eta}\| = \left(1 + \frac{\|\hat{\eta}_i - \bar{\eta}\|}{\|\hat{\eta}_i\|}\right)\|\hat{\eta}_i\| = (1+\theta)\|\hat{\eta}_i\| = (1+\theta)\hat{\varepsilon}_i,$$

also die obere Abschätzung mit $\kappa_2 = 1 + \theta_0$. □

Beachte, dass die Gültigkeit des Fehlerschätzers entscheidend von (2.31) abhängt, also nur für bereits hinreichend kleine Schrittweiten gilt.

### 2.7.2 Schrittweitenberechnung und adaptiver Algorithmus

Wir wollen nun untersuchen, wie man aus dem geschätzten Fehler effektiv eine neue Schrittweite berechnen kann. Hierzu benötigen wir eine weitere Annahme, nämlich dass der Fehler $\hat{\varepsilon}_i$ für kleine $h_i$ von der Form

$$\hat{\varepsilon}_i \approx c_i h_i^{\hat{p}+1} \tag{2.33}$$

ist. Für Runge–Kutta–Verfahren ist dies erfüllt, falls $f$ $\hat{p}+2$–mal stetig differenzierbar ist, wobei sich die $c_i$ gerade aus dem zu $h_i^{\hat{p}+1}$ gehörigen Koeffizienten der Taylor–Entwicklung ergeben. Allerdings ist der exakte Wert von $c_i$ unbekannt bzw. kann nur mit unverhältnismäßig großem Aufwand berechnet werden.

Sei nun eine Fehlerschranke $tol > 0$ für den lokalen Fehler vorgegeben. Wir führen jeweils einen Schritt mit beiden Verfahren $\widehat{\Phi}$ und $\Phi$ zum Zeitschritt $h_i$ durch. Sei $\bar{\varepsilon}$ der gemäß (2.32) berechnete Fehlerschätzer. Für kleine Schrittweiten gilt $\kappa_1 \approx \kappa_2 \approx 1$, also $\bar{\varepsilon} \approx \hat{\varepsilon}_i \approx c_i h_i^{\hat{p}+1}$. Hieraus können wir einen Schätzwert

$$\bar{c}_i = \frac{\bar{\varepsilon}}{h_i^{\hat{p}+1}}$$

für $c_i$ berechnen. Die gewünschte Fehlertoleranz wird damit (approximativ) für diejenige Schrittweite $h_{i,neu}$ eingehalten, für die die Gleichung

$$tol = \bar{c}_i h_{i,neu}^{\hat{p}+1} = \frac{\bar{\varepsilon}}{h_i^{\hat{p}+1}} h_{i,neu}^{\hat{p}+1}$$

bzw.

$$h_{neu} = \sqrt[\hat{p}+1]{\frac{tol}{\bar{\varepsilon}}}\, h$$

gilt. Da diese Gleichungen (wegen der verschiedenen "$\approx$") nur näherungsweise gelten, führt man in der Praxis noch einen "Sicherheitsfaktor" $fac \in (0,1)$ ein, um die Fehlerquellen bei der Fehlerschätzung zu kompensieren: man setzt

$$h_{i,neu} = \sqrt[\hat{p}+1]{fac\frac{tol}{\bar{\varepsilon}}}h_i.$$

Eine typische Wahl hierfür ist $fac = 0.9$.

Nach der Durchführung eines Schrittes mit Schrittweite $h_i$ und der Schätzung des Fehlers $\bar{\varepsilon}$ können nun zwei Fälle auftreten:

(i) $\bar{\varepsilon} > tol$:
In diesem Fall wird der Schritt mit $h_i = h_{i,neu}$ erneut durchgeführt ("zurückweisen und wiederholen").

(ii) $\bar{\varepsilon} \leq tol$:
In diesem Fall wurde die gewünschte Genauigkeit $tol$ erreicht. Der Schritt wird akzeptiert und die neue Schrittweite $h_{i,neu}$ wird als Schrittweite $h_{i+1}$ für den nächsten Schritt verwendet ("akzeptieren").

Beachte, dass die Schrittweite in Schritt (i) immer verkleinert wird. Die Wahl von $h_{i,neu}$ als Schrittweitenvorschlag für $h_{i+1}$ in (ii) ist also ein notwendiger Schritt, damit auch Vergrößerungen der Schrittweite ermöglicht werden und darf daher auf keinen Fall weggelassen werden.

Formal lassen sich unsere Überlegungen in dem folgenden Grundalgorithmus zusammenfassen.

**Algorithmus 2.39 (Einschrittverfahren mit Schrittweitensteuerung)**

**Eingabe:** Anfangsbedingung $(t_0, x_0)$, Endzeit $T$, Toleranz $tol > 0$, Sicherheitsfaktor $fac$, Einschrittverfahren $\widehat{\Phi}$ und $\Phi$ mit unterschiedlichen Kosistenzordnungen $p \geq \hat{p}+1$, Schrittweitenvorschlag $h_0$ für den ersten Schritt

(1) Setze $\tilde{x}_0 = x_0$, $i = 0$

(2) Falls $t_i = T$, beende den Algorithmus; falls $t_i + h_i > T$, setze $h_i = T - t_i$.

(3) Berechne $t_{i+1} = t_i + h_i$, $\tilde{x}^1_{i+1} = \Phi(t_i, \tilde{x}_i, h_i)$, $\tilde{x}^2_{i+1} = \widehat{\Phi}(t_i, \tilde{x}_i, h_i)$, den Fehlerschätzer $\bar{\varepsilon}$ und den Schrittweitenvorschlag $h_{i,neu}$

(4) Falls $\bar{\varepsilon} > tol$ setze $h_i = h_{i,neu}$ und gehe zu (3)

(5) Falls $\bar{\varepsilon} \leq tol$ setze $\tilde{x}_{i+1} := \tilde{x}^1_{i+1}$, $h_{i+1} := h_{i,neu}$, $i := i+1$ und gehe zu (2)

**Ausgabe:** Werte der Gitterfunktion $\tilde{x}(t_i) = \tilde{x}_i$ in $t_0, \ldots, t_N = T$, □

Beachte, dass wir in (5) die genauere Lösung $\tilde{x}^1_{i+1}$ zum Weiterrechnen und für die Ausgabe verwenden. Diese Praxis wurde früher (und zum Teil noch heute) abgelehnt, da der Fehlerschätzer ja den Fehler in $\tilde{x}^2_{i+1}$ misst. Da das gesamte Verfahren aber auf der Annahme (2.31) beruht, die gerade besagt, dass $\Phi$ (also $\tilde{x}^1_{i+1}$) eine genauere Approximation ist, ist es durchaus gerechtfertigt, diesen Wert zu verwenden.

In der Praxis wird der Algorithmus in mehreren Punkten verfeinert:

(i) Statt in der euklidischen Norm wird $\bar{\varepsilon}$ in der Maximumsnorm

$$\bar{\varepsilon} = \|\bar{\eta}\|_\infty = \max_{i=1,\ldots,n} |\bar{\eta}_i|$$

berechnet, da diese schneller auszuwerten ist.

(ii) Der Bruch $tol/\bar{\varepsilon}$ in der Berechnung der neuen Schrittweite wird durch einen Wert ersetzt, in dem der absolute und der relative Fehler eingeht. Z.B. verwendet man statt $tol/\bar{\varepsilon}$ den Wert $1/err$ mit

$$err = \max_{j=1,\ldots,n} \frac{|\bar{\eta}_j|}{atol + |\widehat{\Phi}_j| \cdot rtol}$$

für absolute und relative Fehlertoleranzen $atol$ und $rtol > 0$; das Fehlerkriterium $\bar{\varepsilon} \leq tol$ wird dabei zu $err \leq 1$. Damit wird bei betragsmäßig großen Lösungskomponenten $|\widehat{\Phi}_j|$ ein größerer Fehler erlaubt, was Probleme mit Rundungsfehlern vermeidet, die bei sehr großen Komponenten ebenfalls groß werden können, weswegen eine rein absolute Fehlertoleranz in diesem Fall nicht einzuhalten wäre.

(iii) Die erlaubte Schrittweite wird durch Schranken $h_{min}$ und $h_{max}$ nach unten und oben beschränkt. Falls für die berechnete Schrittweite $h_{neu} < h_{min}$ gilt, so wird eine Warnung ausgegeben oder mit einer Fehlermeldung abgebrochen.

(iv) Der Variationsfaktor der Schrittweite, der durch

$$\sqrt[\hat{p}+1]{fac\frac{tol}{\bar{\varepsilon}}} \quad \text{bzw. allgemeiner durch} \quad \sqrt[\hat{p}+1]{fac\frac{1}{err}}$$

gegeben ist, wird durch Schranken $fac_{min}$ und $fac_{max}$ nach unten und oben beschränkt. Dadurch werden starke Schwankungen der Schrittweite vermieden.

(v) Im Falle eines Fehlberg–Verfahrens (vgl. Abschnitt 2.7.3) setzt man in Schritt (5) $h_{i+1} = h_i$ falls $h_{i,neu} \approx h_i$. Damit kann man Zwischenergebnisse aus dem $i$–ten Schritt effizient im $i+1$–ten Schritt verwenden.

Einige dieser Punkte werden in der Programmieraufgabe auf dem aktuellen Übungsblatt berücksichtigt; dort ist auch der oben angegebene Algorithmus noch einmal in etwas anderer Form dargestellt.

Abbildung 2.9 zeigt die Anwendung dieses Algorithmus auf das aus den Übungen bekannte restringierte Dreikörperproblem (Satellitenlaufbahn). Die Gitterpunkte $t_i$ sind auf der in Kurvenform dargestellten Lösung markiert. Das Beispiel wurde mit der Routine `ode45` in MATLAB mit $atol = rtol = 10^{-7}$ gerechnet; die Routine verwendet zwei Runge–Kutta–Verfahren der Konsistenzordnung 4 und 5.

### 2.7.3 Eingebettete Verfahren

Die in vielen Beispielen sehr effiziente Schrittweitensteuerung hat den Nachteil, dass man zur Berechnung des Fehlerschätzers zwei Einschrittverfahren $\widehat{\Phi}$ und $\Phi$ in jedem Schritt

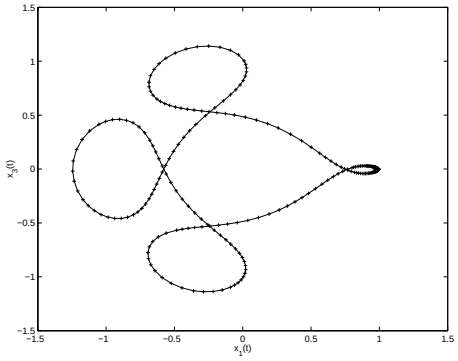



Abbildung 2.9: Adaptive Schrittweitensteuerung an einem Beispiel

auswerten muss. Der Aufwand dieser Auswertungen kann allerdings beträchtlich reduziert werden, wenn man hierfür geschickt gewählte Verfahren verwendet, die sogenannten *eingebetteten Runge–Kutta–Verfahren.*

Wir betrachten zur Erläuterung zwei Verfahren $\widehat{\Phi}$ und $\Phi$ mit Konsistenzordnungen $\hat{p}$ und $p \geq \hat{p} + 1$. Bezeichnen wir die Stufen der Verfahren mit $\hat{k}_i$ bzw. $k_i$, so besteht die Idee der Einbettung einfach darin, dass die Verfahren so konstruiert werden, dass $\hat{k}_i = k_i$ für $i = 1, \ldots, s$ gilt. Für die Koeffizienten der Verfahren muss also $\hat{a}_{ij} = a_{ij}$ und $\hat{c}_i = c_i$ gelten, weswegen wir bei den alten Bezeichnugen $a_{ij}$ und $c_i$ bleiben. Lediglich $\hat{b}_i$ und $b_i$ unterscheiden sich. Ein solches Paar $(\Phi, \widehat{\Phi})$ eingebetteter Verfahren wird mit RK$p(\hat{p})$ bezeichnet. Sie werden in einem Butcher–Tableau der Form

| | | | | | |
|---|---|---|---|---|---|
| $c_1$ | | | | | |
| $c_2$ | $a_{2\,1}$ | | | | |
| $c_3$ | $a_{3\,1}$ | $a_{3\,2}$ | | | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | | |
| $c_s$ | $a_{s\,1}$ | $a_{s\,2}$ | $\cdots$ | $a_{s\,s-1}$ | |
| | $b_1$ | $b_2$ | $\cdots$ | $b_{s-1}$ | $b_s$ |
| | $\hat{b}_1$ | $\hat{b}_2$ | $\cdots$ | $\hat{b}_{s-1}$ | $\hat{b}_s$ |

dargestellt. Um zu zeigen, dass eine solche Einbettung nicht ganz trivial ist, betrachten wir das klassische Runge–Kutta–Verfahren mit Ordnung 4, das durch die Koeffizienten

| | | | | |
|---|---|---|---|---|
| 0 | | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | | |
| 1 | 0 | 0 | 1 | |
| | $\frac{1}{6}$ | $\frac{2}{6}$ | $\frac{2}{6}$ | $\frac{1}{6}$ |

gegeben ist. Wir wollen dieses als Verfahren $\Phi$ der Ordnung $p = 4$ verwenden und versuchen, Koeffizienten $\hat{b}^T = (\hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4)$ zu finden, so dass

$$\widehat{\Phi}(t, x, h) = x + h \sum_{i=1}^{4} \hat{b}_i k_i$$

ein Verfahren $\widehat{\Phi}$ der Ordnung $p = 3$ ergibt, womit wir ein RK4(3)–Verfahren erhalten würden. Wenn man die Bedingungsgleichungen aus Satz 2.25 (iii) löst, so stellt man fest, dass die einzige Lösung durch $\hat{b}^T = (1/6, 1/3, 1/3, 1/6)$ gegeben ist. Wir erhalten damit $\widehat{\Phi} = \Phi$, was keine sinnvolle Lösung ist, da sich die zwei Verfahren in der Konsistenzordnung echt unterscheiden müssen. So paradox es erscheinen mag: Um ein Verfahren niedrigerer Konsistenzordnung zu erhalten, müssen wir eine Stufe hinzunehmen, also $s$ um 1 erhöhen.

Um die Berechnung der nötigen weiteren Stufe (nun wieder mit $k_s$ bezeichnet) möglichst effizient zu gestalten, hilft ein Trick, den E. Fehlberg Ende der 1960er Jahre entwickelt hat: Wir wählen die letze Stufe gerade so, dass

$$k_s = k_1^* \tag{2.34}$$

gilt, wobei $k_1^*$ die erste Stufe des nächsten Schritts des Verfahrens bezeichnet. Damit muss man trotzdem eine Stufe mehr berechnen, kann diese aber speichern und im nächsten Schritt des Verfahrens verwenden, wenn die Schrittweite $h_{i+1} = h_i$ gewählt werden kann (vgl. Punkt (v) in den praktischen Anmerkungen zu Algorithmus 2.39). Ein $s$–stufiges Verfahren mit diesem Trick ist also effektiv ein $s-1$–stufiges Verfahren.

Der Fehlberg–Trick lässt sich in Bedingungen an die Koeffizienten der letzten Stufe $s$ ausdrücken. Wegen Konsistenz und Autonomieinvarianz gilt $k_1 = f(t, x)$, also $k_1^* = f(t + h, \Phi(t, x, h))$. Damit ergibt sich (2.34) zu

$$\underbrace{f(t + c_s h, x + h \sum_{j=1}^{s-1} a_{sj} k_j)}_{=k_s} = \underbrace{f(t + h, x + h \sum_{j=1}^{s} b_j k_j)}_{=k_1^*},$$

was gerade dann der Fall ist, wenn für die Koeffizienten der $s$–ten Stufe die Bedingungen

$$c_s = 1, \quad b_s = 0, \quad a_{sj} = b_j \text{ für } j = 1, \ldots, s-1 \tag{2.35}$$

gelten. Beachte dass es keine Garantie gibt, dass dieser Trick wirklich auf eine sinnvolle Lösung für $\hat{b}$ führt; wenn dies aber gelingt, so liefert er eine sehr effiziente Lösung.

Wir wollen diesen Trick auf das klassische Runge–Kutta–Verfahren anwenden. Dazu müssen wir die unbestimmten Koeffizienten in dem Butcher–Tableau

| | | | | | |
|---|---|---|---|---|---|
| 0 | | | | | |
| $\frac{1}{2}$ | $\frac{1}{2}$ | | | | |
| $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | | | |
| 1 | 0 | 0 | 1 | | |
| $c_5$ | $a_{51}$ | $a_{52}$ | $a_{53}$ | $a_{54}$ | |
| | $\frac{1}{6}$ | $\frac{2}{6}$ | $\frac{2}{6}$ | $\frac{1}{6}$ | 0 |
| | $\hat{b}_1$ | $\hat{b}_2$ | $\hat{b}_3$ | $\hat{b}_4$ | $\hat{b}_5$ |

bestimmen. Aus (2.35) erhalten wir

$$c_5 = 1, \quad a_{51} = a_{54} = \frac{1}{6}, \quad a_{52} = a_{53} = \frac{1}{3}$$

und aus den Bedingungsgleichungen von Satz 2.25 (iii) erhält man die Lösungen

$$\hat{b}^T = (1/6, 1/3, 1/3, 1/6, 0) \quad \text{und} \quad \hat{b}^T = (1/6, 1/3, 1/3, 0, 1/6).$$

Die erste führt wiederum auf $\widehat{\Phi} = \Phi$, die zweite hingegen führt tatsächlich auf ein Verfahren mit maximaler Konsistenzordnung 3. Zusammenfassend erhalten wir

$$\begin{array}{c|ccccc}
0 & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & & \\
1 & 0 & 0 & 1 & & \\
1 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} & \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} & 0 \\
\hline
 & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & 0 & \frac{1}{6}
\end{array}$$

Dieses von Fehlberg Ende der 1960er Jahre entwickelte Verfahren ist für anspruchsvolle numerische DGL–Probleme durchaus schon zu gebrauchen.

Die heutzutage gebräuchlichsten eingebetteten Runge–Kutta–Verfahren wurden allerdings erst Anfang der 1980er Jahren von J.R. Dormand und P.J. Prince entwickelt. Es handelt sich um ein effektiv 6–stufiges RK5(4)–Verfahren mit den Koeffizienten

$$\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{5} & \frac{1}{5} & & & & & & \\
\frac{3}{10} & \frac{3}{40} & \frac{9}{40} & & & & & \\
\frac{4}{5} & \frac{44}{45} & -\frac{56}{15} & \frac{32}{9} & & & & \\
\frac{8}{9} & \frac{19372}{6561} & -\frac{25360}{2187} & \frac{64448}{6561} & -\frac{212}{729} & & & \\
1 & \frac{9017}{3168} & -\frac{355}{33} & \frac{46732}{5247} & \frac{49}{176} & -\frac{5103}{18656} & & \\
1 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & \\
\hline
 & \frac{35}{384} & 0 & \frac{500}{1113} & \frac{125}{192} & -\frac{2187}{6784} & \frac{11}{84} & 0 \\
\hline
 & \frac{5179}{57600} & 0 & \frac{7571}{16695} & \frac{393}{640} & -\frac{92097}{339200} & \frac{187}{2100} & \frac{1}{40}
\end{array}$$

sowie um ein 13–stufiges RK8(7)–Verfahren, das sich z.B. im Abschnitt 5.4 des Buches von Deuflhard/Bornemann findet. Diese Verfahren sind deswegen besonders gut, weil der von $f$ unabhängige Anteil der Konstanten $E$ in der Konsistenzabschätzung für $\Phi$ sehr klein im Vergleich zu anderen Verfahren ist. Das Dormand–Prince–RK5(4)–Verfahren ist MATLABs "Standard–Löser" und ist dort unter dem Namen `ode45` implementiert. Im Internet finden sich MATLAB Implementierungen des RK8(7)–Verfahrens unter dem Namen `ode87.m` (zu finden unter `http://www.mathworks.com/matlabcentral/files/3616/ode87.m` oder mit Google mit dem Suchbegriff `ode87 matlab`).

## 2.8 Extrapolationsverfahren

Die Konstruktion von Runge–Kutta–Verfahren über die in Satz 2.25 angegebenen Bedingungsgleichungen an die Koeffizienten ist i.A. kompliziert und für Konsistenzordnungen $p \geq 10$ kaum durchführbar. Als Alternative gibt es Einschrittverfahren, die mit anderen Methoden hergeleitet und implementiert werden. Ein Beispiel hierfür sind die expliziten Extrapolationsverfahren, die wir in diesem Abschnitt betrachten werden[6]. Tatsächlich liefert die Extrapolationsidee aber nicht etwa eine neue Verfahrensklasse, sondern wieder Runge-Kutta-Verfahren, die allerdings ganz anders implementiert werden. Der Zusammenhang wird auf dem aktuellen Übungsblatt genauer untersucht.

### 2.8.1 Theoretische Grundlagen

Die Extrapolationsverfahren für DGL beruhen auf der gleichen Idee wie die in der Numerik I behandelten Extrapolationsverfahren zur numerischen Integration. Die Grundlage der Verfahren bildet der folgende Satz von Gragg (bewiesen im Jahre 1964, eine frühere Version wurde 1962 von Henrici bewiesen).

**Satz 2.40** Betrachte ein Einschrittverfahren $\Phi$ mit Konvergenzordnung $p$. Wir bezeichnen die zugehörige approximative Lösung mit Anfangsbedingung $(t_0, x_0)$ und äquidistantem Zeitschritt $h > 0$ zur Zeit $t > t_0$ als $\tilde{x}_h(t)$. Dann gilt: Falls das Vektorfeld $f$ und die Abbildung $\Phi$ mindestens $p+k$–mal stetig differenzierbar sind, so existieren stetig differenzierbare Funktionen $e_0, \ldots, e_{k-1} : \mathbb{R} \to \mathbb{R}^n$, so dass die asymptotische Entwicklung

$$\tilde{x}_h(t) = x(t; t_0, x_0) + e_0(t)h^p + \ldots + e_{k-1}(t)h^{p+k-1} + O(h^{p+k}) \tag{2.36}$$

gilt.

Der Beweis, der auf einer geschickt gewählten Taylor–Entwicklung beruht, findet sich in [2, Satz 4.37].

**Bemerkung 2.41** Diese Entwicklung muss für $k \to \infty$ nicht konvergieren, selbst wenn $f$ und $\Phi$ analytisch sind. Für unsere Zwecke sind wir allerdings auch nicht am Verhalten für $k \to \infty$, sondern am Verhalten für festes $k$ und $h \to 0$ interessiert. □

Wir werden uns bei der Beschreibung der Extrapolation auf einen Spezialfall von (2.36) einschränken, bei dem die Konstruktion besonders einfach wird. Wir nehmen dazu an, dass das Verfahren eine asymptotische Entwicklung der Form

$$\tilde{x}_h(t) = x(t; t_0, x_0) + e_0(t)h^p + e_p(t)h^{2p} + \ldots + e_{p(m-2)}(t)h^{p(m-1)} + O(h^{pm}) \tag{2.37}$$

besitzt. Diese Form folgt unter geeigneten Bedingungen aus (2.36), z.B. wenn $p = 1$ ist oder wenn $e_i = 0$ gilt für alle $i$ mit $i \neq lp$ für alle $l \in \mathbb{N}$. Gleichung (2.37) gilt für das

[6] Ein anderes Beispiel sind die impliziten Kollokationsverfahren, die auf dem aktuellen Übungsblatt behandelt werden.

Euler-Verfahren mit $p = 1$; andere Verfahren, die diese Bedingung erfüllen, diskutieren wir am Ende dieses Abschnitts.

Die Idee der Extrapolation ist nun, aus einem weniger genauen Verfahren $\Phi$ und der asymptotischen Entwicklung (2.37) eine genauere Approximation zu erhalten. Die Grundidee verläuft dabei wie folgt:

- Für ein Verfahren $\Phi$ berechnen wir approximative Lösungen $\tilde{x}_{h_i}(t)$ für $x(t; t_0, x_0)$ zur Zeit $t > t_0$ und verschiedenen Schrittweiten $h_1 > \ldots > h_{k+1} > 0$ und erhalten so Wertepaare $(h_i^p, \tilde{x}_{h_i}(t))$, $i = 1, \ldots, k+1$.
- Durch diese Werte legen wir ein Interpolationspolynom $P(h^p)$ und werten dieses in $h^p = 0$ aus. Da $h^p = 0$ *außerhalb* der Stützstellen $h_1^p, \ldots, h_{k+1}^p$ des Polynoms liegt, spricht man von *Extra*polation.

Der folgende Satz zeigt, dass dieses Vorgehen tatsächlich eine Approximation höherer Genauigkeit liefert.

**Satz 2.42** Betrachte ein Einschrittverfahren mit Konvergenzordnung $p$ und asymptotischer Entwicklung (2.37). Dann liefert die oben beschriebene Extrapolation mit $k = m - 1$ eine Approximation $P(0)$ von $x(t; t_0, x_0)$ der Ordnung $O(h_1^{mp})$.

**Beweis:** Wir betrachten zwei Interpolationspolynome

$$\begin{aligned} Q(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k \\ \widetilde{Q}(x) &= \tilde{a}_0 + \tilde{a}_1 x + \tilde{a}_2 x^2 + \ldots + \tilde{a}_k x^k \end{aligned}$$

zu Daten $(x_i, f_i)$ und $(x_i, \tilde{f}_i)$, $i = 0, \ldots, k$. Aus den Lagrange–Darstellungen

$$\begin{aligned} Q(x) &= \sum_{i=0}^{k} L_i(x) f_i \\ \widetilde{Q}(x) &= \sum_{i=0}^{k} L_i(x) \tilde{f}_i \end{aligned}$$

sieht man durch Ausmultiplizieren, dass die Koeffizienten $a_i$ und $\tilde{a}_i$ die Abschätzung

$$|a_i - \tilde{a}_i| \leq C \max_{j=0,\ldots,k} |f_j - \tilde{f}_j|$$

für eine von den $L_i$ abhängige Konstante $C > 0$ erfüllen.

Wir betrachten nun das durch die Daten $(h_i^p, \tilde{x}_{h_i}(t))$, $i = 1, \ldots, m$ definierte Interpolationspolynom

$$P(h^p) = a_0 + a_1 h^p + a_2 (h^p)^2 + \ldots + a_{m-1} (h^p)^{m-1}$$

und vergleichen dieses mit dem durch Abschneiden von (2.37) gewonnenen Polynom

$$\begin{aligned} \widetilde{P}(h^p) &= \tilde{a}_0 + \tilde{a}_1 h^p + \tilde{a}_2 (h^p)^2 + \ldots + \tilde{a}_{m-1} (h^p)^{m-1} \\ &= x(t; t_0, x_0) + e_0(t) h^p + e_p(t) h^{2p} + \ldots + e_{p(m-2)}(t) h^{p(m-1)}. \end{aligned}$$

An den Stützstellen $h_i^p$ unterscheiden sich die Werte dieser Polynome (komponentenweise betrachtet, da $f_i \in \mathbb{R}^n$) um $O(h_i^{mp})$, weswegen sich auch die Komponenten der (vektorwertigen) Koeffizienten $a_0$ und $\tilde{a}_0$ um höchstens

$$C \max_{j=1,\ldots,m} O(h_j^{mp}) = O(h_1^{mp})$$

unterscheiden. Damit folgt

$$P(0) = a_0 = \tilde{a}_0 + O(h_1^{mp}) = x(t; t_0, x_0) + O(h_1^{mp}),$$

also die Behauptung. □

### 2.8.2 Algorithmische Umsetzung

Die im vorherigen Abschnitt skizzierte Extrapolationsidee kann ganz analog zur Integration in der Numerik I als Diagonalschema implementiert werden. Dieses Schema ermöglicht die iterative Berechnung von $P(0)$ für eine wachsende Anzahl von Stützstellen, ohne dass wir diese Polynome explizit aufstellen müssen.

Wir wählen dazu eine aufsteigende Folge $n_i \in \mathbb{N}$ und setzen $h_i = (t - t_0)/n_i$. Wir bezeichnen die mit dem Verfahren $\Phi$ zur Anfangsbedingung $(t_0, x_0)$ und Zeitschritt $h_i$ erhaltenen Lösungen mit $T_{i,1} = \tilde{x}_{h_i}(t)$. Mit

$$P_{i,k}(h^p)$$

bezeichnen wir die durch die Stützstellen $(h_{i-k+1}^p, T_{i-k+1,1}), \ldots, (h_i^p, T_{i,1})$ definierten Interpolationspolynome in $h^p$ und mit $T_{i,k} = P_{i,k}(0)$ ihre Werte in $h^p = 0$. Gemäß Satz 2.42 liefern die Diagonalwerte $T_{k,k}$ also eine Approximation der Ordnung $O(h_1^{kp})$ für die Lösung $x(t; t_0, x_0)$ zur Zeit $t$. Das folgende Lemma zeigt, wie die Werte $T_{i,k}$ iterativ berechnet werden können.

**Lemma 2.43** Für die Werte $T_{i,k}$ gilt die Rekursionsformel

$$T_{i,k} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left(\frac{h_{i-k+1}}{h_i}\right)^p - 1}, \quad k = 2, 3, \ldots; \;\; i = k, k+1, \ldots$$

**Beweis:** Durch Nachprüfen der Interpolationseigenschaft sieht man leicht, dass für die Interpolationspolynome $P_{i,k}$ die Gleichung

$$P_{i,k}(h^p) = \frac{(h_{i-k+1}^p - h^p) P_{i,k-1}(h^p) - (h_i^p - h^p) P_{i-1,k-1}(h^p)}{h_{i-k+1}^p - h_i^p}$$

gilt, die auch als *Lemma von Aitken* bekannt ist. Damit folgt die oben angegebene Formel durch Auswerten in $h^p = 0$ und Kürzen des Bruchs mit $h_i^p$. □

Die Berechnung, die als *Extrapolationsschema* bezeichnet wird, lässt sich ganz analog zur Integration in der Numerik I grafisch darstellen, siehe Abb. 2.10 (vgl. auch Abb. 5.1 im Skript zur Vorlesung Numerik I).

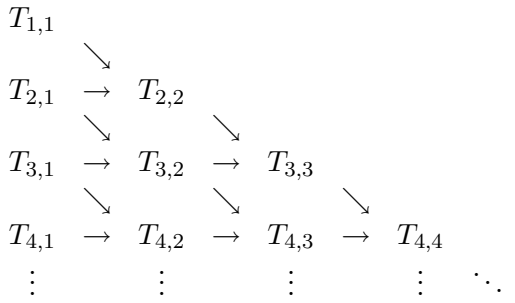


Abbildung 2.10: Illustration des Extrapolationsschemas

In der Praxis verwendet man zur Berechnung der $h_i$ oft die naheliegendste Folge $n_i = (1, 2, 3, 4, 5, 6, \ldots)$. Eine weitere Möglichkeit ist die Halbierung der Schrittweite, also die Folge $n_i = (1, 2, 4, 8, 16, \ldots)$.

Natürlich will man auch mit Extrapolationsverfahren i.A. nicht nur *einen* Wert $\tilde{x}(t)$ sondern eine Gitterfunktion $\tilde{x}(t_i)$ auf einem Zeitgitter $(t_i)_{i\in\mathbb{N}}$ auf $[t_0, T]$ berechnen. Wenn wir eine gewünschte Extrapolationsordnung $kp$ fixieren und das Verfahren mit $t = h$ anwenden, so kann man mittels $\Phi_E(t_0, x_0, h) := T_{k,k}$ ein neues Einschrittverfahren definieren, mit dem sich in üblicher Form die gewünschte Gitterfunktion berechnen lässt. Die Schrittweitensteuerung ist hier besonders effizient zu implementieren, da man mit $T_{k,k-1}$ und $T_{k-1,k-1}$ gleich zwei Approximation niedrigerer Ordnung zur Verfügung hat, ohne weitere Berechnungen durchführen zu müssen. In der Praxis wählt man üblicherweise $\widehat{\Phi}_E(t_0, x_0, h) := T_{k,k-1}$ für die Fehlerschätzung, da dieser Ausdruck eine genauere Approximation liefert (vgl. auch die Diskussion der adaptiven Romberg–Quadratur).

Wir wollen abschließend untersuchen, welche Einschrittverfahren sich als Basisverfahren $\Phi$ der Extrapolation eignen. Betrachtet man die zu Grunde liegende asymptotische Entwicklung (2.37), so sieht man (aus Satz 2.40), dass das Euler–Verfahren die Voraussetzung für $p = 1$ erfüllt. Dieses Verfahren kann also als Basis der Extrapolation verwendet werden.

Effizienter wäre es aber sicherlich, wenn wir ein Verfahren $\Phi$ verwendeten, welches (2.37) für $p > 1$ erfüllt, da wir mit jedem Extrapolationsschritt die Ordnung um den Faktor $p$ erhöhen. Leider kann man nachweisen, dass es kein explizites Runge–Kutta–Verfahren $\Phi$ gibt, für das dieses gilt.

Wir wollen den Fall $p = 2$ genauer untersuchen. Hier lässt sich ein Kriterium angeben, unter dem (2.37) gilt, wobei wir annehmen, dass das betrachtete Einschrittverfahren von der Form $\Phi(t, x, h) = x + h\varphi(t, x, h)$ ist, vgl. Lemma 2.10.

**Satz 2.44** Falls das Einschrittverfahren *reversibel* ist, d.h. die Bedingung

$$\Phi(t + h, \Phi(t, x, h), -h) = x \tag{2.38}$$

erfüllt und die Konsistenz– und Konvergenzordnung $p = 2$ besitzt, so existiert für hinreichend glattes Vektorfeld $f$ eine asymptotische Entwicklung der Form (2.37) mit $p = 2$.

**Beweisskizze:** Betrachte die exakte Lösung $x(t)$ und die $e_i(t)$, $i = 0, 1, 2, \ldots$ aus Satz 2.40. Für ein beliebiges $k \in \mathbb{N}$ definieren wir

$$x^*(t) = x(t) + e_0(t)h^2 + e_1(t)h^3 + \ldots + e_{k-1}(t)h^{2+k-1}.$$

Mittels Taylor–Entwicklung von $\Phi(t, x^*(t), h)$ nach $x$ und $h$ im Punkt $(t, x(t), 0)$ sieht man, dass dann eine differenzierbare Funktion $d_k(t)$ existiert, so dass die Gleichungen

$$\begin{aligned} x^*(t+h) - \Phi(t, x^*(t), h) &= d_k(t)h^{2+k+1} + O(h^{2+k+2}) \\ x^*(t) - \Phi(t+h, x^*(t+h), -h) &= d_k(t+h)(-h)^{2+k+1} + O(h^{2+k+2}) \end{aligned}$$

gelten. Durch Koeffizientenvergleich erhält man dabei die Gleichung

$$d_k(t) = e_k(t),$$

wobei $e_k$ der Koeffizient aus Satz 2.40 für $k = i$ ist. Da $d_k$ differenzierbar ist, folgt

$$d_k(t+h)h^{2+k+1} = d_k(t)h^{2+k+1} + O(h^{2+k+2})$$

und da $\Phi$ differenzierbar ist auch

$$\begin{aligned} &\Phi(t+h, x^*(t+h) + d_k(t)h^{2+k+1}, -h) \\ &\quad = x^*(t+h) + d_k(t)h^{2+k+1} - h\varphi(t+h, x^*(t+h) + d_k(t)h^{2+k+1}, -h) \\ &\quad = x^*(t+h) + d_k(t)h^{2+k+1} - h\varphi(t+h, x^*(t+h), -h) + O(h^{2+k+2}). \end{aligned}$$

Aus der Reversibilität von $\Phi$ folgt damit

$$\begin{aligned} x^*(t) &= \Phi(t, \Phi(t, x^*(t), h), -h) \\ &= \Phi(t, x^*(t+h) - d_k(t)h^{2+k+1} + O(h^{2+k+2}), -h) \\ &= \Phi(t, x^*(t+h), -h) - d_k(t)h^{2+k+1} + O(h^{2+k+2}) \\ &= x^*(t) - d_k(t+h)(-h)^{2+k+1} - d_k(t)h^{2+k+1} + O(h^{2+k+2}) \\ &= x^*(t) + ((-1)^{2+k} - 1)d_k(t)h^{2+k+1} + O(h^{2+k+2}). \end{aligned}$$

Da dies für alle $h > 0$ gelten muss, folgt $((-1)^{2+k} - 1)d_k(t)h^{2+k+1} = 0$. Falls $k$ ungerade ist, folgt damit $d_k = 0$, also $e_k(t) = 0$. Damit erhalten wir (2.37) für $p = 2$. □

Leider ist Reversibilität eine Eigenschaft, die kein explizites Runge–Kutta–Verfahren besitzt. Wir haben im Beweis von Lemma 2.23 gesehen, dass jede explizite Runge–Kutta–Approximation der Gleichung $\dot{x}(t) = x(t)$ mit $x(0) = x_0$ für alle $t \in \mathbb{R}$ von der Form $\Phi(t, x_0, h) = P(h)x_0$ für ein Polynom in $h$ ist. Die Bedingung (2.38) würde $P(h)P(-h) = 1$ erzwingen, was für nichtkonstante Polynome unmöglich ist.

Als Ausweg müssen wir eine andere Klasse von Verfahren betrachten. Hier kann man z.B. ein Verfahren $\Phi$ verwenden, das als *explizite Mittelpunktregel* bekannt ist und für äquidistante Stützstellen $h_i = h$ durch

$$\tilde{x}(t_{i+2}) = \tilde{x}(t_i) + 2hf(t_{i+1}, \tilde{x}(t_{i+1}))$$

gegeben ist. Dieses Verfahren ist reversibel, wenn man es als Abbildung von $\tilde{x}(t_i)$ nach $\tilde{x}(t_{i+2})$ auffasst.

Allerdings ist dies kein Einschrittverfahren, da die rechte Seite von $\tilde{x}(t_{i+1})$ und $\tilde{x}(t_i)$ abhängt. Wir haben es hier mit einem sogenannten *Mehrschrittverfahren* zu tun, einer Klasse von Verfahren, die wir im Folgenden systematisch untersuchen wollen. Um das Verfahren zu starten, benötigen wir neben dem Anfangswert $\tilde{x}(t_0) = x_0$ noch den Wert $\tilde{x}(t_1)$, der (um die Konvergenzordnung $p = 2$ zu erhalten) mindestens mit Genauigkeit $O(h^2)$ bestimmt werden muss. Dies kann durch einen einfachen Euler–Schritt, also mittels $\tilde{x}(t_1) = x_0 + hf(t_0, x_0)$ geschehen.

## 2.9 Mehrschrittverfahren

Die Mehrschrittverfahren unterscheiden sich von den Einschrittverfahren dadurch, dass der Wert $\tilde{x}(t_{i+1})$ nicht nur von $\tilde{x}(t_i)$ sondern von einer ganzen Reihe von Vorgängerwerten $\tilde{x}(t_{i-k+1}), \ldots, \tilde{x}(t_i)$ abhängt. Wie schon bei den Einschrittverfahren gibt es explizite und implizite Mehsrchrittverfahren; erstere geben einen expliziten Ausdruck für $\tilde{x}(t_{i+1})$, während bei letzteren noch eine Fixpunktgleichung zu lösen ist. Man hofft dabei, dass man – da ja durch die größere Anzahl von Punkten mehr Information zur Verfügung steht – im Vergleich zu Einschrittverfahren gleicher Konsistenzordnung mit weniger Auswertungen von $f$ pro Schritt auskommt. Tatsächlich werden wir sehen, dass diese Hoffnung berechtigt ist.

Zur Motivation betrachten wir wieder Verfahren, die wir heuristisch aus numerischen Integrationsformeln ableiten. Wir nehmen dabei konstante Schrittweite $h_i = h$ an. Wenn wir in der Integralgleichung

$$x(t_{i+1}) = x(t_{i-1}) + \int_{t_{i-1}}^{t_{i+1}} f(t, x(t))dt$$

das Integral durch die Mittelpunktregel

$$\int_{t_{i-1}}^{t_{i+1}} f(t, x(t))dt \approx 2hf(t_i, x(t_i))$$

ersetzen, so erhalten wir die im letzten Abschnitt bereits betrachtete *explizite Mittelpunktregel*

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_{i-1}) + 2hf(t_i, \tilde{x}(t_i)).$$

Wählen wir die Simpson–Regel

$$\int_{t_{i-1}}^{t_{i+1}} f(t, x(t))dt \approx \frac{h}{3}\Big(f(t_{i+1}, x(t_{i+1})) + 4f(t_i, x(t_i)) + f(t_{i-1}, x(t_{i-1}))\Big),$$

so erhalten wir das (implizite) *Milne–Simpson–Verfahren*

$$\tilde{x}(t_{i+1}) = \tilde{x}(t_{i-1}) + \frac{h}{3}(f(t_{i+1}, \tilde{x}(t_{i+1})) + 4f(t_i, \tilde{x}(t_i)) + f(t_{i-1}, \tilde{x}(t_{i-1}))).$$

Eine Verallgemeinerung, die diese beiden Verfahren umfasst, ist die folgende Klasse der *linearen Mehrschrittverfahren (MSV).*

**Definition 2.45** Ein *k–stufiges lineares Mehrschrittverfahren (MSV)* ist gegeben durch die Gleichung

$$\begin{aligned} &a_k\tilde{x}(t_{i+k}) + a_{k-1}\tilde{x}(t_{i+k-1}) + \ldots + a_0\tilde{x}(t_i) \\ &\quad = \quad h\Big(b_k\tilde{f}(t_{i+k}) + b_{k-1}\tilde{f}(t_{i+k-1}) + \ldots + b_0\tilde{f}(t_i)\Big) \end{aligned} \tag{2.39}$$

mit der Abkürzung $\tilde{f}(t_j) = f(t_j, \tilde{x}(t_j))$, wobei $a_k \neq 0$ ist □

Mit dieser Klasse von Verfahren wollen wir uns schwerpunktmäßig beschäftigen. Wenn $b_k = 0$ ist, so ist das Verfahren explizit, da es direkt nach $\tilde{x}(t_{i+k})$ aufgelöst werden kann. Falls $b_k \neq 0$ ist, so kann man die entstehenden Gleichungen analog zu den impliziten Einschrittverfahren lösen (algebraisch, Fixpunkt–Iteration, Newton–Verfahren,...). Wir beschränken uns zunächst auf den Fall äquidistanter Schrittweiten $h_i = h$ und gehen am Schluss dieses Abschnittes (kurz) auf variable Schrittweiten und Schrittweitensteuerung ein.

**Bemerkung 2.46** (i) Zum Start eines Mehrschrittverfahrens benötigt man neben dem Anfangswert $\tilde{x}(t_0)$ noch die Werte $\tilde{x}(t_1), \ldots, \tilde{x}(t_{k-1})$. Diese werden üblicherweise durch ein geeignetes Einschrittverfahren bestimmt. Details dazu besprechen wir etwas später.
(ii) Wenn man die $\tilde{f}$–Werte eines Schrittes zwischenspeichert, so muss in jedem Schritt lediglich der Wert $\tilde{f}(t_{i+k-1})$ neu berechnet werden. Ein explizites lineares MSV kommt also mit einer $f$–Auswertung pro Schritt aus. □

Zur Analyse von MSV hat sich der folgende (aus der Theorie der dynamischen Systeme stammende) Formalismus als sehr geeignet erwiesen.

**Definition 2.47** Auf dem Raum der Gitterfunktionen $\Delta_{\mathcal{T}} := \{f : \mathcal{T} \to \mathbb{R}^n\}$ definieren wir den *Shift–Operator* $E : \Delta_{\mathcal{T}} \to \Delta_{\mathcal{T}}$ mittels

$$E(f)(t_i) = f(t_{i+1}).$$

Hierbei erweitern wir unser Gitter formal zu einem Gitter mit unendlich vielen Gitterpunkten $\mathcal{T} = \{t_0, t_1, t_2, \ldots\}$. □

**Beispiel 2.48** Für eine Gitterfunktion mit $f(t_i) = a_i$ mit $a_i = (2, 4, 8, 16, 32, \ldots)$ gilt also $E(f) = \tilde{f}$ mit $\tilde{f}(t_i) = \tilde{a}_i$ mit $\tilde{a}_i = (4, 8, 16, 32, 64, \ldots)$. Die Wertefolge wird also um eine Stelle nach links verschoben, woraus sich der Name Shift–Operator (manchmal auch 'Linksshift' genannt) ergibt. □

Der Shift–Operator erlaubt die folgende, sehr kompakte Schreibweise von Mehrschrittverfahren: Mit den Polynomen

$$\begin{aligned} P_a(z) &= a_0 + a_1 z + \ldots + a_k z^k \\ P_b(z) &= b_0 + b_1 z + \ldots + b_k z^k \end{aligned}$$

kann man (2.39) als

$$P_a(E)(\tilde{x})(t_i) = h P_b(E)(\tilde{f})(t_i) \tag{2.40}$$

schreiben, wobei die Potenz $E^j$ des Shift–Operators die $j$–malige Hintereinanderausführung des Operators bedeutet.

Wir wollen nun die Konvergenz von Mehrschrittverfahren untersuchen und dabei das für die Einschrittverfahren bewiesene Resultat "Konsistenz + Lipschitzbedingung ⇒ Konvergenz" verallgemeinern. Wir beginnen mit der Konsistenz.

### 2.9.1 Konsistenz

Bei der Untersuchung der Konsistenz bei Einschrittverfahren haben wir mittels

$$\varepsilon := \|\Phi(t,x,h) - x(t+h;t,x)\|$$

den Konsistenzfehler durch Vergleich des numerischen Verfahrens mit der exakten Lösung erhalten. Die Größe $\varepsilon$ lässt sich aber auch anders interpretieren:
Für die numerisch berechnete Gitterfunktion gilt gerade die Gleichung

$$0 = \|\tilde{x}(t_{i+1}) - \Phi(t_i, \tilde{x}(t_i), h)\|$$

Setzen wir hier nun die exakte Lösungsfunktion $x(t) = x(t;t_0,x_0)$ ein, so erhalten wir

$$\|x(t_{i+1}) - \Phi(t_i, x(t_i), h)\| = \varepsilon,$$

also gerade wieder unseren Konsistenzfehler für $x = x(t_i)$. Beachte, dass jede Funktion $x : [t_0, T] \to \mathbb{R}^n$ auch eine Gitterfunktion auf den in $[t_0, T]$ liegenden Gitterpunkten ist.

Dieses Verfahren "Einsetzen der exakten Lösung in die numerische Gleichung" lässt sich auf viele numerische Verfahren anwenden, z.B. auf unsere Mehrschrittverfahren. In der kompakten Schreibweise (2.40) müssen wir also die Norm des Konsistenzfehlers

$$L(x,t,h) = P_a(E)(x)(t) - hP_b(E)(f)(t) = P_a(E)(x)(t) - hP_b(E)(\dot{x})(t)$$

bestimmen. Beachte, dass der Parameter $x$ hier eine Funktion $x : [t_0, T] \to \mathbb{R}^n$ und dass $L$ nur für solche Parametertripel $(x,t,h)$ definiert ist, für die $[t, t+hk] \subset [t_0, T]$ gilt.

**Definition 2.49** Ein lineares Mehrschrittverfahren besitzt die *Konsistenzordnung* $p$, falls für jede $p+1$–mal stetig differenzierbare Lösung $x : [t_0, T] \to \mathbb{R}^n$ der Differentialgleichung (2.1) die Abschätzung

$$L(x,t,h) = O(h^{p+1})$$

gleichmäßig in $t$ gilt für alle $t, h$, in denen $L(x,t,h)$ definiert ist. □

Interessanterweise hängt die Definition des Konsistenzfehlers $L$ *nicht* von $f$ ab, da wir die auftretenden Werte des Vektorfeldes $f$ durch die Ableitungen $\dot{x}$ ersetzt haben. Dies nutzt der folgende Satz aus, der Bedingungen angibt, anhand derer man die Konsistenzordnung eines Mehrschrittverfahrens überprüfen kann.

**Satz 2.50** Ein lineares Mehrschrittverfahren besitzt genau dann die Konsistenzordnung $p \in \mathbb{N}$, wenn eine der folgenden äquivalenten Bedingungen erfüllt ist.

(i) Für jede beliebige $p+1$–mal stetig differenzierbare Funktion $x : [t_0, T] \to \mathbb{R}^n$ gilt die Abschätzung

$$L(x,t,h) = O(h^{p+1})$$

gleichmäßig in $t$ für alle $t, h$, in denen $L(x,t,h)$ definert ist.

(ii) $L(Q, 0, h) = 0$ für alle Polynome $Q \in \mathcal{P}_p$.

(iii) Es gilt

$$\sum_{j=0}^{k} a_j = 0, \ \sum_{j=0}^{k} a_j j^l = l \sum_{j=0}^{k} b_j j^{l-1} \text{ für } l = 1, \ldots, p$$

mit der Konvention $0^0 = 1$.

**Beweis:** Wir zeigen die Äquivalenz durch die Implikationen

$$\text{(i)} \Rightarrow \text{Konsistenzordnung } p \Rightarrow \text{(ii)} \Rightarrow \text{(i)} \Rightarrow \text{(iii)} \Rightarrow \text{(i)}$$

"(i) $\Rightarrow$ Konsistenzordnung $p$": Dies folgt direkt, da mit jeder beliebigen Funktion auch jede Lösung die behauptete Abschätzung erfüllt.

"Konsistenzordnung $p \Rightarrow$ (ii)": Gegeben sei ein beliebiges Polynom $Q \in \mathcal{P}_p$. Mit $f(t,x) = \dot{Q}(t)$ erhalten wir eine "triviale" Differentialgleichung, deren Lösung $Q$ ist. Nach Definition der Konsistenzordnung folgt also

$$L(Q, 0, h) = O(h^{p+1}).$$

Da $Q$ ein Polynom vom Grad $\leq p$ ist, muss auch $L(Q,0,h)$ ein Polynom vom Grad $\leq p$ in $h$ sein, weswegen $L(Q,0,h) = 0$ sein muss.

"(ii) $\Rightarrow$ (i)": Sei $x$ eine beliebige $p+1$–mal differenzierbare Funktion und sei Sei $Q \in \mathcal{P}_p$ das Polynom, das durch die ersten $p$ Terme der Taylorentwicklung von $x$ in $t^*$ definiert ist. Dann gilt

$$x(t) = Q(t) + O(h^{p+1}) \text{ für alle } t \in [t^* - h, t^* + h].$$

Aus der Struktur von $L$ folgt damit sofort die Abschätzung

$$L(x, t, h) = L(Q, t, h) + O(h^{p+1}).$$

Diese Abschätzung ist gleichmäßig in $t \in [t_0, T]$, da das den $O(h^{p+1})$–Term bestimmende Taylor–Restglied gleichmäßig beschränkt auf kompakten Intervallen ist. Aus (ii) wissen wir, dass $L(Q,0,h) = 0$ gilt, woraus (durch "Verschieben" des Polynoms) auch $L(Q,t,h) = 0$ folgt, was schließlich die Behauptung liefert.

"(i) $\Rightarrow$ (iii)": Die Implikation aus (i) gilt insbesondere für konstante Funktionen $x \equiv c$. Für diese gilt

$$O(h^{p+1}) = L(x,0,h) = P_a(E)x(t) - \underbrace{hP_b(E)\dot{x}(t)}_{=0} = \sum_{j=0}^{k} a_j c.$$

Da die rechte Seite unabhängig von $h$ ist, kann dies nur gelten, wenn die Summe der $a_j$ gleich Null ist, was die erste Gleichung in (iii) zeigt.

Für die weiteren Gleichungen in (iii) betrachten wir (i) mit $x(t) = \exp(t)$. Wegen

$$E^j(\exp)(0) = \exp(jh) = \exp(h)^j \quad \text{und} \quad \frac{d}{dt}\exp(t) = \exp(t)$$

folgt

$$L(\exp, 0, h) = P_a(\exp(h)) - hP_b(\exp(h))$$

Wir betrachten die Taylorentwicklung dieses Ausdrucks in $h = 0$. Diese lautet

$$L(\exp, 0, h) = \sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} a_j j^l h^l - \sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^{k} b_j j^l h^{l+1} + O(h^{p+1}).$$

Aus (i) wissen wir $L(\exp, 0, h) = O(h^{p+1})$, weswegen

$$\sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} a_j j^l h^l - \sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^{k} b_j j^l h^{l+1} = O(h^{p+1})$$

sein muss. Dieser Summenausdruck ist ein Polynom vom Grad $\leq p$ in $h$, und kann daher nur von der Ordnung $O(h^{p+1})$ sein, wenn er bereits Null ist. Dies wiederum kann nur dann gelten, wenn sich die Koeffizienten zu gleichen Potenzen von $h$ zu Null addieren, also

$$\frac{1}{l!} \sum_{j=0}^{k} a_j j^l - \frac{1}{(l-1)!} \sum_{j=0}^{k} b_j j^{l-1} = 0$$

gilt. Dies sind gerade die weiteren Gleichungen aus (iii).

"(iii) $\Rightarrow$ (i)": Die Taylorentwicklung von $L$ für allgemeine $x$ in $h = 0$ lautet

$$\begin{aligned} L(x, t, h) &= \sum_{l=0}^{p} \frac{1}{l!} \sum_{j=0}^{k} a_j j^l h^l x^{(l)}(t) \\ &\quad - h \left( \sum_{l=0}^{p-1} \frac{1}{l!} \sum_{j=0}^{k} b_j j^l h^l x^{(l+1)}(t) \right) + O(h^{p+1}). \end{aligned}$$

Wenn die Gleichungen aus (iii) gelten, so fallen alle diese Summanden weg, so dass nur $O(h^{p+1})$ übrig bleibt. Diese Abschätzung ist wegen der gleichmäßigen Beschränktheit des Taylor–Restgliedes gleichmäßig in $t \in [t_0, T]$, weswegen (i) folgt. □

**Bemerkung 2.51** Der Fall $p = 1$ ist hierbei besonders interessant, da er die Frage beantwortet, wann ein Verfahren überhaupt konsistent ist. Für $p = 1$ erhalten wir aus (iii) die Bedingungen

$$\sum_{j=0}^{k} a_j = 0 \quad \text{und} \quad \sum_{j=0}^{k} a_j j = \sum_{j=0}^{k} b_j.$$

Beide Bedingungen lassen sich mit Hilfe der Polynome $P_a$ und $P_b$ ausdrücken, sie sind gerade äquivalent zu

$$P_a(1) = 0 \quad \text{und} \quad P_a'(1) = P_b(1).$$

Diese Bedingungen entsprechen der Bedingung $\sum b_i = 1$ bei den Runge–Kutta–Verfahren. Insbesondere muss für konsistente Verfahren die 1 eine Nullstelle von $P_a$ sein. Wir werden im nächsten Teilabschnitt sehen, dass auch die weiteren Nullstellen von $P_a$ eine wichtige Rolle bei der Konvergenzanalyse von Mehrschrittverfahren spielen. □

### 2.9.2 Stabilität

Wir wollen nun ein geeignetes Analogon der Lipschitzbedingung für Einschrittverfahren entwickeln. In der Konvergenztheorie der Einschrittverfahren haben wir diese Bedingung verwendet, um sicher zu stellen, dass sich die in vergangenen Schritten gemachten Fehler im aktuellen Schritt nicht zu sehr verstärken.

Sicherlich sollte die rechte Seite unseres Mehrschrittverfahrens (2.39) eine ähnliche Lipschitzbedingung erfüllen, diese erhalten wir aber "geschenkt", da wir ja nur Lipschitz–stetige Vektorfelder $f$ betrachten. Leider reicht es aber nicht aus, wenn $f$ Lipschitz–stetig ist. Diese Bedingung besagt ja nur, dass sich kleine Fehler in den vergangenen $\tilde{x}$ in der *rechten* Seite unseres Verfahrens wenig auswirken. Wir benötigen zusätzlich noch eine Bedingung, die uns garantiert, dass kleine Fehler auf der *linken* Seite von (2.39) auch nur kleine Fehler in $\tilde{x}(t_{i+k})$ hervorrufen.

Um zu sehen, dass dies ein nichttriviales Problem ist, betrachten wir zwei Mehrschrittverfahren, die wir auf das Anfangswertproblem

$$\dot{x}(t) = 0, \qquad x(0) = 0 \tag{2.41}$$

anwenden. Da die rechte Seite in (2.39) wegen $f \equiv 0$ verschwindet, reicht es, die Koeffizienten $a_i$ anzugeben. Wir betrachten nun die Verfahren mit

$$a_2 = 1,\, a_1 = -3,\, a_0 = 2 \quad \text{und} \quad \tilde{a}_2 = 1,\, \tilde{a}_1 = -3/2,\, \tilde{a}_0 = 1/2. \tag{2.42}$$

Man sieht leicht, dass beide Verfahren wegen $\sum a_i = 0$ bzw. $\sum \tilde{a}_i = 0$ konsistent sind. Für die DGL (2.41) ergeben sich daraus die Iterationsvorschriften

$$\tilde{x}(t_{i+1}) = -a_1\tilde{x}(t_i) - a_0\tilde{x}(t_{i-1}) = 3\tilde{x}(t_i) - 2\tilde{x}(t_{i-1}) \tag{2.43}$$

und

$$\tilde{x}(t_{i+1}) = -\tilde{a}_1\tilde{x}(t_i) - \tilde{a}_0\tilde{x}(t_{i-1}) = 3/2\tilde{x}(t_i) - 1/2\tilde{x}(t_{i-1}). \tag{2.44}$$

Man sieht leicht, dass beide Verfahren für exakte Startwerte $\tilde{x}(t_0) = \tilde{x}(t_1) = 0$ die exakte Lösung $\tilde{x}(t_i) \equiv 0$ liefern. Wenn wir in den Startwert $\tilde{x}(t_1)$ allerdings leicht stören, so unterscheidet sich das Verhalten der beiden Verfahren erheblich. Abbildung 2.11 zeigt das unterschiedliche Verhalten für $\tilde{x}(t_0) = 0$ und den (nur ganz leicht gestörten Wert) $\tilde{x}(t_1) = 10^{-12}$.

Offenbar reproduziert das zweite Verfahren die exakte konstante Lösung trotz der kleinen Störung in $\tilde{x}(t_1)$ gut, während das erste Verfahren nach nur etwa 35 Schritten riesige Fehler produziert.

Wir wollen nun untersuchen, warum dies so ist und wie man erkennen kann, ob ein Mehrschrittverfahren stabil gegenüber solchen kleinen Fehlern ist. Wegen der Linearität der linken Seite des Verfahrens genügt es, dazu das einfache Anfangswertproblem (2.41) zu betrachten (später im Beweis der Konvergenz werden wir genauer sehen, warum). Aus (2.39) folgt sofort, dass für (2.41) mit $\tilde{x}(t_0) = \ldots = \tilde{x}(t_{k-1}) = 0$ die Gleichung $\tilde{x} \equiv 0$ gilt, d.h. die exakte Lösung wird ohne Fehler reproduziert, falls die Startwerte exakt sind. Wie im obigen Beispiel betrachten wir nun den Fall, dass die bis zum Schritt $i^* \in \mathbb{N}$ erhaltenen Werte $\tilde{x}(t_i)$, $i = 0, \ldots, i^*$ durch Rechenfehler etwas gestört sind, wobei $\|\tilde{x}(t_i)\| \leq \varepsilon$ gelte.

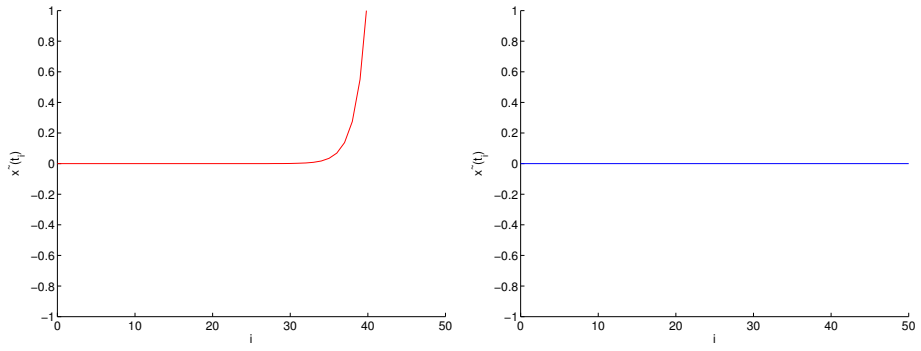



Abbildung 2.11: MSV (2.43) (links) und (2.44) (rechts) mit $\tilde{x}(t_0) = 0$, $\tilde{x}(t_1) = 10^{-12}$

Für kleines $\varepsilon > 0$ sollten nun auch die nachfolgenden Werte $\tilde{x}(t_j)$, $j \geq i^*$ nur leicht gestört werden. Sicherlich kann man das nicht für alle Zeiten verlangen, aber doch zumindest auf vorgegebenen kompakten Zeitintervallen. Eine vernünftige Bedingung an das Verfahren für $f \equiv 0$ wäre also

$$\|\tilde{x}(t_i)\| \leq \varepsilon \text{ für } i = 0, \ldots, i^* \quad \Rightarrow \quad \|\tilde{x}(t_j)\| \leq C\varepsilon \text{ für alle } t_j \in [t_{i^*}, T].$$

Die wesentliche Beobachtung ist nun, dass zwar die Werte $\tilde{x}$ unabhängig von der Schrittweite $h$ sind (dies ist gerade der entscheidende Unterschied zwischen der *linken* und der *rechten* Seite von (2.39)), nicht aber die Bedingung $t_j \in [t_{i^*}, T]$, die im Gegenteil stark von $h$ abhängt: Je kleiner $h$ wird, desto mehr Gitterpunkte $t_j$ liegen in diesem Intervall. Da $h$ beliebig klein werden kann, wird jeder $t_j$–Wert also für geeignetes $h$ in $[t_{i^*}, T]$ liegen, weswegen man die Schranke $\|\tilde{x}(t_j)\| \leq C\varepsilon$ tatsächlich für alle $j \geq i^*$ fordern muss. Dies führt auf die folgende Definition, in der wir die jeweils die $k$ Werte, die im Verfahren in Schritt $i$ verwendet werden, gemeinsam betrachten.

**Definition 2.52** Ein lineares Mehrschrittverfahren heißt *stabil*, falls ein $C > 0$ existiert, so dass für jeden Vektor $\tilde{x}^0 := (\tilde{x}(t_0), \ldots, \tilde{x}(t_{k-1}))^T$ von (reellen) Anfangswerten und alle $i \in \mathbb{N}$ die Ungleichung

$$\left\| \begin{pmatrix} \tilde{x}(t_i) \\ \vdots \\ \tilde{x}(t_{i+k-1}) \end{pmatrix} \right\| \leq C\|\tilde{x}^0\|$$

gilt. Hierbei ist die Folge $\tilde{x}(t_i)$ durch (2.39) bzw. (2.40) mit $\tilde{f}(t_i) = 0$ definiert, also kompakt geschrieben als

$$P_a(E)(\tilde{x})(t_i) = 0 \tag{2.45}$$

oder explizit ausgeschrieben als

$$\tilde{x}(t_{i+k}) = -\frac{a_{k-1}}{a_k}\tilde{x}(t_{i+k-1}) - \ldots - \frac{a_0}{a_k}\tilde{x}(t_i). \tag{2.46}$$

□

Wir werden nun ein einfaches Kriterium herleiten, das uns sagt, ob ein gegebenes Verfahren stabil ist. Hierzu stellen wir die Gleichung (2.46) zunächst in etwas anderer Form dar. Wir erinnern dazu an die linearen Differenzengleichungen (2.28), die durch eine Iterationsvorschrift der Form

$$x(t_{i+1}) = Ax(t_i)$$

mit einer Matrix $A \in \mathbb{R}^{k\times k}$ gegeben sind. Eine solche Gleichung heißt *stabil*, falls die Ungleichung

$$\|x(t_i)\| \leq C\|x(t_0)\|$$

für ein $C > 0$ und alle $i \in \mathbb{N}$ gilt. Das folgende Lemma zeigt, wie sich (2.46) als eine Matrix–Differenzengleichung schreiben lässt.

**Lemma 2.53** Betrachte die lineare Differenzengleichung

$$x(t_{i+1}) = Ax(t_i) \tag{2.47}$$

mit

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ -\frac{a_0}{a_k} & -\frac{a_1}{a_k} & -\frac{a_2}{a_k} & \cdots & \cdots & -\frac{a_{k-1}}{a_k} \end{pmatrix} \in \mathbb{R}^{k\times k}.$$

Dann gilt für die Lösungen von (2.46) mit $\tilde{x}^0 = x(t_0)$ die Gleichung

$$\begin{pmatrix} \tilde{x}(t_i) \\ \vdots \\ \tilde{x}(t_{i+k-1}) \end{pmatrix} = x(t_i).$$

Insbesondere ist das Mehrschrittverfahren genau dann stabil, wenn (2.47) stabil ist.

**Beweis:** Ausschreiben der Differenzengleichung (2.47) liefert für alle $i \in \mathbb{N}_0$ die Gleichungen

$$x_j(t_{i+1}) = x_{j+1}(t_i) \text{ für } j = 1, \ldots, k-1$$

und

$$x_k(t_{i+1}) = -\frac{a_0}{a_k}x_1(t_i) - \ldots - \frac{a_{k-1}}{a_k}x_k(t_i)$$

Hiermit folgt die Behauptung per Induktion über $i$. □

Um ein Stabilitätskriterium für (2.39) zu erhalten, genügt uns also ein Stabilitätskriterium für (2.47). Hier hilft der folgende Satz, der eine Erweiterung von Satz 2.32(ii) darstellt.

Hierbei nennen wir einen Eigenwert *halbeinfach*, wenn seine algebraische und geometrische Vielfachheit übereinstimmen. Dies ist genau dann der Fall ist, wenn er eine einfache Nullstelle des Minimalpolynoms $m_A$ ist. Das Minimalpolynom $m_A$ ist dabei das Polynom mit minimalem Grad $p \geq 1$, für das $m_A(A) = 0$ gilt. Das Minimalpolynom $m_A$ teilt immer das charakteristische Polynom $\chi_A$.

**Satz 2.54** Eine lineare Differenzengleichung $x(t_{i+1}) = Ax(t_i)$ ist genau dann stabil, wenn alle Eigenwerte $\lambda_i$ von $A$ die Bedingung $|\lambda_i| \leq 1$ erfüllen und alle Eigenwerte $\lambda_i$ mit $|\lambda_i| = 1$ halbeinfach sind.

**Beweis:** Wir nummerieren die Eigenwerte gemäß der Ordnung $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_d|$. Sei $J$ die Jordan'sche Normalform von $A$ mit Transformationsmatrix $T$, also $T^{-1}AT = J$. Wir schreiben kurz $x_i = x(t_i)$ und erinnern an die explizite Lösungsdarstellung $x_i = A^i x_0 = TJ^iT^{-1}x_0$. Wir schreiben $y_0 = T^{-1}x_0$ und $y_i = J^i y_0$.

Wir nehmen zunächst an, dass die Eigenwertbedingung erfüllt ist. Der Vektor $y_0$ lässt sich zerlegen in $y_0 = y_0^1 + y_0^2$ mit $y_0^1 = (y_1, \ldots, y_p, 0 \ldots, 0)^T$ und $y_0^2 = (0 \ldots, 0, y_{p+1}, \ldots, y_k)^T$, wobei $|\lambda_p| = 1$ und $|\lambda_{p+1}| < 1$ gilt. Mit $E_1 = \langle e_1, \ldots, e_p \rangle$ und $E_2 = \langle e_{p+1}, \ldots, e_k \rangle$ bezeichen wir die zugehörigen Unterräume. Für den Vektor $y_i$ gilt nun

$$y_i = J^i y_0 = J^i(y_0^1 + y_0^2) = \underbrace{J^i y_0^1}_{=:y_i^1} + \underbrace{J^i y_0^2}_{=:y_i^2}.$$

Beachte, dass $y_i^1 \in E_1$ und $y_i^2 \in E_2$ liegt. Da die Einschränkung von $J$ auf den Unterraum $E_2$ die Bedingung von Satz 2.32(ii) erfüllt (alle Eigenwerte im Betrag kleiner als 1), folgt die Existenz von $C_1 > 0$ und $\sigma > 0$ mit

$$\|y_i^2\| \leq C_1 \underbrace{e^{-\sigma(t_i - t_0)}}_{\leq 1} \|y_0^2\| \leq C_1 \|y_0^2\|.$$

Für $y_i^1$ gilt

$$\|y_i^1\| \leq \|J^i y_0^1\| = \|y_0\|,$$

wobei die letze Gleichung aus der Eigenwertstruktur folgt, denn $J^i$ eingeschränkt auf $E_1$ ist wegen der Halbeinfachheit der Eigenwerte eine Diagonalmatrix mit Diagonalelementen $\lambda_i$ mit $|\lambda_i| = 1$. Zusammen folgt also unter Verwendung der Definition der euklidischen Norm

$$\begin{aligned} \|x_i\| &\leq \|T\|\|y_i\| = \|T\|(\|y_i^1\| + \|y_i^2\|) \leq \|T\|(C_1\|y_0^2\| + |y_0^1\|) \\ &\leq (C_1 + 1)\|T\|\|y_0\| \leq (C_1 + 1)\|T\|\|T^{-1}\|\|x_0\| = C\|x_0\| \end{aligned}$$

für die Konstante $C = (C_1 + 1)\|T\|\|T^{-1}\|$.

Sei umgekehrt die Eigenwertbedingung nicht erfüllt. Falls ein Eigenwert $\lambda_j$ mit $|\lambda_j| > 1$ existiert, so gilt für den zugehörigen Eigenvektor $x_0$

$$\|A^i x_0\| = |\lambda_j|^i \|x_0\| \to \infty \text{ für } i \to \infty,$$

was der Stabilität widerspricht. Falls ein nicht halbeinfacher Eigenwert $\lambda_j$ mit $|\lambda_j| = 1$ existiert, so gibt es einen Eigenvektor $x_0$ sowie einen verallgemeinerten Eigenvektor $x_1$, für die die Gleichungen

$$Ax_0 = \lambda_j x_0 \quad \text{und} \quad Ax_1 = x_0 + \lambda_j x_1$$

gelten (dies folgt, da das Jordan–Kästchen zu dem nicht halbeinfachen Eigenwert $\lambda_j$ eine 1 über der Diagonale besitzt). Per Induktion ergibt sich

$$A^i x_1 = i\lambda_j^{i-1} x_0 + \lambda_j^i x_1.$$

Da $\|\lambda_j^{i-1} x_0\| = \|x_0\|$ und $\|\lambda_j^i x_1\| = \|x_1\|$ (wegen $|\lambda_j| = 1$), folgt

$$\|A^i x_1\| \geq i\|x_0\| - \|x_1\| \to \infty \text{ für } i \to \infty,$$

was wiederum der Stabilität widerspricht. □

Zur Bestimmung der Stabilität genügt es also, die Eigenwerte der Matrix $A$ zu bestimmen. Dies ist aber recht einfach, wie das folgende Lemma zeigt.

**Lemma 2.55** Die Eigenwerte von $A$ aus (2.47) sind genau die Nullstellen des Polynoms $P_a$ aus (2.40). Ihre Vielfachheit im Minimalpolynom stimmt dabei mit ihrer Vielfachheit in $P_a$ überein.

**Beweis:** Man rechnet nach, dass das charakteristische Polynom von $A$ gerade durch

$$\chi_A(z) = z^k + \frac{a_{k-1}}{a_k} z^{k-1} + \ldots + \frac{a_1}{a_k} z + \frac{a_0}{a_k}$$

gegeben ist. Da die ersten Zeilen von $A, A^2, \ldots, A^{k-1}$ linear unabhängig sind (was aus der Verteilung der 0–Einträge leicht zu sehen ist), muss dies auch das Minimalpolynom $m_A$ sein. Da $a_k \neq 0$ ist, stimmen die Nullstellen und Vielfachheiten von $\chi_A$ mit denen von

$$P_a(z) = a_0 + a_1 z + \ldots + a_k z^k = a_k \chi_A(z)$$

überein. □

Unsere Überlegungen führen nun direkt auf den folgenden Satz.

**Satz 2.56** Ein lineares Mehrschrittverfahren (2.39) ist genau dann stabil, wenn alle Nullstellen $\lambda_i$ von $P_a$ die Bedingung $|\lambda_i| \leq 1$ erfüllen und alle Nullstellen $\lambda_i$ von $P_a$ mit $|\lambda_i| = 1$ einfache Nullstellen sind.

**Beweis:** Folgt sofort aus den vorangegangenen Aussagen.

Beachte, dass das Polynom $P_a$ nach Bemerkung 2.51 für jedes konsistente Mehrschrittverfahren die Nullstelle 1 besitzen muss, also mindestens eine Nullstelle mit $|\lambda_i| = 1$ besitzt. Falls dies die einzige Nullstelle mit $|\lambda_i| = 1$ ist, nennt man das Verfahren *strikt stabil*. Falls es weitere Nullstellen $\lambda_i$ mit $|\lambda_i| = 1$ gibt, so heißt das Verfahren *marginal stabil* oder *schwach stabil*. Obwohl sie theoretisch stabil sind, können solche Verfahren für bestimmte Differentialgleichungen numerische Instabilitäten aufweisen, die z.B. durch Rundungsfehler hervorgerufen werden (vgl. das aktuelle Übungsblatt).

Für die explizite Mittelpunktregel z.B. berechnet man $P_a(z) = z^2 - 1$, das Polynom besitzt also die Nullstellen $z_{1/2} = \pm 1$ und ist damit stabil, genauer marginal stabil.

Für Einschrittverfahren, die als Spezialfall der Mehrschrittverfahren aufgefasst werden können, muss das Polynom $P_a$ vom Grad $k = 1$ sein, denn nur $x_{i+1}$ und $x_i$ treten auf. Wegen $P_a(1) = 0$ kommt also nur $P_a(z) = z - 1$ in Frage, das als einzige Nullstelle $\lambda = 1$ besitzt. Also sind alle Einschrittverfahren stabil, weswegen wir die Stabilität dort nicht

betrachten mussten. Dies ist auch der Grund, warum wir die Lipschitzbedingung für Einschrittverfahren nicht (wie in vielen Lehrbüchern) als Stabilitätsbedingung bezeichnet haben: Die Bedingungen bezeichnen verschiedene Sachverhalte, auch wenn sie den gleichen Zweck im Konvergenzbeweis erfüllen, nämlich zu garantieren, dass sich die in jedem Schritt gemachten lokalen Fehler nicht aufschaukeln können.

Auf Basis von Satz 2.56 können wir nun auch verstehen, warum die zwei Mehrschrittverfahren in dem einführenden Beispiel (2.42) so unterschiedliches Verhalten aufweisen. Für das Verfahren mit den Koeffizienten $a_i$ ist das zugehörige Polynom $P_a(z) = z^2 - 3z + 2 = (z-1)(z-2)$, das gerade die Nullstellen 1 und 2 besitzt und das deswegen instabil ist. Für das zweite Verfahren mit den Koeffizienten $\tilde{a}_i$ gilt $P_{\tilde{a}}(z) = z^2 - 3/2z + 1/2 = (z-1)(z-1/2)$. Dieses Polynom hat die Nullstellen 1 und $1/2$, weswegen das Verfahren stabil ist.

### 2.9.3 Konvergenz

Ganz analog zu den Einschrittverfahren werden wir in diesem Abschnitt unser Hauptkonvergenzresultat

"Konsistenz (mit Ordnung $p$) + Stabilität $\Rightarrow$ Konvergenz (mit Ordnung $p$)"

formulieren und beweisen.

Zur Vorbereitung des Konvergenzsatzes benötigen wir noch ein Resultat über Lösungen von Differenzengleichungen, das im folgenden Lemma bereitgestellt wird.

**Lemma 2.57** Betrachte die aus (2.45) hervorgehende *inhomogene Gleichung*

$$P_a(E)(y)(t_i) = c(t_i)$$

für eine Gitterfunktion $c : \mathcal{T} \to \mathbb{R}$ und ein stabiles Mehrschrittverfahren. Dann erfüllen die Lösungen dieser Gleichung die Abschätzung

$$|y(t_{i+k})| \leq C \left( \max_{l=0,\ldots,k-1} |y(t_l)| + \sum_{l=0}^{i} |c(t_l)| \right)$$

für eine geeignete Konstante $C > 0$.

**Beweis:** Für die vektorwertige Funktion $\hat{c}(t_i) = (0, \ldots, 0, c(t_i)/a_k)^T$ kann man die Gleichung in Matrixform

$$x(t_{i+1}) = Ax(t_i) + \hat{c}(t_i)$$

mit der Matrix $A$ aus (2.47) und

$$\begin{pmatrix} y(t_i) \\ \vdots \\ y(t_{i+k-1}) \end{pmatrix} = x(t_i).$$

schreiben. Für diese Gleichung kann man die allgemeine Lösung per Induktion als

$$x(t_i) = A^i x(t_0) + \sum_{k=0}^{i-1} A^k \hat{c}(t_{i-k-1})$$

berechnen. Da $A$ stabil ist, folgt aus der Definition der Matrixnorm sofort $\|A^k\|_\infty \leq \widetilde{C}$ für alle $k \in \mathbb{N}$ für ein $\widetilde{C} > 0$. Damit ergibt sich

$$\begin{aligned} |y(t_{i+k})| \;\leq\; \|x(t_{i+1})\|_\infty &\leq \widetilde{C}\|x(t_0)\|_\infty + \widetilde{C}\sum_{k=0}^{i} \|\hat{c}(t_{i-k})\|_\infty \\ &= \widetilde{C}\|x(t_0)\|_\infty + \widetilde{C}\sum_{k=0}^{i} |c(t_{i-k})/a_k| \\ &\leq \widetilde{C} \max_{l=0,\ldots,k-1} |y(t_l)| + \widetilde{C}/a_k \sum_{k=0}^{i} |c(t_i)|, \end{aligned}$$

also die Behauptung mit $C = \max\{\widetilde{C}, \widetilde{C}/a_k\}$. □

Wir kommen nun zum Konvergenzsatz. Wir formulieren das Resultat hier etwas schwächer als im Satz 2.11, da wir keine kompakte Menge von Anfangswerten, sondern nur einen einzelnen Anfangswert betrachten. Dies dient lediglich der Vermeidung allzu technischer Formulierungen in der Aussage und im Beweis des Satzes und hat keine prinzipiellen Gründe.

**Satz 2.58** Gegeben sei ein Anfangswertproblem (2.1), (2.2) mit Anfangsbedingung $(t_0, x_0)$ und $p$–mal stetig differenzierbarem Vektorfeld $f$. Gegeben seien weiterhin ein $k$–stufiges stabiles und konsistentes lineares Mehrschrittverfahren mit Ordnung $p \in \mathbb{N}$ und Näherungswerte $\tilde{x}(t_1), \ldots, \tilde{x}(t_{k-1})$ mit

$$\|\tilde{x}(t_i) - x(t_i; t_0, x_0)\| \leq \varepsilon_0 \;\text{ für }\; i = 1, \ldots, k-1.$$

Dann gilt für die durch das Verfahren auf dem Gitter $t_i = t_0 + hi$ zur Schrittweite $h$ erzeugte Gitterfunktion $\tilde{x}(t_i)$ für alle Zeiten $t_i \in [t_0, T]$ und alle hinreichend kleinen $h > 0$ die Abschätzung

$$\|\tilde{x}(t_i) - x(t_i; t_0, x_0)\| \leq C(\varepsilon_0 + h^p)$$

für eine geeignete Konstante $C > 0$.

**Beweis:** Wir bezeichnen die exakte Lösung kurz mit $x(t)$ und wählen eine kompakte Umgebung $K \subset \mathbb{R} \times \mathbb{R}^n$ des exakten Lösungsgraphen $\{(t, x(t)) \,|\, t \in [t_0, T]\}$. Dann existiert ein $\delta_K > 0$, so dass für alle $t \in [t_0, T]$ die Folgerung $\|x - x(t)\| \leq \delta_K \Rightarrow (t, x) \in K$ gilt. Zudem existiert eine Konstante $L > 0$, so dass $f$ auf $K$ Lipschitz–stetig in $x$ mit Konstante $L$ ist. Mit $N$ bezeichnen wir die größte ganze Zahl mit $N \leq (T - t_0)/h$.

Wie im Beweis von Satz 2.11 nehmen wir zunächst an, dass die numerische Lösung für alle $t_i \in [t_0, T]$ in $K$ verläuft. Wir definieren den vektorwertigen Fehler als

$$\varepsilon_h(t_i) := x(t_i) - \tilde{x}(t_i).$$

Aus der Definition des Konsistenzfehlers folgt

$$P_a(E)(x)(t_i) = L(x,t_i,h) + hP_b(E)(\dot{x})(t_i) = L(x,t_i,h) + hP_b(E)(f)(t_i)$$

(wiederum mit der Abkürzung $f(t_i) = f(t_i, x(t_i))$). Von dieser Gleichung subtrahieren wir die Gleichung (2.40)

$$P_a(E)(\tilde{x})(t_i) = hP_b(E)(\tilde{f})(t_i).$$

Dies ergibt

$$P_a(E)(\varepsilon_h)(t_i) = L(x,t_i,h) + hP_b(E)\left(f(t_i) - \tilde{f}(t_i)\right).$$

Dies ist eine inhomogene (vektorwertige) Gleichung für $\varepsilon_h$. Indem wir Lemma 2.57 auf die einzelnen Komponenten von $\varepsilon_h(t_i)$ anwenden und $\|\varepsilon_h(t_j)\| \le \varepsilon_0$ für $j = 0,\dots,k-1$ ausnutzen, erhalten wir

$$\|\varepsilon_h(t_{i+k})\|_\infty \le C\left(\varepsilon_0 + \sum_{l=0}^{i} \|L(x,t_l,h)\|_\infty + h\left\|P_b(E)\left(f(t_l) - \tilde{f}(t_l)\right)\right\|_\infty\right). \tag{2.48}$$

für alle $i = 0,\dots,N-k$. Aus der Konsistenz folgt nun die Abschätzung

$$\|L(x,t_l,h)\|_\infty \le C_p h^{p+1}$$

und aus der Lipschitz–Stetigkeit und der Definition von $P_b$ und $E$ folgt

$$\left\|P_b(E)\left(f(t_l) - \tilde{f}(t_l)\right)\right\|_\infty \le L\sum_{m=0}^{k} |b_m| \, \|\varepsilon_h(t_{l+m})\|_\infty.$$

Setzen wir diese beiden Ungleichungen in (2.48) ein, so folgt

$$\begin{aligned}
\|\varepsilon_h(t_{i+k})\|_\infty &\le C\left(\varepsilon_0 + \underbrace{\sum_{l=0}^{i} C_p h^{p+1}}_{\le NC_ph^{p+1} \le (T-t_0)C_ph^p} + h\sum_{l=0}^{i} L\sum_{m=0}^{k} |b_m| \, \|\varepsilon_h(t_{l+m})\|_\infty\right) \\
&\le \widehat{C}_1\varepsilon_0 + \widehat{C}_2 h^p + h\widehat{C}_3 \sum_{l=0}^{i+k} \|\varepsilon_h(t_l)\|_\infty
\end{aligned}$$

für geeignete Konstanten $\widehat{C}_q > 0$. Beschränken wir nun die Schrittweite durch $h \le 1/(2\widehat{C}_3)$, so können wir nach $\|\varepsilon_h(t_{i+k})\|_\infty$ auflösen und erhalten mit $j = i+k$ die Ungleichung

$$\|\varepsilon_h(t_j)\|_\infty \le C_1\varepsilon_0 + C_2 h^p + hC_3 \sum_{l=0}^{j-1} \|\varepsilon_h(t_l)\|_\infty$$

mit $C_q = 2\widehat{C}_q$. Beachte, dass diese Ungleichung auch für $j = 1,\dots,k-1$ stimmt wenn wir o.B.d.A. $C_1 \ge 1$ annehmen.

Per Induktion (wie im Beweis von Satz 2.11) ergibt sich daraus die Abschätzung

$$\|\varepsilon_h(t_j)\|_\infty \le (C_1\varepsilon_0 + C_2h^p)e^{jhC_3} = (C_1\varepsilon_0 + C_2h^p)e^{(t_j - t_0)C_3},$$

also die gewünschte Behauptung.

Der induktive Beweis, dass die numerische Lösung für hinreichend kleine $h > 0$ tatsächlich in $K$ liegt, verläuft für explizite Verfahren ganz analog zum Beweis von Satz 2.11. Für implizite Verfahren muss dieser Beweis in jedem Schritt um ein Fixpunktargument erweitert werden, das wir hier aber nicht ausführen wollen. □

**Bemerkung 2.59** (i) Das Konvergenzresultat zeigt insbesondere, wie die Startwerte $\tilde{x}(t_1)$, $\ldots$, $\tilde{x}(t_{k-1})$ bestimmt werden müssen. Um für das Mehrschrittverfahren die Konvergenzordnung $p$ zu garantieren, müssen diese ebenfalls mit der Genauigkeit $O(h^p)$ bestimmt werden. Da es sich hier nur um endlich viele Werte handelt, deren Anzahl unabhängig von $h$ ist, genügt es dazu, ein Einschrittverfahren mit Konsistenzordnung $p-1$ zu verwenden. Der Beweis von Satz 2.11 zeigt nämlich, dass die ersten $k$ Werte durch ein solches Verfahren immer die Genauigkeit $O(h^p)$ besitzen, falls $k$ unabhängig von $h$ ist. Der "Verlust" einer Ordnung beim Übergang von der Konsistenz– zur Konvergenzordnung ergibt sich erst dadurch, dass die Anzahl der nötigen Schritte von $h$ abhängt.

(ii) Eine genauere Analyse zeigt, dass sogar die stärkere Aussage

$$\text{Konsistenz} + \text{Stabilität} \Leftrightarrow \text{Konvergenz}$$

gilt. Konsistenz und Stabilität sind also *notwendig und hinreichend* für die Konvergenz eines Verfahrens. □

### 2.9.4 Verfahren in der Praxis

In der Praxis haben sich zwei Klassen von Mehrschrittverfahren durchgsetzt. Beide Klassen haben gewisse Eigenschaften, die sie für gewisse Problemklassen besonders auszeichnen.

#### Die Adams–Verfahren

Historisch haben sich die Adams–Verfahren aus Quadraturformeln zur numerischen Integration entwickelt. Wir motivieren die Herleitung hier allerdings aus ihrer besonderen Eigenschaft, die ihre Vorteile in der Praxis begründet.

Wir haben gesehen, dass das Polynom $P_a$ eines Mehrschrittfervahrens stabil sein muss, also — abgesehen von einer Nullstelle $= 1$ — nur Nullstellen mit Betrag $|\lambda_i| \leq 1$ besitzen darf. Je kleiner die Eigenwerte dabei im Betrag sind, desto "stabiler" wird das Verfahren. Bei den Adams–Verfahren wählt man $P_a$ deswegen so, dass neben der $\lambda_1 = 1$ nur Nullstellen $\lambda_i = 0$ auftreten, also

$$P_a(z) = z^{k-1}(z-1) = z^k - z^{k-1}$$

ist. Beachte, dass damit auf der linken Seite von (2.39) nur die Werte $\tilde{x}(t_{i+k})$ und $\tilde{x}(t_{i+k-1})$ stehen bleiben.

Für jede beliebige Stufenanzahl $k$ liefert Satz 2.50(iii) nun ein Gleichungssystem mit genau zwei Lösungen, nämlich

- genau ein *explizites* Adams–Verfahren der Konsistenzordnung $p = k$ (auch *Adams–Bashforth–Verfahren* genannt)
- genau ein *implizites* Adams–Verfahren der Konsistenzordnung $p = k + 1$ (auch *Adams–Moulton–Verfahren* genannt)

Z.B. lauten die Polynome $P_b$ der ersten vier expliziten Adams–Verfahren

$$\begin{aligned} k = 1: \quad & P_b(z) &=& \ 1 \\ k = 2: \quad & P_b(z) &=& \ (3z - 1)/2 \\ k = 3: \quad & P_b(z) &=& \ (23z^2 - 16z + 5)/12 \\ k = 4: \quad & P_b(z) &=& \ (55z^3 - 59z^2 + 37z - 9)/24 \end{aligned}$$

Interessanterweise ist das explizite Adams–Verfahren für $k = 1$ gerade das explizite Euler–Verfahren.

Für diese Verfahren hat sich ein Algorithmus durchgsetzt, der als *Prädiktor–Korrektor–Verfahren* bezeichnet wird. Ein Schritt dieses Algorithmus verläuft wie folgt:

**Algorithmus 2.60 Prädiktor–Korrektor–Verfahren** Gegeben seien das explizite und das implizite Adams–Verfahren der Stufe $k$.

**1) Prädiktor–Schritt:** Berechne $\tilde{x}(t_{i+k})$ mit dem expliziten Adams–Verfahren

**2) Korrektor–Schritt:** Führe *einen Schritt* der Fixpunktiteration zur Lösung des impliziten Verfahrens mit Startwert $\tilde{x}(t_{i+k})$ durch. □

Der Prädiktor–Schritt liefert hierbei eine Approximation mit Konsistenzfehler $O(h^{k+1})$. Für hinreichend kleine Schrittweite $h$ ist die Kontraktionskonstante der Fixpunktiteration gleich $Ch$ für ein $C > 0$. Also liefert der eine Iterationsschritt eine Approximation mit dem Konsistenzfehler

$$\frac{Ch}{1 - Ch} O(h^{k+1}) = O(h^{k+2}).$$

Das Prädiktor–Korrektor–Verfahren besitzt also die Konsistenzordnung $k + 1$.

## BDF–Verfahren

Obwohl die Familie der Adams–Verfahren implizite Verfahren enthält, sind diese (wegen ihrer recht kleinen Stabilitätsgebiete $\mathcal{S}$) schlecht für steife DGL geeignet.

Tatsächlich kann man beweisen, dass kein Mehrschrittverfahren der Ordnung $p > 2$ A–stabil ist. Die zur Lösung steifer DGL so nützliche Eigenschaft $\mathbb{C}^- \subseteq \mathcal{S}$ lässt sich also nicht erreichen. Es gibt allerdings eine Klasse impliziter Mehrschrittverfahren, die zumindest unendlich große Stabilitätsgebiete $\mathcal{S}$ besitzt, und die deswegen zur Lösung steifer DGL recht gut geeignet sind.

Dies ist die Klasse der BDF–Verfahren (BDF="backwards difference"). Hier wird gefordert, dass ein Kegel der Form $\{a + ib \in \mathbb{C}^- \,|\, |b| \leq c|a|\}$ für ein $c > 0$ in $\mathcal{S}$ liegt. Dies führt auf die Bedingung

$$P_b(z) = z^k.$$

Wiederum mit Satz 2.50(iii) erhält man dann Bedingungen, nun an die Koeffizienten von $P_a$, die die Konstruktion von Verfahren beliebig hoher Konsistenzordnung $p = k$ ermöglichen. Die ersten vier Polynome lauten hier

$$\begin{aligned} k = 1: \quad P_a(z) &= z - 1 \\ k = 2: \quad P_a(z) &= \frac{3}{2}z^2 - 2z + \frac{1}{2} \\ k = 3: \quad P_a(z) &= \frac{11}{6}z^3 - 3z^2 + \frac{3}{2}z - \frac{1}{3} \\ k = 4: \quad P_a(z) &= \frac{25}{12}z^4 - 4z^3 + 3z^2 - \frac{4}{3}z + \frac{1}{4} \end{aligned}$$

Für $k = 1$ ergibt sich gerade das implizite Euler–Verfahren. Die BDF–Verfahren sind allerdings nur bis $p = k = 6$ praktikabel, da die Verfahren für höhere Stufenzahlen instabil werden (beachte, dass die Bedingungen aus 2.50(iii) nur die Konsistenz, nicht aber die Stabilität sicher stellen).

### Schrittweitensteuerung

Zuletzt wollen wir ganz kurz die Schrittweitensteuerung für Mehrschrittverfahren diskutieren. Sicherlich kann man die Fehlerschätzertheorie für Einschrittverfahren eins zu eins auf Mehrschrittverfahren übertragen und ebenso wie dort neue Schrittweiten berechnen und damit die Schrittweite adaptiv steuern.

Es ergibt sich aber ein technisches Problem, da die Schrittweite im aktuellen Schritt mit den Schrittweiten der $k - 1$ vorangegangenen Schritten übereinstimmen muss, weil ansonsten die definierende Gleichung (2.39) nicht sinnvoll ausgewertet werden kann.

Abhilfe schafft hier eine alternative Darstellung, die wir für die Adams–Verfahren illustrieren: Wenn die Werte $\tilde{x}(t_i), \ldots, \tilde{x}(t_{i+k-1})$ eine Approximation der Ordnung $p$ an die differenzierbare Funktion $x(t)$ in den Punkten $t_i, \ldots, t_{i+k-1}$ darstellen, so ist das durch die Daten

$$(t_i, \tilde{x}(t_i)), \ldots, (t_{i+k-1}, \tilde{x}(t_{i+k-1}))$$

definierte Interpolationspolynom $q(t)$ eine Approximation der Ordnung $p$ an $x(t)$, und zwar für alle $t$ aus einem vorgegebenen kompakten Intervall.

Für die Adams–Verfahren kann man nachrechnen, dass die Verfahren mit diesem Interpolationspolynom $q$ gerade als

$$\tilde{x}(t_{i+k}) = \tilde{x}(t_{i+k-1}) + \int_{t_{i+k-1}}^{t_{i+k}} q(t)dt$$

gegeben sind (zum Beweis betrachtet man die Lagrange–Polynomdarstellung von $q$ und integriert). Diese Gleichung ist nun unabhängig von der zur Berechnung von $q$ verwendeten Schrittweite und kann daher für variable Schrittweiten ausgewertet werden.

Für die BDF–Verfahren ist ein ähnlicher Trick möglich, so dass auch hier die Schrittweitensteuerung anwendbar ist.

In MATLAB finden sich schrittweitengesteuerte Adams–Verfahren unter dem Namen `ode113` und BDF–Verfahren unter dem Namen `ode15s`.

## 2.10 Randwertprobleme

Bisher haben wir uns ausschließlich mit der Lösung von Anfangswertproblemen

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0$$

beschäftigt. In diesem Abschnitt wollen wir eine weitere Problemstellung bei gewöhnlichen Differentialgleichungen betrachten, nämlich die sogenannten *Randwertprobleme*. Zur Einführung soll das folgende Beispiel dienen:

**Beispiel 2.61** Betrachte die zweidimensionale Gleichung

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} x_2(t) \\ -kx_2(t) - \sin x_1(t) \end{pmatrix},$$

die die Bewegung eines Pendels beschreibt, bei dem $x_1$ den Winkel des Pendels und $x_2$ die Winkelgeschwindigkeit des Pendels beschreibt. Die Konstante $k \geq 0$ gibt die Stärke der Reibung an, der das Pendel unterliegt.

Bei einem Anfangswertproblem gibt man nun eine Zeit $t_0$ und eine Anfangsbedingung $x_0 = (x_1^0, x_2^0)^T$ vor, was bedeutet, dass man Position und Geschwindigkeit des Pendels im Zeitpunkt $t_0$ festlegt und dann errechnet, wie sich das Pendel ausgehend von diese Anfangsbedingung in der Zukunft bewegt.

Bei einem Randwertproblem ist die Problemstellung anders: Hier gibt man sich zwei Zeitpunkte $t_0 < t_1$ vor, einen Anfangs- und einen Endzeitpunkt, und stellt zu beiden Zeitpunkten Bedingungen an die Lösung. Im Pendelmodell könnte man also zum Beispiel Winkel $x_1^0$ und $x_1^1$ vorgeben und nun eine Lösung $x^*(t) = (x_1^*(t), x_2^*(t))^T$ der Pendelgleichung berechnen wollen, für die $x_1^*(t_0) = x_1^0$ und $x_1^*(t_1) = x_1^1$ gilt. Gesucht ist also eine Pendelbewegung, die im Zeitpunkt $t_0$ den Winkel $x_1^0$ und im Zeitpunkt $t_1$ den Winkel $x_1^1$ annimmt. Die zugehörigen Geschwindigkeiten sind nicht festgelegt, sondern spielen hier vielmehr die Rolle freier Parameter, die während der numerischen Lösung so bestimmt werden müssen, dass die zugehörige Lösung die geforderten Bedingungen auch erfüllt. □

Allgemein formulieren wir das Randwertproblem wie folgt.

**Definition 2.62** Ein *Randwertproblem* für eine gewöhnliche Differentialgleichung (2.1) im $\mathbb{R}^n$ besteht darin, eine Lösung $x^*(t)$ der Gleichung zu finden, die für Zeiten $t_0 < t_1$ und eine Funktion $r(x, y)$, $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ die Bedingung

$$r(x^*(t_0), x^*(t_1)) = 0$$

erfüllt. □

Für unser Pendelbeispiel 2.61 könnten wir die Funktion $r$ z.B. als

$$r(x, y) = \begin{pmatrix} x_1 - x_1^0 \\ y_1 - x_1^1 \end{pmatrix}$$

definieren.

### 2.10.1 Lösbarkeit des Problems

Ob ein gegebenes Randwertproblem tatsächlich lösbar ist, ist im Allgemeinen sehr schwer zu überprüfen. Wir beschränken uns daher hier auf einen Existenzsatz für den speziellen Fall linearer Differentialgleichungen und beweisen im allgemeinen nichtlinearen Fall nur einen lokalen Eindeutigkeitssatz.

Aus der Theorie der Differentialgleichungen ist bekannt, dass die Lösungen linearer homogener Differentialgleichungen der Form

$$\dot{x}(t) = A(t)x(t) \tag{2.49}$$

als

$$x(t; t_0, x_0) = \Phi(t, t_0)x_0$$

geschrieben werden können, wobei die sogenannte *Fundamentalmatrix* $\Phi(t; t_0) \in \mathbb{R}^{n\times n}$ eine Lösung des matrixwertigen Anfangswertproblems

$$\dot{\Phi}(t) = A(t)\Phi(t), \qquad \Phi(t_0) = \mathrm{Id} \tag{2.50}$$

ist. Bezeichnet man die $i$-te Spalte dieser Matrix mit $\Phi_i(t; t_0)$, so sieht man leicht, dass $\Phi_i$ Lösung des Anfangswertproblems

$$\dot{\Phi}_i(t) = A(t)\Phi_i(t), \qquad \Phi_i(t_0) = e_i$$

ist, bei dem $e_i$ den $i$-ten Einheitsvektor bezeichnet. Auf diese Weise kann man die Spalten der Matrix $\Phi(t; t_0)$ auch numerisch berechnen.

**Satz 2.63** Gegeben sei eine inhomogene lineare Differentialgleichung der Form

$$\dot{x}(t) = A(t)x(t) + b(t) \tag{2.51}$$

und eine Randbedingung der Form

$$r(x, y) = Bx + Cy + d$$

für Matrizen $A(t), B, C \in \mathbb{R}^{n\times n}$ und Vektoren $b(t), d \in \mathbb{R}^n$. Es sei $\Phi$ die Fundamentalmatrix der zugehörigen homogenen Gleichung (2.49) und $x(t; t_0, x_0)$ eine Lösung von (2.51) mit beliebigem Anfangswert $x_0 \in \mathbb{R}^n$. Dann ist

$$x^*(t) = x(t; t_0, x_0^*)$$

genau dann eine Lösung des Randwertproblems, wenn der Anfangswert $x^*$ eine Lösung des linearen Gleichungssystems

$$(B + C\Phi(t_1, t_0))(x_0^* - x_0) = -(Bx_0 + Cx(t_1; t_0, x_0) + d) \tag{2.52}$$

ist. Insbesondere existiert also genau dann eine eindeutige Lösung des Randwertproblems, wenn die Matrix $B + C\Phi(t_1, t_0)$ vollen Rang besitzt.

**Beweis:** Für zwei beliebige Anfangswerte $x_0, x_0^* \in \mathbb{R}^n$ gilt für die Differenz der zugehörigen Lösungen von (2.51)

$$\begin{aligned}\frac{d}{dt}(x(t;t_0,x_0^*) - x(t;t_0,x_0)) &= A(t)x(t;t_0,x_0^*) + b(t) - A(t)x(t;t_0,x_0) - b(t)\\ &= A(t)(x(t;t_0,x_0^*) - x(t;t_0,x_0))\end{aligned}$$

und damit

$$x(t;t_0,x_0^*) - x(t;t_0,x_0) = \Phi(t,t_0)(x_0^* - x_0)$$

und folglich auch

$$x(t;t_0,x_0^*) = x(t;t_0,x_0) + \Phi(t,t_0)(x_0^* - x_0).$$

Einsetzen in die Randbedingung ergibt

$$\begin{aligned}0 &= Bx(t_0;t_0,x_0^*) + Cx(t_1;t_0,x_0^*) + d\\ &= Bx_0^* + C\Big(x(t_1;t_0,x_0) + \Phi(t_1,t_0)(x_0^* - x_0)\Big) + d\\ &= \Big(B + C\Phi(t_1,t_0)\Big)(x_0^* - x_0) + Bx_0 + Cx(t_1;t_0,x_0) + d\end{aligned}$$

Die Randbedingung ist also genau dann erfüllt, wenn $x_0^*$ eine Lösung des linearen Gleichungssystems (2.52) ist. □

Beachte, dass sich das Gleichungsystem (2.52) deutlich vereinfacht, wenn wir $x_0 = 0$ wählen. Wir werden später sehen, warum es dennoch nützlich ist, den Satz für allgemeine $x_0 \in \mathbb{R}^n$ zu formulieren.

**Beispiel 2.64** Wenn wir die Pendelgleichung aus Beispiel 2.61 durch die lineare Pendelgleichung

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} x_2(t) \\ -kx_2(t) - x_1(t) \end{pmatrix}$$

ersetzen, so erhalten wir ein Problem der Form aus Satz 2.63 mit

$$A = \begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix},\ b = 0,\ B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ C = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \text{ und } d = \begin{pmatrix} -x_1^0 \\ -x_1^1 \end{pmatrix}.$$

□

Für allgemeine nichtlineare Differentialgleichungen ist ein solcher Satz nicht beweisbar. Wir können aber, wenn wir annehmen, dass eine Lösung $x^*(t)$ des Randwertproblems existiert, zumindest Bedingungen für die lokale Eindeutigkeit der Lösung angeben und beweisen.

Dazu — aber auch für die numerische Lösung des Problems im nächsten Abschnitt — benötigen wir die partielle Ableitung

$$\frac{\partial}{\partial x_0} x(t;x_0,x_0)$$

der Lösung eines Anfangswertproblems. Wir hatten bereits im Abschnitt 2.2.4 über die Kondition verwendet, dass diese Ableitung über die sogenannte *Variationsgleichung*

$$\dot{y}(t) = A(t)y(t), \qquad A(t) = \frac{\partial f}{\partial x}(t, x(t; t_0, x_0)) \tag{2.53}$$

berechnet werden kann. Genauer gilt, dass die Fundamentalmatrix $\Phi(t, t_0)$ (vgl. (2.50)) der Variationsgleichung (2.53) gerade die (matrixwertige) Ableitung nach dem Anfangswert ist: Es gilt

$$\frac{\partial}{\partial x_0} x(t; x_0, x_0) = \Phi(t; t_0).$$

Diesen Zusammenhang nutzen wir in dem folgenden Satz.

**Satz 2.65** Es sei $x^* : [t_0, t_1] \to \mathbb{R}^n$ eine Lösung des Randwertproblems aus Definition 2.62 mit $f \in C^1(\mathbb{R} \times \mathbb{R}^n, \mathbb{R}^n)$ und $r \in C^1(\mathbb{R}^n \times \mathbb{R}^n, \mathbb{R}^n)$. Es sei $\Phi^*(t, t_0)$ die Fundamentalmatrix (2.50) der Variationsgleichung (2.53) mit $x(t) = x^*(t)$. Zudem definieren wir die $n \times n$-Matrizen

$$B^* := \frac{\partial r}{\partial x}(x^*(t_0), x^*(t_1)) \text{ und } C^* := \frac{\partial r}{\partial y}(x^*(t_0), x^*(t_1))$$

über die Ableitungen der Randwertfunktion $r(x, y)$.

Dann gilt: Fall die *Sensitivitätsmatrix*

$$E^*(t) := B^*\Phi^*(t_0, t) + C^*\Phi^*(t_1, t)$$

für ein $t = \tau_0 \in [t_0, t_1]$ vollen Rang besitzt, so besitzt sie für alle $t \in [t_0, t_1]$ vollen Rang und $x^*$ ist eine lokal eindeutige Lösung des Randwertproblems.

**Beweis:** Wir zeigen zunächst die lokale Eindeutigkeit. Definieren wir für eine beliebige Lösung $x(t; \tau_0, x_0)$ und die Randbedingungsfunktion $r$ die Funktion

$$F(x_0) = r(x(t_0; \tau_0, x_0), x(t_1; \tau_0, x_0)),$$

so ist eine beliebige Lösung $x(t)$ der Differentialgleichung genau dann eine Lösung des Randwertproblems, wenn

$$F(x(\tau_0)) = 0 \tag{2.54}$$

gilt. Um die lokale Eindeutigkeit der Lösung zu zeigen, müssen wir also beweisen, dass eine Umgebung $U$ um $x^*(\tau_0)$ existiert, so dass

$$F(x) \neq 0 \text{ für alle } x \in U \setminus x^*(\tau_0)$$

gilt.

Gleichung (2.54) ist ein nichtlineares Gleichungssystem mit $n$ Gleichungen und $n$ Unbekannten. Nach dem Satz über inverse Funktionen gibt es genau dann eine lokal eindeutige Lösung, wenn die Jacobi-Matrix

$$DF(x^*(\tau_0))$$

vollen Rang besitzt. Diese ist aber für $x_0 = x^*(\tau_0)$ gerade gegeben durch

$$\begin{aligned} DF(x_0) &= \frac{d}{dx_0} r(x(t_0;\tau_0,x_0), x(t_1;\tau_0,x_0)) \\ &= \frac{\partial r}{\partial x}(x(t_0;\tau_0,x_0), x(t_1;\tau_0,x_0))\frac{\partial}{\partial x_0}x(t_0,\tau_0,x_0) \\ &\quad + \frac{\partial r}{\partial y}(x(t_0;\tau_0,x_0), x(t_1;\tau_0,x_0))\frac{\partial}{\partial x_0}x(t_1,\tau_0,x_0)) \\ &= B^*\Phi^*(t_0,\tau_0) + C^*\Phi(t_1,\tau_0) \end{aligned}$$

und besitzt daher vollen Rang. Daraus folgt die lokale Eindeutigkeit.

Würde nun ein $\tau_1 \in [t_0, t_1]$ existieren, für dass die Sensitivitätsmatrix keinen vollen Rang besitzt, so würden nach dem Satz über implizite Funktionen Werte $x_0$ beliebige nahe an $x^*(t)$ existieren, so dass $x(t;\tau_1,x_0)$ das Randwertproblem löst. Damit hätten wir Werte $x(\tau_0;\tau_1,x_0)$ gefunden, die in beliebig kleinen Umgebungen von $x^*(\tau_0)$ liegen und (2.54) lösen, was ein Widerspruch zur lokalen Eindeutigkeit ist. □

Auch wenn die Bedingungen dieses Satzes i.A. schwer zu überprüfen sind, so liefert er doch die Begründung dafür, dass eine numerische Berechnung der Lösung des Randwertproblems möglich ist, da das Problem zumindest lokal eine eindeutige Lösung besitzt und damit wohldefiniert ist. Zudem liefert er eine wichtige Einsicht in die Struktur des Problems, die wir im folgenden Abschnitt numerisch nutzen werden.

### 2.10.2 Schießverfahren

Der Beweis von Satz 2.65 zeigt bereits die Richtung auf, die wir bei der numerischen Lösung des Problems einschlagen können. Das Problem, eine Lösungsfunktion zu finden, die zwei vorgegebene Punkte verbindet, wurde dort reduziert auf das Problem, einen Anfangswert $x_0 \in \mathbb{R}^n$ zu finden, der das $n$-dimensionale nichtlineare Gleichungssystem (2.54) löst. Die dort definierte Abbildung $F$ vereinfacht sich für $\tau_0 = t_0$ zu

$$F(x_0) = r(x_0, x(t_1;t_0,x_0)). \tag{2.55}$$

Diese Form wollen wir im Folgenden verwenden.

Unser Ziel ist nun, das Problem zu lösen, indem wir das Nullstellenproblem (2.55) numerisch lösen. Dieses Vorgehen — also die Lösung eines Randwertproblems durch die Lösung eines durch ein Anfangswertproblem bestimmtes Gleichungssystem — wird als *Schießverfahren* bezeichnet. Ursprung dieses etwas martialischen Namens ist tatsächlich das Schießen im militärischen Sinne, genauer die Artillerie. Auch hier hat man eine Endbedingung gegeben (nämlich ein zu treffendes Ziel) und variiert die Anfangsbedingung (Winkel des Geschützes oder Schussstärke), um die Endbedingung zu erfüllen.

Algorithmen zur Lösung nichtlinearer Gleichungssysteme kennen wir aus der Numerik I, nämlich die Fixpunktiteration und das Newton-Verfahren. Während erstere nur unter relativ einschränkenden Bedingungen funktioniert (die wir hier realistischerweise nicht unbedingt annehmen können), funktioniert die zweite lokal immer, benötigt aber die Information über die Ableitung von $F$. Hier kommt als weitere Schwierigkeit hinzu, dass die Definition

von $F$ neben der — gegebenen — Abbildung $r$ auch die — im Allgemeinen unbekannte — Lösung $x(t_1; t_0, x_0)$ enthält. Wie können $F$ und die Ableitung $DF$ aber numerisch auswerten, denn da $x(t_1; t_0, x_0)$ und $\Phi(t_1, t_0)$ ja gerade die Lösung von Anfangswertproblemen sind, können wir diese mit jedem der bisher behandelten Algorithmen berechnen.

Zunächst erinnern wir an das Newton-Verfahren im $\mathbb{R}^n$, vgl. Algorithmus 6.14 im Skript zur Numerik I:

Gegeben sei eine Funktion $F : \mathbb{R}^n \to \mathbb{R}^n$, ihre Ableitung $DF : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ sowie ein Startwert $x^{(0)} \in \mathbb{R}^n$ und eine gewünschte Genauigkeit $\varepsilon > 0$. Setze $i = 0$.

(1) Löse das lineare Gleichungssystem $DF(x^{(i)})\Delta x^{(i)} = F(x^{(i)})$
und berechne $x^{(i+1)} = x^{(i)} - \Delta x^{(i)}$

(2) Falls $\|\Delta x^{(i)}\| < \varepsilon$, beende den Algorithmus,
ansonsten setze $i = i + 1$ und gehe zu (1)

Um dies auf unser Problem anzuwenden, müssen wir nun klären, wie wir $F$ und $DF$ numerisch berechnen.

Die Berechnung von $F$ aus (2.55) stellt dabei kein größeres Problem dar: Für gegebenes $x^{(i)}$ berechnen wir numerisch die Lösung $\tilde{x} = x(t_1; t_0, x^{(i)})$ mittels eines Ein- oder Mehrschrittverfahrens und berechnen damit

$$F(x^{(i)}) \approx r(x^{(i)}, \tilde{x})$$

Komplizierter ist die Berechnung von $DF$. Zunächst gilt nach der Rechnung im Beweis von Satz 2.65 mit $\tau = t_0$

$$DF(x^{(i)}) = B + C\Phi(t_1, t_0)$$

mit Matrizen $B$ und $C$ gegeben durch

$$B = \frac{\partial r}{\partial x}(x^{(i)}, x(t; t_0, x^{(i)})) \text{ und } C = \frac{\partial r}{\partial y}(x^{(i)}, x(t; t_0, x^{(i)}))$$

für die Randwertfunktion $r(x, y)$. Die $i$-te Spalte der Matrix $\Phi(t_1, t_0)$ ist nun gerade die Lösung des Anfangswertproblems

$$\dot{y}_i(t) = \frac{\partial f}{\partial x}(t, x(t_1; t_0, x^{(i)}))y_i(t), \quad y_i(t_0) = e_i,$$

wobei $e_i$ der $i$-te Einheitsvektor ist.

Die numerische Berechnung von $F$ und $DF$ kann also wie folgt geschehen. Vorab berechnen wir (analytisch) die Ableitungen

$$\frac{\partial f}{\partial x}(t, x), \quad \frac{\partial r}{\partial x}(x, y), \quad \frac{\partial r}{\partial y}(x, y).$$

In jedem Schritt des Newton-Verfahrens approximieren wir dann numerisch die Lösung $z(t_1)$ des $n(n + 1)$–dimensionalen Anfangswertproblems

$$\dot{z}(t) = g(t, z(t)), \quad z(t_0) = z_0$$

mit

$$z(t) = \begin{pmatrix} x(t) \\ y_1(t) \\ \vdots \\ y_n(t) \end{pmatrix}$$

und

$$g(t, z(t)) = \begin{pmatrix} f(t, x(t)) \\ \frac{\partial f}{\partial x}(t, x(t)) y_1(t) \\ \vdots \\ \frac{\partial f}{\partial x}(t, x(t)) y_n(t) \end{pmatrix}, \quad z_0 = \begin{pmatrix} x^{(i)} \\ e_1 \\ \vdots \\ e_n \end{pmatrix}.$$

Mit Hilfe der numerischen Approximation

$$\tilde{z} = \begin{pmatrix} \tilde{x} \\ \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{pmatrix} \approx z(t_1)$$

berechnen wir dann die Approximationen

$$F(x^{(i)}) \approx r(x^{(i)}, \tilde{x})$$

und

$$DF(x^{(i)}) \approx \widetilde{B} + \widetilde{C}\widetilde{\Phi}(t_1, t_0)$$

mit

$$\widetilde{B} = \frac{\partial r}{\partial x}(x^{(i)}, \tilde{x}), \;\; C = \frac{\partial r}{\partial y}(x^{(i)}, \tilde{x}) \;\text{ und }\; \widetilde{\Phi}(t_1, t_0) = \Big(\tilde{y}_1, \ldots, \tilde{y}_n\Big).$$

Damit kann das Newton-Verfahren nun vollständig implementiert werden.

In Beispiel 2.61 lautet das zu lösende Differentialgleichungssystem also

$$\dot{z}(t) = \begin{pmatrix} z_2(t) \\ -kz_2(t) - \sin(z_1(t)) \\ z_4(t) \\ -kz_4(t) - \cos(z_1(t)) z_3(t) \\ z_6(t) \\ -kz_6(t) - \cos(z_1(t)) z_5(t) \end{pmatrix} \quad \text{mit} \quad z(t_0) = z_0 = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Interessant ist, was im Falle eines linearen Problems im Sinne von Satz 2.63 passiert. In diesem Fall ist ein Newton-Schritt ausgehend von einem beliebigen Startwert $x^{(0)}$ gerade äquivalent zu dem linearen Gleichungssystem (2.52). Wir erhalten die (bis auf numerische Diskretisierungsfehler) exakte Lösung des Randwertproblems also nach genau einem Schritt des Newton-Verfahrens. Auf die genauen Auswirkungen der Diskretisierungsfehler im Linearen und im Nichtlinearen können wir hier aus Zeitgründen nicht genauer eingehen.

### 2.10.3 Mehrzielmethode

Die oben beschriebene Methode funktioniert theoretisch gut, hat aber in der Praxis den nicht zu unterschätzenden Nachteil, dass die Lösung $x(t_1; t_0, x_0)$ in vielen Beispielen sehr sensitiv vom Anfangswert $x_0$ abhängt.

Als Beispiel betrachten wir das Randwertproblem

$$\dot{x}(t) = x^2,\ t_0 = 0,\ t_1 = 1,\ r(x, y) = y - 9.$$

Gesucht ist also eine Lösung $x^*(t)$ dieser Gleichung mit $x(1) = 9$. Da die allgemeine Lösung hier leicht als

$$x(t; 0, x_0) = \frac{x_0}{1 - x_0 t}$$

ausgerechnet werden kann, sieht man, dass die gesuchte Lösung gerade

$$x^*(t) = \frac{0.9}{1 - 0.9t}, \quad \text{also} \quad x^*(0) = 0.9$$

lautet. Liegen wir mit unserem Anfangswert nur um 10% oberhalb dieses Wertes, also $x_0 = 0.99$, so erhalten wir

$$x(1; 0, 0.99) = 99 \quad \text{und damit} \quad r(0.9, x(1; 0, 0.99)) = 90.$$

Für $x_0 = 1$ ist die Sache noch schlimmer, da die Lösung dann zum Zeitpunkt $t_1 = 1$ gar nicht mehr existiert.

Die Schätzlösung $r(x^{(0)}, x(t_1; t_0, x^{(0)}))$ im Newton-Verfahren kann also selbst bei einer relativ guten Startschätzung $x^{(0)} \approx x^*(t_0)$ weit von $r(x^*(t_0), x^*(t_1)) = 0$ abweichen oder sogar undefiniert sein. Es ist leicht einzusehen, dass dies große numerische Konvergenzprobleme im Newton-Verfahren nach sich zieht und der Bereich der lokalen Konvergenz des Verfahrens dadurch sehr klein wird.

Eine Abhilfe ist die in den 1960er Jahren zuerst vorgeschlagene und in den 1970er Jahren vor allem durch Roland Bulirsch[7] weiterentwickelte *Mehrzielmethode* (auch *Mehrfachschießverfahren*). Die Idee dabei ist, das Intervall $[t_0, t_1]$ in $d \in \mathbb{N}$ Teilintervalle $[\tau_i, \tau_{i+1}]$ zu zerlegen mit

$$t_0 = \tau_0 < \tau_1 < \ldots < \tau_d = t_1.$$

Statt die Lösung $x(t_0; t_1, x_0)$ für einen Anfangswert $x_0$ auf dem gesamten Intervall $[t_0, t_1]$ zu berechnen, wählt man nun $d$ Anfangswerte $x_0, \ldots, x_{d-1}$ und berechnet separat die Lösungen

$$x(\tau_k; \tau_{k-1}, x_{k-1}), \quad k = 1, \ldots, d$$

auf den Teilintervallen $[\tau_{k-1}, \tau_k]$. Damit sich diese Lösungen zu einer Gesamtlösung auf dem Intervall $[t_0, t_1]$ zusammensetzen lassen, müssen die Stetigkeitsbedingungen

$$x(\tau_k; \tau_{k-1}, x_{k-1}) = x_{k+1}, \quad k = 1, \ldots, d-2$$

gelten und damit diese Gesamtlösung eine Lösung des Randwertproblems ist, muss zusätzlich noch die ursprüngliche Randbedingung

$$r(x_0, x(\tau_d; \tau_{d-1}, x_{d-1})) = 0$$

[7] deutscher Mathematiker, geb. 1932

gelten.

Definieren wir nun eine neue Randbedingungsfunktion $R : \mathbb{R}^{2dn} \to \mathbb{R}^{dn}$ mittels

$$R(x_0, x_0', \ldots, x_{d-1}, x_{d-1}') = \begin{pmatrix} x_0 - x_0' \\ \vdots \\ x_{d-1} - x_{d-1}' \\ r(x_0, x_{d-1}') \end{pmatrix},$$

so liefert die Lösung des Nullstellenproblems

$$F(x_0, \ldots, x_{d-1}) = 0$$

mit $F : \mathbb{R}^{dn} \to \mathbb{R}^{dn}$ definiert durch

$$F(x_0, \ldots, x_{d-1}) = R(x_0, x(\tau_1; \tau_0, x_0), x_1, x(\tau_2; \tau_1, x_1), \ldots, x_{d-1}, x(\tau_d; \tau_{d-1}, x_{d-1})$$

eine Lösung des ursprünglichen Randwertproblems. Da die Lösungen der Differentialgleichung hier nur auf kurzen Intervallen $[\tau_{k-1}, \tau_k]$ berechnet werden müssen, sind sie deutlich weniger sensitiv gegenüber Änderungen in den Anfangswerten. Dies überträgt sich auf die Randwertfunktion, weswegen die Mehrzielmethode einen deutlich größeren Konvergenzbereich besitzt.

Der Preis dafür ist natürlich die Erhöhung der Dimension des Nullstellenproblems von $n$ auf $dn$, die sich insbesondere bei der Lösung der linearen Gleichungssysteme im Newton-Verfahren bemerkbar macht (beachte, dass die numerische Berechnung der höheren Anzahl von Differentialgleichungslösungen i.A. kaum mehr Aufwand verursacht, weil diese auf entsprechend kürzeren Intervallen zu lösen sind). Hier kann insbesondere durch Ausnutzen der speziellen Bandstruktur der entstehenden Gleichungssysteme viel Rechenzeit gespart werden, für Details der algorithmischen Umsetzung verweisen wir z.B. auf das Buch von Deuflhard und Bornemann [2, Abschnitt 8.2.2].

# Kapitel 3

# Stochastische Differentialgleichungen

In diesem Kapitel beschäftigen wir uns mit der numerischen Lösung von gewöhnlichen Differentialgleichungen, die von einer zufälligen Funktion abhängen. Informell kann man diese Gleichungen in Erweiterung von (2.1) als

$$\frac{d}{dt}X(t) = a(t, X(t)) + b(t, X(t))g(t, \omega)$$

schreiben, wobei die Vektorfelder $a, b : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ "gewöhnliche" (also deterministische) Funktionen sind, während die Funktion $g : \mathbb{R} \times \Omega \to \mathbb{R}$ von einem zufälligen Parameter $\omega \in \Omega$ abhängt. Solche zufälligen Gleichungen treten überall dort auf, wo man nicht genügend Informationen für ein exaktes deterministisches Modell besitzt und die bestehenden Unsicherheiten (z.B. unbekannte Umwelteinflüsse, unsichere zukünftige Entwicklung, unbekannte variierende Parameter ...) stochastisch modelliert, z.B. in der Biologie, in der Ingenieurmathematik oder in der Finanzmathematik.

Eine typische Aufgabe der Numerik stochastischer Differentialgleichungen besteht nun darin, dass man eine große Anzahl zufälliger Lösungen $X(t; t_0, x_0, \omega_1)$, ..., $X(t; t_0, x_0, \omega_M)$ numerisch berechnet und daraus gewisse statistische Größen wie z.B. den *Erwartungswert*

$$\mathbb{E}\{X(t; t_0, x_0, \omega)\} \approx \frac{1}{M}\sum_{j=1}^{M} X(t; t_0, x_0, \omega_j)$$

berechnet. Hierzu ist es i.A. nicht nötig, jede Funktion $x(t; t_0, x_0, \omega_j)$ möglichst genau zu berechnen (was sich im Übrigen als sehr schwierig herausstellen wird); es genügt, dass der Fehler "im Mittel" klein wird.

Um dies etwas präziser zu machen und zu zeigen, wie man geeignete zufällige Funktionen definiert und am Computer erzeugt, beginnen wir mit einigen Grundlagen.

## 3.1 Zufallsvariablen und Zufallszahlen

Die Grundobjekte, die man benötigt, um den Zufall in der Mathematik zu formalisieren, sind der *Wahrscheinlichkeitsraum* und die *Zufallsvariable*. Der Wahrscheinlichkeitsraum

bestehe aus einer (endlichen oder unendlichen) Menge $\Omega$ von *Ereignissen*, aus dem man zufällig Elemente $\omega \in \Omega$, die *Elementarereignisse* ziehen kann. Diese Ereignisse gehorchen einer gewissen Verteilung, die durch ein *Wahrscheinlichkeitsmaß* $\mathbb{P}$ gegeben ist. Das Maß $\mathbb{P}$ ist dabei eine Abbildung von $\mathcal{P}(\Omega)$ (der Menge aller Teilmengen von $\Omega$)[1] nach $[0, 1]$. Für eine gegebene Teilmenge $A \subset \Omega$ gibt $\mathbb{P}(A)$ gerade die Wahrscheinlichkeit an, mit der ein zufällig gezogenes Ereignis $\omega$ in $A$ liegt.

**Beispiel 3.1** Für ein Modell eines idealen Würfels setzt man $\Omega = \{1, 2, 3, 4, 5, 6\}$. Das Wahrscheinlichkeitsmaß $\mathbb{P}$ kann man dann mittels

$$\mathbb{P}(\{\omega\}) := \frac{1}{6} \text{ für alle } \omega \in \Omega$$

und

$$\mathbb{P}(B) := \sum_{\omega \in B} \mathbb{P}(\{\omega\})$$

für beliebige Teilmengen $B \subseteq \Omega$ definiert werden. □

Schwieriger wird die Sache, wenn wir es mit unendlichen Mengen $\Omega$ zu tun haben, da eine solche direkte Definition dann nicht mehr möglich ist, wir uns genau überlegen müssen, welche Mengen $A$ hier überhaupt zulässig sein sollen etc. Wir wollen uns mit diesen technischen Feinheiten hier nicht näher aufhalten, da wir nicht direkt mit Wahrscheinlichkeitsräumen, sondern mit Zufallsvariablen arbeiten werden.

### 3.1.1 Zufallsvariablen

Anstelle der Wahrscheinlichkeitsräume werden wir hier die *Zufallsvariablen* $X : \Omega \to \mathbb{R}$ in den Mittelpunkt stellen. Wie die Schreibweise bereits andeutet, sind Zufallsvariablen keine Variablen, sondern Abbildungen von der Ereignismenge $\Omega$ in die reellen Zahlen. Das Wahrscheinlichkeitsmaß $\mathbb{P}$ auf $\Omega$ induziert dann eine *Wahrscheinlichkeitsverteilung* $\mathbb{P}_X$ für $X$: Für jede Teilmenge $B \subseteq \mathbb{R}$ gibt

$$\mathbb{P}_X(B) := \mathbb{P}(\{\omega \in \Omega \,|\, X(\omega) \in B\})$$

gerade die Wahrscheinlichkeit an, dass sich eine *Realisierung* $X(\omega)$ *von* $X$, also der Wert $X(\omega)$ für ein zufällig gezogenes $\omega \in \Omega$ in $B$ befindet.

Im Prinzip kann man $\mathbb{P}_X$ durch die obige Formel definieren. In der Praxis geht man allerdings meist anders vor, indem man eine geeignete Formel für $\mathbb{P}_X$ angibt. Dies hat mehrere Vorteile: Zum einen befinden wir uns im Wertebereich von $X$ auf den reellen Zahlen, also auf analytisch "sicherem Boden", zum anderen brauchen wir in diesem Fall weder die genaue Abbildung $X : \Omega \to \mathbb{R}$ noch das zugrundeliegende Wahrscheinlichkeitsmaß $\mathbb{P}$ auf $\Omega$ zu kennen (wir müssen aber immer im Hinterkopf behalten, dass eine Menge $\Omega$ und ein Maß $\mathbb{P}$ existieren).

Hier werden wir uns mit zwei verschiedenen Verteilungen von Zufallsvariablen beschäftigen. Die erste ist die *gleichverteilte Zufallsvariable*. Hier muss man zwischen Zufallsvariablen

[1] Tatsächlich ist das Maß i.A. nur auf einer Teilmenge von $\mathcal{P}(\Omega)$ definiert, einer sogenannten Sigma-Algebra. Für genaue Definitionen verweisen wir auf die einführende Stochastik-Vorlesung.

unterscheiden, die nur endlich viele Werte annehmen können und solchen, die unendlich viele Werte annehmen können.

Wir betrachten zunächst den endlichen Fall. Hier kann $X(\omega)$ gerade $N$ verschiedene Werte $x_1, \ldots, x_N$ annehmen. Gleichverteilt bedeutet nun, dass

$$\mathbb{P}_X(\{x_i\}) := \frac{1}{N} \text{ für } i = 1, \ldots, N$$

und

$$\mathbb{P}_X(B) := \sum_{x_i \in B} \mathbb{P}_X(\{x_i\}) = \sum_{i=1}^{N} \chi_B(x_i) \mathbb{P}_X(\{x_i\})$$

ist, wobei

$$\chi_B(x) = \begin{cases} 1, & x \in B \\ 0, & x \notin B \end{cases}$$

die sogenannte *charakteristische Funktion* der Menge $B$ ist. Jeder Wert $x_i$ wird also mit der gleichen Wahrscheinlichkeit angenommen.

Falls $X(\Omega)$ unendlich viele verschiedene Werte annehmen kann, so versagt diese Konstruktion, da sie für $N = \infty$ keine sinnvolle Definition liefert. Wir betrachten hier nur den Fall, dass $X(\Omega)$ genau die Werte in einem kompakten Intervall $[a, b]$ annimmt. Dann lässt sich $\mathbb{P}_X$ für Teilintervalle $[c, d] \subset [a, b]$ als

$$\mathbb{P}_X([c, d]) = (d - c)/(b - a)$$

angeben, die Wahrscheinlichkeit ist also linear proportional zur Intervallgröße. Dies kann man auch als

$$\mathbb{P}_X([c, d]) = \int_c^d p(x)dx = \int_{\mathbb{R}} \chi_{[c,d]}(x) p(x) dx$$

schreiben, wobei

$$p(x) = \begin{cases} 1/(b - a), & x \in [a, b] \\ 0, & x \notin [a, b] \end{cases}$$

ist und *Dichtefunktion* heißt. Diese komplizierte Schreibweise lässt sich mittels

$$\mathbb{P}_X(B) = \int_a^b \chi_B(x) p(x) dx \tag{3.1}$$

auf beliebige Mengen verallgemeinern. Natürlich kann $B$ nicht völlig beliebig sein, denn das Integral sollte natürlich existieren. In der Stochastik verwendet man dabei das Lebesgue–Integral, die Funktion $\chi_B$ muss also Lebesgue–messbar sein. Folglich nennen wir eine Menge $B \subset \mathbb{R}$ *Lebesgue–messbar*, falls ihre charakteristische Funktion $\chi_B$ Lebesgue–messbar ist.

Viele Verteilungen von Zufallsvariablen lassen sich durch Dichtefunktionen $p : \mathbb{R} \to \mathbb{R}$ beschreiben. Auch die zweite Klasse von Zufallsvariablen, die wir betrachten werden und die für die stochastischen DGL von zentraler Bedeutung ist, lässt sich so beschreiben. Dies ist die Klasse der Gauß–verteilten Zufallsvariablen, deren Wahrscheinlichkeitsverteilung durch (3.1) mit

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right). \tag{3.2}$$

gegeben ist. Diese Dichtefunktion hängt von zwei Parametern $\mu \in \mathbb{R}$ und $\sigma \geq 0$ ab. Für Gauß–verteilte (auch *normalverteilte* oder kurz *Gauß'sche*) Zufallsvariablen $X$ schreibt man oft $X \sim N(\mu, \sigma^2)$. Eine Zufallsvariable $X \sim N(0,1)$ heißt *standard–normalverteilt*. Abbildung 3.1 zeigt die Dichtefunktion $p$ für die Standard–Normalverteilung.

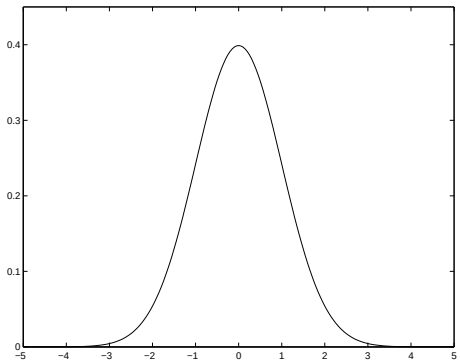



Abbildung 3.1: Dichtefunktion der Gauß'schen Zufallsvariablen für $\mu = 0$ und $\sigma = 1$

Die Gauß'sche Zufallsvariable kann also Werte in ganz $\mathbb{R}$ annehmen ($p$ ist nirgends gleich 0), allerdings ist die Wahrscheinlichkeit betragsmäßig großer Werte sehr gering, da $p$ für große $|x|$ sehr klein wird.

Auf einem Wahrscheinlichkeitsraum können viele Zufallsvariablen $X_1, X_2, \ldots$ gleichzeitig definiert sein. Wichtig ist in diesem Zusammenhang das Prinzip der *Unabhängigkeit*: Zwei auf dem gleichen Wahrscheinlichkeitsraum definierte Zufallsvariablen $X_1 : \Omega \to \mathbb{R}$ und $X_2 : \Omega \to \mathbb{R}$ heißen *unabhängig*, wenn eine Realisierung $X_1(\omega)$ von $X_1$ keine Informationen über die Realisierung $X_2(\omega)$ von $X_2$ (für das gleiche $\omega$) enthält, und umgekehrt. Auf die mathematisch präzise Beschreibung dieser informellen Definition wollen wir hier verzichten; wir wollen sie aber an einem Beispiel erläutern.

**Beispiel 3.2** Betrachte einen Wahrscheinlichkeitsraum, der das (gleichzeitige) Würfeln mit zwei Würfeln beschreibt. Jedes Elementarereignis ist also ein Paar $\omega = (\omega_1, \omega_2)$ von Würfelwerten, wobei die erste Komponente $\omega_1$ das Ergebnis des ersten Würfels und $\omega_2$ das Ergebnis des zweiten Würfels beschreibt (wir nehmen an, dass die zwei Würfel unterscheidbar sind). Dies führt zu $\Omega = \{(1,1), (1,2), \ldots, (6,6)\} \subset \mathbb{N}^2$. Offenbar ist (für ideale Würfel) $\mathbb{P}(\{\omega\}) = 1/36$ für jedes Element $\omega \in \Omega$, da $\Omega$ gerade $36 = 6 \times 6$ Elementarereignisse enthält, die alle gleich wahrscheinlich sind. Wir betrachten nun drei Zufallsvariablen, die als Wert das Ergebnis des ersten Würfels, des zweiten Würfels und die Summe der Würfelwerte ausgeben, also $X_1(\omega) = \omega_1$, $X_2(\omega) = \omega_2$ und $X_3(\omega) = \omega_1 + \omega_2$. Jede Realisierung dieser Variablen entspricht nun einem (simultanen) Wurf der zwei Würfel. Offenbar kann man aus dem Ergebnis des ersten Würfels keinerlei Rückschlüsse auf das Ergebnis des zweiten Würfels ziehen: Unabhängig davon, was beim ersten Würfel herauskam, ist beim zweiten Würfel jeder Wert $1, \ldots, 6$ weiterhin gleich wahrscheinlich. Die Zufallsvariablen $X_1$ und $X_2$ sind also *unabhängig*. Wenn wir aber $X_1$ und $X_3$ betrachten, so ändert sich die Situation: Die Wahrscheinlichkeit, dass $X_3(\omega)$ den Wert 12 annimmt, ist gerade $1/36$ (von den 36 möglichen Würfelkombinationen liefert gerade eine die Summe 12). Wenn wir aber wissen,

dass die Realisierung $X_1(\omega) = \omega_1 = 1$ ist, so kann die Summe $X_3(\omega) = \omega_1 + \omega_2 = 1 + \omega_2$ höchstens den Wert 7 annehmen; die Wahrscheinlichkeit für $X_3(\omega) = 12$ ist unter dieser Vorinformation also gleich 0. Diese Wahrscheinlichkeit unter zusätzlicher Vorinformation nennt man *bedingte Wahrscheinlichkeit*. Da sich nun die bedingte Wahrscheinlichkeit der Zufallsvariablen $X_3$ unter Vorinformation aus $X_1$ von der unbedingten Wahrscheinlichkeit unterscheidet, ist hier gerade der Fall *nicht unabhängiger* Zufallsvariablen gegeben. □

Für eine Folge $X_i$ von Zufallsvariablen mit identischer Verteilung lässt sich die Dichtefunktion grafisch wie folgt approximieren: Man teilt die reellen Zahlen in äquidistante Intervalle $I_j = [jh, (j+1)h]$, $j \in \mathbb{Z}$, $h > 0$ ein und betrachtet eine große Zahl $N$ von Realisierungen $X_1(\omega)$, $X_2(\omega)$, ... und zählt die Häufigkeiten $n_j$, mit denen diese Werte im Intervall $n_j$ liegen. Stellt man die Häufigkeiten $n_j/N$ dann in einem Balkendiagramm dar (man spricht von einem *Histogramm*), so erhält man eine grafische Approximation des Graphen der Dichtefunktion $p$.

### 3.1.2 Zufallszahlen

Am Rechner können wir numerische Approximationen von Zufallsvariablen mit dem Zufallsgenerator erzeugen. Alle höheren Programmiersprachen besitzen zumindest einen einfachen Zufallsgenerator, der eine Folge von (approximativ) unabhängigen gleichverteilten (Pseudo–)Zufallszahlen erzeugt.

Das $\omega$ aus der theoretischen Definition entspricht dabei dem *Seed* ("Samenkorn") des Zufallsgenerators, den man als BenutzerIn mit einem entsprechenden Befehl selbst auswählen kann. Nachdem der Seed gesetzt wurde (was üblicherweise beim Booten des Rechners oder dem Starten der entsprechenden Software automatisch geschieht), liefert jeder Aufruf des Zufallsgenerators eine Realisierung einer neuen unabhängigen Pseudo–Zufallsvariablen. In `C`, z.B. lautet der Befehl zum Setzen des Seed `srand(seed)`, wobei `seed` eine nichtnegative ganze Zahl ist. Zur Generierung von Zufallszahlen dient die Funktion `rand`, die gleichverteilte Integer–Zufallszahlen zwischen `0` und `RAND_MAX` liefert. Andere Programmiersprachen liefern Gleitkomma–Zufallszahlen zwischen 0 und 1, z.B. MATLAB, wo der Befehl zur Zufallszahlenerzeugung `rand(1)` lautet und der Seed durch `rand('state',seed)` gesetzt wird. Die Beispiel–Aufruffolge

```
srand(20); x1=rand(); x2=rand(); ...
```

liefert in der mathematischen Notation gerade

$$\omega = 20, \;\; x1 = X_1(\omega), \;\; x2 = X_2(\omega), \ldots$$

Abhängig von $\omega$ werden also Realisierungen einer Folge unabhängiger Zufallsvariablen erzeugt.

Tatsächlich kann man bei einer gegebenen Zufallszahlenfolge aber gar nicht unterscheiden, ob diese von einer *Folge* unabhängiger Zufallsvariablen $X_i$ stammen oder von *einer* Zufallsvariablen $X$, die für verschiedene unabhängig gezogene $\omega_i$ ausgewertet wird. Die Folge $x_i$ lässt sich daher auch als

$$\omega_1 = 20, \;\; x_1 = X(\omega_1), \;\; x_2 = X(\omega_2), \ldots$$

oder auch als

$$\omega_1 = 20,\ \ x_1 = X_1(\omega_1), \ldots,\ \ x_N = X_N(\omega_1),\ \ \ldots x_{N+1} = X_1(\omega_2), \ldots$$

interpretieren. Bei diesen Interpretationen ist allerdings etwas Vorsicht geboten, da das Computermodell nicht ganz mit dem mathematischen Modell überein stimmt. Die vom Rechner erzeugten $\omega_i$–Werte sind nämlich nur *pseudo–zufällig* sind: Sie liefern zwar (approximativ) die richtigen statistischen Eigenschaften, hängen aber deterministisch von $\omega_1$ ab. Um "echten" Zufall zu erzielen, muss man also auch den Wert $\omega_1$ zufällig wählen, z.B. indem man den Seed abhängig von der internen Uhr des Rechners setzt. Aufgrund der obigen Interpretation genügt es dabei, den Seed einmal beim Start des Programms bzw. beim Start einer Berechnung zu setzen. Beim Testen eines Programms empfiehlt es sich übrigens, den Seed bei Programmbeginn auf einen festen Wert zu setzen. Damit erhält man bei jedem Durchlauf die gleiche Zufallszahlenfolge, womit die Ergebnisse reproduzierbar werden, was z.B. bei der Fehlersuche eine sehr wichtige Eigenschaft ist.

Wir haben bereits erwähnt, dass die Gauß–verteilten Zufallsvariablen für uns besonders wichtig sind. Um aus den gleichverteilten Zufallszahlen $x_i$ zu Gauß–verteilten Zufallszahlen zu kommen, benötigt man eine geeignete Transformation. Zunächst transformieren wir die Zahlen mittels

$$u_i = x_i/\texttt{RAND\_MAX}$$

auf das Intervall $[0, 1]$. Um nun Gauß'sche Zufallszahlen zu erzeugen, muss man jeweils zwei gleichverteilte $u_i$ verwenden und diese mittels

$$y_i = \mu + \sigma\sqrt{-2\ln(u_i)}\cos(2\pi u_{i+1}), \quad y_{i+1} = \mu + \sigma\sqrt{-2\ln(u_i)}\sin(2\pi u_{i+1}),$$

für $i = 1, 3, 5, 7, \ldots$ transformieren. Diese Transformation heißt *Box–Muller–Methode* und liefert eine Folge von $N(\mu; \sigma^2)$–verteilten Zufallszahlen $y_1, y_2, y_3, \ldots$.

### 3.1.3 Der approximative Wiener–Prozess

Um stochastische Differentialgleichungen numerisch lösen zu können, müssen wir nicht nur einzelne Zufallsvariablen, sondern auch geeignete zufällige Funktionen im Rechner erzeugen können. Eine zufällige Funktion ist dabei nichts anderes als eine Funktion $X : \mathbb{R}_0^+ \times \Omega \to \mathbb{R}$, so dass $X(t, \cdot) : \Omega \to \mathbb{R}$ für jedes $t \in \mathbb{R}_0^+$ eine Zufallsvariable ist. Eine solche zeitabhängige Zufallsvariable heißt *stochastischer Prozess*, eine "Funktionsrealisierung" $X(\cdot, \omega) : \mathbb{R} \to \mathbb{R}$ heißt *Pfad* des Prozesses. Der für uns wichtige Prozess ist dabei der sogenannte *Wiener–Prozess*, der mit $W(t, \omega)$ bezeichnet wird und den wir etwas später präzise definieren werden. Hier wollen wir allerdings bereits einen Algorithmus bereit stellen, mit dem man einen solchen Wiener–Prozess auf einem Gitter approximieren kann.

**Algorithmus 3.3 Approximation des Wiener–Prozesses durch Gitterfunktion**
Gegeben: Schrittweite $h$, Gitter $\mathcal{T} = \{t_0, \ldots, t_N\}$ mit $t_i = ih$
Gesucht: $M$ approximative Pfade $\widetilde{W}(t_i, \omega_1)$, …, $\widetilde{W}(t_i, \omega_M)$ des Wiener Prozesses auf $\mathcal{T}$.

(1) Für $j = 1, \ldots, M$:

(2a) Erzeuge $N(0, h)$–verteilte Zufallszahlen $\Delta W_i(\omega_j)$ für $i = 0, \ldots, N-1$

(2b) Erzeuge Gitterfunktionen $\widetilde{W}(t_i, \omega_j)$ für mittels der Rekursion

$$\widetilde{W}(t_0, \omega_j) = 0, \;\; \widetilde{W}(t_{i+1}, \omega_j) = \widetilde{W}(t_i, \omega_j) + \Delta W_i(\omega_j), \;\; \text{für } \; i = 0, \ldots, N-1$$

(3) Ende der $j$–Schleife □

### 3.1.4 Erwartungswert und Varianz

Ausgehend von ihren Verteilungen kann man eine Reihe von Kenngrößen für $X$ definieren; zwei davon sind für uns wichtig.

Die erste wichtige Größe ist der *Erwartungswert* von $X$, der mit $\mathbb{E}\{X\}$ bezeichnet wird. Anschaulich ist dies nichts anderes als der Mittelwert, den man erhält, wenn man über alle möglichen Realisierungen $X(\omega)$ von $X$ bezüglich ihrer Wahrscheinlichkeit mittelt, für eine Folge $X(\omega_1), X(\omega_2), \ldots$ von Realisierungen also

$$\mathbb{E}\{X\} := \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} X(\omega_i). \tag{3.3}$$

Formal kann man den Erwartungwert auch über die Dichtefunktion erklären, nämlich als

$$\mathbb{E}\{X\} := \sum_{x_i \in X(\Omega)} x_i \mathbb{P}_X(x_i) \tag{3.4}$$

für diskrete und, analog,

$$\mathbb{E}\{X\} := \int_{-\infty}^{\infty} x p(x) dx, \tag{3.5}$$

für kontinuierlichne Zufallsvariablen, wobei wir voraussetzen, dass dieses Integral existiert. Für die Gauß–Verteilung errechnet man $\mathbb{E}\{X\} = \mu$. Es ist übrigens nicht trivial zu beweisen, dass diese beiden Definitionen von $\mathbb{E}\{X\}$ übereinstimmen.

Wir betrachten einige Rechenregeln für den Erwartungswert. Aus den Integrationsregeln und der Definition der Dichtefunktion kann man errechnen, dass für eine Funktion $g : \mathbb{R} \to \mathbb{R}$ und eine Zufallsvariable $X$ der Erwartungswert von $g(X)$ durch

$$\mathbb{E}\{g(X)\} = \int_{-\infty}^{\infty} g(x) p(x) dx$$

gegeben ist. Aus (3.3) sieht man leicht, dass der Erwartungswert linear in $X$ ist, d.h. für zwei Zufallsvariablen $X_1$ und $X_2$ und $\alpha_1, \alpha_2 \in \mathbb{R}$ gilt $\mathbb{E}\{\alpha_1 X_1 + \alpha_2 X_2\} = \alpha_1 \mathbb{E}\{X_1\} + \alpha_2 \mathbb{E}\{X_2\}$. Beachte hierbei, dass Summen, Differenzen und reelle Vielfache von Zufallsvariablen $X_1$ und $X_2$ wieder Zufallsvariablen sind, speziell sind Summen von Gauß–verteilten Zufallsvariablen wieder Gauß–verteilte Zufallsvariablen. Falls für $X_1$ und $X_2$ dabei Dichtefunktionen $p_1$ und $p_2$ existieren, so existieren für die kombinierten Zufallsvariablen wiederum Dichtefunktionen. Diese sind i.A. nicht leicht zu berechnen, aber allein die Tatsache, dass sie existieren, ist oft hilfreich.

Eine weitere i.A. nur schwer berechenbare Größe ist der Erwartungswert des Produktes $X_1 X_2$, es sei denn, $X_1$ und $X_2$ sind *unabhängige Zufallsvariablen*. In diesem Fall gilt die einfache Gleichung

$$\mathbb{E}\{X_1 X_2\} = \mathbb{E}\{X_1\} \mathbb{E}\{X_2\}, \tag{3.6}$$

die für beliebige Zufallsvariablen im Allgemeinen nicht gilt.

Die zweite wichtige Größe ist die *Varianz* von $X$, geschrieben $\mathrm{Var}(X)$. Sie gibt an, wie weit sich die einzelnen Realisierungen im Mittel vom Erwartungswert entfernen können, man sagt auch, wie weit die Werte "streuen". Mit $\mu = \mathbb{E}\{X\}$ ist sie definiert durch

$$\mathrm{Var}(X) = \mathbb{E}\{(X-\mu)^2\}.$$

Eine einfache Rechnung ergibt

$$\mathrm{Var}(X) = \mathbb{E}\{(X-\mu)^2\} = \mathbb{E}\{X^2 - 2\mu X + \mu^2\} = \mathbb{E}\{X^2\} - 2\mu\mathbb{E}\{X\} + \mu^2 = \mathbb{E}\{X^2\} - \mu^2.$$

Für die Gauß–Verteilung errechnet man $\mathrm{Var}(X) = \sigma^2$.

Erwartungswert und Varianz stellen zwei wesentliche charakteristische Größen von Zufallsvariablen dar. In unseren Betrachtungen wollen wir uns deswegen auf die numerische Berechnung dieser Größen konzentrieren.

### 3.1.5 Der Wiener–Prozess

Mit Hilfe des Erwartungswertes und der Varianz kann man nun den Wiener–Prozess formal definieren. Dieser wird üblicherweise mit $W$ bezeichnet wird und ist für $t \geq 0$ definiert. Die Definition des Wiener Prozesses ergibt sich aus der von N. Wiener[2] eingeführten mathematischen Beschreibung der *Brown'schen Bewegung*, die in der Physik die zufällige Bewegung eines auf einer Wasseroberfläche schwimmenden Teilchens beschreibt. Formal verlangt man die folgenden Bedingungen:

(i) $W(t,\cdot)$ ist eine Gauß–verteilte Zufallsvariable mit $\mathbb{E}\{W(t,\cdot)\} = 0$ und $\mathrm{Var}(W(t,\cdot)) = t$, also $W(t,\cdot) \sim N(0,t)$

(ii) Für $t_2 \geq t_1 \geq 0$ sind die *Inkremente* $W(t_2,\cdot) - W(t_1,\cdot)$ Gauß–verteilte Zufallsvariablen mit $\mathbb{E}\{W(t_2,\cdot) - W(t_1,\cdot\} = 0$ und $\mathrm{Var}(W(t_2,\cdot) - W(t_1,\cdot))) = t_2 - t_1$, also $W(t_2,\cdot) - W(t_1,\cdot) \sim N(0, t_2 - t_1)$

(iii) Für $s_2 \geq s_1 \geq t_2 \geq t_1 \geq 0$ sind die Inkremente $W(t_2,\cdot) - W(t_1,\cdot)$ und $W(s_2,\cdot) - W(s_1,\cdot)$ unabhängige Zufallsvariablen.

Beachte, dass wir oben die Gauß–Verteilung nur für $\sigma > 0$ definiert haben. Hier erhalten wir für $t = 0$ die Bedingung $\mathrm{Var}(W(0)) = 0$, womit einfach $\mathbb{P}_{W(0)}(\mu) = 1$ gemeint ist. Mit anderen Worten ist die Zufallsvariable $W(0)$ hier also konstant gleich $\mu$, hier also gleich $\mu = 0$.

Man kann beweisen, dass der approximative Wiener–Prozess aus Algorithmus 3.3 die obigen Bedingungen an allen Gitterpunkten $t_i$ erfüllt ((i) und (ii) sind Übungsaufgabe, (iii) folgt aus der Unabhängigkeit der Zufallszahlen in Algorithmus 3.3). Die Approximation ist also an den Gitterpunkten tatsächlich exakt, lediglich zwischen den Gitterpunkten stimmen die Approximation und der exakte Prozess nicht überein.

[2] US-amerikanischer Mathematiker, 1894–1964

## 3.2 Konvergenz– und Approximationsbegriffe

In der Einleitung wurde bereits erwähnt, dass wir bei der numerischen Lösung stochastischer DGL i.A. nicht benötigen, dass *jeder* Lösungspfad $X(t,\omega) = X(t,t_0,x_0,\omega)$ möglichst genau numerisch approximiert wird. Statt dessen genügt es uns üblicherweise, dass gewisse statistische Größen mit hinreichend hoher Genauigkeit widergegeben werden.

Je nachdem, welche Größen man ausrechnen möchte, benötigt man verschiedene Konvergenzbegriffe. Tatsächlich gibt es in der stochastischen Numerik eine riesige Menge von Konvergenzbegriffen, von denen wir hier nur zwei herausheben wollen, die für unsere Zwecke besonders wichtig sind.

Wir betrachten dabei einen reellwertigen[3] stochastischen Prozess $X$ auf dem Intervall $[0,T]$. Dieser soll durch eine Folge numerischer Approximationen $\widetilde{X}_i$ auf Gittern $\mathcal{T}_i = \{0, h_i, 2h_i, \ldots, N_i h_i\}$ mit Schrittweiten $h_i = T/N_i$ (also $N_i h_i = T$) approximiert werden mit $N_i \to \infty$ für $i \to \infty$. Wir schreiben kurz $X(T) = X(T,\omega)$ und $\widetilde{X}_i(T) = \widetilde{X}_i(T,\omega)$.

**Definition 3.4** (i) Die Folge $\widetilde{X}_i$ von stochastischen Prozessen heißt *starke Approximation* für $X$ zur Zeit $T$ bzgl. einer Funktion $g : \mathbb{R} \to \mathbb{R}$, wenn die Bedingung

$$\lim_{i\to\infty} \mathbb{E}\{|g(X(T)) - g(\widetilde{X}_i(T))|\} = 0$$

gilt. Sie heißt *starke Approximation mit Ordnung* $\gamma > 0$, falls für alle $i \geq i_0$ zusätzlich die Abschätzung

$$\mathbb{E}\{|g(X(T)) - g(\widetilde{X}_i(T))|\} \leq C h_i^\gamma$$

für ein $C > 0$ gilt.

(ii) Die Folge $\widetilde{X}_i$ von stochastischen Prozessen heißt *schwache Approximation* für $X$ zur Zeit $T$ bzgl. einer Funktion $g : \mathbb{R} \to \mathbb{R}$, wenn die Bedingung

$$\lim_{i\to\infty} |\mathbb{E}\{g(X(T))\} - \mathbb{E}\{g(\widetilde{X}_i(T))\}| = 0$$

gilt. Sie heißt *schwache Approximation mit Ordnung* $\beta > 0$, falls für alle $i \geq i_0$ zusätzlich die Abschätzung

$$|\mathbb{E}\{g(X(T))\} - \mathbb{E}\{g(\widetilde{X}_i(T))\}| \leq C h_i^\beta$$

für ein $C > 0$ gilt. □

Eine starke Approximation liefert also eine Approximation der Pfade, bei denen zwar u.U. nicht jeder Pfad gut approximiert wird, der Fehler aber zumindest im Mittel klein werden muss. Dies ist die natürliche Verallgemeinerung der deterministischen Approximation.

Bei einer schwachen Approximation hingegen (die sich, wie wir sehen werden, mit deutlich geringerem numerischen Aufwand berechnen lässt), können die einzelnen Pfade ganz anders aussehen, wichtig ist hier nur, dass die bezüglich $g$ ermittelten statistischen Eigenschaften gleich sind.

[3] Natürlich lässt sich dies auf vektorwertige Prozesse verallgemeinern, wir wollen die Definition aber technisch einfach halten.

Kurz gesagt können wir festhalten: Eine starke Approximation liefert eine *Approximation der Pfade* von $X$ während eine schwache Approximation eine *Approximation der statistischen Eigenschaften* von $X$ liefert, jeweils gemessen bzgl. $g$.

Das folgende Beispiel zeigt typische Anwendungen starker und schwacher Approximationen.

**Beispiel 3.5** (i) Wenn wir den Erwartungswert $\mathbb{E}\{X(T)\}$ numerisch berechnen wollen, genügt eine schwache Approximation bzgl. $g(x) = x$, denn dafür gilt

$$\lim_{i\to\infty} |\mathbb{E}\{X(T))\} - \mathbb{E}\{\widetilde{X}_i(T)\}| = 0,$$

und damit

$$\lim_{i\to\infty} \mathbb{E}\{\widetilde{X}_i(T)\} = \mathbb{E}\{X(T)\}.$$

(ii) Wenn wir die Varianz $\mathrm{Var}(X(T))$ numerisch berechnen wollen, genügt eine schwache Approximation bzgl. $g(x) = x$ und $g(x) = x^2$ (Übungsaufgabe).

(iii) Für kompliziertere Berechnungen reichen schwache Approximationen allerdings i.A. nicht mehr aus. Wenn wir z.B. für einen Wert $c \in \mathbb{R}$ die "minimale Überschreitungszeit"

$$t(\omega) := \inf\{t \geq 0 \,|\, X(t,\omega) \geq c\}$$

für ein vorgegebenes $c \in \mathbb{R}$ definieren, so erhalten wir eine weitere Zufallsvariable $t(\omega)$. Dies könnte z.B. die (zufällige) Zeit des Ausfalls einer Maschine in einem mechanischen Modell oder die Überschreitung eines vorgegebenen Kursniveaus in der Finanzmathematik bedeuten.

Ziel einer numerischen Berechnung könnte es nun sein, die mittlere Zeit $\mathbb{E}\{t(\omega)\}$ zu ermitteln. Da sich diese Größe nicht als $\mathbb{E}\{g(X(T))\}$ schreiben lässt, genügt hier eine schwache Approximation nicht mehr. □

Neben der oben definierten Approximationen gibt es noch weitere Approximationsbegriffe, z.B. die *(starke) Quadratmittel–Approximation*

$$\lim_{i\to\infty} \mathbb{E}\{|X(T) - \widetilde{X}_i(T)|^2\} = 0.$$

Man kann zeigen, dass hieraus die starke Approximation bzgl. $g(x) = x$ und die schwache Approximation bzgl. $g(x) = x^2$ folgt.

Die Namen "stark" und "schwach" legen nahe, dass (i) eine stärkere Eigenschaft als (ii) ist. Das folgende Lemma zeigt, dass dies tatsächlich stimmt.

**Lemma 3.6** Wenn $\widetilde{X}_i(T)$ eine starke Approximation von $X(T)$ bzgl. $g$ ist, so ist $\widetilde{X}_i(T)$ auch eine schwache Approximation bzgl. $g$. Hierbei bleibt die Konvergenzordnung erhalten, d.h. es gilt $\beta \geq \gamma$.

**Beweis:** Aus der Stochastik weiß man, dass eine Dichtefunktion $p_g$ mit der Eigenschaft

$$\mathbb{E}\{g(q(X(T)))\} - \mathbb{E}\{g(q(\widetilde{X}_i(T)))\} = \int_{-\infty}^{\infty} q(x) p_g(x) dx$$

für alle reellen integrierbaren Funktionen $q$ existiert. Nach Dreiecksungleichung gilt damit

$$\begin{aligned} |\mathbb{E}\{g(X(T))\} - \mathbb{E}\{g(\widetilde{X}_i(T))\}| &= \left| \int_{-\infty}^{\infty} x p_g(x) dx \right| \\ &\leq \int_{-\infty}^{\infty} |x| p_g(x) dx \\ &= \mathbb{E}\{|g(X(T)) - g(\widetilde{X}_i(T))|\}. \end{aligned}$$

Hieraus folgt die Behauptung. □

Für den approximativen Wiener–Prozess aus Algorithmus 3.3 haben wir gesehen, dass er an den Stützstellen gerade die Definition des exakten Wiener–Prozesses erfüllt. Daher kann man jedem Pfad $\widetilde{W}(\cdot, \omega)$ des approximierten Wiener–Prozesses gerade einen Pfad des exakten Prozesses $W(\cdot, \omega)$ mit $\widetilde{W}(t_i, \omega) = W(t_i, \omega)$ für jedes $t_i \in \mathcal{T}$ zuordnen, so dass die Approximation für jedes $g : \mathbb{R} \to \mathbb{R}$ und jeden Gitterpunkt $T \in \mathcal{T}$ die Bedingung

$$\mathbb{E}\{|g(\widetilde{W}(T, \omega)) - g(W(T, \omega))|\} = 0$$

erfüllt. Es handelt sich also um eine "besonders starke" Form der starken Konvergenz.

### 3.2.1 Schwache Approximation des Wiener–Prozesses

Wir wollen nun untersuchen, wie man eine schwache Approximation des Wiener–Prozesses erhalten kann — in der Hoffnung, dass sich der Algorithmus dabei vereinfacht. Der folgende Algorithmus zeigt, wie dies geht. Wir benötigen eine spezielle Form von gleichverteilten diskreten Zufallsvariablen, nämlich die durch

$$X(\Omega) = \{x_1, x_2\}, \quad \mathbb{P}_X(\{x_1\}) = \mathbb{P}_X(\{x_2\}) = \frac{1}{2}$$

definierte *zweipunktverteilte Zufallsvariable* mit den zwei Werten $\{x_1, x_2\}$.

**Algorithmus 3.7 Schwache Approximation des Wiener–Prozesses durch Gitterfunktion**
Gegeben: Schrittweite $h$, Gitter $\mathcal{T} = \{t_0, \ldots, t_N\}$ mit $t_i = ih$
Gesucht: $M$ schwach approximierende Pfade $\widetilde{W}(T, \omega_1)$, …, $\widetilde{W}(T, \omega_M)$ des Wiener Prozesses.

(1) Für $j = 1, \ldots, M$:

(2a) Erzeuge zweipunktverteilte Zufallszahlen $\Delta W_i(\omega_j)$ für $i = 0, \ldots, N-1$ mit $x_1 = -\sqrt{h}$, $x_2 = \sqrt{h}$.

(2b) Erzeuge Gitterfunktionen $\widetilde{W}(t_i, \omega_j)$ für mittels der Rekursion

$$\widetilde{W}(t_0, \omega_j) = 0, \;\; \widetilde{W}(t_{i+1}, \omega_j) = \widetilde{W}(t_i, \omega_j) + \Delta W_i(\omega_j), \;\; \text{für } \; i = 0, \ldots, N-1$$

(3) Ende der $j$–Schleife □

Der große Unterschied in der Konstruktion besteht darin, dass wir hier in jedem Schritt nur endlich viele (nämlich gerade 2) Möglichkeiten für $\Delta W_i(\omega_j)$ haben. Die Anzahl der möglichen Pfade ist demnach endlich, weswegen man hier tatsächlich alle möglichen Pfade mitsamt der Wahrscheinlichkeiten ihres Auftretens berechnet.

Abbildung 3.2 zeigt einen Pfad des Wienerprozesses (durchgezogen) sowie seine schwache Approximation (gestrichelt). Man sieht, dass die Pfade erheblich voneinander abweichen können.

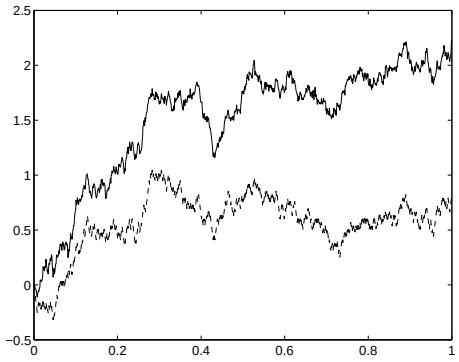



Abbildung 3.2: Schwache Approximation (- -) eines Pfades eines Wiener–Prozesses (—)

Dass dieser Algorithmus trotzdem die richtigen statistischen Eigenschaften besitzt, zeigt das folgende Lemma.

**Lemma 3.8** Der Algorithmus 3.7 liefert eine schwache Approximation für den Wiener–Prozess bzgl. $g(x) = x$ und $g(x) = x^2$ für jeden Gitterpunkt $T = t_i$.

**Beweis:** Für die zweipunktverteilten Zufallszahlen $\Delta W_i$ berechnet man

$$\mathbb{E}\{\Delta W_i(\cdot)\} = -\sqrt{h}/2 + \sqrt{h}/2 = 0 \text{ und } \mathbb{E}\{\Delta W_i(\cdot)^2\} = h/2 + h/2 = h.$$

Aus der Definition folgt

$$\widetilde{W}(t_i, \omega) = \sum_{k=0}^{i-1} \Delta W_k(\omega).$$

Also ergibt sich

$$\mathbb{E}\{\widetilde{W}(t_i, \cdot)\} = \sum_{k=0}^{i-1} \mathbb{E}\{\Delta W_k(\cdot)\} = 0 = \mathbb{E}\{W(t_i, \cdot)\}$$

und, indem wir die Unabhängigkeit der Zufallszahlen und (3.6) ausnutzen, auch

$$\begin{aligned} \mathbb{E}\{\widetilde{W}(t_i, \cdot)^2\} &= \mathbb{E}\left\{\left(\sum_{k=0}^{i-1} \Delta W_k(\cdot)\right)^2\right\} \\ &= \mathbb{E}\left\{\sum_{k=0}^{i-1} \Delta W_k(\cdot)^2 + \sum_{k \neq j} \Delta W_k(\cdot)\Delta W_j(\cdot)\right\} \end{aligned}$$

$$\begin{aligned} &= \sum_{k=1}^{i} \underbrace{\mathbb{E}\{\Delta W_k(\cdot)^2\}}_{=h} + \sum_{k\neq j} \underbrace{\mathbb{E}\{\Delta W_k(\cdot)\}\mathbb{E}\{\Delta W_j(\cdot)\}}_{=0} \\ &= hi \;=\; t_i \;=\; \mathbb{E}\{W(t_i,\cdot)^2\}, \end{aligned}$$

denn es gilt

$$\mathbb{E}\{W(t_i,\cdot)^2\} = \mathbb{E}\{W(t_i,\cdot)^2\} - \underbrace{\mathbb{E}\{W(t_i,\cdot)\}^2}_{=0} = \mathrm{Var}(W(t_i,\cdot)) = t_i.$$

□

## 3.3 Stochastische Differentialgleichungen

Wir werden nun die stochastischen Differentialgleichungen, die wir später numerisch lösen wollen, präzise definieren. Als zufällige Funktion auf der rechten Seite der SDG wollen wir dabei den Wiener–Prozess bzw. von ihm abgeleitete Größen erlauben.

Zur Erläuterung der dabei auftretenden Schwierigkeiten und des nötigen Lösungskonzeptes wollen wir zunächst versuchen, eine SDG aufzustellen, deren Lösung gerade der Wiener–Prozess ist.

Zunächst einige Anmerkungen zur Notation: Da die Lösung einer SDG über den eingehenden Wiener Prozess wieder eine zufällige Funktion — also ein stochastischer Prozess — ist, verwenden wir hier für den gesuchten unbekannten Prozess die groß geschriebene Bezeichnung $X(t)$, bzw. mit Anfangswert $X_0 \in \mathbb{R}^n$ und Anfangszeit $t_0 \in \mathbb{R}$ die Schreibweise $X(t; t_0, X_0)$. Wir erlauben hierbei, dass $X$ vektorwertig, also aus dem $\mathbb{R}^n$ ist, was einfach bedeutet, dass $X = (X_1, X_2, \ldots, X_n)^T$ ist, wobei die $X_i$ reellwertige stochastische Prozesse im oben eingeführten Sinne sind. Zu jedem Pfad $W(t,\omega)$ des eingehenden Wiener Prozesses gehört dann ein Lösungspfad des $X$–Prozesses, den wir mit $X(t; t_0, X_0, \omega)$ bezeichnen.

Die technische Hauptschwierigkeit in der mathematischen Formulierung stochastischer Differentialgleichungen zeigt sich nun bereits bei einer scheinbar trivialen Aufgabe, nämlich dem Problem, eine stochastische Differentialgleichung aufzustellen, deren Lösung gerade der Wiener Prozess ist. Scheinbar trivial ist die Aufgabe deswegen, weil wir ja den Wiener Prozess als gegeben voraussetzen und in der Formulierung verwenden dürfen, weswegen es nahe liegt, einfach die Differentialgleichung

$$\frac{d}{dt}X(t) = \frac{d}{dt}W(t) \tag{3.7}$$

mit Anfangsbedingung $X_0 = W(0)$ zur Anfangszeit $t_0 = 0$ zu verwenden. Das Problem ist jetzt aber: Was verstehen wir unter "$\frac{d}{dt}W(t)$"? Man würde vielleicht versuchen, die Ableitung pfadweise auffassen, d.h., wir berechnen die Ableitung für jeden Pfad $W(t,\omega)$. Nur ist ein typischer Pfad $W(t,\omega)$, wie oben erwähnt, nirgends differenzierbar.

Zunächst einmal bietet es sich an, die Gleichung (3.7) analog zu (2.3) in Integralform

$$X(t) = X_0 + \int_0^t \frac{d}{d\tau}W(\tau)d\tau$$

zu schreiben. Jetzt könnten wir formal integrieren, was uns aber bei der Frage "was ist $\frac{d}{dt}W(t)$?" nicht weiter bringt. Für das obige sogenannte *stochastische Integral* hat sich in der Literatur die kürzere Schreibweise

$$\int_0^t dW(\tau)$$

eingebürgert, die wir hier übernehmen wollen. Dies zeigt die Richtung auf, die wir zur Lösung unseres Problems einschlagen wollen: Anstatt die Ableitung $\frac{d}{dt}W(t)$ zu betrachten, werden wir versuchen, diesem stochastischen Integral eine mathematische Definition zu geben, die

(i) wohldefiniert ist, obwohl $\frac{d}{dt}W(t)$ nicht existiert

(ii) das gewünschte Ergebnis, nämlich $X(t) = W(t)$, liefert

(iii) sich auf allgemeinere Integrale der Form

$$I(F) := \int_{t_0}^{t_1} F(t)dW(t) \tag{3.8}$$

verallgemeinern lässt, damit wir auch kompliziertere SDGs formulieren können. Hierbei ist $F$ wiederum ein stochastischer Prozess, der auf demselben Wahrscheinlichkeitsraum wie $W$ definiert ist.

Wir wollen dieses Konzept nun für Integrale der Form (3.8) angeben. Die hier vorgestellte Lösung geht auf Kiyosi Ito[4] zurück und wurde in den 1940er Jahren entwickelt. Die Idee besteht darin, das Integral (3.8) für jedes Paar von Pfaden $F(t,\omega)$ und $W(t,\omega)$ durch den Limes einer geeigneten Summe zu approximieren. Wir wählen dazu ein $N \in \mathbb{N}$ und eine Folge von Zeiten $\tau_i^{(N)}$, $i = 0, 1, \ldots, N$ mit

$$t_0 = \tau_0^{(N)} < \tau_1^{(N)} < \ldots < \tau_N^{(N)} = t_1$$

und definieren für jedes $\omega \in \Omega$

$$I^{(N)}(F)(\omega) := \sum_{i=0}^{N-1} F(\tau_i^{(N)}, \omega)(W(\tau_{i+1}^{(N)}, \omega) - W(\tau_i^{(N)}, \omega)).$$

Das Integral (3.8) wird nun über den Limes dieser Summe definiert. Betrachte eine Familie von Folgen $\tau_i^{(N)}$ für $N \in \mathbb{N}$ mit $\lim_{N\to\infty} \delta(N) = 0$, wobei $\delta(N) := \max_{i=1,\ldots,N} \tau_i^{(N)} - \tau_{i-1}^{(N)}$ ist. Dann definieren wir

$$I(F) := \lim_{N\to\infty} I^{(N)}(F). \tag{3.9}$$

Diese Definition wirft zunächst eine Reihe von Fragen auf, denn da die Pfade des Wiener Prozesses sehr unangenehme Funktionen sein können, ist nicht garantiert, dass dieser Limes für jedes $\omega$ überhaupt existiert. Tatsächlich lag der Haupttrick von Ito darin, zu definieren, was der Limes in (3.9) eigentlich bedeuten soll. Dieser Limes ist nämlich *nicht*

[4] japanischer Mathematiker, *1915, oft auch Itô geschrieben

pfadweise zu verstehen (in dem Sinne, dass wir $\lim_{N\to\infty} I^{(N)}(F)(\omega)$ für jedes $\omega \in \Omega$ bilden), sondern man muss die Werte $I^{(N)}(F)$ ebenso wie das Integral $I(F)$ wieder als Zufallsvariablen $I^{(N)}(F) : \Omega \to \mathbb{R}$ bzw. $I(F) : \Omega \to \mathbb{R}$ auffassen. Für Zufallsvariablen gibt es verschiedene Konvergenzbegriffe und der hier geeignete ist der oben bereits definierte Begriff der *Quadratmittel–Konvergenz*, der hier wie folgt verwendet wird: Eine Folge von Zufallsvariablen $X_N : \Omega \to \mathbb{R}$ konvergiert im Quadratmittel–Sinne gegen eine Zufallsvariable $X : \Omega \to \mathbb{R}$, falls

$$\lim_{N\to\infty} \mathbb{E}(|X_N - X|^2) = 0$$

gilt. Mit diesem Konvergenzbegriff kann man zeigen, dass die Folge $I^{(N)}(F)$ (unter geeigneten Bedingungen an $F$) tatsächlich konvergiert und (3.9) also wohldefiniert ist. Das resultierende Integral wird *Ito–Integral* genannt und es besitzt tatsächlich die oben aufgeführten gewünschten Eigenschaften (i)–(iii). Insbesondere gilt die Eigenschaft

$$\int_{t_1}^{t_2} dW(t) = \int_{t_0}^{t_2} dW(t) - \int_{t_0}^{t_1} dW(t) = W(t_2) - W(t_1). \tag{3.10}$$

Mit Hilfe des Ito–Integrals können wir die informelle Schreibweise aus der Einleitung dieses Kapitels mathematisch präzise schreiben. Statt der üblichen Differentialgleichungsschreibweise schreibt man *Ito–stochastische Differentialgleichungen* nämlich als

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t). \tag{3.11}$$

Dies ist nur eine symbolische Schreibweise; was mit (3.11) tatsächlich gemeint ist, ist die längere Integralschreibweise

$$X(t) = X(t_0) + \int_{t_0}^{t} a(t, X(t))dt + \int_{t_0}^{t} b(t, X(t))dW(t),$$

bei der das zweite Integral gerade das Ito–Integral ist. Diese Definition liefert eine mathematisch fundierte und brauchbare Definition stochastischer Differentialgleichungen. Falls $b(t, x) \equiv 0$ ist, also kein stochastischer Anteil vorhanden ist, reduziert sich (3.11) auf

$$X(t) = X(t_0) + \int_{t_0}^{t} a(t, X(t))dt \quad \Longleftrightarrow \quad \frac{d}{dt}X(t) = a(t, X(t)),$$

also auf die wohlbekannte deterministische gewöhnliche Differentialgleichung. Der deterministische Anteil $a(t, x)$ wird auch "Drift" genannt, während der stochastische Anteil $b(t, x)$ oft als "Diffusion" bezeichnet wird.

Natürlich lässt sich (3.11) in vielfacher Hinsicht erweitern, z.B. kann man statt nur einem $W$ mehrere unabhängige Wiener Prozesse $W^1, \ldots, W^m$ eingehen lassen, was zur Gleichung

$$dX(t) = a(t, X(t))dt + \sum_{j=1}^{m} b_j(t, X(t))dW^j(t)$$

führt.

**Bemerkung 3.9** Es sollte hier erwähnt werden, dass es eine weitere sinnvolle stochastische Integraldefinition gibt, die auf R. Stratonovich[5] zurück geht. Das Stratonovich–Integral

$$\int_{t_0}^{t_1} F(t) \circ dW(t)$$

wird über eine ähnliche Limes–Bildung wie das Ito–Integral definiert und liefert ebenfalls eine mathematisch fundierte Definition stochastischer DGLs. Die beiden Integrale unterscheiden sich allerdings in den Rechenregeln ebenso wie in der Form der Lösungen. Die zugehörigen *Stratonovich–SDGs* werden in der Form

$$dX(t) = a(t, X(t))dt + b(t, X(t)) \circ dW(t)$$

geschrieben. Aus Zeitgründen können wir auf diese zweite Definition und auf die Gemeinsamkeiten und Unterschiede zum Ito–Integral hier nicht näher eingehen. □

## 3.4 Numerische Verfahren

### 3.4.1 Das stochastische Euler–Verfahren

Das stochastische Euler–Verfahren lässt sich genau wie sein deterministisches Gegenstück heuristisch herleiten:

Für die Lösung zur Zeit $t + h$ gilt die Integralgleichung

$$X(t+h) = X(t) + \int_t^{t+h} a(\tau, X(\tau))d\tau + \int_t^{t+h} b(\tau, X(\tau))dW(\tau),$$

für die mit $a(\tau, X(\tau)) \approx a(t, X(t))$, $b(\tau, X(\tau)) \approx b(t, X(t))$ und (3.10) die Approximation

$$\approx X(t) + ha(t, X(t)) + \Delta W(t)b(t, X(t))$$

mit $\Delta W(t) = W(t + h) - W(t)$ gilt.

Dies führt auf das stochastische Euler–Verfahren

$$\Phi(t, X, h, W, \omega) = X(\omega) + ha(t, X(\omega)) + \Delta W(t, \omega)b(t, X(\omega)). \tag{3.12}$$

Dies ist das einfachste Beispiel eines *stochastischen Einschrittverfahrens*, deren Anwendung wir als Algorithmus formulieren wollen.

**Algorithmus 3.10 (Lösung einer SDG mit stoch. Einschritt–Verfahren)**
Gegeben: Schrittweite $h$, Gitter $\mathcal{T} = \{t_0, \ldots, t_N\}$ mit $t_i = ih$
Gesucht: $M$ approximative Pfade $\widetilde{X}(\cdot, \omega_1)$, …, $\widetilde{X}(\cdot, \omega_M)$ für die SDG (3.11) auf $\mathcal{T}$ mit $X(t_0, \omega_j) = x_0$.

(0) Erzeuge $M$ approximative Pfade $\widetilde{W}(\cdot, \omega_j)$ des Wiener–Prozesses auf $\mathcal{T}$

(1) Für $j = 1, \ldots, M$:

[5] russischer Mathematiker, 1930–1997

(2) Erzeuge Gitterfunktionen $\widetilde{X}(t_i, \omega_j)$ mittels der Rekursion $\widetilde{X}(t_0, \omega_j) = x_0$,

$$\widetilde{X}(t_{i+1}, \omega_j) = \Phi(t_i, \widetilde{X}(t_i, \omega_j), h, \widetilde{W}, \omega_j)$$

für $i = 0, \ldots, N-1$

(3) Ende der $j$–Schleife □

Natürlich kann man die Implementierung noch optimieren, statt z.B. die Pfade $\widetilde{W}(\cdot, \omega_j)$ im Voraus zu berechnen, kann man die im Euler–Verfahren benötigten Zufallszahlen $\Delta W$ auch direkt an der Stelle per Zufallsgenerator erzeugen, an der man sie benötigt. Der Algorithmus 3.10 soll in erster Linie den prinzipiellen Ablauf einer solchen Simulation verdeutlichen.

### 3.4.2 Die stochastische Taylor–Entwicklung

Um die Konvergenzordnung dieses Verfahrens zu untersuchen, benötigen wir noch etwas Vorbereitung. Wie im deterministischen Fall werden wir die Taylor–Entwicklung der Lösung $X$ betrachten, wobei wir uns auf den eindimensionalen Fall beschränken werden, da die höherdimensionalen Versionen einen sehr großen technischen Aufwand bei der Notation bedeuten.

Um die Taylor–Entwicklung herzuleiten, benötigen wir zunächst eine geeignete Kettenregel. Zur Erinnerung: Im deterministischen Fall, haben wir die aus der Kettenregel folgende Gleichung

$$\frac{d}{dt} g(t, x(t)) = \frac{\partial g}{\partial t}(t, x(t)) + \frac{\partial g}{\partial x}(t, x(t)) f(t, x(t))$$

verwendet, um die Taylor–Entwicklung entlang von Lösungen herzuleiten (vgl. den Beweis von Satz 2.13).

Man kann diese Gleichung als gewöhnliche Differentialgleichung für die Funktion $y(t) = g(t, x(t))$ interpretieren, und in diesem Sinne lässt sich die Gleichung auf den stochastischen Fall verallgemeinern, wie das folgende Lemma zeigt, das als Ito–Lemma bekannt ist.

**Lemma 3.11** Sei $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ eine zweimal stetig differenzierbare Funktion und sei $X(t)$ die Lösung einer reellwertigen Ito–SDG vom Typ (3.11). Dann erfüllt $k(t, X(t))$ die Gleichung

$$\begin{aligned} dk(t, X(t)) &= \left( \frac{\partial k}{\partial t}(t, X(t)) + \frac{\partial k}{\partial x}(t, X(t)) a(t, X(t)) + \frac{1}{2} \frac{\partial^2 k}{\partial x^2} b(t, X(t))^2 \right) dt \\ &+ \frac{\partial k}{\partial x}(t, X(t)) b(t, X(t)) dW(t), \end{aligned}$$

wobei $W$ hier gerade der Wiener Prozess aus der SDG ist, die $X(t)$ erfüllt. Diese Formel wird auch *Ito–Formel* genannt.

**Beweisidee:** Wir werden hier keinen vollständigen Beweis betrachten, wollen aber begründen, warum die (im Vergleich zur deterministischen Formel ungewöhnliche) zweite

Ableitung von $k$ nach $x$ hier auftritt. Wir betrachten dazu zwei Zeitpunkte $t$ und $t+\Delta t$. Dann gilt $X(t+\Delta t) = X(t) + \Delta X(t)$ mit

$$\begin{aligned}\Delta X(t) &= \int_t^{t+\Delta t} a(s, X(s))ds + \int_t^{t+\Delta t} b(s, X(s))dW_s \\ &\approx a(t, X(t))\Delta t + b(t, X(t))\Delta W(t), \end{aligned} \tag{3.13}$$

wobei $\Delta W(t) = W(t+\Delta t) - W(t)$ ist. Die Approximation (3.13) folgt hierbei aus der Limes–Definition des Ito–Integrals, wenn wir $\tau_{i+1} - \tau_i = \Delta t$ setzen. Wir betrachten nun die Größe

$$\Delta Y := k(t+\Delta t, X(t) + \Delta X(t)) - k(t, X(t)).$$

Aus der Taylor–Entwicklung von $k$ folgt

$$\begin{aligned}\Delta Y &\approx \frac{\partial k}{\partial x}\Delta X + \frac{\partial k}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 k}{\partial x^2}(\Delta X)^2 \\ &\approx \frac{\partial k}{\partial x}\Big(a(t, X(t))\Delta t + b(t, X(t))\Delta W(t)\Big) + \frac{\partial k}{\partial t}\Delta t \\ &\quad + \frac{1}{2}\frac{\partial^2 k}{\partial x^2}\Big(a(t, X(t))^2(\Delta t)^2 + 2a(t, X(t))b(t, X(t))\Delta t\Delta W(t) + (b(t, X(t))\Delta W(t))^2\Big),\end{aligned}$$

wobei alle Ableitungen in $(t, X(t))$ ausgewertet werden. Aus den (hier nicht näher behandelten) Rechenregeln für die stochastischen Integrale folgt, dass man zur Berechnung des Limes

$$dk(t, x(t)) = \lim_{\Delta t \to 0} \Delta Y$$

gerade alle Terme der Ordnung $O(\Delta t)$ berücksichtigen muss. Wären alle Größen deterministisch, so blieben hier gerade die Terme mit den ersten Ableitungen von $k$ stehen. Im stochastischen Fall ist aber auch $(\Delta W(t))^2$ ein Term der Ordnung $O(\Delta t)$, denn es gilt

$$\mathbb{E}((\Delta W(t))^2) = \mathbb{E}(\Delta W(t)^2) - \underbrace{\mathbb{E}(\Delta W(t))^2}_{=0} = \mathrm{Var}(\Delta W(t)) = \mathrm{Var}(W(t+\Delta t) - W(t)) = \Delta t.$$

Der Term

$$\frac{1}{2}\frac{\partial^2 k}{\partial x^2} b(t, X(t))^2 (\Delta W(t))^2$$

muss also mit berücksichtigt werden. Führt man nun den Grenzübergang für $\Delta t \to 0$ im richtigen stochastischen Sinne durch, so erhält man gerade die behauptete Formel. □

Analog zum deterministischen Fall führen wir nun geeignete Differentialoperatoren ein. Zunächst definieren wir

$$\begin{aligned}L^{(0)}k(t,x) &= \frac{\partial k}{\partial t}(t,x) + a(t,x)\frac{\partial k}{\partial x}(t,x) + \frac{1}{2}b(t,x)^2\frac{\partial^2 k}{\partial x^2}(t,x) \\ L^{(1)}k(t,x) &= b(t,x)\frac{\partial k}{\partial x}(t,x)\end{aligned}$$

Die Ito–Formel lässt sich damit kurz als

$$dk(t,x) = L^{(0)}k(t,x)dt + L^{(1)}k(t,x)dW(t)$$

schreiben.

Für die Taylor–Formel benötigen wir iterierte Anwendungen dieser Operatoren, allerdings nicht nur für jeden der zwei Operatoren, sondern auch für gemischte Iterierte. Dazu definieren wir für einen *Multiindex–Vektor* $\alpha = (\alpha_1, \ldots, \alpha_l)$ mit $\alpha_i \in \{0, 1\}$ den Operator

$$L^\alpha := L^{(\alpha_1)} L^{(\alpha_2)} \cdots L^{(\alpha_l)}.$$

Mit $l(\alpha)$ bezeichnen wir dabei die Länge von $\alpha$ und mit $n(\alpha)$ die Anzahl der Null–Einträge, also z.B. $l((0,0,1)) = 3$, $n((0,0,1)) = 2$. Die Menge aller solchen Multiindizes bezeichnen wir mit $\mathcal{M}$.

Zusätzlich zu den Differentialoperatoren benötigen wir geeignete von $\alpha$ abhängige Integralausdrücke, die definiert sind durch

$$I^\alpha[t_0, t_1] := \int_{t_0}^{t_1} \int_{t_0}^{s_l} \cdots \int_{t_0}^{s_2} 1 dv^{\alpha_1}(s_1) \cdots dv^{\alpha_{l-1}}(s_{l-1}) dv^{\alpha_l}(s_l)$$

mit $v^0(s_i) = s_i$ und $v^1(s_i) = W(s_i)$. Für $\alpha = (0, 1)$ erhalten wir also z.B.

$$I^\alpha[t_0, t_1] = \int_{t_0}^{t_1} \int_{t_0}^{s_2} 1 ds_1 dW(s_2)$$

(die Integrationsvariablen zählen "von innen nach außen"). Statt für die "1" kann man dies auch für beliebige reellwertige Funktionen definieren, was wir für unsere Zwecke aber nicht benötigen. Beachte, dass für $\alpha = (0, \ldots, 0)$ mit $l(\alpha) = l$ gerade die Gleichung

$$I^\alpha[t_0, t_1] = \frac{1}{l!}(t_1 - t_0)^l$$

gilt. Dies sind gerade die Integrale, die in der deterministischen Taylor–Entwicklung auftreten.

Mit Hilfe des Ito–Lemmas kann man nun die folgenden stochastischen Taylor–Approximationen beweisen. Hierbei nehmen wir an, dass $a$ und $b$ jeweils hinreichend oft differenzierbar sind, durch eine lineare Schranke beschränkt sind und ihre Ableitungen durch Polynome beschränkt sind.

**Satz 3.12** (i) Für jedes $\gamma = 0.5,\ 1,\ 1.5,\ 2, \ldots$ ist der stochastische Prozess

$$T_\gamma(h) := X_0 + \sum_{\alpha \in \mathcal{A}_\gamma^{stark}} I^\alpha[t_0, t_0 + h] L^\alpha k(t_0, X_0)$$

mit

$$\mathcal{A}_\gamma^{stark} := \{\alpha \in \mathcal{M} \,|\, l(\alpha) + n(\alpha) \leq 2\gamma \text{ oder } l(\alpha) = n(\alpha) = \gamma + 1/2\}$$

für die Funktion $k(t, x) = x$ eine starke Approximation von $X(t_0 + h; t_0, X_0)$ der Ordnung $O(h^{2\gamma+1})$ im Quadratmittel–Sinne.

(ii) Für jedes $\beta \in \mathbb{N}$ ist der stochastische Prozess

$$S_\beta(h) := X_0 + \sum_{\alpha \in \mathcal{A}_\beta^{schwach}} I^\alpha[t_0, t_0 + h] L^\alpha k(t_0, X_0)$$

mit

$$\mathcal{A}_\beta^{schwach} := \{\alpha \in \mathcal{M} \,|\, l(\alpha) \leq \beta\}$$

für die Funktion $k(t,x) = x$ eine schwache Approximation von $X(t_0+h; t_0, X_0)$ der Ordnung $O(h^{\beta+1})$ bzgl. jedes Polynoms $g : \mathbb{R} \to \mathbb{R}$.

Der Ausdruck $L^\alpha k(t_0, X_0)$ ist hierbei wie folgt zu verstehen: Wir wenden den oben definierten Operator $L^\alpha$ auf die Funktion $k(t,x) = x$ an und werten die entstehende Funktion $L^\alpha k$ in $(t,x) = (t_0, X_0)$ aus.

Wir wollen diesen Satz nicht beweisen; er beruht auf einer mittels der Ito–Formel hergeleiteten Taylor–Entwicklung für die Integranden der Gleichung

$$X(t+h) = X(t) + \int_t^{t+h} a(t, X(t))dt + \int_t^{t+h} b(t, X(t))dW(t),$$

bei der die Restterme der geeigneten Ordnung weggelassen werden. Die Mehrfachintegrale treten dabei deswegen auf, weil wir die Integranden durch approximative Integralformeln ersetzen.

### 3.4.3 Stochastische Taylor–Verfahren

Aus den Taylor–Approximationen kann man direkt die sogenannten *stochastischen Taylor–Verfahren* erhalten, indem wir

$$\Phi(t, X, h, W, \omega) = X + \sum_{\alpha \in \mathcal{A}} I^\alpha[t, t+h] L^\alpha k(t, X)$$

für die jeweilige Indexmenge $\mathcal{A}$ setzen, also gerade die Prozesse $T_\gamma$ bzw. $S_\beta$ aus Satz 3.12 mit $t_0 = t$ und $X(t_0) = X$ verwenden. Die Anwendung der Verfahren auf gegebene stochastische Differentialgleichungen erfolgt gemäß Algorithmus 3.10.

Für diese Verfahren gilt der folgende Konvergenzsatz.

**Satz 3.13** Unter geeigneten Lipschitz– und Differenzierbarkeitsbedingungen an $a$ und $b$ gelten die folgenden Aussagen.

(i) Das auf der starken Taylor–Approximation $T_\gamma$ beruhende Taylor–Verfahren liefert eine starke Approximation bzgl. $g(x) = x$ mit der Ordnung $\gamma$.

(ii) Das auf der schwachen Taylor–Approximation $S_\beta$ beruhende Taylor–Verfahren liefert eine schwache Approximation bzgl. jedes Polynoms $g : \mathbb{R} \to \mathbb{R}$ mit der Ordnung $\beta$.

**Beweisidee:** Der Beweis beruht auf einer iterativen Anwendung von Satz 3.12, der hier die Rolle der Konsistenz bei den deterministischen Verfahren spielt.

Aus den (hier nicht näher ausgeführten) Lipschitz–Bedingungen erhält man eine geeignete Lipschitz–Eigenschaft von $\Phi$ in $x$, die es erlaubt, Satz 3.12 analog zur Konsistenzabschätzung im deterministischen Fall iterativ anzuwenden.

Genau wie dort "verliert" man dabei durch die Aufsummierung von $\sim 1/h$ Summanden eine Ordnung, so dass man im schwachen Fall gerade die behauptete Approximationsordnung $\beta$ erhält.

Im starken Fall erhält man damit zunächst eine Quadratmittel–Approximation der Ordnung $O(h^{2\gamma})$, woraus man mit Rechenregeln für den Erwartungswert für ein geeignetes $C > 0$ die Ordnung $C\sqrt{O(h^{2\gamma})}) = O(h^{\gamma})$ für die starke Approximation bzgl. $g(x) = x$ berechnet. □

Als Beispiel betrachten wir die Fälle $\gamma = 0.5$, $\gamma = 1$ und $\beta = 1$. Die Indexmengen sind dabei gegeben durch

$$\begin{aligned} \mathcal{A}_{0.5}^{stark} &= \{(0), (1)\} \\ \mathcal{A}_{1}^{stark} &= \{(0), (1), (1,1)\} \\ \mathcal{A}_{1}^{schwach} &= \{(0), (1)\} = \mathcal{A}_{0.5}^{stark} \end{aligned}$$

Damit erhalten wir für $t = t_0$ und $X = X(t_0)$ die Approximationen

$$\begin{aligned} T_{0.5} = S_1 &= X + ha(t,X) + \Delta W(t) b(t,X) \\ T_1 &= X + ha(t,X) + \Delta W(t) b(t,X) + \frac{1}{2}((\Delta W(t))^2 - h) b(t,X) b'(t,X) \end{aligned}$$

wobei $b' = \partial b / \partial x$ ist (Übungsaufgabe).

Wir sehen, dass $T_{0.5} = S_1$ gerade unser stochastisches Euler–Verfahren ist. Dieses Verfahren besitzt also die starke Ordnung $\gamma = 0.5$ und die schwache Ordnung $\beta = 1$.

Das Verfahren für $T_1$ ist ein neues Verfahren, das sogenannte *Milstein–Verfahren*. Es ist das einfachste Verfahren mit starker Ordnung $\gamma = 1$. Wegen

$$T_1 = S_1 + \frac{1}{2}((\Delta W(t))^2 - h) b(t,X) b'(t,X)$$

sieht man hier deutlich, welcher Term beim Euler–Verfahren ergänzt werden muss, um die starke Ordnung 1 zu erhalten.

Bei diesem Verfahren kann man geeignete Rechenregeln für das Integral $I^{(1,1)}$ ausnutzen, um dieses mittels $\Delta W$ auszudrücken. Für höhere gemischte Mehrfachintegrale ist dies i.A. nicht mehr möglich, so dass man die hierbei auftretenden Integrale u.U. geeignet approximieren muss.

**Bemerkung 3.14** Bei den schwachen Taylor–Verfahren genügt es, den eingehenden Wiener–Prozess durch eine schwache Approximation numerisch zu bestimmen, wobei sich aus dem Konvergenzbeweis allerdings gewisse Bedingungen ergeben, die von $\Delta\widetilde{W}$ erfüllt werden müssen. Die schwache Approximation aus Algorithmus 3.7 z.B. ist für das Euler–Verfahren $S_1$ geeignet, allerdings nicht mehr für das Verfahren $S_2$, da hierfür die schwache Approximation von $W$ bzgl. weiterer Funktionen $g$ (zusätzlich zu $g(x) = x$ und $g(x) = x^2$) benötigt wird. □

### 3.4.4 Verfahren vom Runge–Kutta–Typ

Wie bereits bei den deterministischen Differentialgleichungen ist es auch bei den stochastischen Gleichungen i.A. unpraktisch, wenn die Ableitungen der Funktionen $a$ und $b$ in den numerischen Schemata auftauchen.

In diesem abschließenden Abschnitt wollen wir daher kurz erläutern, wie man die Ableitungen in den Taylor–Verfahren durch nur von $a$ und $b$ abhängige Ausdrücke ersetzen kann. Während die Runge–Kutta–Verfahren für deterministische Differentialgleichungen weitgehend unabhängig von den Taylor–Verfahren entwickelt wurden, verlief die Entwicklung bei den stochastischen DGL ungekehrt: Die ableitungsfreien "stochastischen Runge–Kutta–Verfahren" werden aus den entsprechenden Taylor–Verfahren abgeleitet.

Die Idee ist dabei denkbar einfach: Die Ableitungen in den Taylor–Approximationen werden durch geeignete Differenzenquotienten so approximiert, dass die allgemeine Konvergenzordnung erhalten bleibt.

Wir wollen dies am Beispiel des Milstein–Verfahrens erläutern, das durch

$$\Phi(t,X,h,W) \;=\; X + ha(t,X) + \Delta W(t) b(t,X) + \frac{1}{2}((\Delta W(t))^2 - h) b(t,X) b'(t,X)$$

gegeben ist.

Der Term mit der Ableitung $bb'$ kann alternativ als $b'b$ geschrieben werden, was gerade die Richtungsableitung von $b$ in Richtung von $b$ darstellt. Durch (deterministische) Taylor–Entwicklung von $b(t, X + \tau b(t,X))$ in $\tau$ sieht man, dass

$$b'(t,X)b(t,X) = \frac{d}{d\tau} b(t, X + \tau b(t,X)) = \frac{1}{\tau}(b(\tau, X + \tau b(t,X)) - b(\tau, X)) + O(\tau)$$

gilt. Mit $\tau = \sqrt{h}$ können wir so das Verfahren

$$\begin{aligned} X_1 &= b(\sqrt{h}, X + \sqrt{h} b(t,X)) \\ \Phi(t,X,h,W) &= X + ha(t,X) \Delta W(t) b(t,X) + \frac{1}{2\sqrt{h}} (\Delta W(t)^2 - h)(X_1 - b(t,X)) \end{aligned}$$

definieren. Wegen des Vorfaktors vor dem abgeänderten Term $bb'$ unterscheiden sich die zwei Verfahren um $O(h^{1.5})$, woraus folgt, dass sie sich im Quadratmittel–Sinne gerade um $O(h^3)$ unterscheiden. Deswegen gilt Satz 3.12 und damit auch Satz 3.13 für $\gamma = 1$, womit das ableitungsfreie Verfahren eine starke Approximation der Ordnung $\gamma = 1$ darstellt.

Auf diese Art und Weise kann man ableitungsfreie Schemata aus den Taylor–Schemata konstruieren, wobei auch hier anzumerken ist, dass dieses Verfahren für große Ordnungen sehr aufwändig wird, da die Anzahl der Terme des Taylor–Verfahrens exponentiell mit der Ordnung wächst.

### 3.4.5 Abschließende Bemerkungen

Abschließend sollte angemerkt werden, dass viele weitere Themen, die wir bei den deterministischen DGL angesprochen haben, auch bei den stochastischen Differentielgleichungen

eine Rolle spielen: So gibt es z.B. implizite Verfahren für steife Gleichungen, Mehrschrittverfahren, Schrittweitensteuerung und viele weitere Aspekte, auf die wir in unserer kurzen Einführung nicht näher eingehen können, da sie den Rahmen einer einführenden Numerik–Vorlesung bei weitem sprengen würden.

# Kapitel 4

# Partielle Differentialgleichungen

Partielle Differentialgleichungen unterscheiden sich von den bisher betrachteten Gleichungen dadurch, dass hier Ableitungen nach mehr als einer eindimensionalen Variablen auftreten.

Im Vergleich zu gewöhnlichen Differentialgleichungen gibt es also viel mehr Möglichkeiten, Ableitungen einzuführen, weswegen es nicht überraschend ist, dass hier keine schöne abgeschlossene Theorie existiert, weder z.B. für Existenz– und Eindeutigkeitsresultate noch für die numerische Behandlung.

Speziell die numerische Herangehensweise hängt ganz entscheidend davon ab, was für Ableitungen mit welchen Koeffizienten in der Gleichung auftreten. Gleichungen mit zweiten Ableitungen werden i.A. ganz anders behandelt als Gleichungen, in denen nur erste Ableitungen auftreten, und wenn zweite Ableitungen auftreten, kommt es auf die Struktur der zugehörigen Koeffizienten an, welchen numerischen Ansatz man verfolgt.

Eine genaue Klassifizierung verschiedener PDGen und die Diskussion geeigneter numerischer Verfahren wäre Thema einer (u.U. sogar mehrsemestrigen) Vorlesung und kann nicht in den wenigen uns zur Verfügung stehenden Wochen durchgeführt werden.

Statt also einen Überblick über verschiedene Gleichungstypen und numerische Verfahren zu geben, werden wir uns hier auf eine Gleichung und ein Verfahren beschränken, dieses aber genauer besprechen und herleiten.

## 4.1 Die Wärmeleitungsgleichung

Unsere "Prototypgleichung" ist dabei die aus der Physik stammende *Wärmeleitungsgleichung*, die wir hier im $\mathbb{R}^2$ betrachten wollen. Man kann sich dabei z.B. eine Platte aus Metall vorstellen, die an einer oder mehreren Stellen erhitzt wird. Im Laufe der Zeit wird sich die Temperatur in der Platte verteilen und gegen eine Endverteilung konvergieren, den *stationären Zustand*.

Wir wollen uns hier mit diesem *stationären Problem* beschäftigen, d.h. die Frage beantworten, welche Temperaturverteilung sich nach einer langen Zeit in der Platte einstellen wird.

Wir wollen uns hier aus Zeitgründen nicht genauer mit der mathematischen Modellierung beschäftigen, sondern gleich die den stationären Zustand beschreibende Gleichung angeben. Dazu benötigen wir allerdings einige eingehende Funktionen. Ganz allgemein suchen wir für eine Menge $\Omega \subset \mathbb{R}^2$ eine Funktion $u : \Omega \to \mathbb{R}$, die uns in jedem Punkt $(x, y) \in \mathbb{R}^2$ die Endtemperatur $u(x, y)$ beschreibt. Diese Temperatur hängt von der Wärmeleitfähigkeit des Materials ab, die in jedem Punkt $(x, y) \in \Omega$ unterschiedlich sein kann und mit $\lambda(x, y) \geq 0$ beschrieben wird: Je größer $\lambda$ ist, desto besser ist die Wärmeleitfähigkeit. Außerdem müssen wir die Wärmequellen beschreiben, was durch eine Funktion $g : \Omega \to \mathbb{R}$ geschieht. Die Funktion $g$ heißt *spezifische Ergiebigkeit der Wärmequelle*, in der Literatur findet man hierfür meist die Schreibweise $g = \dot{q}_e$, die aber etwas unschön ist, weswegen wir sie durch das neutralere $g$ ersetzen. Der Wert $g(x, y)$ gibt an, wie stark die Platte im Punkt $(x, y)$ durch die Wärmequelle erwärmt wird.

Mit diesen Größen erfüllt die stationäre Wärmeverteilung $u(x, y)$ die Differentialgleichung

$$\frac{\partial}{\partial x}\left(\lambda(x,y)\frac{\partial u}{\partial x}(x,y)\right) + \frac{\partial}{\partial y}\left(\lambda(x,y)\frac{\partial u}{\partial y}(x,y)\right) + g(x,y) = 0 \tag{4.1}$$

oder, in der oft üblichen Kurzschreibweise ohne Argumente

$$\frac{\partial}{\partial x}\left(\lambda\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda\frac{\partial u}{\partial y}\right) + g = 0. \tag{4.2}$$

Diese Gleichung allein liefert allerdings noch keine brauchbare Lösung des Problems. Wenn wir die Gleichung numerisch lösen wollen, müssen wir uns nämlich auf ein beschränktes Gebiet $\Omega \subset \mathbb{R}^2$ einschränken (auch für die Modellierung realer Werkstücke ist dies natürlich nötig). Genauso, wie wir bei den gewöhnlichen Differentialgleichungen einen Anfangswert zur Zeit $t_0$ festlegen müssen, müssen wir hier eine Bedingung am Rand von $\Omega$ definieren, um eine eindeutige Lösung zu erhalten.

Wir werden den Rand $\partial\Omega$ hier in zwei Teile $C_a$ und $C_b$ aufteilen, in denen wir unterschiedliche Randbedingungen festlegen wollen.

**Dirichlet–Randbedingung:**
Von den gewöhlichen Differentialgleichungen kommend wird zunächst die sogenannte *Dirichlet–Randbedingung* als natürliche Wahl erscheinen. Bei dieser wird eine Funktion $v(x, y)$ vorgegeben, und es wird verlangt, dass auf dem Rand von $\Omega$ die Bedingung

$$u(x, y) = v(x, y) \tag{4.3}$$

gilt. Diese Bedingung werden wir auf $C_a$ annehmen.

**Neumann–Randbedingung:**
Die zweite Randbedingung, die wir betrachten wollen, macht eine Annahme über den Wärmestrom am Rand, also die "Flussrichtung" der Temperatur. In gewissen physikalischen Situationen kann man annehmen, dass diese parallel zum Rand verläuft, also kein Wärmeaustausch über den Rand hinweg stattfindet. Mathematisch wird dies durch die Bedingung

$$0 = n(x,y)^T \nabla u(x,y) = n_x(x,y)\frac{\partial u}{\partial x}(x,y) + n_y(x,y)\frac{\partial u}{\partial y}(x,y) \tag{4.4}$$

beschrieben, wobei $n = (n_x, n_y)^T$ der nach außen zeigende Normalenvektor des Randpunktes $(x, y)$ ist. Hierbei muss man natürlich annehmen, dass der Rand im Punkte $(x, y)$ durch eine differenzierbare Kurve beschrieben wird. Diese Bedingung werden wir auf $C_b$ annehmen.

Man kann beweisen, dass die Gleichung (4.2) zusammen mit den Randbedingungen (4.3) und (4.4) eine eindeutige zweimal stetig differenzierbare Lösung $u : \Omega \to \mathbb{R}$ besitzt, also $u \in C^2(\Omega, \mathbb{R})$, wenn die eingehenden Funktionen geeignete Regularitätseigenschaften besitzen, auf die wir hier nicht näher eingehen wollen.

**Beispiel 4.1 (Wärmedämmung am Fenster)**

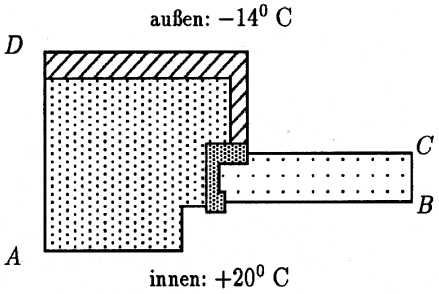



Abbildung 4.1: Beispiel: Wärmedämmung am Fenster

In Abbildung 4.1 ist ein Problem der Wärmedämmung an einem Fenster illustriert. Hier gibt es verschiedene Materialien mit verschiedenen Wärmeleitfähigkeiten: Das Fenster (hell gepunktet), das Mauerwerk (dunkler gepunktet) sowie zwei Dämmmaterialien (schraffiert bzw. ganz dunkel gepunktet). An den "realen" Außen– und Innenrändern (also den Randabschnitten zwischen $C$ und $D$ bzw. $A$ und $B$) wählt man in diesem Modell Dirichlet–Randbedingungen, während man an den "unechten" Rändern (also den Randabschnitten zwischen $A$ und $D$ und zwischen $B$ und $C$) Neumann–Randbedingungen wählen kann, die die physikalische Realität hier zumindest approximativ gut modellieren, da die Wärme vom Innen– zum Außenrand fließt, also parallel zu den unechten Rändern.

Beachte, dass sich die Wärmeleitfähigkeit hier an den Übergängen der unterschiedlichen Materialien unstetig ändert, was typisch für Wärmeleitungsprobleme mit verschiedenen Materialien ist. In diesem Fall kann man keine globale $C^2$–Lösung $u$ mehr erwarten, allerdings immer noch eine global stetige Lösung, die auf jedem Teilgebiet $C^2$ ist. □

**Bemerkung 4.2** Die Wärmeleitungsgleichung ist deswegen ein guter Prototyp einer PDG, weil viele weitere partielle Differentialgleichungen aus ganz unterschiedlichen Anwendungsgebieten der Mathematik eine ähnliche Struktur besitzen und mit ähnlichen numerischen Methoden gelöst werden können. □

## 4.2 Finite Elemente

Das numerische Lösungsverfahren, welches wir hier betrachten wollen, ist die *Methode der Finiten Elemente*. Diese Methode stammt ursprünglich aus den Ingenieurwissenschaften,

wo sie etwa seit den 1960er Jahren zur numerischen Lösung vieler angewandter Phänomene verwendet wurde. Die mathematische Theorie der Finiten Elemente wurde erst etwas später, etwa ab den 1970er Jahren systematisch entwickelt.

Mathematisch gesehen kann die Methode der finiten Elemente als eine sehr allgemeine Interpolationsmethode auf $n$–dimensionalen Gebieten interpretiert werden. Wie auch die in der Numerik I behandelte Polynominterpolation eignet sie sich gut zur Lösung von Integrationsproblemen. Hierdurch wird der Zusammenhang zur numerischen Lösung von partiellen Differentialgleichungen hergestellt, denn auch diese können — wie wir im folgenden Abschnitt sehen werden — als geeignete Integralgleichungen formuliert werden.

Wir wollen die Methode der finiten Elemente am Beispiel $n = 2$ nun etwas genauer darstellen. Wir beschränken uns dabei auf die einfachste Klasse der *linearen Finiten Elemente*.

Zur Einführung wiederholen wir kurz einige Tatsachen aus der Numerik 1:
Wir betrachten eine Funktion $f : [a, b] \to \mathbb{R}$ sowie Stützstellen $a = x_0 < x_1 < \ldots < x_n = b$. Mit Hilfe dieser Stützstellen konnten wir die Funktion $f$ mittels Interpolation der Daten $(x_i, f_i)$ mit $f_i = f(x_i)$ durch ein Polynom $P \in \mathcal{P}_n$ approximieren. Auf Basis dieser Polynominterpolation haben wir dann numerische Integrationsformeln (die Newton–Cotes–Formeln) entwickelt, wobei wir gesehen haben, dass es i.A. nicht sinnvoll ist, Polynome sehr hohen Grades zu verwenden. Statt dessen haben wir das Integrationsintervall in kleine Teilintervalle mit je $m$ Stützstellen zerlegt und auf diesen Teilintervallen das Interpolationspolynom erzeugt.

Im einfachsten Fall wählen wir dabei $m = 2$ und können dabei $f$ auf jedem Intervall $E_i = [x_i, x_{i+1}]$ durch ein lineares Polynom der Form

$$P_i(x) = f_i + \frac{x - x_i}{x_{i+1} - x_i}(f_{i+1} - f_i)$$

approximieren. Die zugehörige Integrationsregel

$$\int_a^b f(x)dx \approx \frac{1}{2}\sum_{i=0}^{n-1}(x_{i+1} - x_i)\Big(f(x_i) + f(x_{i+1})\Big)$$

ist gerade die Trapezregel.

Wir wollen nun versuchen, diese Technik auf Funktionen $f : \Omega \to \mathbb{R}$ für $\Omega \subset \mathbb{R}^2$ zu verallgemeinern. Hierzu nehmen wir an, dass sich das Gebiet $\Omega$ durch eine Vereinigung endlich vieler Dreiecke darstellen lässt. Dies wird nicht immer der Fall sein (z.B. wenn $\Omega$ eine Kreisscheibe ist), weswegen man die Menge $\Omega$ oft zunächst durch eine einfachere Menge approximieren muss (im Falle der Kreisscheibe z.B. durch ein approximierendes $M$–Eck). Abbildung 4.2 zeigt eine mögliche Zerlegung eines einfachen Rechteckgebietes. Eine solche Zerlegung wird als *Triangulierung* bezeichnet.

Jedes Dreieck in diesem Gitter wird nun als ein *Element* bezeichnet. Da es offenbar nur endlich viele davon gibt (im Gegensatz zu den unendlich vielen Punkten in $\Omega$), wird die Menge der Elemente (und damit das Verfahren) als *Finite Elemente* bezeichnet. Wir bezeichnen die Eckpunkte der Dreiecke nun als die *Knoten* des Gitters und bezeichnen ihre Koordinaten mit $(x_i, y_i)$.

Betrachten wir nun eine Funktion $f : \Omega \to \mathbb{R}$. Analog zur obigen Interpolation durch lineare Polynome wollen wir diese Funktion durch eine Funktion approximieren, die auf

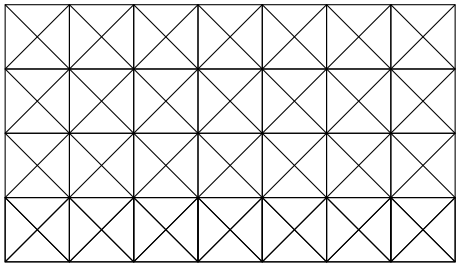


Abbildung 4.2: Triangulierung eines Rechtecks

jedem Element linear ist und an den Knotenpunkten mit $f(x_i, y_i)$ übereinstimmt. Wir betrachten dazu zunächst ein Element $E$ mit den Eckpunkten $(x_i, y_i)$, $i = 1, \ldots, 3$.

**Lemma 4.3** Jeder Punkt $(x, y) \in E$ lässt sich als

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{3} \mu_i \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

mit eindeutigen Koeffizienten $\mu_i$ mit $\mu_i \geq 0$ und $\sum_{i=1}^{3} \mu_i = 1$ schreiben.

**Beweis:** Beachte, dass ein Dreiecks–Element gerade die konvexe Hülle seiner Eckpunkte ist, also gerade durch

$$E = \left\{ \sum_{i=1}^{3} \mu_i \begin{pmatrix} x_i \\ y_i \end{pmatrix} \;\middle|\; \sum_{i=1}^{3} \mu_i = 1,\ \mu_i \geq 0, i = 1, 2, 3 \right\}$$

gegeben ist. Hieraus folgt, dass $\mu_i$ mit der angegebenen Eigenschaft existieren.

Um zu zeigen, dass sie eindeutig sind, betrachten wir das lineare Gleichungssystem

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Da die Punkte $(x_i, y_i)$ ein Dreieck bilden, ist die Matrix invertierbar (denn je zwei Seitenkanten des Dreiecks definieren linear unabhängige Vektoren). Also existiert eine eindeutige Lösung $\mu_1$, $\mu_2$, $\mu_3$. □

Mit Hilfe dieser Koordinaten $\mu_i$, die *baryzentrische Koordinaten* genannt werden, können wir nun auf jedem Element eine lineare Approximation von $f$ konstruieren: Für jeden Punkt $(x, y) \in E$ mit baryzentrischen Koordinaten $\mu_i$ definieren wir die approximierende Funktion $P : E \to \mathbb{R}$ als

$$P(x, y) = \sum_{i=1}^{3} \mu_i f(x_i, y_i) \tag{4.5}$$

**Lemma 4.4** (i) Die mittels (4.5) definierte Funktion $P : E \to \mathbb{R}$ ist affin linear in $(x, y)$ und erfüllt $P(x_j, y_j) = f(x_j, y_j)$ für $j = 1, \ldots, 3$.

(ii) Falls $f$ aus $C^2(E, \mathbb{R})$ ist, so gilt die Abschätzung

$$|P(x, y) - f(x, y)| \leq Ck(E)^2, \tag{4.6}$$

wobei $C$ von der zweiten Ableitung von $f$ abhängt und $k(E)$ den maximalen Euklidischen Abstand zweier Punkte in $E$ bezeichnet.

(iii) Falls $f$ aus $C^2(E, \mathbb{R})$ ist, so gelten die Abschätzungen

$$\left|\frac{\partial f}{\partial x}(x, y) - \frac{\partial P}{\partial x}(x, y)\right| \leq Cc(E)k(E) \text{ und } \left|\frac{\partial f}{\partial y}(x, y) - \frac{\partial P}{\partial y}(x, y)\right| \leq Cc(E)k(E) \tag{4.7}$$

für alle $(x, y) \in E$. Hierbei sind $C$ und $k(E)$ wie in (ii), während die Konstante $c(E)$ von den Winkeln des Dreieckselementes $E$ abhängt.

**Beweis:** (i) Zum Beweis der Linearität müssen wir nachweisen, dass sich $P(x, y) = a_0 + a_1 x + a_2 y$ schreiben lässt. Wenn wir das lineare Gleichungssystem aus dem Beweis von Lemma 4.3 als $A\mu = b$ schreiben, so ergibt sich $\mu = A^{-1}b$. Wenn wir $A^{-1} = (\alpha^*_{ij})$ schreiben, also

$$\mu_i = \alpha^*_{i1} x + \alpha^*_{i2} y + \alpha^*_{i3}.$$

Damit folgt

$$P(x, y) = \underbrace{\sum_{i=1}^{3} \alpha^*_{i1} f(x_i, y_i)}_{=a_1} x + \underbrace{\sum_{i=1}^{3} \alpha^*_{i2} f(x_i, y_i)}_{=a_2} y + \underbrace{\sum_{i=1}^{3} \alpha^*_{i3} f(x_i, y_i)}_{=a_0},$$

also die gewünschte Form. Für $(x, y) = (x_j, y_j)$ sieht man leicht, dass $\mu_j = 1$ und $\mu_i = 0$ für $i \neq j$ sein muss. Also folgt

$$P(x_j, y_j) = \mu_j f(x_j, y_j) = f(x_j, y_j).$$

(ii) Sei $(x, y) \in E$ beliebig und $(x_i, y_i)$ ein beliebiger Eckpunkt. Die Taylor–Entwicklung von $f$ liefert

$$f(x, y) = f(x_i, y_i) + (x - x_i)\frac{\partial f}{\partial x}(x_i, y_i) + (y - y_i)\frac{\partial f}{\partial y}(x_i, y_i) + O(k(E)^2).$$

Wählen wir nun einen beliebigen Punkt $(x^*, y^*) \in E$, so folgt aus der Lipschitz–Stetigkeit der ersten Ableitungen die Abschätzung

$$(x - x_i)\frac{\partial f}{\partial x}(x_i, y_i) = (x - x_i)\frac{\partial f}{\partial x}(x^*, y^*) + (x - x_i)O(k(E)) = (x - x_i)\frac{\partial f}{\partial x}(x^*, y^*) + O(k(E)^2)$$

und analog für $\partial f / \partial y$.

Damit folgt für die zu $(x, y)$ gehörigen baryzentrischen Koordinaten $\mu_i$ die Gleichung

$$\begin{aligned} f(x,y) &= \sum_{i=1}^{3} \mu_i f(x,y) \\ &= \sum_{i=1}^{3} \mu_i f(x_i,y_i) + \sum_{i=1}^{3} \mu_i (x-x_i)\frac{\partial f}{\partial x}(x_i,y_i) + \sum_{i=1}^{3} \mu_i (y-y_i)\frac{\partial f}{\partial y}(x_i,y_i) + O(k(E)^2) \\ &= \sum_{i=1}^{3} \mu_i f(x_i,y_i) + \sum_{i=1}^{3} \mu_i (x-x_i)\frac{\partial f}{\partial x}(x^*,y^*) + \sum_{i=1}^{3} \mu_i (y-y_i)\frac{\partial f}{\partial y}(x^*,y^*) + O(k(E)^2) \\ &= \sum_{i=1}^{3} \mu_i f(x_i,y_i) + O(k(E)^2) \;=\; P(x,y) + O(k(E)^2), \end{aligned}$$

denn es gilt

$$\sum_{i=1}^{3} \mu_i (x - x_i) = x - \sum_{i=1}^{3} \mu_i x_i = x - x = 0$$

und analog für $(y - y_i)$. Beachte, dass der $O(k(E)^2)$–Term hier von der Form $Ck(E)^2$ ist, wobei $C$ durch das Maximum der Norm der zweiten Ableitung bestimmt ist.

(iii) Wir zeigen die Behauptung für die Ableitung nach $x$. Betrachte die Eckpunkte $(x_i, y_i)$, die hier o.B.d.A. nach ihrer $y$–Komponente aufsteigend nummeriert seien. Dann gibt es für den Eckpunkt $(x_2, y_2)$ ein $\delta_x > 0$, so dass entweder $(x_2 + \delta_x, y_2)$ oder $(x_2 - \delta_x, y_2)$ ein Randpunkt des Dreiecks ist, vgl. Abbildung 4.3.

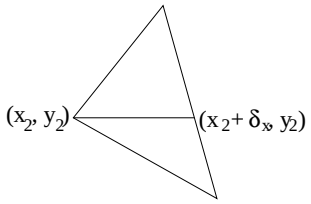



Abbildung 4.3: Punkte $(x_2, y_2)$ und $(x_2 + \delta_x, y_2)$ im Beweis von Lemma 4.4(iii)

Abhängig von den Winkeln des Dreiecks existiert nun eine Konstante $\tilde{c}(E) > 0$, so dass die Ungleichung $\delta_x \geq k(E)/\tilde{c}(E)$ gilt. Aus dem Mittelwertsatz der Differentialrechnung folgt nun

$$\frac{f(x_2 + \delta_x, y_2) - f(x_2, y_2)}{\delta_x} = \frac{\partial f}{\partial x}(x^*, y^*)$$

für einen Punkt $(x, y)$ auf der Verbindungsgeraden von $(x_2, y_2)$ nach $(x_2 + \delta_x, y_2)$.

Andererseits gilt wegen der affinen Linearität von $P$ (die Ableitung ist konstant) für jeden Punkt $(x, y) \in E$, also insbesondere für den obigen Punkt $(x^*, y^*)$ die Gleichung

$$\frac{P(x_2 + \delta_x, y_2) - P(x_2, y_2)}{\delta_x} = \frac{\partial P}{\partial x}(x, y) = a_1 = \frac{\partial P}{\partial x}(x^*, y^*).$$

Also folgt mit (ii) die Abschätzung

$$\begin{aligned}\left|\frac{\partial f}{\partial x}(x^*,y^*)-\frac{\partial P}{\partial x}(x^*,y^*)\right| &= \frac{|f(x_2+\delta x_2,y_2)-P(x_2+\delta x_2,y_2)|}{\delta_x}\\ &\le \frac{Ck(E)^2}{\delta_x} \le C\tilde{c}(E)k(E)\end{aligned}$$

Da $f$ eine $C^2$–Funktion ist, ist die erste Ableitung Lipschitz–stetig mit einer Konstanten $C$, die wiederum durch die maximale Norm der zweiten Ableitung bestimmt ist. Für einen beliebigen Punkt $(x,y)\in E$ gilt damit

$$\begin{aligned}\left|\frac{\partial f}{\partial x}(x,y)-\frac{\partial P}{\partial x}(x,y)\right| &\le \left|\frac{\partial f}{\partial x}(x^*,y^*)-\frac{\partial P}{\partial x}(x^*,y^*)\right|+\left|\frac{\partial f}{\partial x}(x,y)-\frac{\partial f}{\partial x}(x^*,y^*)\right|\\ &\le C\tilde{c}(E)k(E)+Ck(E),\end{aligned}$$

also die Behauptung mit $c(E)=\tilde{c}(E)+1$. □

**Bemerkung 4.5** Beachte, dass die Konstante $c(E)$ in (iii) um so größer wird, je kleinere Winkel im Dreieck $E$ auftreten. Für eine gute Approximation der ersten Ableitungen von $f$ sollte man daher Elemente mit kleinen Winkeln vermeiden. □

Unser Ziel ist die Integration auf den Finiten Elementen, weswegen wir uns im nächsten Schritt überlegen wollen, wie man das Integral der Funktion $P$ auf einem Element $E$ berechnet und welchen Fehler man dabei gegenüber der Integration von $f$ auf $E$ macht.

Im nächsten Abschnitt werden wir sehen, dass wir zur numerischen Lösung von (4.2) zwei Integrale lösen müssen, nämlich zum einen

$$\iint_E f(x,y)\,dx\,dy \tag{4.8}$$

und zum anderen

$$\iint_E \left(\frac{\partial f}{\partial x}(x,y)\right)^2+\left(\frac{\partial f}{\partial y}(x,y)\right)^2 dx\,dy. \tag{4.9}$$

Beide Integrale sollen dabei durch Integration von $P$ anstelle von $f$ numerisch approximiert werden.

Wir beginnen mit (4.8). Wir müssen dazu das Integral

$$\iint_E P(x,y)\,dx\,dy := \int_{y\in E_1}\int_{x\in E_2(y)} P(x,y)\,dx\,dy$$

auf einem Element $E$ berechnen, wobei

$$E_1=\{y\in\mathbb{R}\,|\,(x,y)\in E \text{ für ein } x\in\mathbb{R}\} \text{ und } E_2(y)=\{x\in\mathbb{R}\,|\,(x,y)\in E\}$$

abgeschlossene Intervalle sind. Um auf eine einfache Formel zu kommen, betrachten wir zunächst die Fläche des Elementes $E$, die durch

$$A(E) = \frac{1}{2} \det \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{pmatrix}$$

gegeben ist, wobei wir die Konvention machen, dass die Ecken von $E$ entgegen dem Uhrzeigersinn nummeriert sind. Beachte, dass wir für $k(E)$ aus Lemma 4.4 die Abschätzung $|x_i - x_j| \leq k(E)$ und $|y_i - y_j| \leq k(E)$ erhalten, weswegen

$$A(E) \leq \frac{1}{2} k(E)^2 \tag{4.10}$$

gilt, wobei Gleichheit genau dann gilt, wenn das Dreieck rechtwinklig und gleichschenklig ist.

Das gesuchte Integral gibt nun gerade das Volumen des dreidimensionalen Körpers an, der als Grundfläche das Dreieckselement besitzt und nach oben durch die Funktion $P(x, y)$ begrenzt ist. Dieses wiederum ist gerade durch die Formel

$$\frac{1}{3}\Big(P(x_1, y_1) + P(x_2, y_2) + P(x_3, y_3)\Big)\, A(E)$$

gegeben, was damit die Verallgemeinerung der oben angegebenen Trapezregel darstellt.

Der folgende Satz fasst dies zusammen.

**Satz 4.6** Betrachte ein Dreieckselement $E$ mit Eckpunkten $(x_i, y_i)$, $i = 1, 2, 3$ und eine Funktion $f$ aus $C^2(E, \mathbb{R})$. Dann gilt die Abschätzung

$$\left| \iint_E f(x, y)\, dx\, dy - \frac{1}{3}(f(x_1, y_1) + f(x_2, y_2) + f(x_3, y_3))A(E) \right| \leq \frac{1}{2} C k(E)^4,$$

wobei $A(E)$ die Fläche von $E$ und $k(E)$ den maximalen Abstand zweier Punkte aus $E$ bezeichnet und $C$ die Konstante aus Lemma 4.4(ii) ist.

**Beweis:** Es sei $P$ die lineare Approximation von $f$ auf $E$ aus Lemma 4.4. Nach den obigen Überlegungen gilt dann

$$\begin{aligned} \iint_E P(x, y)\, dx\, dy &= \frac{1}{3}(P(x_1, y_1) + P(x_2, y_2) + P(x_3, y_3))A(E) \\ &= \frac{1}{3}(f(x_1, y_1) + f(x_2, y_2) + f(x_3, y_3))A(E). \end{aligned}$$

Es bleibt also zu zeigen, dass

$$\left| \iint_E P(x, y) - f(x, y)\, dx\, dy \right| \leq \frac{1}{2} C k(E)^4$$

ist. Diese Ungleichung folgt unter Verwendung von Lemma 4.4 und Abschätzung (4.10) aus

$$\begin{aligned}\left|\iint_E P(x,y) - f(x,y)\,dx\,dy\right| &\leq \iint_E |P(x,y) - f(x,y)|\,dx\,dy \\ &\leq \iint_E Ck(E)^2\,dx\,dy \;=\; Ck(E)^2 \underbrace{\iint_E dx\,dy}_{=A(E)} \\ &= Ck(E)^2 A(E) \;\leq\; \frac{1}{2}Ck(E)^4.\end{aligned}$$

□

Wir betrachten nun (4.9). Hierzu überlegen wir uns zunächst, wie das Integral

$$\iint_E \left(\frac{\partial P}{\partial x}(x,y)\right)^2 + \left(\frac{\partial P}{\partial y}(x,y)\right)^2 dx\,dy$$

numerisch gelöst werden kann. In der Schreibweise

$$P(x,y) = a_0 + a_1 x + a_2 y$$

ist dies leicht, denn es gilt $\partial P/\partial x \equiv a_1$ und $\partial P/\partial y \equiv a_2$ und damit

$$\iint_E \left(\frac{\partial P}{\partial x}(x,y)\right)^2 + \left(\frac{\partial P}{\partial y}(x,y)\right)^2 dx\,dy = (a_1^2 + a_2^2)A(E).$$

Für $P$ gilt nun

$$\begin{aligned} a_0 + a_1 x_1 + a_2 y_1 &= f_1 \\ a_0 + a_1 x_2 + a_2 y_2 &= f_2 \\ a_0 + a_1 x_3 + a_2 y_3 &= f_3 \end{aligned}$$

mit $f_i = f(x_i, y_i) = P(x_i, y_i)$, woraus man durch Lösen mit der Cramer'schen Regel und einigen Umformungen die Lösung

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{2A(E)} Bf$$

mit

$$B = \begin{pmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{pmatrix} \tag{4.11}$$

berechnet. Also ergibt sich

$$\iint_E \left(\frac{\partial P}{\partial x}(x,y)\right)^2 + \left(\frac{\partial P}{\partial y}(x,y)\right)^2 dx\,dy = A(E)(a_1^2 + a_2^2) = \frac{1}{4A(E)} f^T B^T B f.$$

Diese Berechnungen führen auf den folgenden Satz.

**Satz 4.7** Betrachte ein Dreieckselement $E$ mit Eckpunkten $(x_i, y_i)$, $i = 1, 2, 3$ und eine Funktion $f$ aus $C^2(E, \mathbb{R})$. Dann gilt die Abschätzung

$$\left| \iint\limits_E \left(\frac{\partial f}{\partial x}(x,y)\right)^2 + \left(\frac{\partial f}{\partial y}(x,y)\right)^2 dx\, dy - \frac{1}{4A(E)} f^T B^T B f \right| \leq Dc(E)k(E)^3.$$

Hierbei sind $A(E)$ die Fläche von $E$ und $k(E)$ den maximalen Abstand zweier Eckpunkte von $E$, $f = (f(x_1, y_1), f(x_2, y_2), f(x_3, y_3))^T$, $B$ die Matrix aus (4.11), $c(E)$ die von den Winkeln von $E$ abhängige Konstante aus Lemma 4.4(iii) und $D$ eine von den Ableitungen von $f$ abhängige Konstante.

**Beweis:** Der Beweis verläuft völlig analog zum Beweis von Satz 4.6, wobei wir am Ende das Integral über den Fehler

$$\left| a_1^2 + a_2^2 - \left(\frac{\partial f}{\partial x}\right)^2 - \left(\frac{\partial f}{\partial y}\right)^2 \right|$$

abschätzen müssen. Dieser Ausdruck lässt sich mittels

$$\left| a_1^2 + a_2^2 - \left(\frac{\partial f}{\partial x}\right)^2 - \left(\frac{\partial f}{\partial y}\right)^2 \right| \leq \left| a_1^2 - \left(\frac{\partial f}{\partial x}\right)^2 \right| + \left| a_2^2 - \left(\frac{\partial f}{\partial y}\right)^2 \right|$$

und mit Lemma 4.4(iii) weiter durch

$$\left| a_1^2 - \left(\frac{\partial f}{\partial x}\right)^2 \right| = \left| \left(a_1 + \frac{\partial f}{\partial x}\right)\left(a_1 - \frac{\partial f}{\partial x}\right) \right| \leq Dc(E)k(E)$$

und

$$\left| a_2^2 - \left(\frac{\partial f}{\partial y}\right)^2 \right| = \left| \left(a_2 + \frac{\partial f}{\partial y}\right)\left(a_2 - \frac{\partial f}{\partial y}\right) \right| \leq Dc(E)k(E)$$

für ein geeignetes $D > 0$ abschätzen. Integration über $E$ analog zum Beweis von Satz 4.6 liefert dann die Aussage. □

**Bemerkung 4.8** Wenn wir das Integral auf ganz $\Omega$ auf diese Weise durch Summierung über die einzelnen Elemente approximieren wollen, so ergibt sich der Fehler der Ordnung $O(Mk^4)$ für (4.8) bzw. $O(Mk^3)$ für (4.9), wobei $k$ das Maximum von $k(E)$ über alle Elemente $E$ ist und $M$ die Anzahl der Elemente ist. Man überlegt sich leicht, dass man z.B. bei einem Quadrat mit der Fläche $A$ und bei einer Triangulierung wie in Abbildung 4.2 gerade $M = 4A/k^2$ Dreiecks–Elemente benötigt, um die Größe $k(E) \leq k$ für jedes Element $E$ sicher zu stellen. Allgemein benötigt man $M \sim 1/k^2$ Elemente um diese Größenbedingung zu garantieren, weswegen man für die Gesamtintegrale einen Fehler der Ordnung $O(k^2)$ für (4.8) und $O(k)$ für (4.9) erhält.

Beachte, dass diese Abschätzungen auch dann gültig sind, wenn die Funktion $f$ nur auf den Elementen $E$, aber nicht global $C^2$ ist. Auf den Rändern der Elemente braucht die Funktion nur stetig zu sein. □

## 4.3 Die Wärmeleitungsgleichung als Integralgleichung

Wir wollen nun eine Integralgleichung angeben, die äquivalent zur PDG (4.2) ist. Hierfür gibt es viele verschiedene Möglichkeiten; wir wollen hier eine Variante verwenden, die gerade so gewählt ist, dass die zweiten Ableitungen in (4.2) verschwinden. Dafür müssen wir allerdings einen Preis zahlen, denn wir werden keine Integralgleichung erhalten, die von $u$ erfüllt wird, sondern eine Integralgleichung, die von $u$ *minimiert* wird.

**Satz 4.9** Die eindeutige Lösung $u$ von (4.2) ist das eindeutige Minimum des Funktionals

$$J(u) := \iint\limits_{\Omega} \left[ \frac{\lambda}{2} \left( \frac{\partial u}{\partial x} \right)^2 + \frac{\lambda}{2} \left( \frac{\partial u}{\partial y} \right)^2 - gu \right] dx\, dy,$$

wobei alle im Integranden auftretenden Funktionen, also $\lambda$, $u$ und $g$, von $(x, y)$ abhängen.

Hierbei wird das Minimum über alle stückweise stetig differenzierbaren Funktionen $u$ gebildet, die auf $C_a$ der Dirichlet–Randbedingung $u = v$ genügen. Die Lösung erfüllt dabei auf $C_b = \partial\Omega \setminus C_a$ die Neumann–Randbedingungen.

Stückweise stetig differenzierbar bedeutet hierbei, dass wir eine feste Zerlegung von $\Omega$ in Teilgebiete mit 1–dimensionalen Rändern betrachten, auf denen die betrachteten Funktionen $C^1$ sind. Die Ränder der Teilgebiete, in denen die Funktion nicht differenzierbar ist, lassen wir dabei bei der Integration wegfallen; da diese eine niedrigere Dimension besitzen, ändert dies nichts am Integralwert. Die resultierende minimierende Funktion $u$ von $J$ ist dann "automatisch" zwei mal stetig differenzierbar (zumindest dort, wo $\lambda$ und $g$ stetig sind).

Diese Art von Integralgleichungen nennt man *Variationsformulierung* der PDG. Sie lässt sich für eine große Klasse von PDGen, die sogenannten *elliptischen PDGen* erhalten.

Der Name "Variationsformulierung" ergibt sich aus dem Beweis des Satzes, von dem wir hier nur die Idee angeben wollen: Man betrachtet Variationen $u + \alpha w$ der Funktion $u$ und zeigt, dass die Ableitung von $J$ gerade durch (4.2) gegeben ist. Ein Minimum von $J$ muss also Lösung von (4.2) sein. Umgekehrt analysiert man, dass jeder Extremalpunkt von $J$ ein Minimum sein muss, woraus man die Äquivalenz erhält.

Für die Wärmeleitungsgleichung kann man diese Gleichung auch physikalisch interpretieren. Das Funktional $J(u)$ misst gerade die Wärmeenergie, die im stationären Zustand minimal wird.

## 4.4 Approximation auf den Finiten Elementen

Wir nehmen in diesem Abschnitt an, dass eine Triangulierung des Gebietes $\Omega$ vorgegeben ist. Diese Triangulierung bestehe aus $M$ Elementen $E_j$, $j = 1, \ldots, M$ mit insgesamt $N$ Knotenpunkten $(x_i, y_i)$, $i = 1, \ldots, N$. Mit $k$ sei die Größe des maximalen Elementes bezeichnet und die Eckpunkte des Elementes $E_j$ seien mit $(x_{i_{j,1}}, y_{i_{j,1}})$, $(x_{i_{j,2}}, y_{i_{j,2}})$, $(x_{i_{j,3}}, y_{i_{j,3}})$ bezeichnet.

Das Integral $J(u)$ können wir nun zunächst auf die Elemente $E_j$ "aufteilen", es gilt

$$J(u) = \sum_{j=1}^{M} J_j(u)$$

mit

$$J_j(u) := \iint\limits_{E_j} \left[ \frac{\lambda}{2} \left( \frac{\partial u}{\partial x} \right)^2 + \frac{\lambda}{2} \left( \frac{\partial u}{\partial y} \right)^2 - gu \right] dx\, dy,$$

Auf jedem Element $E_j$ approximieren wir nun $\lambda$ und $g$ durch Konstanten $\lambda_j$ und $g_j$. Wenn wir z.B. hierfür den Wert in einem beliebigen Punkt $(x, y)$ des Elementes verwenden, und annehmen, dass $\lambda$ und $g$ auf jedem Element Lipschitz–stetig sind, so machen wir dabei einen Fehler der Ordnung $O(k)$ unter dem Integral, also der Ordnung $O(k^3)$ in der Integralberechnung.

Nun verwenden wir die Integralapproximation auf den finiten Elementen. Wir schreiben dabei $u_i = u(x_i, y_i)$ für den Wert von $u$ im $i$–ten Knotenpunkt. Auf einem Element $E_j$ mit den Eckpunkten $(x_i, y_i)$ mit Indizes $i = i_{j,1}, i_{j,2}, i_{j,3}$ erhalten wir damit

$$\begin{aligned} J_j(u) &:= \iint\limits_{E_j} \frac{\lambda}{2} \left( \frac{\partial u}{\partial x} \right)^2 + \frac{\lambda}{2} \left( \frac{\partial u}{\partial y} \right)^2 dx\, dy - \iint\limits_{E_j} gu dx\, dy \\ &= \frac{\lambda_j}{2} \iint\limits_{E_j} \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 dx\, dy - g_j \iint\limits_{E_j} u dx\, dy + O(k^3) \\ &= \frac{1}{2} \lambda_j \frac{1}{4A(E_j)} \mathbf{u}_j^T B_j^T B_j \mathbf{u}_j - \frac{1}{3} A(E_j) g_j \sum_{k=1}^{3} u_{i_{j,k}} + O(k^3) \end{aligned}$$

mit $\mathbf{u}_j = (u_{i_{j,1}}, u_{i_{j,2}}, u_{i_{j,3}})^T \in \mathbb{R}^3$.

Mit den Kurzschreibweisen

$$S_j = \frac{\lambda_j}{4A(E_j)} B_j^T B_j \quad \text{und} \quad b_j = \frac{1}{3} A(E_j) g_j \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

können wir diese Approximation kurz als

$$J_j(u) = \frac{1}{2} \mathbf{u}_j^T S_j \mathbf{u}_j - b^T \mathbf{u}_j + O(k^3)$$

schreiben. Auf ganz $\Omega$ gilt daher — unter der Annahme $M \sim 1/k^2$ — die Abschätzung

$$J(u) = \sum_{j=1}^{M} \frac{1}{2} \mathbf{u}_j^T S_j \mathbf{u}_j - \sum_{j=1}^{M} b_j^T \mathbf{u}_j + O(k).$$

Wenn wir den Vektor $\mathbf{u} := (u_1, u_2, \ldots, u_N)^T$ definieren, so kann man Matrizen $\widetilde{S}_j$ mittels

$$[\widetilde{S}_j]_{i_{j,k}, i_{j,l}} = [S_j]_{k,l}$$

und Vektoren $\tilde{b}_j$ mittels

$$[\tilde{b}_j]_{i_{j,k}} = [b_j]_k$$

für $j = 1, \ldots, M$ und $k, l = 1, \ldots, 3$ definieren. Damit ergibt sich

$$\mathbf{u}_j^T S_j \mathbf{u}_j = \mathbf{u}^T \widetilde{S}_j \mathbf{u} \quad \text{und } b_j^T \mathbf{u}_j = \tilde{b}_j^T \mathbf{u},$$

also

$$\sum_{j=1}^{M} \frac{1}{2} \mathbf{u}_j^T S_j \mathbf{u}_j - \sum_{j=1}^{M} b_j^T \mathbf{u}_j = \sum_{j=1}^{M} \frac{1}{2} \mathbf{u}^T \widetilde{S}_j \mathbf{u} - \sum_{j=1}^{M} \tilde{b}_j^T \mathbf{u} = \frac{1}{2} \mathbf{u}^T S \mathbf{u} + b^T \mathbf{u}$$

mit

$$S = \sum_{j=1}^{M} \widetilde{S}_j \quad \text{und} \quad b = \sum_{j=1}^{M} \tilde{b}_j.$$

Definieren wir schließlich

$$J_{app}(\mathbf{u}) := \frac{1}{2} \mathbf{u}^T S \mathbf{u} - b^T \mathbf{u}, \tag{4.12}$$

so erhalten wir eine Approximation von $J$ der Form

$$J(u) = J_{app}(\mathbf{u}) + O(k). \tag{4.13}$$

Die Matrix $S$ wird hierbei *Steifigkeitsmatrix* genannt; dieser Ausdruck stammt aus der Modellierung mechanischer Objekte durch finite Elemente, bei der diese Matrix von gewissen Materialeigenschaften — eben der Steifigkeit — abhängt. Man sieht leicht, dass die Matrix $S$ symmetrisch ist, da ihre Komponenten aus Produkten der Form $B_j^T B_j$ stammen.

Statt $J$ über Funktionen $u : \Omega \to \mathbb{R}$ zu minimieren, werden wir jetzt $J_{app}$ über Vektoren $\mathbf{u} \in \mathbb{R}^N$ minimieren. Wir wollen uns zunächst überlegen, ob das Problem eine eindeutige Lösung besitzt. Um die Funktion $J_{app}$ zu minimieren, kann man wie gewohnt vorgehen: Ableiten von $J_{app}$ nach $\mathbf{u}$ und Null–setzen liefert

$$0 = \nabla J_{app}(u) = S\mathbf{u} - b.$$

Falls wir eine Lösung dieses linearen Gleichungssystems finden, so ist dies ein Minimum, falls

$$\frac{d^2}{d\mathbf{u}^2} J_{app}(u) = S$$

positiv definit ist. In diesem Fall folgt auch gleich, dass das obige Gleichungssystem $S\mathbf{u} = b$ eine eindeutige Lösung besitzt, da positiv definite Matrizen invertierbar sind.

Tatsächlich ist die Matrix $S$ positiv semidefinit, denn für jeden Vektor $\mathbf{u} \neq 0$ stellt $\mathbf{u}^T S \mathbf{u}$ gerade das Integral über $(\partial P/\partial x)^2 + (\partial P/\partial y)^2$ dar, wobei $P$ die Interpolierende der Knotenwerte $\mathbf{u}$ ist. Sie ist aber *nicht* positiv definit, denn wenn alle Einträge in $\mathbf{u}$ den gleichen Wert besitzen, ist die Funktion $P$ konstant, weswegen alle Ableitungen und damit das Integral gleich Null sind.

Der Grund dafür liegt in der Tatsache, dass wir noch keine Dirichlet–Randbedingunen berücksichtigt haben. Ohne die Festlegung von zumindest einem Punkt am Rand besitzt die Gleichung (4.2) keine eindeutige Lösung, weswegen wir das auch nicht von unserer Diskretisierung erwarten können. Wir wollen dies nun nachholen.

O.B.d.A. seien die Knoten dabei so nummeriert, dass die Dirichlet–Randbedingungen gerade die letzten $N-K$ Knoten festlegen, d.h. die Werte $u_{K+1},\ldots,u_N$ sind gerade durch die Randbedingungen $u_i = v(x_i,y_i)$, $i = K+1,\ldots,N$, festgelegt. Wir zerlegen den Vektor $\mathbf{u}$ nun in die freien Knotenwerte $\mathbf{u}_f = (u_1,\ldots,u_K)^T$ und die "randbedingten" Werte $\mathbf{u}_r = (u_{K+1},\ldots,u_N)^T$. Passend dazu zerlegen wir die Matrix $S$ in Teilmatrizen

$$S = \begin{pmatrix} S_f & S_{fr} \\ S_{fr}^T & S_r \end{pmatrix}$$

mit $S_f \in \mathbb{R}^{K\times K}$ und passenden Dimensionen für die restlichen Matrizen, sowie den Vektor $b$ in $b_f \in \mathbb{R}^K$ und $b_r \in \mathbb{R}^{N-K}$.

Dann gilt

$$J_{app}(\mathbf{u}) = \frac{1}{2}\mathbf{u}_f^T S_f \mathbf{u}_f + \mathbf{u}_r^T S_{fr}^T \mathbf{u}_f + \frac{1}{2}\mathbf{u}_r^T S_r \mathbf{u}_r - b_f^T \mathbf{u}_f - b_r^T \mathbf{u}_r.$$

Mit der Abkürzung $d = -S_{fr}\mathbf{u}_r + b_f$ und $e = \frac{1}{2}\mathbf{u}_r^T S_r \mathbf{u}_r - b_r^T \mathbf{u}_r$ (beachte, dass diese Werte nur von den durch die Dirichlet–Randbedingung festgelegten Werten in $\mathbf{u}_r$ abhängen) erhalten wir

$$J_{app}(\mathbf{u}) = J_{app}^f(\mathbf{u}_f) = \frac{1}{2}\mathbf{u}_f^T S_f \mathbf{u}_f - d^T \mathbf{u}_f + e.$$

Wie oben müssen wir nun zur Lösung des Minimierungproblems $\min_{\mathbf{u}_f\in\mathbb{R}^K} J_{app}^f(\mathbf{u}_f)$ das lineare Gleichungssystem

$$S_f \mathbf{u}_f = d \tag{4.14}$$

lösen, dessen Lösung ein eindeutiges Minimum liefert, falls $S_f$ positiv definit ist.

Wir zeigen nun, dass $S_f$ positiv definit ist. Betrachte dazu einen beliebigen Vektor $\mathbf{u}_f \in \mathbb{R}^K$ mit $\mathbf{u}_f \neq 0$. Wir ergänzen diesen durch Anhängen von $N-K$ Null–Einträgen zu einem Vektor $\mathbf{u} \in \mathbb{R}^N$. Die zugehörige interpolierende Funktion $P$ ist dann nicht–konstant, weil es mindestens ein Element gibt, das sowohl Knotenwerte $\neq 0$ als auch Knotenwerte $= 0$ besitzt. Folglich ist $\mathbf{u}^T S\mathbf{u} > 0$, da dies gerade das Integral über $(\partial P/\partial x)^2 + (\partial P/\partial y)^2$ ist. Da die Einträge $u_i$ für $i \geq K+1$ gleich Null sind, gilt $\mathbf{u}_f^T S_f \mathbf{u}_f = \mathbf{u}^T S\mathbf{u} > 0$, was gerade die Bedingung für die positive Definitheit von $S_f$ ist.

Wir fassen das bisher hergeleitete in dem folgenden Satz zusammen.

**Satz 4.10** Das Minimierungsproblem

$$\min_{w\in\mathbb{R}^N} J_{app}(w) \quad \text{mit } J_{app}(w) = w^T S w - b^T w$$

mit der Dirichlet–Randbedingung $w_i = v(x_i,y_i)$ für $i = K+1,\ldots,N$ für ein $K < N$ besitzt genau eine Lösung $\mathbf{u} \in \mathbb{R}^N$. Diese ist gegeben durch $\mathbf{u}^T = (\mathbf{u}_f^T, \mathbf{u}_r^T)$ mit $\mathbf{u}_f \in \mathbb{R}^K$ und $\mathbf{u}_r \in \mathbb{R}^{N-K}$, wobei $\mathbf{u}_r = (u_{K+1},\ldots,u_N)$ durch die Dirichlet–Randbedingung festgelegt ist und $\mathbf{u}_f$ die eindeutige Lösung des linearen Gleichungssystems

$$S_f \mathbf{u}_f = d$$

mit der oben eingeführten symmetrischen Matrix $S_f \in \mathbb{R}^{K\times K}$ und dem Vektor $d \in \mathbb{R}^K$ ist.

Da $S_f$ symmetrisch und positiv definit ist, kommen eine ganze Reihe numerischer Verfahren zur Lösung dieses Gleichungssystems in Frage, z.B. das Choleski–Verfahren, falls die freie Knotenanzahl $K$ nicht zu groß ist. Falls $K$ groß ist, kommen iterative Verfahren wie Gauß–Seidel oder das CG–Verfahren in Frage. Beachte, dass $S_f$ schwach besetzt ist, also in jeder Zeile nur wenige Einträge $\neq 0$ enthält, weswegen iterative Verfahren hier besonders effizient sind.

Wir werden später noch eine Abschätzung für die positive Definitheit von $S_f$ benötigen, die wir nun angeben wollen.

**Lemma 4.11** Es existiert eine vom Gitter unabhängige Konstante $C > 0$, für die die Abschätzung

$$\mathbf{u}_f^T S_f \mathbf{u}_f \geq C \frac{\|\mathbf{u}_f\|^2}{K} \tag{4.15}$$

gilt.

**Beweis:** Betrachte einen Vektor $\mathbf{u}_f$ mit $\|\mathbf{u}_f\| =: c > 0$. Dann gilt $\sum_{i=1}^K u_i^2 = c^2$, folglich gibt es mindestens einen Eintrag $u_i$ in $\mathbf{u}_f$ mit $u_i^2 \geq c^2/K$, also $|u_i| \geq c/\sqrt{K}$. Mit $u_{i^*}$ bezeichnen wir den betragsmäßig maximalen Eintrag von $\mathbf{u}_f$, o.B.d.A. sei $u_{i^*} > 0$. Wie oben ergänzen wir den Vektor mit Null–Einträgen zu einem Vektor $\mathbf{u} \in \mathbb{R}^N$.

Die zugehörige Interpolierende $P$ besitzt also ein Maximum $(x^*, y^*)$ mit $P(x^*, y^*) =: \tilde{c} \geq c/\sqrt{K}$ sowie einen Punkt $(x^0, y^0)$ mit $P(x^0, y^0) = 0$. Wir definieren $\Delta x = x^* - x^0$ und $\Delta y = y^* - y^0$.

Wir betrachten nun alle auf den Elementen stückweisen $C^1$–Funktionen $w : \Omega \to \mathbb{R}^n$, für die $w(x^*, y^*) = \tilde{c}$ und $w(x^0, y^0) = 0$ gilt. Unter all diesen Funktionen minimieren gerade die Funktionen mit stückweise konstanter Ableitung das Integral

$$J(w) = \iint_\Omega \left[ \left( \frac{\partial w}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right] dx\, dy,$$

denn für diese gilt ja gerade $J(w) = 0$, weil die zweiten Ableitungen auf jedem Element verschwinden. Insbesondere wird $J$ also für jede global lineare Funktion $w(x, y) = a_0 + a_1 x + a_2 y$ minimiert. Wir wählen nun die $a_i$ so dass, $w(x^*, y^*) = \tilde{c}$ und $w(x^0, y^0) = 0$ gilt.

Da unser $P$ in der Menge der Funktionen liegt, über die wir minimiert haben, gilt dann

$$\mathbf{u}_f^T S_f \mathbf{u}_f = \mathbf{u}^T S \mathbf{u} = J_{app}(P) = J(P) \geq J(w).$$

Damit $w$ die Bedingungen in $(x^*, y^*)$ und $(x^0, y^0)$ erfüllt, muss die Gleichung

$$\tilde{c} = w(x^*, y^*) - w(x^0, y^0) = \nabla w(x, y)^T \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = (a_1, a_2) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

gelten, woraus wegen

$$(a_1, a_2) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \leq \left\| \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right\| \left\| \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right\|$$

die Ungleichung

$$a_1^2 + a_2^2 = \left\| \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right\|^2 \geq \frac{\tilde{c}^2}{\left\| \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right\|^2} = \frac{\tilde{c}^2}{\Delta x^2 + \Delta y^2}$$

folgt. Also ergibt sich

$$\mathbf{u}_f^T S_f \mathbf{u}_f \geq J(w) = \iint_\Omega a_1^2 + a_2^2 dx\, dy \geq A(\Omega) \frac{\tilde{c}^2}{\Delta x^2 + \Delta y^2}.$$

Da $\Delta x^2 + \Delta y^2$ durch die Größe von $\Omega$ nach oben beschränkt ist, erhalten wir

$$\mathbf{u}_f^T S_f \mathbf{u}_f \geq C\tilde{c}^2 \geq C\frac{c^2}{K} = C\frac{\|\mathbf{u}_f\|^2}{K},$$

also die behauptete Abschätzung. □

## 4.5 Konvergenzbeweis

Wir wollen in diesem abschließenden Abschnitt den Konvergenzbeweis für unsere Finite Elemente Approximation betrachten.

Wie schon bei den Mehrschrittverfahren wird sich die Konvergenz aus den zwei Eigenschaften *Konsistenz* und *Stabilität* ableiten. Wir beginnen mit der Konsistenz.

**Satz 4.12** Betrachte eine Familie von Dreiecksgittern auf $\Omega$, für die die Winkel der Elemente gleichmäßig von Null wegbeschränkt sind (d.h. es existiert ein $\theta > 0$, so dass alle auftretenden Dreiecke durch $\theta$ nach unten beschränkte Winkel besitzen).

Es sei $u : \Omega \to \mathbb{R}$ die Lösung der Wärmeleitungsgleichung (4.2) und $\mathbf{u}_{ex} \in \mathbb{R}^N$ der zugehörige Vektor der Knotenwerte, also $u_{ex,i} = u(x_i, y_i)$, $i = 1, \ldots, N$. Dann existiert eine von $k$ unabhängige Konstante $C > 0$, so dass für $J_{app}^f$ die Abschätzung

$$J_{app}^f(\mathbf{u}_{ex,f}) \leq \min_{w \in \mathbb{R}^K} J_{app}^f(w) + Ck$$

gilt, wobei $k$ die maximale Größe eines Dreieckselementes im Gitter ist.

**Beweis:** Aus Gleichung (4.13) folgt

$$\min_{v:\Omega\to\mathbb{R}} J(v) \leq \min_{w \in \mathbb{R}^K} J_{app}^f(w) + \frac{C}{2}k,$$

da wir jedem Vektor $w \in \mathbb{R}^K$ durch Interpolation unter Berücksichtigung der Dirichlet–Randbedingungen eine stückweise $C^1$–Funktion $v : \Omega \to \mathbb{R}$ zuordnen können und umgekehrt. Aus Satz 4.9 folgt, dass $J(u)$ gerade minimal ist, also

$$J_{app}^f(\mathbf{u}_{ex,f}) \leq J(u) + \frac{C}{2}k = \min_{v:\Omega\to\mathbb{R}} J(v) + \frac{C}{2}k \leq \min_{w \in \mathbb{R}^K} J_{app}^f(w) + Ck,$$

d.h. die Behauptung, wobei die erste Ungleichung wiederum aus (4.13) folgt. □

**Bemerkung 4.13** Beachte, dass in der hier verwendeten Abschätzung (4.13) die Konstante $c(E)$ eingeht, weswegen wir hier explizit verlangt haben, dass die Winkel und damit $c(E)$ für alle auftretenden Dreiecke $E$ gleichmäßig beschränkt sind. □

Dies ist tatsächlich genau die gleiche Art und Weise, wie wir die Konsistenz auch für Mehrschrittverfahren definiert haben: Wir setzen die exakte Lösung in das numerische Schema ein und messen die Abweichung (auch Residuum genannt) bzgl. der numerischen Lösung.

Als zweite Zutat des Konvergenzresultates benötigen wir eine geeignete Stabilitätseigenschaft. Genauer benötigen wir die Tatsache, dass zwei Funktionen dann nahe beieinander liegen, wenn sie das Minimierungsproblem approximativ lösen. Dies wird ziemlich einfach, wenn man den richtigen Begriff für "nahe beieinander" verwendet, also die richtige Norm für den Funktionenraum. Wir machen uns die Sache hier etwas einfacher und vermeiden die Definition auf dem Funktionenraum, stattdessen verwenden wir eine geeignete Norm auf dem Vektorraum $\mathbb{R}^K$ und zeigen, in welchem Sinne die Vektoren der Knotenwerte nahe beieinander liegen.

**Satz 4.14** Es sei $\mathbf{u} \in \mathbb{R}^N$ die numerische Lösung gemäß Satz 4.10. Dann gilt für jeden Vektor $v \in \mathbb{R}^K$, für den die Ungleichung

$$J^f_{app}(v) \leq J^f_{app}(\mathbf{u}_f) + \varepsilon$$

gilt, die Ungleichung

$$\|v - \mathbf{u}_f\| \leq \kappa\sqrt{K}\sqrt{\varepsilon}$$

für eine vom Gitter unabhängige Konstante $\kappa > 0$.

**Beweis:** Die quadratische Funktion $J^f_{app}$ beschreibt gerade einen Paraboloid im $\mathbb{R}^{K+1}$. Mit der Koordinatenverschiebung $(w, J^f_{app}(w)) \to (w - \mathbf{u}_f, J^f_{app}(w) - J^f_{app}(\mathbf{u}_f))$ können wir das Minimum des Paraboloiden in den Punkt $0 \in \mathbb{R}^{K+1}$ verschieben. Die optimale Lösung $\mathbf{u}_f$ verschiebt sich damit in den Nullpunkt, ebenso wie das Minimum von $J^f_{app}$. In den neuen Koordinaten mit $\tilde{w} = w - \mathbf{u}_f$ gilt dann

$$\begin{aligned}
J^{f,neu}_{app}(\tilde{w}) &= J^f_{app}(w) - J^f_{app}(\mathbf{u}) \;=\; J^f_{app}(\tilde{w} + \mathbf{u}_f) - J^f_{app}(\mathbf{u}_f) \\
&= \frac{1}{2}(\tilde{w} + \mathbf{u}_f)^T S_f(\tilde{w} + \mathbf{u}_f) - d^T(\tilde{w} + \mathbf{u}_f) + e - J^f_{app}(\mathbf{u}_f) \\
&= \frac{1}{2}\tilde{w}^T S_f \tilde{w} - \underbrace{(d^T - \mathbf{u}_f^T S_f)}_{=0}\tilde{w} + \underbrace{\frac{1}{2}\mathbf{u}_f^T S_f \mathbf{u}_f - d^T\mathbf{u}_f + e}_{=J^f_{app}(\mathbf{u}_f)} - J^f_{app}(\mathbf{u}_f) \\
&= \frac{1}{2}\tilde{w}^T S_f \tilde{w}
\end{aligned}$$

für alle $\tilde{w} \in \mathbb{R}^K$. Die Bedingung an $v$ bedeutet mittels $J^{f,neu}_{app}$ ausgedrückt gerade

$$J^{f,neu}_{app}(v - \mathbf{u}_f) \leq \varepsilon.$$

Aus (4.15) erhalten wir nun die Ungleichung

$$\varepsilon \geq J_{app}^{f,neu}(v - \mathbf{u}_f) \geq \frac{C}{K}\|v - \mathbf{u}_f\|^2,$$

also

$$\|v - \mathbf{u}_f\| \leq \frac{\sqrt{K}}{\sqrt{C}}\sqrt{\varepsilon}$$

und damit die Behauptung. □

Aus diesen zwei Sätzen können wir nun unseren Konvergenzsatz erhalten.

**Satz 4.15** Betrachte die numerische Lösung $\mathbf{u}$ gemäß Satz 4.10 und die Knotenwerte $\mathbf{u}_{ex}$ der exakten Lösung der Wärmeleitungsgleichung auf den finiten Elementen. Dann gilt für alle Gitter mit $N \sim 1/k^2$ Knoten und gleichmäßig von Null wegbeschränkten Winkeln die Abschätzung

$$\frac{1}{N}\|\mathbf{u} - \mathbf{u}_{ex}\| \leq Ck\sqrt{k}$$

für eine von $k$ unabhängige Konstante $C$.

**Beweis:** Aus Satz 4.12 folgt

$$J_{app}^f(\mathbf{u}_{ex,f}) \leq \min_{w \in \mathbb{R}^K} J_{app}^f(w) + C_1 k = J_{app}^f(\mathbf{u}_f) + C_1 k.$$

Aus Satz 4.14 folgt damit

$$\|\mathbf{u} - \mathbf{u}_{ex}\| = \|\mathbf{u}_f - \mathbf{u}_{ex,f}\| \leq \kappa\sqrt{C_1}\sqrt{K}\sqrt{k} \leq \kappa\sqrt{C_1}\sqrt{N}\sqrt{k},$$

da die Vektoren in den durch die Dirichlet–Randbedingung festgelegten Knoten übereinstimmen. Also folgt

$$\frac{1}{N}\|\mathbf{u} - \mathbf{u}_{ex}\| \leq \kappa\sqrt{C_1}\frac{\sqrt{k}}{\sqrt{N}} \leq Ck\sqrt{k}.$$

□

Wir können also erwarten, dass der *gemittelte* Fehler über alle Knoten superlinear gegen Null konvergiert, wobei es vorkommen kann, dass in einzelnen Knoten durchaus sehr große Fehler auftreten.

**Bemerkung 4.16** Mit einer feineren Abschätzung des Fehlers bei der numerischen Approximation des quadratischen Integralanteils von $J$ kann man zeigen, dass der Fehler tatsächlich $\leq Ck^2$ ist. Diese genauere Analyse benötigt allerdings tiefere Kenntnisse der Funktionalanalysis, die wir hier nicht voraussetzen oder einführen wollten. ▫

# Literaturverzeichnis


[1] Aulbach, B.: *Gewöhnliche Differenzialgleichungen.* 2. Elsevier–Spektrum Verlag, Heidelberg, 2004

[2] Deuflhard, P. ; Bornemann, F.: *Numerische Mathematik. II: Integration gewöhnlicher Differentialgleichungen.* 2. Auflage. de Gruyter, Berlin, 2002

[3] Hairer, E. ; Wanner, G.: *Solving ordinary differential equations. II. Stiff and differential-algebraic problems.* 2nd edition. Springer-Verlag, Berlin, 1996. – (2nd revised and updated printing, 2002)

[4] Kloeden, P. E. ; Platen, E.: *Numerical Solution of Stochastic Differential Equations.* Springer–Verlag, Heidelberg, 1992. – (3rd revised and updated printing, 1999)

[5] Kloeden, P. E. ; Platen, E. ; Schurz, H.: *Numerical solution of SDE through computer experiments.* Springer-Verlag, Berlin, 1994 (Universitext). – (2nd revised and updated printing, 1997)

[6] Stoffel, A.: *Finite Elemente und Wärmeleitung.* VCH Verlagsgesellschaft, Weinheim, 1992

# Index