



Fachbereich:
Angewandte Naturwissenschaften
und Technik

Studiengang:
Technomathematik

Schwerpunkt:
Operations Research

Implementierung numerischer Lösungsverfahren und graphischer Darstellungsmethoden für nicht lineare Differentialgleichungen

Diplomarbeit von Anne Meckelholt

Jülich, November 2006

Die Arbeit ist von mir selbstständig angefertigt und verfasst. Es sind keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden.

(Anne Meckelholt)

Diese Arbeit wurde betreut von:

und Herrn Prof. Dr. Martin Reißel
 Herrn Dr.-Ing. Ralf Sievering

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 9 |
| 2 | Mathematische Grundlagen | 10 |
| 2.1 | Differentialgleichungen | 10 |
| 2.2 | Taylorentwicklung | 14 |
| 2.3 | Quadraturformeln | 14 |
| 2.4 | Fouriertransformation | 17 |
| 2.5 | Wavelettransformation | 23 |
| 2.5.1 | Einleitung | 23 |
| 2.5.2 | Kontinuierliche Wavelettransformation | 23 |
| 2.5.3 | Diskrete Wavelettransformation | 26 |
| 2.5.4 | Multiskalenanalyse | 27 |
| 2.5.5 | Daubechies-Wavelets | 31 |
| 3 | Numerische Lösungsverfahren (Einschrittverfahren) | 34 |
| 3.1 | Herleitung | 34 |
| 3.2 | Konvergenz | 35 |
| 3.3 | Runge-Kutta-Verfahren | 37 |
| 3.3.1 | Konstante Schrittweite | 37 |
| 3.3.2 | Variable Schrittweite | 40 |
| 4 | Graphische Darstellungsmethoden | 46 |
| 4.1 | Ort-Zeit-Verlauf | 46 |
| 4.2 | Phasendiagramm | 47 |
| 4.3 | Leistungsspektrum | 50 |
| 4.4 | Waveletanalyse | 51 |
| 4.5 | Vergleich der Darstellungsmethoden | 54 |
| 5 | Benutzeranleitung | 57 |
| 5.1 | Die graphische Oberfläche | 57 |
| 5.2 | Verwendung im Worksheet | 63 |

| | | |
|----------|--|-----------|
| 6 | Vergleich der Runge-Kutta-Verfahren für ausgewählte nicht lineare Differentialgleichungen | 69 |
| 6.1 | Stabilität | 70 |
| 6.2 | Rechenzeit | 72 |
| 6.3 | Genauigkeit | 75 |
| 7 | Zusammenfassung | 82 |
| A | Quellcode | 85 |

Abbildungsverzeichnis

| | | |
|------|---|----|
| 2.1 | Mittelpunktregel | 16 |
| 2.2 | Haar-Wavelet | 24 |
| 2.3 | Tochterfunktionen des Haar-Wavelets | 25 |
| 3.1 | Algorithmus des Runge-Kutta-Verfahrens mit konstanter Schrittweite | 40 |
| 3.2 | Algorithmus des schrittweitengesteuerten Runge-Kutta-Verfahrens | 45 |
| 4.1 | Ort-Zeit-Verlauf | 47 |
| 4.2 | Phasenportrait einer ungedämpften Schwingung . . . | 48 |
| 4.3 | Phasenportrait einer gedämpften Schwingung | 49 |
| 4.4 | Phasenportrait einer stark gedämpften Schwingung . | 49 |
| 4.5 | Phasenportrait einer Schwingung mit Sattelpunkt . . | 50 |
| 4.6 | Algorithmus des Leistungsspektrums | 50 |
| 4.7 | Diskrete Waveletanalyse | 51 |
| 4.8 | Algorithmus der diskreten Wavelettransformation . . | 52 |
| 4.9 | Algorithmus der schnellen Wavelettransformation . . | 54 |
| 4.10 | Vergleich der Darstellungsmethoden | 56 |
| 5.1 | Hauptmaplet | 58 |
| 5.2 | Maplet: Integration fortsetzen | 60 |
| 5.3 | Maplet: diskrete Waveletanalyse | 61 |
| 5.4 | Maplet: schnelle Waveletanalyse | 62 |
| 6.1 | Stabilitätsgebiete der Runge-Kutta-Verfahren | 71 |
| 6.2 | Rechenzeit der Runge-Kutta-Verfahren mit konstanter Schrittweite | 72 |
| 6.3 | Vergleich der Genauigkeit der Lösungen für verschiedene Schrittweiten | 76 |
| 6.4 | Vergleich der Genauigkeit der schrittweitengesteuerten Verfahren | 77 |
| 6.5 | Vergleich der Genauigkeit der Lösungen für verschiedene Schrittweiten (lineare Differentialgleichung) . . . | 78 |

Tabellenverzeichnis

| | | |
|-----|--|----|
| 2.1 | Waveletkoeffizienten der Daubechies-Wavelets | 33 |
| 5.1 | Parameter der Funktion „RK_feste_Schrittweite“ . . | 64 |
| 5.2 | Aufgaben der Plotfunktionen | 65 |
| 5.3 | Parameter der Plotfunktionen | 66 |
| 5.4 | Parameter der Funktion „dwt“ | 67 |
| 5.5 | Parameter der Funktion „fwt“ | 68 |
| 6.1 | Butcher-Schemata der verwendeten Runge-Kutta-Verfahren mit konstanter Schrittweite | 69 |
| 6.2 | Butcher-Schemata der verwendeten Runge-Kutta-Verfahren mit variabler Schrittweite | 70 |
| 6.3 | Anzahl der Funktionsauswertungen der Runge-Kutta-Verfahren mit konstanter Schrittweite | 73 |
| 6.4 | Rechenzeit der Runge-Kutta-Verfahren mit variabler Schrittweite | 74 |
| 6.5 | Fehlerverhalten | 81 |

Kapitel 1

Einleitung

Diese Diplomarbeit wurde im Zusammenhang zu einem Bericht für den Verein Deutscher Ingenieure erstellt. Der Bericht über die Analyse von Schwingungen soll es ermöglichen, anhand von Schwingungsdarstellungen wie dem Ort-Zeit-Verlauf, dem Phasendiagramm oder dem Leistungsspektrum auf die zu Grunde liegende Schwingung zu schließen. Um dies zu erreichen, ist es sinnvoll, den umgekehrten Weg zu gehen, d.h. eine gegebene Schwingung zu untersuchen und die Ergebnisse darzustellen. In der Diplomarbeit wurde dieser Ansatz umgesetzt. Mit Hilfe eines Mapleprogramms wurde für eine Vielzahl von Schwingungstypen das Ermitteln des Ort-Zeit-Verlaufs, des Phasendiagramms, des Leistungsspektrums und einiger anderer Darstellungsarten der Schwingung ermöglicht.

Schwingungen werden in der Regel durch nicht lineare Differentialgleichungen beschrieben und können nicht oder nur sehr schwer analytisch gelöst werden. Deshalb werden in dieser Diplomarbeit die numerischen Lösungsmethoden von Differentialgleichungen im Vordergrund stehen.

Schwingungen spielen in vielen Bereichen des Lebens eine wichtige Rolle. Alle mehr oder weniger regelmäßig erfolgenden zeitlichen Änderungen von Zustandsgrößen können als Schwingung bezeichnet werden. Typische Alltagsbeispiele hierfür sind das Schwanken der Tageshelligkeit in einem 24 stündigen Rhythmus, das Schwingen von Uhrenpendeln oder von Saiten eines Musikinstrumentes, das Ausbreiten von Wasserwellen und vieles mehr. Vor allem in der Physik werden Schwingungen häufig beobachtet. Zum Beispiel üben Atome und Luftmolekülen bei Schallwellen Schwingungen aus. Auch bei elektrischen Strömen in Radio- und Fernsehgeräten oder in magnetischen Feldern kann man Schwingungen feststellen. [1]

Kapitel 2

Mathematische Grundlagen

Nachdem in der Einleitung die Bedeutung der Schwingungen und damit der Differentialgleichungen, die diese beschreiben, deutlich gemacht wurde, werden in diesem Kapitel die mathematischen Grundlagen, die für die Analyse der Differentialgleichungen benötigt werden, erläutert. Zunächst wird eine kurze Einführung in die Mathematik der Differentialgleichungen gegeben, d.h. es werden Begriffe und einige Eigenschaften der Differentialgleichungen erklärt. Anschließend werden die Taylorentwicklung und die Quadraturformeln näher erläutert. Diese sind beide für die Herleitung der Verfahren zur Lösung der Differentialgleichung (Kapitel 3) nötig. Außerdem ist die Fouriertransformation und die Wavelettransformation für das Verständnis der folgenden Kapitel erforderlich. Sie werden beide bei der Waveletanalyse (Kapitel 4.4) eingesetzt. Die Fouriertransformation benötigt man außerdem beim Erzeugen des Leistungsspektrums (Kapitel 4.3).

2.1 Differentialgleichungen

Es wird zunächst auf die Differentialgleichungen näher eingegangen. Die in diesem Kapitel aufgeführten Beschreibungen basieren auf [3]. Dort können auch weiterführende Einzelheiten nachgelesen werden.

Bei einer Differentialgleichung wird eine Funktion durch ihre Ableitungen beschrieben. Eine Differentialgleichung ist somit eine Gleichung, die eine Funktion und deren Ableitungen enthält.

Definition 1 *Die allgemeine Form einer gewöhnlichen Differentialgleichung erster Ordnung für $x : [t_0, t_e] \rightarrow \mathbb{R}$ ist*

$$F(t, x(t), \dot{x}(t)) = 0 \quad \forall t \in [t_0, t_e]$$

wobei F eine gegebene Funktion ist.

Bemerkung 1 Sie kann in den meisten Fällen auch in der expliziten Form

$$\dot{x} = f(t, x(t)) \quad \forall t \in [t_0, t_e]$$

angegeben werden.

Definition 2 Treten nur Ableitungen von x nach einem Parameter t auf, dann bezeichnet man die Differentialgleichung als gewöhnlich.

Definition 3 Ist $x(t) \in \mathbb{R}$, dann nennt man die Differentialgleichung skalar.

Bezeichnung 1 Die Differentialgleichung hat die Ordnung n , falls auch die n -te Ableitung von x in der Differentialgleichung auftritt, d.h.

$$x^{(n)} = f(t, x, \dot{x}, \ddot{x}, \dots, x^{(n-1)})$$

Die Lösung der Differentialgleichung ist in der Regel nicht eindeutig. Um eine bestimmte Lösung zu fixieren, gibt man einen zusätzlichen Punkt an, durch den die Lösungskurve verlaufen soll, sowie die Ableitungen in diesem Punkt:

$$x^{(i)}(t_0) = x_{0,i}, \quad i = 0, \dots, n-1$$

Die $x_{0,i}, i = 0, \dots, n-1$ bezeichnet man als Anfangswerte.

Definition 4 Ein Anfangswertproblem ist durch eine Differentialgleichung der Ordnung n und n Anfangswerte charakterisiert.

$$x^{(n)}(t) = f(t, x, \dot{x}, \ddot{x}, \dots, x^{(n-1)})$$

$$x^{(i)}(t_0) = x_{0,i}, \quad i = 0, \dots, n-1$$

Oft betrachtet man zeitliche Änderungen mehrerer Größen, die sich gegenseitig beeinflussen. Dies führt zu einem System von Differentialgleichungen.

Bezeichnung 2 Ein Anfangswertproblem auf $T = [t_0, t_e]$ für ein System erster Ordnung der Dimension m ist gegeben durch

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

mit

$$x : T \rightarrow \mathbb{R}^m, \quad f : T \times \mathbb{R}^m \rightarrow \mathbb{R}^m, \quad x_0 \in \mathbb{R}^m$$

Durch eine Dimensionserweiterung kann jedes System mit beliebiger Ordnung in ein System erster Ordnung transformiert werden, wie folgender Satz zeigt.

Satz 1 *Ein Differentialgleichungssystem p -ter Ordnung der Dimension m*

$$z : T \rightarrow \mathbb{R}^m, \quad z^{(p)} = h(t, z, \dot{z}, \ddot{z}, \dots, z^{(p-1)})$$

kann immer in ein System erster Ordnung der Dimension mp umgewandelt werden durch

$$x = (x_1, x_2, \dots, x_p)^T = (z, \dot{z}, \dots, z^{(p-1)})^T, \quad x : T \rightarrow \mathbb{R}^{mp},$$

mit

$$\dot{x} = f(t, x), \quad f(t, x) = (x_2, \dots, x_p, h(t, x_1, x_2, \dots, x_p))^T$$

Da man jedes System in ein System erster Ordnung umwandeln kann, wird im Folgenden nur noch die Differentialgleichung

$$x : T \rightarrow \mathbb{R}^m, \quad T = [t_0, t_e]$$

mit

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

betrachtet.

Anfangswertprobleme müssen bestimmte Eigenschaften besitzen, damit sie ohne größere Probleme gelöst werden können. Zum einen muss eine Lösung existieren und falls eine existiert, sollte sie auch eindeutig sein. Außerdem sollte die Lösung vom Anfangswert stetig abhängen. Es ist weiterhin wünschenswert, dass leicht gestörte Eingangsdaten nicht zu einem großen Fehler in der Lösung führen. Über diese Eigenschaften geben die folgenden Sätze Auskunft.

Satz 2 (Existenzsatz von Peano) *Sei $\Omega \subset \mathbb{R}^{m+1}$ eine Umgebung von (t_0, x_0) und $f : \Omega \rightarrow \mathbb{R}^m$ stetig auf Ω . Dann hat das Anfangswertproblem mindestens eine Lösung, die in Ω existiert. Jede Lösung kann bis zum Rand $\partial\Omega$ von Ω fortgesetzt werden.*

Die Stetigkeit der rechten Seite f der Differentialgleichung ist demzufolge eine hinreichende Bedingung für die Existenz einer Lösung der Differentialgleichung. Ist f zu dem Lipschitz-stetig, so ist das Anfangswertproblem eindeutig lösbar.

Satz 3 (Picard-Lindelöf) *Sei $\Omega \subset \mathbb{R}^{m+1}$ eine Umgebung von (t_0, x_0) , f stetig auf Ω und f lokal Lipschitz-stetig bezüglich x , d.h. für alle $K \subset \Omega$ kompakt existiert eine Konstante $L_K > 0$ mit*

$$\|f(t, \tilde{x}) - f(t, x)\| \leq L_K \|\tilde{x} - x\| \quad \forall (t, x), (t, \tilde{x}) \in K,$$

dann hat das Anfangswertproblem genau eine Lösung x . Sie ist bis $\partial\Omega$ fortsetzbar.

Bemerkung 2 (Stetige Abhängigkeit von x_0) Unter der Voraussetzung von Satz 3 hängt x stetig von x_0 ab, d.h. das Anfangswertproblem ist gut gestellt.

Das Verhalten des Anfangswertproblems in Bezug auf gestörte Eingangsdaten kann durch folgende Definition näher beschrieben werden.

Definition 5 Ist $\dot{y}(t) = f(t, y(t)) + \delta(t)$ mit $y(t_0) = x_0 + \delta_0$ das gestörte Anfangswertproblem mit Störungen $\delta(t)$ und δ_0 im Vergleich zu $\dot{x}(t) = f(t, x(t))$, $x(t_0) = x_0$, dann heißt das Anfangswertproblem Ljapunov stabil, falls $\forall \delta_0$ und δ mit

$$|\delta_0| < \varepsilon, \quad \|\delta(t)\| < \varepsilon \quad \forall t \in T = [t_0, t_e]$$

ein $C > 0$ unabhängig von ε existiert mit

$$\|y(t) - x(t)\| < C\varepsilon \quad \forall t \in T.$$

Ist $t_e = \infty$ und $\lim_{t \rightarrow \infty} \|y(t) - x(t)\| = 0$, so heißt das Anfangswertproblem asymptotisch stabil.

Satz 4 Ist $f(t, x)$ gleichmäßig L -stetig bezüglich x , dann ist das Anfangswertproblem Ljapunov stabil.

Um in den folgenden Kapiteln ein Verfahren zur Lösung von Differentialgleichungen herleiten zu können, benötigt man eine andere Darstellung der Differentialgleichung, die so genannte Volterrasche Integralgleichung.

Satz 5 Das Anfangswertproblem kann in die äquivalente Volterrasche Integralgleichung umgeformt werden:

$$\begin{aligned} \dot{x}(\tau) = f(\tau, x(\tau)), \quad x(t_0) = x_0 &\Leftrightarrow \int_{t_0}^t \dot{x}(\tau) d\tau = \int_{t_0}^t f(\tau, x(\tau)) d\tau \\ &\Leftrightarrow x(t) - x(t_0) = \int_{t_0}^t f(\tau, x(\tau)) d\tau \\ &\Leftrightarrow x(t) = x_0 + \int_{t_0}^t f(\tau, x(\tau)) d\tau \end{aligned}$$

Die Volterrasche-Integralgleichung ist eine implizite Darstellung für eine Evolution Φ . Diese Evolution bildet (t_0, x_0, t) auf $x(t)$ ab, falls Existenz und Eindeutigkeit der Differentialgleichung gegeben sind.

$$\begin{aligned} \Phi : t_0, x_0, t &\rightarrow x(t) \\ \Phi(t, t_0, x_0) &= x_0 + \int_{t_0}^t f(\tau, x(\tau)) d\tau \end{aligned}$$

Für die Evolution Φ kann man eine sehr nützliche Eigenschaft, die Halbgruppeneigenschaft, nachweisen.

Satz 6 Φ besitzt die Halbgruppeneigenschaft, d.h.

$$\Phi(t, t_0, x_0) = \Phi(t, s, \Phi(s, t_0, x_0)) \quad \forall t, t_0, s \quad (2.1)$$

Beweis 1

$$\begin{aligned} \Phi(t, t_0, x_0) &= x_0 + \int_{t_0}^t f(\tau, x(\tau)) d\tau \\ &= x_0 + \int_{t_0}^s f(\tau, x(\tau)) d\tau + \int_s^t f(\tau, x(\tau)) d\tau \\ &= \Phi(s, t_0, x_0) + \int_s^t f(\tau, x(\tau)) d\tau \\ &= \Phi(t, s, \Phi(s, t_0, x_0)) \end{aligned}$$

2.2 Taylorentwicklung

Die Taylorentwicklung wird in der Mathematik oft verwendet. Mit ihrer Hilfe können Funktionen als Reihen dargestellt werden. Bricht man die Reihe nach dem n-ten Glied ab, erhält man eine Näherung für die Funktion, für die der Fehler bekannt ist.

Satz 7 (Taylorsche Formel) Die Funktion $f : (a, b) \rightarrow \mathbb{R}$ sei $(n+1)$ -mal stetig differenzierbar und $x_0 \in (a, b)$ sei gegeben. Dann gilt für alle $x \in (a, b)$

$$\begin{aligned} f(x) = f(x_0) &+ f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots \\ &+ \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x) \end{aligned}$$

wobei für das Restglied $R_n(x)$ gilt:

Es existiert ein $\xi \in (a, b)$ mit

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

[4]

2.3 Quadraturformeln

Als nächstes werden die Quadraturformeln beschrieben. Die Ausführungen stützen sich auf [2].

Mit Hilfe der Quadraturformeln werden Integrale numerisch angenähert. Die zu integrierende Funktion wird interpoliert und von der interpolierten Funktion das Integral bestimmt. Man wertet dabei die zu integrierende Funktion f an Stützstellen aus und gewichtet diese. Die aufsummierten Werte bilden anschließend eine Näherung für das Integral. Um die Stützstellen und Gewichte zu erhalten, kann man zum Beispiel Orthogonalpolynome verwenden.

Definition 6 Polynome $P_k(x) = p_{k0} + p_{k1}x + \dots + p_{kk-1}x^{k-1} + x^k$ heißen orthogonal, falls gilt:

$$\langle P_i, P_j \rangle = \int_a^b P_i(x) P_j(x) dx = \gamma_i \delta_{ij}, \quad \gamma_i \neq 0$$

Die Polynome können mit Hilfe einer einfachen Rekursion berechnet werden.

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x - \frac{\langle x, 1 \rangle}{\langle 1, 1 \rangle} \\ P_k(x) &= (x - a_k) P_{k-1}(x) - b_k P_{k-2}(x) \\ a_k &= \frac{\langle x P_{k-1}, P_{k-1} \rangle}{\langle P_{k-1}, P_{k-1} \rangle} \\ b_k &= \frac{\langle P_{k-1}, P_{k-1} \rangle}{\langle P_{k-2}, P_{k-2} \rangle} \end{aligned}$$

Aus den Orthogonalpolynomen lassen sich die Stützstellen und Gewichte berechnen:

Satz 8 Sei P_{n+1} das $(n+1)$ -te Orthogonalpolynom auf (a, b) mit Nullstellen x_0, \dots, x_n . Dann gilt für die Gauß-Quadratur

$$G_n(f) = \sum_{i=0}^n \beta_i f(x_i)$$

mit den Gewichten

$$\beta_i = \langle P_{n+1}(x), \frac{1}{(x - x_i) P'_{n+1}(x_i)} \rangle, \quad i = 0, \dots, n$$

Im einfachsten Fall, d.h. für $n = 0$, erhält man die Mittelpunkregel. Hierzu wird zunächst $P_1(x)$ berechnet.

$$\begin{aligned} P_1(x) &= x - \frac{\int_a^b x dx}{\int_a^b dx} = x - \frac{[\frac{1}{2}x^2]_a^b}{[x]_a^b} \\ &= x - \frac{1}{2} \frac{b^2 - a^2}{b - a} = x - \frac{1}{2}(b + a) \end{aligned}$$

$P_1(x)$ hat die Nullstelle $x_0 = \frac{1}{2}(b+a)$. Die Stützstelle ist demzufolge die Mitte des Intervalls, für das das Integral berechnet werden soll. Berechnet man die Gewichtung der Stützstelle, erhält man die Intervallbreite:

$$\begin{aligned}\beta_0 &= \left\langle P_1(x), \frac{1}{(x-x_0)P_1'(x_0)} \right\rangle \\ &= \left\langle x - \frac{1}{2}(b+a), \frac{1}{(x - \frac{1}{2}(b+a))1} \right\rangle \\ &= \int_a^b dx \\ &= b - a\end{aligned}$$

Das Integral wird somit insgesamt durch ein Rechteck mit der Intervallbreite als Länge und dem Funktionswert in der Mitte des Intervalls als Breite angenähert. (vgl. Abbildung 2.1)

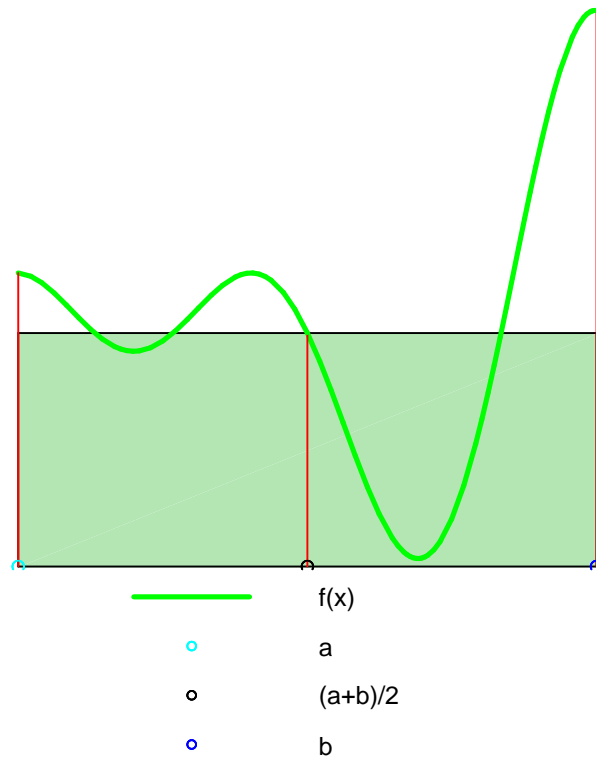


Abbildung 2.1: Mittelpunkregel

Da die Rechnung zur Bestimmung der Stützstellen und Gewichte sehr aufwändig ist, werden die x_i und β_i für das Intervall $[-1, 1]$ tabelliert. Aus diesen können die Stützstellen und Gewichte für ein

beliebiges Intervall $[a, b]$ berechnet werden:

$$\begin{aligned}\tilde{x}_i &= \frac{b-a}{2}x_i + \frac{b+a}{2} \\ \tilde{\beta}_i &= \frac{b-a}{2}\beta_i\end{aligned}$$

2.4 Fouriertransformation

Die Fouriertransformation ist eine Projektion vom Orts- bzw. Zeitraum auf den Frequenzraum. Durch die Fouriertransformation kann man die enthaltenen Frequenzen einer Funktion bestimmen.

Die Fouriertransformation basiert auf den Fourierreihen, durch die periodische Funktionen in eine Reihendarstellung mit sin- und cos-Thermen entwickelt werden. Die Funktion wird dadurch in seine Grundschiwingung und Oberschiwingungen mit den zugehörigen Amplituden zerlegt.

Definition 7 Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ eine T -periodische Funktion, die stetig und stückweise stetig differenzierbar ist, dann konvergiert die Fourierreihe

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n \frac{2\pi}{T} t) + \sum_{n=1}^{\infty} b_n \sin(n \frac{2\pi}{T} t) \quad (2.2)$$

für alle $t \in \mathbb{R}$ und stimmt mit der Funktion f überein. Die Koeffizienten sind

$$a_0 = \frac{1}{T} \int_0^T f(t) dt \quad (2.3)$$

$$a_n = \frac{2}{T} \int_0^T f(t) \cos(n \frac{2\pi}{T} t) dt \quad n = 1, 2, 3, \dots \quad (2.4)$$

$$b_n = \frac{2}{T} \int_0^T f(t) \sin(n \frac{2\pi}{T} t) dt \quad n = 1, 2, 3, \dots \quad (2.5)$$

a_0 gibt die Amplitude zur Grundschiwingung $\omega_0 = \frac{2\pi}{T}$ wieder. Die Koeffizienten $a_n, n = 1, 2, \dots$ und $b_n, n = 1, 2, \dots$ enthalten die Amplituden zu den Oberschiwingungen $n\omega_0 = n\frac{2\pi}{T}$. Die Koeffizienten a_0, a_1, \dots und b_1, b_2, \dots heißen Fourierkoeffizienten.

Zu jeder Kreisfrequenz $n\omega_0$ erhält man durch diese Darstellung zwei Fourierkoeffizienten: a_n und b_n . Durch eine Umformung der Gleichung (2.2) mit Hilfe der Additionstheoreme kann man dies vermeiden. Man erhält folgende Darstellung:

$$f(t) = a_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \varphi_n) \quad (2.6)$$

Anhand dieser Darstellung kann man direkt die Amplituden $A_n = \sqrt{a_n^2 + b_n^2}$ zu den jeweiligen Kreisfrequenzen ablesen. $\varphi_n = \arctan \frac{b_n}{a_n}$ gibt die Phasenverschiebung zum Nullpunkt an.

Eine andere Vereinfachung kann man mit Hilfe der Eulerschen Formel erreichen. Nach Euler gilt:

$$\begin{aligned}\cos(x) &= \frac{1}{2}(e^{ix} + e^{-ix}) \\ \sin(x) &= \frac{1}{2i}(e^{ix} - e^{-ix})\end{aligned}$$

Substituiert man in der Gleichung (2.2) $\cos(x)$ und $\sin(x)$ durch diese Formeln, erhält man die Fourierreihe als eine komplexwertige Funktion.

$$\begin{aligned}f(t) &= a_0 + \sum_{n=1}^{\infty} a_n \frac{1}{2}(e^{in\frac{2\pi}{T}t} + e^{-in\frac{2\pi}{T}t}) + \sum_{n=1}^{\infty} b_n \frac{1}{2i}(e^{in\frac{2\pi}{T}t} - e^{-in\frac{2\pi}{T}t}) \\ &= a_0 + \sum_{n=1}^{\infty} \frac{1}{2}(a_n - ib_n)e^{in\frac{2\pi}{T}t} + \sum_{n=1}^{\infty} \frac{1}{2}(a_n + ib_n)e^{-in\frac{2\pi}{T}t}\end{aligned}$$

Durch einsetzen der Gleichungen (2.4) und (2.5) können die Ausdrücke $\frac{1}{2}(a_n - ib_n)$ und $\frac{1}{2}(a_n + ib_n)$ noch vereinfacht werden.

$$\begin{aligned}c_n &= \frac{1}{2}(a_n - ib_n) \\ &= \frac{1}{2} \left(\frac{2}{T} \int_0^T f(t) \cos(n\frac{2\pi}{T}t) dt - i \frac{2}{T} \int_0^T f(t) \sin(n\frac{2\pi}{T}t) dt \right) \\ &= \frac{1}{T} \int_0^T f(t) e^{-in\frac{2\pi}{T}t} dt\end{aligned}$$

$$\begin{aligned}c_{-n} &= \frac{1}{2}(a_n + ib_n) \\ &= \frac{1}{2} \left(\frac{2}{T} \int_0^T f(t) \cos(n\frac{2\pi}{T}t) dt + i \frac{2}{T} \int_0^T f(t) \sin(n\frac{2\pi}{T}t) dt \right) \\ &= \frac{1}{T} \int_0^T f(t) e^{in\frac{2\pi}{T}t} dt\end{aligned}$$

Mit c_n und c_{-n} kann die Fourierreihe als eine Summe geschrieben werden.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\frac{2\pi}{T}t}, \quad c_n = \frac{1}{T} \int_0^T f(t) e^{-in\frac{2\pi}{T}t} dt \quad (2.7)$$

Der Betrag der Koeffizienten c_n repräsentiert das Amplitudenspektrum, da

$$|c_n| = \left| \frac{1}{2}(a_n - ib_n) \right| = \frac{1}{2} \sqrt{a_n^2 + b_n^2} = \frac{1}{2} A_n$$

Basierend auf Gleichung (2.7) kann die Fouriertransformation hergeleitet werden. Durch die Fouriertransformation wird untersucht, welche Frequenzen mit welchen Amplituden in einem beliebigen Zeitsignal $f(t)$ enthalten sind. Die Funktion muss demzufolge nicht periodisch sein, wie dies bei den Fourierreihen der Fall war.

Um diese Untersuchung durchführen zu können, wird die Funktion als eine periodische Funktion mit der Periode $T \rightarrow \infty$ interpretiert. Hierzu geht man zunächst von einer $2p$ - periodischen Funktion aus und betrachtet den Grenzübergang $p \rightarrow \infty$. Für eine $2p$ - periodische Funktion gilt Gleichung (2.7).

$$\begin{aligned} f(t) &= \sum_{n=-\infty}^{\infty} \left[\frac{1}{2p} \int_0^{2p} f(t) e^{-in \frac{2\pi}{2p} t} dt \right] e^{in \frac{2\pi}{2p} t} \\ &\stackrel{\omega_n = n \frac{\pi}{p}}{=} \sum_{n=-\infty}^{\infty} \left[\frac{1}{2p} \int_0^{2p} f(t) e^{-i\omega_n t} dt \right] e^{i\omega_n t} \\ &\stackrel{f(t) \text{ periodisch}}{=} \sum_{n=-\infty}^{\infty} \left[\frac{1}{2p} \int_{-p}^p f(t) e^{-i\omega_n t} dt \right] e^{i\omega_n t} \\ &\stackrel{\Delta\omega = \omega_n - \omega_{n-1} = \frac{\pi}{p}}{=} \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \Delta\omega \left[\int_{-\pi/\Delta\omega}^{\pi/\Delta\omega} f(t) e^{-i\omega_n t} dt \right] e^{i\omega_n t} \\ &\stackrel{p \rightarrow \infty}{\rightarrow} \frac{1}{2\pi} \int_{-\infty}^{\infty} \underbrace{\left[\int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \right]}_{F(\omega)} e^{i\omega t} d\omega \end{aligned}$$

Die Funktion $F(\omega)$ bezeichnet man als Fouriertransformierte von f . Sie gibt die Amplituden der jeweiligen Kreisfrequenzen der Funktion f wieder.

Satz 9 (Fouriertransformation) Sei $f : \mathbb{R} \rightarrow \mathbb{R}$ stückweise stetig differenzierbar und $\int_{-\infty}^{\infty} |f(t)| dt < \infty$, dann existiert für jedes $\omega \in \mathbb{R}$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (2.8)$$

Durch die Fouriertransformation kann nun der Zeitbereich auf den Frequenzbereich auch für nicht periodische Funktionen abgebildet

werden, jedoch muss die Funktion f in kontinuierlicher Form vorliegen und auf ganz \mathbb{R} definiert sein. In dieser Diplomarbeit werden Funktionen untersucht, von denen nur Werte an diskreten Stellen in einem endlichen Zeitintervall bekannt sind. Für diese abgetasteten, diskreten Funktionen kann die diskrete Fouriertransformation hergeleitet werden. Zunächst geht man hierfür von einem endlichen, kontinuierlichen Zeitsignal $f(t)$ im Zeitintervall $[0, T]$ aus. Außerhalb dieses Intervalls wird die Funktion 0 gesetzt. Mit Hilfe der Funktion definiert man die T - periodische Erweiterung

$$\tilde{f}(t) = \sum_{n=-\infty}^{\infty} f(t + nT).$$

Für diese Funktion kann die Fourierreihe mit den Koeffizienten

$$\begin{aligned} c_m &= \frac{1}{T} \int_0^T \tilde{f}(t) e^{-im\omega_0 t} dt, \quad \omega_0 = \frac{2\pi}{T} \\ &= \frac{1}{T} \int_0^T f(t) e^{-im\omega_0 t} dt \\ &= \frac{1}{T} \int_{-\infty}^{\infty} f(t) e^{-im\omega_0 t} dt \end{aligned}$$

ermittelt werden. Bis auf den Faktor $\frac{1}{T}$ stellen diese Koeffizienten die Fouriertransformierte an den Stellen $m\omega_0$ dar.

$$F(m\omega_0) = Tc_m = \int_0^T f(t) e^{-im\omega_0 t} dt$$

Liegt $f(t)$ nicht als kontinuierliche Funktion, sondern nur als diskretes Signal vor, kann das Integral nur numerisch gelöst werden. Wurden N Abtastzeitpunkte äquidistant gewählt, dass heißt kennt man die Funktion an den Stellen $t_n = \frac{T}{N}n, n = 0, \dots, N-1$, kann das Integral wie folgt mit Hilfe der linksseitigen Rechteckregel angenähert werden:

$$\begin{aligned} \int_0^T f(t) e^{-im\omega_0 t} dt &\approx \sum_{n=0}^{N-1} f(t_n) e^{-im\omega_0 t_n} \Delta t \\ &= \frac{T}{N} \sum_{n=0}^{N-1} f(t_n) e^{-imn \frac{2\pi}{N}} \\ &=: \hat{F}(m\omega_0) \end{aligned}$$

$\hat{F}(m\omega_0)$ ist die diskrete Fouriertransformierte von f .

Formal könnte man nun die Amplituden zu allen Kreisfrequenzen

$m\omega_0$ untersuchen. Tatsächlich erhält man jedoch nur Informationen für die Kreisfrequenzen $m\omega_0$ mit $m = 0, 1, \dots, \frac{N}{2}$, da sich die Spektren für $m > N$ wiederholen und die Amplituden der Kreisfrequenzen für $m = \frac{N}{2} + 1, \dots, N$ konjugiert komplex symmetrisch zur mittleren Frequenz liegen. Dies gilt, da für $k \in \mathbb{Z}$ folgendes nachgewiesen werden kann:

$$\begin{aligned}\hat{F}((N+k)\omega_0) &= \frac{T}{N} \sum_{n=0}^{N-1} e^{-i(N+k)n \frac{2\pi}{N}} = \frac{T}{N} \sum_{n=0}^{N-1} e^{-ikn \frac{2\pi}{N}} e^{-iNn \frac{2\pi}{N}} \\ &= \frac{T}{N} \sum_{n=0}^{N-1} e^{-ikn \frac{2\pi}{N}} = \hat{F}(k\omega_0)\end{aligned}\quad (2.9)$$

$$\begin{aligned}\hat{F}((N-k)\omega_0) &= \frac{T}{N} \sum_{n=0}^{N-1} e^{-i(N-k)n \frac{2\pi}{N}} = \frac{T}{N} \sum_{n=0}^{N-1} e^{+ikn \frac{2\pi}{N}} e^{-iNn \frac{2\pi}{N}} \\ &= \frac{T}{N} \sum_{n=0}^{N-1} e^{+ikn \frac{2\pi}{N}} = \overline{\frac{T}{N} \sum_{n=0}^{N-1} e^{-ikn \frac{2\pi}{N}}} = \overline{\hat{F}(k\omega_0)}\end{aligned}$$

Man erhält also nur die Amplituden für die Kreisfrequenzen $m\omega_0$ mit $m = 0, 1, \dots, \frac{N}{2}$ also für die Frequenzen $f = \frac{m}{T}$, $m = 0, 1, \dots, \frac{N}{2}$. Über welche Frequenzen man Informationen erhält, hängt daher entscheidend von der Intervallbreite und der Anzahl der Abtastpunkte ab. Die kleinste Frequenz ungleich 0, die untersucht wird, ist $f = \frac{1}{T}$. Je größer das zu untersuchende Intervall, um so kleinere Frequenzen werden betrachtet. Die Anzahl der Abtastpunkte N beeinflusst hingegen die Grenze der untersuchten Frequenzen nach oben. Je mehr Abtastpunkte berücksichtigt werden, das heißt je größer die Abtastfrequenz ist, um so größere Frequenzen werden untersucht.

Die diskrete Fouriertransformation ist mit einer Ordnung von $O(N^2)$ sehr rechenintensiv. Mit Hilfe der schnellen Fouriertransformation kann der Aufwand erheblich verringert werden. Die schnelle Fouriertransformation ist ein rekursiver Algorithmus. Die zu berechnende Summe $\frac{T}{N} \sum_{n=0}^{N-1} f(t_n) e^{-imn \frac{2\pi}{N}}$ bei der diskreten Fouriertransformation wird in zwei Summen aufgespalten. Die eine Summe enthält alle Summanden mit geraden Indizes und die andere Summe alle Summanden mit ungeraden Indizes. Durch geschicktes Umformen erhält man einen Ausdruck mit zwei Summen, die wiederum diskre-

te Fouriertransformierte darstellen.

$$\begin{aligned}
\hat{F}(m\omega_0) &= \frac{T}{N} \sum_{n=0}^{N-1} f(t_n) e^{-imn \frac{2\pi}{N}} \\
&= \frac{T}{N} \sum_{n=0}^{\frac{N}{2}-1} f(t_{2n}) e^{-im2n \frac{2\pi}{N}} + \frac{T}{N} \sum_{n=0}^{\frac{N}{2}-1} f(t_{2n+1}) e^{-im(2n+1) \frac{2\pi}{N}} \\
&= \frac{1}{2} \frac{T}{\frac{N}{2}} \sum_{n=0}^{\frac{N}{2}-1} f(t_{2n}) e^{-imn \frac{2\pi}{N}} + e^{-im \frac{2\pi}{N}} \frac{1}{2} \frac{T}{\frac{N}{2}} \sum_{n=0}^{\frac{N}{2}-1} f(t_{2n+1}) e^{-imn \frac{2\pi}{N}} \\
&= \frac{1}{2} \hat{F}_1(m\omega_0) + e^{-im \frac{2\pi}{N}} \frac{1}{2} \hat{F}_2(m\omega_0)
\end{aligned}$$

Da $\hat{F}_1(m\omega_0)$ und $\hat{F}_2(m\omega_0)$ nach Gleichung(2.9) periodisch zur Periode $\frac{N}{2}$ sind, gilt für $m \in [0, \dots, \frac{N}{2} - 1]$:

$$\begin{aligned}
\hat{F}(m\omega_0) &= \frac{1}{2} \hat{F}_1(m\omega_0) + e^{-im \frac{2\pi}{N}} \frac{1}{2} \hat{F}_2(m\omega_0) \\
\hat{F}((\frac{N}{2} + m)\omega_0) &= \frac{1}{2} \hat{F}_1((\frac{N}{2} + m)\omega_0) + e^{-i(\frac{N}{2} + m) \frac{2\pi}{N}} \frac{1}{2} \hat{F}_2((\frac{N}{2} + m)\omega_0) \\
&= \frac{1}{2} \hat{F}_1(m\omega_0) + e^{-i\pi} e^{im \frac{2\pi}{N}} \frac{1}{2} \hat{F}_2(m\omega_0) \\
&= \frac{1}{2} \hat{F}_1(m\omega_0) - e^{-im \frac{2\pi}{N}} \frac{1}{2} \hat{F}_2(m\omega_0)
\end{aligned}$$

Das Berechnen der diskreten Fouriertransformierten \hat{F} mit einem Aufwand von $O(N^2)$ hat man auf diese Weise auf die Auswertung zweier diskreter Fouriertransformierten \hat{F}_1 und \hat{F}_2 mit jeweils $N^2/4$ Operationen zurückgeführt. Durch zusätzliche N Operationen kann aus \hat{F}_1 und \hat{F}_2 \hat{F} gewonnen werden. Insgesamt wurde der Aufwand somit von $O(N^2)$ auf $O(N + 2(N^2/4))$ reduziert.

Natürlich kann man die Berechnung von \hat{F}_1 und \hat{F}_2 auch auf die gleiche Weise durchführen wie \hat{F} , so dass sich ein rekursiver Algorithmus ergibt. Um die Summen immer wieder in zwei gleichgroße Summen aufspalten zu können, fordert man, dass die Anzahl der Abtastpunkte eine Zweierpotenz ist: $N = 2^p$, $p \in \mathbb{N}$. In diesem Fall verringert sich die Komplexität des Algorithmus auf $O(N \log(N))$. Die in diesem Kapitel aufgeführten Beschreibungen basieren auf [12] und [13].

2.5 Wavelettransformation

In diesem Kapitel wird die Wavelettransformation näher betrachtet. Die Ergebnisse, die in diesem Kapitel aufgeführt werden, wurden mit Hilfe von [5],[6],[7], [8] und [9] ermittelt.

2.5.1 Einleitung

Mit Hilfe der Wavelettransformation versucht man, ähnlich wie bei der Fouriertransformation, durch Projektion der Funktion f auf einen Funktionenraum Informationen über die enthaltenen Frequenzen zu erlangen. Die Wavelettransformation kann dabei als lineare Abbildung zwischen gewichteten L^2 -Räumen interpretiert werden. Der große Vorteil bei der Wavelettransformation im Vergleich zur Fouriertransformation ist dabei, dass die Zeitinformation nicht verloren geht. Das heißt, man weiß nicht nur welche Frequenzen in der Funktion enthalten sind, sondern auch wann diese auftreten.

Unter Wavelets versteht man dabei Funktionen, die um die x-Achse oszillieren und schnell abklingen. Der Graph der Funktion ähnelt aus diesem Grund kleinen Wellen, wodurch sich der Name Wavelets erklären lässt.

Definition 8 Eine Funktion $\psi(t) \in L^2(\mathbb{R})$ mit ihrer Fouriertransformierten $\hat{\psi}(\omega)$, die die Gleichung

$$0 < c_\psi := 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty$$

erfüllt, heißt Wavelet.

Bemerkung 3 Diese Bedingung kann nur erfüllt sein, wenn gilt:

$$\hat{\psi}(0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(t) dt = 0$$

Das erste Wavelet wurde von Haar 1910 benutzt (siehe Abbildung 2.2).

2.5.2 Kontinuierliche Wavelettransformation

Bei der kontinuierlichen Wavelettransformation geht man von einem beliebigen Wavelet ψ , zum Beispiel dem Haar-Wavelet, aus. Aus diesem Ausgangswavelet, dem so genannten Mutterwavelet, werden durch kontinuierliche Skalierung und Verschiebung auf der Zeitachse

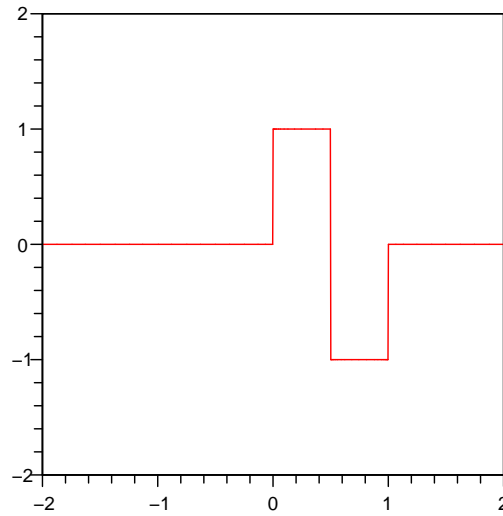


Abbildung 2.2: Haar-Wavelet

so genannte Tochterwavelets generiert:

$$\psi_{ab}(t) = \frac{1}{\sqrt{|a|}} \bar{\psi}\left(\frac{t-b}{a}\right)$$

$\bar{\psi}$ steht für die konjugiert komplexe Funktion von ψ . Der Faktor $\frac{1}{\sqrt{|a|}}$ dient der Normierung.

Das Mutterwavelet wird in diesem Fall um den Faktor b auf der Zeitachse verschoben und mit dem Faktor a skaliert. Einige Tochterwavelets des Haar-Wavelets sind in Abbildung 2.3 dargestellt. Man kann erkennen, dass kleine Skalierungen zu hohen Frequenzen in der Funktion führen. Die Skalierung ist demzufolge invers proportional zur Frequenz:

$$a = \frac{\beta_\psi}{\omega} \quad \text{mit einer Konstanten } \beta_\psi$$

Die Tochterwavelets $\psi_{ab}(t)$ bilden ein Erzeugendensystem des $L^2(\mathbb{R})$. Es ist also möglich eine Funktion als Linearkombination der $\psi_{ab}(t)$ darzustellen. Projiziert man die Ausgangsfunktion f auf diese Funktionen, das heißt, bildet man die Skalarprodukte

$$\langle \psi_{ab}, f(t) \rangle_{L^2},$$

erhält man die kontinuierliche Wavelettransformation.

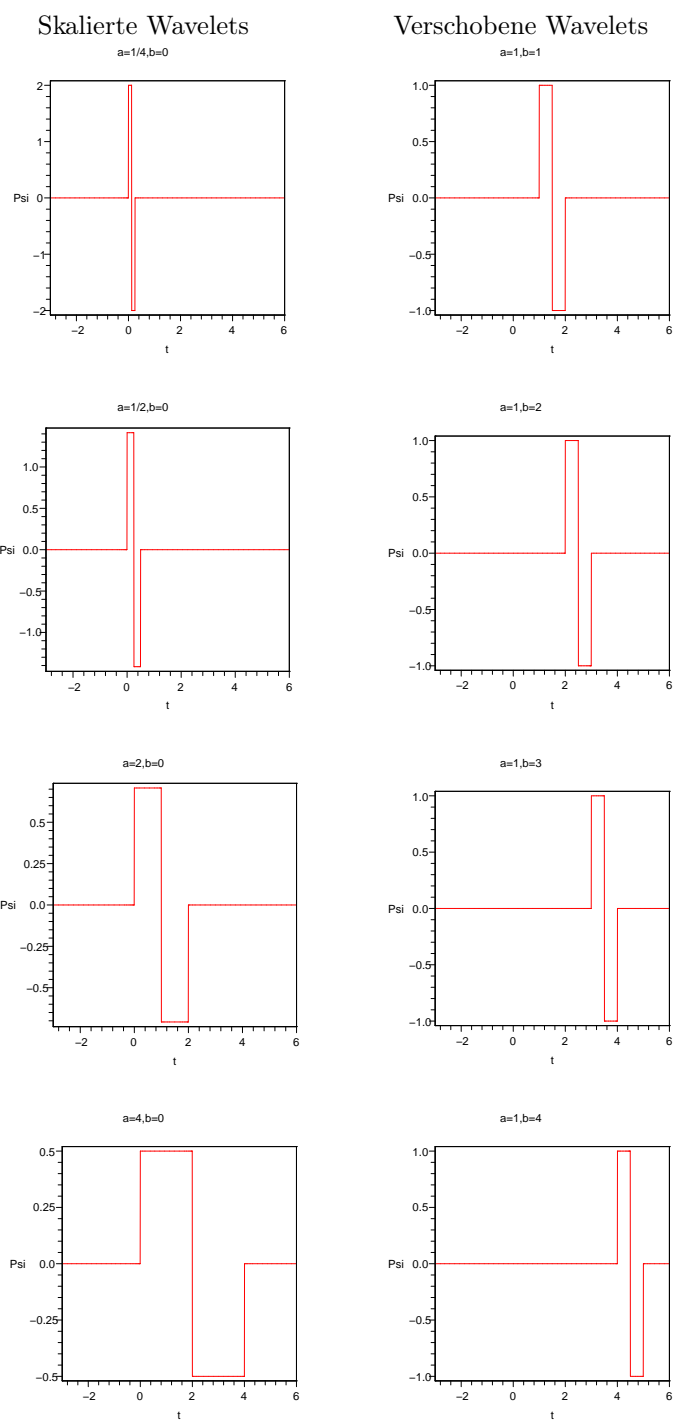


Abbildung 2.3: Tochterfunktionen des Haar-Wavelets

Definition 9 Die kontinuierliche Wavelettransformation ist definiert durch:

$$L_\psi f(a, b) = \frac{1}{\sqrt{c_\psi}} \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \bar{\psi}\left(\frac{t-b}{a}\right) f(t) dt \quad a, b \in \mathbb{R}, a \neq 0$$

Hat $L_\psi f$ an der Stelle (a, b) einen großen Wert, enthält die Funktion f zur Zeit b in etwa die Frequenz $\frac{1}{a}$. Die Wavelettransformation gibt somit darüber Auskunft, welche Frequenz zu welcher Zeit in einer Funktion auftritt.

2.5.3 Diskrete Wavelettransformation

Da es mit Hilfe eines Rechners nicht möglich ist, die kontinuierliche Wavelettransformation durchzuführen, ist es notwendig, diese zu diskretisieren, um eine Implementierung zu ermöglichen. Um den Aufwand bei der kontinuierlichen Wavelettransformation zu verringern, benutzt man bei der diskreten Wavelettransformation ein Erzeugendensystem, das aus weniger Funktionen besteht. Man skaliert und verschiebt das Mutterwavelet nicht mehr kontinuierlich, sondern benutzt diskrete Skalierungs- und Verschiebungsparameter. Die Redundanz, die durch die Transformation entsteht, wird dadurch verringert. Es muss jedoch noch gewährleistet sein, dass der $L^2(\mathbb{R})$ durch die so erzeugten Funktionen aufgespannt werden kann und somit keine Informationen über die Funktion f durch die Transformation verloren gehen. Dies ist gegeben, wenn die Menge der Funktionen einen Frame bilden.

Definition 10 Es sei $\psi \in L^2(\mathbb{R})$ ein Wavelet, $S \subset \mathbb{R}^2$ eine diskrete Menge von reellen Zwei-Tupeln und $B_\psi = \{\psi_{mn} | (m, n) \in S\}$ eine diskrete Familie von Wavelets. B_ψ bildet einen Frame, wenn Konstanten $A > 0$ und $B < \infty$ existieren, so dass für alle $f(x) \in L^2(\mathbb{R})$ gilt:

$$A \|f\|^2 \leq \sum_{(m,n) \in S} \left| \int_{-\infty}^{\infty} \psi_{(m,n)}(t) f(t) dt \right|^2 \leq B \|f\|^2$$

Hat man einen Frame gefunden, kann die diskrete Wavelettransformation analog zur kontinuierlichen definiert werden. Die Skalierungs- und Verschiebungsparameter durchlaufen bei der diskreten Transformation jedoch nicht ganz \mathbb{R} , sondern nur alle Element aus dem Frame.

Definition 11 Die diskrete Wavelettransformation ist definiert durch:

$$L_\psi f(a, b) = \frac{1}{\sqrt{c_\psi}} \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \bar{\psi}\left(\frac{t-b}{a}\right) f(t) dt, \quad (a, b) \in S$$

2.5.4 Multiskalenanalyse

Ein Spezialfall der diskreten Wavelettransformation wird durch die Multiskalenanalyse ermöglicht. Aus dem Mutterwavelet wird kein beliebiges Erzeugendensystem mehr generiert, sondern eine orthogonale Waveletbasis. Der Frame des Wavelets ist somit eine Orthogonalbasis des $L^2(\mathbb{R})$. Dadurch wird die Redundanz auf ein Minimum reduziert. Man kann Wavelets so konstruieren, dass die Waveletfamilien

$$\left\{ \psi_{m,k}(t) = \frac{1}{\sqrt{2^m}} \psi\left(\frac{t - 2^m k}{2^m}\right) \right\}_{(m,k) \in \mathbb{Z}^2}$$

orthogonale Basen des $L^2(\mathbb{R})$ bilden. Die Wavelets, die zu einer Orthogonalbasis führen, können über die Multi-Skalen-Analyse erzeugt werden.

Bei einer Multi-Skalen-Analyse wird das zu untersuchende Signal $f \in V_{-1} \subset L^2(\mathbb{R})$ in seine hoch- und niederfrequenten Anteile aufgespalten. Durch eine orthogonale Projektion $P_0 f$ auf einen Unterraum $V_0 \subset V_{-1}$ wird der niederfrequente (glatte) Anteil des Signals bestimmt. Die hochfrequenten (rauen) Anteile werden analog durch eine orthogonale Projektion $Q_0 f$ auf einen Unterraum W_0 , der das orthogonale Komplement zu V_0 in V_{-1} darstellt, bestimmt. Das Signal kann durch die beiden Projektionen dargestellt werden:

$$\begin{aligned} f &= P_0 f + Q_0 f \\ V_{-1} &= V_0 \oplus W_0 \end{aligned}$$

$P_0 f$ kann nun auf die gleiche Weise wieder in seine hoch- und niederfrequenten Anteile zerlegt werden: $P_0 f = P_1 f + Q_1 f$. Wendet man dieses Verfahren rekursiv an, erhält man folgendes Schema:

$$\begin{array}{ccccccc} V_{-1} & \xrightarrow{P_0} & V_0 & \xrightarrow{P_1} & V_1 & \xrightarrow{P_2} & V_2 & \cdots \\ & \searrow Q_0 & & \searrow Q_1 & & \searrow Q_2 & & \cdots \\ & & W_0 & & W_1 & & W_2 & \cdots \end{array}$$

Das Signal f wird durch die Projektionen immer feiner zerlegt:

$$\begin{aligned} f &= P_0 f + Q_0 f \\ &= P_1 f + Q_1 f + Q_0 f \\ &\dots \\ &= P_m f + \sum_{k=0}^m Q_k f \\ V_{-1} &= \bigoplus_{k=0}^m W_k \oplus V_m \end{aligned}$$

Mathematisch wird die Multi-Skalen-Analyse über die Unterräume V_m definiert:

Definition 12 *Eine Multi-Skalen-Analyse des $L^2(\mathbb{R})$ ist eine aufsteigende Folge abgeschlossener Unterräume $V_m \subset L^2(\mathbb{R})$*

$$\{0\} \subset \dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots \subset L^2(\mathbb{R}),$$

so dass gilt:

$$\overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(\mathbb{R}), \quad (2.10)$$

$$\bigcap_{m \in \mathbb{Z}} V_m = \{0\}, \quad (2.11)$$

$$f(\cdot) \in V_m \Leftrightarrow f(2^m \cdot) \in V_0. \quad (2.12)$$

Es gibt eine Funktion $\phi \in L^2(\mathbb{R})$, deren ganzzahlige Verschiebung eine Riesz-Basis von V_0 erzeugen, das heißt:

$$V_0 = \overline{\text{span}\{\phi(\cdot - k) | k \in \mathbb{Z}\}} \quad (2.13)$$

und

$$A \sum_{k \in \mathbb{Z}} c_k^2 \leq \left\| \sum_{k \in \mathbb{Z}} c_k \phi(\cdot - k) \right\|_{L^2}^2 \leq B \sum_{k \in \mathbb{Z}} c_k^2 \quad (2.14)$$

für alle $\{c_k\}_{k \in \mathbb{Z}} \in l^2(\mathbb{Z})$. A und B sind positive Konstanten.

Die Gleichung (2.10) garantiert die Vollständigkeit, das heißt der $L^2(\mathbb{R})$ kann vollständig durch die Unterräume V_m wiedergegeben werden. Die Unterräume V_m sind zudem möglichst unterschiedlich (Gleichung (2.11)). Das einzige Element, das in allen Räumen vorkommen darf, ist 0. Hierdurch wird Redundanz vermieden. Außerdem sind die Unterräume skalierte Versionen des Grundraums V_0 (Gleichung (2.12)). Wichtiger Bestandteil der Definition ist die Funktion ϕ , die auch als Skalierungsfunktion bezeichnet wird. Durch Verschiebung der Skalierungsfunktion um ganze Zahlen kann der Grundraum V_0 aufgespannt werden (Gleichung (2.13)). Diese Funktion ist entscheidend für die Generierung der orthogonalen Waveletbasen.

Um dies zu erkennen, benötigt man einige Eigenschaften der Räume V_m und der Skalierungsfunktion, die aus der Definition hergeleitet werden können.

$$f \in V_0 \Leftrightarrow f(\cdot - k) \in V_0 \quad k \in \mathbb{Z}$$

Mit Gleichung (2.12) ergibt sich weiter

$$f \in V_m \Leftrightarrow f(\cdot - 2^m k) \in V_m \quad k \in \mathbb{Z}$$

Aus dieser Gleichung kann man schließen, dass der Raum V_m durch die Funktionen

$$\phi_{m,k}(x) = 2^{-\frac{m}{2}} \phi(2^{-m}x - k) \quad (2.15)$$

aufgespannt werden kann. Da $\phi \in V_0$ und V_0 ein Unterraum von V_{-1} ist, liegt ϕ auch in V_{-1} und kann laut Gleichung (2.15) durch die Funktionen $\phi_{-1,k}$ wiedergegeben werden. Es existiert somit eine Folge $\{h_k\}_{k \in \mathbb{Z}}$ reeller Zahlen mit

$$\phi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi(2x - k). \quad (2.16)$$

Die Koeffizienten der Skalierungsgleichung h_k erfüllen die Gleichung:

$$\sum_{k \in \mathbb{Z}} h_k h_{k+2s} = \delta_{0,s} \quad (2.17)$$

da

$$\begin{aligned} \delta_{0,s} &= \langle \phi(\cdot), \phi(\cdot + s) \rangle_{L^2} \\ &\stackrel{(2.16)}{=} \langle \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi(2 \cdot - k), \sqrt{2} \sum_{l \in \mathbb{Z}} h_l \phi(2(\cdot + s) - l) \rangle_{L^2} \\ &= 2 \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} h_k h_l \langle \phi(2 \cdot - k), \phi(2 \cdot + 2s - l) \rangle_{L^2} \\ &\stackrel{\phi \text{ orthonormal}}{=} \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} h_k h_l \delta_{k, l-2s} \\ &= \sum_{k \in \mathbb{Z}} h_k h_{k+2s} \end{aligned}$$

Mit Hilfe dieser Eigenschaften ist es möglich, orthogonale Waveletbasen zu generieren. Die Voraussetzungen und Vorschriften zur Generierung dieser Basis werden im folgenden Satz wiedergegeben.

Satz 10 Sei $\{V_m\}_{m \in \mathbb{Z}}$ eine Multi-Skalen-Analyse, die von der orthogonalen Skalierungsfunktion $\phi \in V_0$ erzeugt wird. Die Funktion $\psi \in V_{-1}$ definiert durch

$$\psi(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \phi(2x - k) = \sum_{k \in \mathbb{Z}} g_k \phi_{-1,k}(x) \quad (2.18)$$

$$g_k = (-1)^k h_{1-k} \quad (2.19)$$

wobei $\{h_k\}_{k \in \mathbb{Z}}$ die Koeffizienten der Skalierungsgleichung (2.16) sind, besitzt die folgenden Eigenschaften:

(i) $\{\psi_{m,k}(\cdot) = 2^{-\frac{m}{2}}\psi(2^{-m} \cdot -k) | k \in \mathbb{Z}\}$ ist eine Orthonormalbasis für W_m

(ii) $\{\psi_{m,k} | m, k \in \mathbb{Z}\}$ ist eine Orthonormalbasis für $L^2(\mathbb{R})$

(iii) ψ ist ein Wavelet mit $c_\psi = 2\pi \int_{\mathbb{R}} |\omega|^{-1} |\hat{\psi}(\omega)|^2 d\omega = 2 \ln 2$

Der Beweis zu diesem Satz kann in [5] nachgelesen werden.

Mit Hilfe des Satzes können nun orthogonale Waveletbasen nach Gleichung (2.18) generiert werden. Außerdem bildet er die Grundlage für eine schnelle Implementierung der Waveletanalyse, die nun im Folgenden beschrieben wird.

Die Voraussetzungen von Satz 10 seien erfüllt. Das heißt V_0 bildet den Grundraum einer Multiskalenanalyse zur orthogonalen Skalierungsfunktion ϕ und ψ ist das Wavelet, das nach der Vorschrift erzeugt wurde. Außerdem sei das zu untersuchende Signal $f \in V_0$. Da V_0 auf Grund der Voraussetzungen der Grundraum einer Multiskalen-Analyse ist und somit nach Definition 12 Gleichung (2.13) durch die Funktionen $\{\phi(\cdot - k) | k \in \mathbb{Z}\}$ aufgespannt werden kann, ist es auch möglich, das Signal f , das in V_0 liegt, als Linearkombination dieser Funktionen darzustellen.

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \phi(x - k)$$

Bei der Wavelettransformation muss nun diese Funktion gemäß Definition 11 auf den Raum, der durch einen Frame aufgespannt wird, projiziert werden. Der Frame ist in diesem Fall die Orthonormalbasis $\{\psi_{m,k} | m, k \in \mathbb{Z}\}$, so dass die Skalarprodukte

$$\langle f, \psi_{m,k} \rangle_{L^2} = \sqrt{c_\psi} L_\psi f(2^m, 2^m k), \quad m \in \mathbb{N}_0, K \in \mathbb{Z}$$

berechnet werden müssen. Dies ist nun auf Grund der Gleichungen (2.16) und (2.18) sehr effizient möglich. Die zu bestimmenden Skalarprodukte $d_k^m = \langle f, \psi_{m,k} \rangle_{L^2}$ können mit Hilfe der Skalarprodukte $c_k^m = \langle f, \phi_{m,k} \rangle_{L^2}$ bestimmt werden, wie folgende Rechnung

zeigt:

$$\begin{aligned}
d_k^m &= \langle f, \psi_{m,k} \rangle_{L^2} & (2.20) \\
&\stackrel{\text{Satz 10(i)}}{=} \langle f, 2^{\frac{-m}{2}} \psi(2^{-m}x - k) \rangle \\
&\stackrel{(2.18)}{=} \langle f, 2^{\frac{-m}{2}} \sqrt{2} \sum_{l \in \mathbb{Z}} g_l \phi(2(2^{-m}x - k) - l) \rangle \\
&= \langle f, 2^{\frac{-m+1}{2}} \sum_{l \in \mathbb{Z}} g_l \phi(2^{-m+1}x - (2k + l)) \rangle \\
&\stackrel{(2.15)}{=} \sum_{l \in \mathbb{Z}} g_l \langle f, \phi_{m-1, (2k+l)} \rangle \\
&= \sum_{l \in \mathbb{Z}} g_{l-2k} \langle f, \phi_{m-1, l} \rangle \\
&= \sum_{l \in \mathbb{Z}} g_{l-2k} c_l^{m-1}
\end{aligned}$$

$$\begin{aligned}
c_k^m &= \langle f, \phi_{m,k} \rangle_{L^2} & (2.21) \\
&\stackrel{(2.15)}{=} \langle f, 2^{\frac{-m}{2}} \phi(2^{-m}x - k) \rangle \\
&\stackrel{(2.16)}{=} \langle f, 2^{\frac{-m}{2}} \sqrt{2} \sum_{l \in \mathbb{Z}} h_l \phi(2(2^{-m}x - k) - l) \rangle \\
&= \langle f, 2^{\frac{-m+1}{2}} \sum_{l \in \mathbb{Z}} h_l \phi(2^{-m+1}x - (2k + l)) \rangle \\
&\stackrel{(2.15)}{=} \sum_{l \in \mathbb{Z}} h_l \langle f, \phi_{m-1, (2k+l)} \rangle \\
&= \sum_{l \in \mathbb{Z}} h_{l-2k} \langle f, \phi_{m-1, l} \rangle \\
&= \sum_{l \in \mathbb{Z}} h_{l-2k} c_l^{m-1}
\end{aligned}$$

Es ist nun möglich, die Skalarprodukte und damit die Wavelettransformation durch die Berechnung der c_k^m und d_k^m zu bestimmen.

2.5.5 Daubechies-Wavelets

Eine spezielle Familie der orthogonalen Wavelets hat Ingrid Daubechies entdeckt. Sie zeichnen sich durch einen kompakten Träger und Glattheit aus. Der Träger eines Wavelets bezeichnet den Bereich, in dem die Waveletfunktion ungleich Null ist. Ist der Träger kompakt, ist dieser Bereich endlich. Bei kurzen Trägern steigt die Lokalisierungseigenschaft, d.h. die Frequenzen werden zeitlich besser lokalisiert.

Außerdem sind bei kompakten Trägern nur wenige Koeffizienten der Skalierungsgleichung ungleich Null. Aus diesem Grund kann die Wavelettransformation bei kompakten Trägern durch vergleichsweise wenige Rechenoperationen ermittelt werden. Die Rechenzeit wird dadurch entscheidend reduziert.

Die Glattheit eines Wavelets ist die Anzahl der stetigen Ableitungen der Waveletfunktion. Damit das Ausgangssignal nach der Transformation wieder rekonstruiert werden kann, sollte die Waveletfunktion in etwa die Glattheit des zu untersuchenden Signals besitzen. Ansonsten können bei der Rekonstruktion Fehler auftreten.

Ingrid Daubechies hat eine ganze Klasse orthogonaler Wavelets mit kompakten Trägern und Glattheitseigenschaften beschrieben. Die einzelnen Daubechies-Wavelets unterscheiden sich dabei durch ihre Ordnung. Die Wavelets werden meist mit "DA" abgekürzt. Die Zahl A entspricht der Anzahl der Waveletkoeffizienten h_k und ist doppelt so groß wie die Ordnung N des Wavelets. Die Ordnung des Wavelets gibt die Anzahl der verschwindenden Momente an. Das k -te Moment ist das Skalarprodukt einer Funktion, hier des Wavelets $\psi(t)$, mit der k -ten Potenz der Variablen t . Ist das Skalarprodukt Null, so spricht man von einem verschwindenden Moment.

Definition 13 Sei ψ_N das Daubechies-Wavelet mit Ordnung N . Dann gilt:

$$\int_{\mathbb{R}} t^m \psi_N(t) dt = 0, \quad m = 0, \dots, N - 1$$

Ein Wavelet N -ter Ordnung steht senkrecht zu allen Polynomen mit Grad höchstens $N - 1$. Das D2-Wavelet (Haar-Wavelet) hat zum Beispiel ein verschwindendes Moment und steht senkrecht auf allen konstanten Funktionen. Das D4-Wavelet hat zwei verschwindende Momente und steht senkrecht zu allen linearen Funktionen. Entsprechendes gilt für höhere Ordnungen.

Die Polynome vom Grad m , $0 \leq m \leq N - 1$ liegen im Grundraum V_0 der Multiskalenanalyse, was eine Voraussetzung zur Durchführung der schnellen Wavelettransformation ist.

Die Waveletkoeffizienten h_k zu den Daubechies-Wavelets der Ordnung 1 bis 4 sind in folgender Tabelle wiedergegeben.

| | D2 | D4 | D6 | D8 | D10 |
|----------|----------------------|--------------------------------|-----------|-----------|------------|
| h_1 | $\frac{1}{\sqrt{2}}$ | $\frac{1-\sqrt{3}}{4\sqrt{2}}$ | +0,332671 | +0,230378 | +0,160102 |
| h_2 | $\frac{1}{\sqrt{2}}$ | $\frac{3-\sqrt{3}}{4\sqrt{2}}$ | +0,806892 | +0,714847 | +0,603829 |
| h_3 | | $\frac{3+\sqrt{3}}{4\sqrt{2}}$ | +0,459878 | +0,630881 | +0,724309 |
| h_4 | | $\frac{1+\sqrt{3}}{4\sqrt{2}}$ | -0,135011 | -0,027984 | +0,138428 |
| h_5 | | | -0,085441 | -0,187035 | -0,242295 |
| h_6 | | | +0,035226 | +0,030841 | -0,032245 |
| h_7 | | | | +0,032883 | +0,077571 |
| h_8 | | | | -0,010597 | -0,006241 |
| h_9 | | | | | -0,012582 |
| h_{10} | | | | | +0,003336 |

Tabelle 2.1: Waveletkoeffizienten der Daubechies-Wavelets

Kapitel 3

Numerische Lösungsverfahren (Einschrittverfahren)

In diesem Kapitel werden die Verfahren zur Lösung von Differentialgleichungen vorgestellt. Da die Runge-Kutta-Verfahren weit verbreitet sind, wird speziell auf diese Methode zur Lösung von Differentialgleichungen eingegangen. Die Ausführungen in diesem Kapitel basieren auf [3].

Im ersten Kapitel wurde gezeigt, dass jede Differentialgleichung in ein System erster Ordnung umgewandelt werden kann. Deshalb sei im Folgenden das Anfangswertproblem gegeben durch

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

Es existiere außerdem eine eindeutige Lösung des Problems.

3.1 Herleitung

Zur Herleitung der Verfahren benutzt man die Evolution (Gleichung (2.1)) d.h. die Volterrasche-Integralgleichung.

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

Da für die Evolution die Halbgruppeneigenschaft (Gleichung (2.1)) gilt, gilt auch:

$$x(t+h) = \Phi(t+h, t, x(t)) = x(t) + \int_t^{t+h} f(\tau, x(\tau)) d\tau.$$

Das Integral kann nicht gelöst werden, da $x(\tau)$ für $t+h$ zu diesem Zeitpunkt noch nicht bekannt ist. Aus diesem Grund wird es durch

numerische Verfahren angenähert, zum Beispiel durch Quadraturformeln. Man erhält dadurch für die exakte Evolution Φ eine genäherte Evolution φ , die gelöst werden kann: $x(t+h) = \varphi(t+h, t, x(t))$

Mit Hilfe dieser Gleichung kann man die Differentialgleichung schrittweise lösen. D. h. man berechnet zunächst

$$x_1 = \tilde{x}(t_1) = \varphi(t_1, t_0, x(t_0)) \quad \text{mit } t_1 = t_0 + h_1$$

und aus dieser Näherung die nächste:

$$x_2 = \tilde{x}(t_2) = \varphi(t_2, t_1, x_1) \quad \text{mit } t_2 = t_1 + h_2$$

usw.

Da zur Berechnung von x_{i+1} nur x_i benötigt wird, bezeichnet man diese Verfahren als Einschrittverfahren.

Für die Bestimmung von φ kann das Integral zum Beispiel durch die linksseitige Rechteckregel approximiert werden, d. h.

$$\int_t^{t+h} f(\tau, x(\tau)) d\tau \approx hf(t, x(t)).$$

φ kann mit dieser Integralnäherung durch

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

ausgedrückt werden. Nähert man mit φ die Differentialgleichung schrittweise an, erhält man das explizite Euler-Verfahren.

Definition 14 *Das explizite Euler-Verfahren ist gegeben durch:*

$$x_{i+1} = \varphi(t_i + h_i, t_i, x_i) = x_i + h_i f(t_i, x_i).$$

3.2 Konvergenz

Durch die Näherung der exakten Evolution Φ durch φ entsteht natürlich ein Fehler. Im ersten Schritt entsteht nur ein Fehler durch die Approximation des Integrals. Bei allen weiteren Schritten entsteht zusätzlich ein Fehler durch die Annäherung des Integranden, da für

den exakten Wert $x(t_i)$ der genäherte Wert x_i benutzt werden muss:

$$\begin{aligned}
e_{i+1} &= x(t_{i+1}) - x_{i+1} \\
&= \Phi(t_i + h, t_i, x(t_i)) - \varphi(t_i + h, t_i, x_i) \\
&= \underbrace{\Phi(t_i + h, t_i, x(t_i)) - \varphi(t_i + h, t_i, x(t_i))}_{\text{lokaler Fehler}} \\
&\quad + \underbrace{\varphi(t_i + h, t_i, x(t_i)) - \varphi(t_i + h, t_i, x_i)}_{\text{Verstärkungsfehler}}
\end{aligned}$$

Der lokale Fehler $l(t_i, x(t_i), h) = \Phi(t_i + h, t_i, x(t_i)) - \varphi(t_i + h, t_i, x(t_i))$ beschreibt die Approximationseigenschaft von φ bezüglich Φ . Das heißt, es wird bei gleichem Anfangswert $x(t_i)$ überprüft, wie gut φ das Integral annähert. Der lokale Fehler wird auch Konsistenzfehler genannt.

Der Verstärkungsfehler $\varphi(t_i + h, t_i, x(t_i)) - \varphi(t_i + h, t_i, x_i)$ zeigt hingegen, wie stark φ auf Störungen in den Anfangswerten reagiert. Zur Beschreibung der Güte von φ führt man zwei Begriffe ein, die sich auf diese beiden Fehlerquellen beziehen:

Definition 15 Ein Einschrittverfahren $\varphi(t + h, t, x)$ heißt konsistent mit Ordnung p in x , falls für den lokalen Fehler (Konsistenzfehler) $l(t, x, h) = \Phi(t + h, t, x) - \varphi(t + h, t, x)$ gilt

$$|l(t, x, h)| = O(h^{p+1}), \quad h \rightarrow 0$$

Je höher die Konsistenzordnung, desto besser approximiert φ Φ .

Definition 16 Ein Einschrittverfahren φ heißt 0-stabil, falls für

$$\begin{aligned}
x_{i+1} &= \varphi(t_i + h, t_i, x_i), \quad x(t_0) = x_0 \text{ und} \\
\tilde{x}_{i+1} &= \varphi(t_i + h, t_i, \tilde{x}_i) + h\delta_{i+1}, \quad \tilde{x}(t_0) = x_0 + \delta_0
\end{aligned}$$

mit $|\delta_k| \leq \varepsilon$ ein $h_0 > 0$ und ein $C_\varepsilon > 0$ existieren, so dass

$$|\tilde{x}_i - x_i| \leq C_\varepsilon \quad \forall i \leq n = \frac{t_e - t_0}{h} \quad \forall h < h_0.$$

Ist das Verfahren 0-stabil, reagiert φ somit nur gering auf Störungen in den Anfangswerten und in der rechten Seite f der Differentialgleichung. Störungen führen nicht zu einem explodierenden Fehler. Die Fehlerordnung wird dadurch nur von der Konsistenzordnung bestimmt.

Satz 11 (Lax-Richtmyer) Sei φ konsistent mit Ordnung p und 0-stabil, so konvergiert φ mit Ordnung p und umgekehrt.

3.3 Runge-Kutta-Verfahren

3.3.1 Konstante Schrittweite

Um Verfahren höherer Ordnung zu erreichen, muss die Konsistenzordnung dementsprechend hoch sein. Um dies zu erreichen, betrachtet man den lokalen Fehler $l(t, y, h) = \Phi(t + h, t, y) - \varphi(t + h, t, y)$. Durch Quadraturformeln versucht man, die Evolution Φ so anzunähern, dass $l(t, y, h)$ eine möglichst hohe Ordnung hat. Man kann das Integral in der Evolution

$$\begin{aligned}\Phi(t + h, t, y) &= y + \int_t^{t+h} f(\tau, x(\tau, t, y)) d\tau \\ &= y + \int_0^h f(t + \tau, x(t + \tau, t, y)) d\tau\end{aligned}$$

zum Beispiel über die Mittelpunkregel annähern. Mit der Mittelpunkregel

$$\int_0^h g(\tau) d\tau = hg\left(\frac{h}{2}\right) + O(h^3)$$

erhält man für die Evolution

$$\Phi(t + h, t, y) = y + hf\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right)\right) + O(h^3).$$

$\varphi(t + h, t, y) = y + hf\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right)\right)$ liefert demzufolge ein Verfahren zweiter Ordnung.

Jedoch ist $x\left(t + \frac{h}{2}, t, y\right)$ noch unbekannt. Deshalb muss auch dieser Wert angenähert werden. Damit das Verfahren weiterhin die Ordnung zwei hat, muss der Fehler bei der Näherung von $x\left(t + \frac{h}{2}, t, y\right)$ der Ordnung drei entsprechen.

Sei $\tilde{x} = x\left(t + \frac{h}{2}, t, y\right) + \delta$ eine gestörte Lösung für $x\left(t + \frac{h}{2}, t, y\right)$, dann gilt für den anzunähernden Ausdruck $hf\left(t + \frac{h}{2}, \tilde{x}\right)$:

$$\begin{aligned}hf\left(t + \frac{h}{2}, \tilde{x}\right) &= hf\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right) + \delta\right) \\ &\stackrel{\text{Taylor um } \delta=0}{=} hf\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right)\right) + \\ &\quad \delta h d_x f\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right)\right) + hO(\delta^2)\end{aligned}$$

Da $\delta d_x f\left(t + \frac{h}{2}, x\left(t + \frac{h}{2}, t, y\right)\right)$ mit h multipliziert wird, reicht es aus, wenn $\delta = O(h^2)$ ist. Nähert man $x\left(t + \frac{h}{2}, t, y\right)$ mit einem Verfahren

erster Ordnung an, ist dieses erfüllt und das Verfahren

$$\varphi(t+h, t, y) = y + hf(t + \frac{h}{2}, x(t + \frac{h}{2}, t, y))$$

hat insgesamt die Ordnung zwei und der Fehler die Ordnung drei. Ein Verfahren mit Konsistenzordnung eins ist zum Beispiel das Euler-Verfahren $\varphi(t+h, t, y) = y + hf(t, y)$ (vgl. Definition 14 auf Seite 35). Benutzt man dieses Verfahren, um $x(t + \frac{h}{2}, t, y)$ anzunähern, erhält man insgesamt:

$$\varphi(t+h, t, y) = y + hf(t + \frac{h}{2}, y + \frac{h}{2}f(t, y))$$

Dies ist ein Runge-Kutta-Verfahren mit Konsistenzordnung zwei, das so genannte Collatz-Verfahren und wird normalerweise in anderer Form angegeben:

$$\begin{aligned} k_1 &= f(t, y) \\ k_2 &= f(t + \frac{h}{2}, y + \frac{h}{2}k_1) \\ \varphi(t+h, t, y) &= y + hk_2 \end{aligned}$$

Um höhere Konsistenzordnungen p zu erreichen, kann ähnlich vorgegangen werden. Man schreibt Φ mit Hilfe der Volterraschen-Integralgleichung $\Phi(t+h, t, y) = y + \int_t^{t+h} f(\tau, x(\tau))d\tau$ und approximiert das Integral durch die Quadraturformel der Ordnung $p+1$. Die inneren Punkte werden mit entsprechend niedriger Ordnung angenähert. Durch diese Vorgehensweise ist das s -stufige Runge-Kutta-Verfahren definiert.

Definition 17 *Ein s -stufiges, Runge-Kutta-Verfahren hat die Form:*

$$\begin{aligned} k_i &= f(t + c_i h, y + \sum_{j=1}^{i-1} a_{ij} k_j) \quad i = 1, \dots, s \\ \varphi(t+h, t, y) &= y + h \sum_{i=1}^s b_i k_i \end{aligned}$$

Die Stufenzahl s gibt dabei an, wie viele Funktionsauswertungen pro Schritt durchgeführt werden müssen.

Die Verfahrensparameter

$$A = \begin{pmatrix} 0 & \cdots & 0 \\ a_{21} & \ddots & \\ \vdots & \ddots & \ddots \\ a_{s1} & \cdots & a_{ss-1} & 0 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_s \end{pmatrix}, \quad c = \begin{pmatrix} c_1 \\ \vdots \\ c_s \end{pmatrix}$$

werden in einem so genannten Butcher-Schema aufgelistet. Ein Butcher-Schema hat folgende Form:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

Das eben hergeleitete Collatz-Verfahren hat das Butcher-Schema

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

Anhand dieses Schemas kann man einige Eigenschaften bestimmen.

Satz 12 *Ein explizites Runge-Kutta-Verfahren ist konsistent für alle $f \in C(T, \mathbb{R}^m)$ genau dann, wenn*

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

Lemma 1 *Hat ein s -stufiges Runge-Kutta-Verfahren die Konsistenzordnung $p \forall f \in C^\infty$, dann ist $p \leq s$, d.h. für Ordnung p braucht man mindestens p Stufen.*

Hat man das Butcher-Schema gegeben, kann eine Differentialgleichung mit Hilfe der Runge-Kutta-Verfahren mit konstanter Schrittweite durch folgenden Algorithmus gelöst werden.

Algorithmus 1 *Sei f die Differentialgleichung, die für den Zeitbereich von $t_0 \in \mathbb{R}$ bis $t_e \in \mathbb{R}$ mit den Anfangsbedingungen $x_0 \in \mathbb{R}^z$ und der Schrittweite h gelöst werden soll. Sei weiterhin $abc \in \mathbb{R}^{m \times n}$ das Butcher-Schema, das dazu verwendet werden soll, dann kann der Algorithmus der Runge-Kutta-Verfahren wie in Abbildung 3.1 beschrieben werden.*

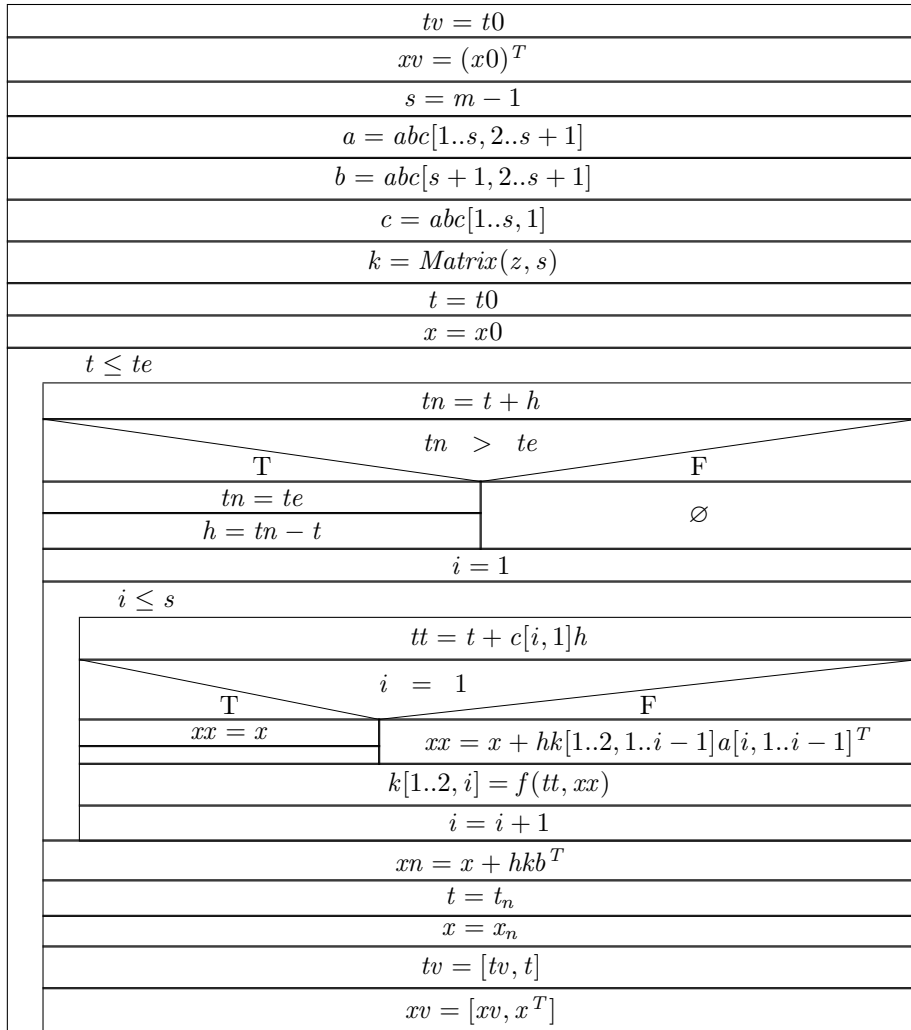


Abbildung 3.1: Algorithmus des Runge-Kutta-Verfahrens mit konstanter Schrittweite

3.3.2 Variable Schrittweite

Bisher war die Schrittweite h in jedem Schritt konstant. Um den Fehler klein zu halten, müsste man konstant eine kleine Schrittweite verwenden. Dies ist aber sehr rechenintensiv und damit teuer. Man versucht deshalb, die Schrittweite zu variieren. Finden im aktuellen Zeitschritt geringe Veränderungen in der wahren Lösung statt, sollte man eine große Schrittweite benutzen. Kann man davon ausgehen, dass im aktuellen Zeitschritt große Veränderungen in der wahren Lösung auftreten, sollte man eine kleine Schrittweite nehmen, um den Fehler unter Kontrolle zu halten.

Um zu überprüfen, ob durch eine kleinere Schrittweite der Fehler im

aktuellen Schritt kleiner wird, schaut man sich den Fehler in jedem Schritt an:

$$\begin{aligned}
e_{i+1} &= x(t_{i+1}) - x_{i+1} \\
&= \Phi(t_i + h_i, t_i, x(t_i)) - \varphi(t_i + h_i, t_i, x_i) \\
&= \underbrace{\Phi(t_i + h_i, t_i, x_i) - \varphi(t_i + h_i, t_i, x_i)}_{l(t_i, x_i, h_i)} + \\
&\quad \underbrace{\Phi(t_i + h_i, t_i, x(t_i)) - \Phi(t_i + h_i, t_i, x_i)}_{v(t_i, x_i, h_i)}
\end{aligned}$$

Den Verstärkungsfehler v der exakten Evolution Φ kann man durch eine kleine Schrittweite im aktuellen Schritt nicht beeinflussen, da

$$v(t_i, x_i, h_i) = \Phi_x(t_i + h_i, t_i, \xi) \underbrace{(x(t_i) - x_i)}_{e_i} \xrightarrow{h_i \rightarrow 0} e_i, \quad \xi \in co(x_i, x(t_i))$$

D.h., auch bei kleinem h geht v nicht gegen 0, sondern gegen den Fehler im vorherigen Schritt. Ist v zu groß, müsste man mit kleinerer Schrittweite von vorne beginnen. Dies ist sehr ineffizient. Man geht deshalb davon aus, dass der Verstärkungsfehler nicht zu stark anwächst, wenn der lokale Fehler l klein gehalten wird.

Der lokale Fehler kann durch Steuerung der Schrittweite beeinflusst werden. Ist φ konsistent mit Ordnung p , so gilt für den lokalen Fehler:

$$l(t_i, x_i, h_i) = c(t_i, x_i)h_i^{p+1} + O(h^{p+2}) \quad (3.1)$$

Da c unabhängig von h ist, kann der lokale Fehler durch h gesteuert werden.

Insgesamt steuert man die Schrittweite somit nur über den lokalen Fehler. Man wählt die Schrittweite, so dass gilt:

$$||l(t_i, x_i, h_i^{opt})|| = \varepsilon \quad (3.2)$$

wobei ε eine vom Benutzer vorgegebene Fehlertoleranz ist. Da der lokale Fehler vom ersten Schritt an kontrolliert wird, geht man davon aus, dass auch v nicht zu stark anwächst und der gesamte Fehler unter Kontrolle gehalten wird.

Zur Ermittlung der optimalen Schrittweite h_i^{opt} wird die Gleichung (3.1) verwendet.

Für die aktuelle Schrittweite gilt:

$$||l(t_i, x_i, h_i)|| \approx ||c(t_i, x_i)||h^{p+1}$$

Für die optimale Schrittweite gilt:

$$\varepsilon = ||l(t_i, x_i, h_i^{opt})|| \approx ||c(t_i, x_i)|| (h_i^{opt})^{p+1}$$

Dividiert man die zweite Gleichung durch die erste, erhält man eine Näherung für h_i^{opt}

$$\begin{aligned} \left(\frac{h_i^{opt}}{h_i} \right)^{p+1} &= \frac{\varepsilon}{\|l(t_i, x_i, h_i)\|} \\ \Leftrightarrow h_i^{opt} &= h_i^{p+1} \sqrt[p+1]{\frac{\varepsilon}{\|l(t_i, x_i, h_i)\|}} \end{aligned}$$

Da für die Berechnung von h_i^{opt} h_i benötigt wird, kann h_i^{opt} erst im Nachhinein, also a-posteriori, bestimmt werden.

Insgesamt hat man ein Verfahren zur Schrittweitensteuerung erhalten.

Man führt zunächst den Schritt von t_i nach $t_i + h_i$ mit gegebenen h_i durch und berechnet $l(t_i, x_i, h_i)$. Ist $\|l(t_i, x_i, h_i)\| \leq \varepsilon$, dann ist x_{i+1} genau genug und kann verwendet werden. Man berechnet außerdem $h_i^{opt} = h_i^{p+1} \sqrt[p+1]{\frac{\varepsilon}{\|l(t_i, x_i, h_i)\|}}$ und benutzt h_i^{opt} als Anfangsschrittweite im nächsten Schritt, d.h. $h_{i+1} = h_i^{opt}$. Ist hingegen $\|l(t_i, x_i, h_i)\| > \varepsilon$, ist x_{i+1} zu ungenau. Man berechnet deshalb h_i^{opt} und bestimmt damit x_{i+1} noch einmal.

Für die Berechnung von $l(t_i, x_i, h_i) = \Phi(t_i + h_i, t_i, x_i) - \varphi(t_i + h_i, t_i, x_i)$ braucht man jedoch die exakte Evolution Φ , die nicht bekannt ist. Deshalb muss l geschätzt werden. Hierzu betrachtet man zwei Verfahren φ und $\hat{\varphi}$ und bildet die Differenz der lokalen Fehler.

$$\begin{aligned} \bar{l} &= \hat{l}(t_i, x_i, h_i) - l(t_i, x_i, h_i) \\ &= \Phi(t_i + h_i, t_i, x_i) - \hat{\varphi}(t_i + h_i, t_i, x_i) - \\ &\quad (\Phi(t_i + h_i, t_i, x_i) - \varphi(t_i + h_i, t_i, x_i)) \\ &= \varphi(t_i + h_i, t_i, x_i) - \hat{\varphi}(t_i + h_i, t_i, x_i) \\ &= x_{i+1} - \hat{x}_{i+1} \end{aligned}$$

Die Schätzung \bar{l} hängt nur noch von den Verfahren φ und $\hat{\varphi}$ ab und kann durch diese berechnet werden.

Ist φ ein genaueres Verfahren als $\hat{\varphi}$, dann ist auch der lokale Fehler $\|l\|$ wesentlich kleiner als $\|\hat{l}\|$. Mit Hilfe der Dreiecksungleichung erhält man dadurch:

$$\|\bar{l}\| = \|\hat{l} - l\| \leq \|\hat{l}\| + \|l\| \approx \|\hat{l}\|$$

Die Schätzung \bar{l} nähert somit den lokalen Fehler des schlechteren Verfahrens $\hat{\varphi}$ an.

$\hat{\varphi}$ sollte so gewählt werden, dass \bar{l} ein asymptotisch exakter Schätzer für \hat{l} ist, d.h.

$$e_r(\bar{l}) = \frac{\|\hat{l} - \bar{l}\|}{\|\hat{l}\|} \xrightarrow{h \rightarrow 0} 0$$

Dies kann man erreichen, wenn $\hat{\varphi}$ so gewählt wird, dass die Ordnung \hat{p} von $\hat{\varphi}$ geringer als die Ordnung p von φ ist. In diesem Fall gilt für den Schätzer \bar{l} :

$$e_r(\bar{l}) = \frac{\|\hat{l} - \bar{l}\|}{\|\hat{l}\|} = \frac{\|l\|}{\|\hat{l}\|} = \frac{ch^{p+1}}{\hat{c}h^{\hat{p}+1}} = \frac{c}{\hat{c}}h^{p-\hat{p}} \xrightarrow{h \rightarrow 0} 0$$

In der Praxis nimmt man zwei Runge-Kutta-Verfahren mit unterschiedlicher Ordnung. Da ein s -stufiges Runge-Kutta-Verfahren gegeben ist durch

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

mit

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

kann man den Aufwand erheblich verringern, wenn die beiden Verfahren die selben k_i benutzen, d.h. die a_{ij} und c_i der beiden Verfahren müssen übereinstimmen. Die k_i müssen dadurch nur einmal berechnet werden. Sind die a_{ij} und c_i identisch, spricht man von einem eingebetteten Runge-Kutta-Verfahren.

Da das Butcher-Schema des Verfahrens der höheren Ordnung das Butcher-Schema der geringeren Ordnung bis auf die b_j enthält, muss man das Butcher-Schema nur noch um diese b_j erweitern, um das eingebettete Runge-Kutta-Verfahren eindeutig zu beschreiben:

| | |
|-----------|-------------|
| c | A |
| p | b^T |
| \hat{p} | \hat{b}^T |

Da man bei der Herleitung der Verfahren φ und $\hat{\varphi}$ nun berücksichtigen muss, dass die a_{ij} und c_i identisch sind, steigt in den meisten Fällen die Stufenzahl für eine gewünschte Konsistenzordnung im Vergleich zum nicht eingebetteten Verfahren an. Dadurch erhöhen sich die Anzahl der Auswertungen von f und damit die Rechenzeit. Dies kann man teilweise umgehen, in dem man fordert:

$$k_s(t_i) = k_1(t_{i+1})$$

Dadurch muss in jeden Schritt ein k weniger berechnet werden. Das Verfahren nennt man deshalb effektiv $(s-1)$ -stufig. Dieses Vorgehen nennt man den Fehlerbergtrick.

Satz 13 *Ein s -stufiges Runge-Kutta-Verfahren ist effektiv $(s-1)$ -stufig, wenn gilt:*

$$c_s = 1, \quad b_s = 0, \quad a_{sj} = b_j, \quad j = 1 \dots s-1$$

Beweis 2

$$\begin{aligned} k_s(t_i) &= k_1(t_{i+1}) \\ \Leftrightarrow f(t_i + c_s h, x + h \sum_{j=1}^{s-1} a_{sj} k_j) &= f(t_{i+1}, x_{i+1}) \\ \Leftrightarrow f(t_i + c_s h, x + h \sum_{j=1}^{s-1} a_{sj} k_j) &= f(t_i + h, x + h \sum_{j=1}^s b_j k_j) \end{aligned}$$

Durch Koeffizientenvergleich erhält man:

$$c_s = 1, \quad b_s = 0, \quad a_{sj} = b_j, \quad j = 1 \dots s-1$$

Durch die eingebetteten Runge-Kutta-Verfahren kann die Schrittweite nun gesteuert werden, ohne die Anzahl der Funktionsauswertungen und die damit verbundene Rechenzeit deutlich zu erhöhen. Insgesamt hat man einen Algorithmus erhalten, mit dem effizient Differentialgleichungen gelöst werden können.

Algorithmus 2 *Sei f die Differentialgleichung, die für den Zeitbereich von $t_0 \in \mathbb{R}$ bis $t_e \in \mathbb{R}$ mit den Anfangsbedingungen $x_0 \in \mathbb{R}^z$ gelöst werden soll. Sei weiterhin $abbc \in \mathbb{R}^{m \times n}$ das eingebettete Butcher-Schema, das dazu verwendet werden soll. Außerdem seien eps_{rel} und $eps_{abs} \in \mathbb{R}$ relative und absolute Genauigkeiten zur Steuerung der Fehlerschranke ε und q_{min} , q_{max} und $\rho \in \mathbb{R}$ seien Sicherheitsfaktoren zur Kontrolle der Schrittweite. Dann kann der Algorithmus der schrittweitengesteuerten Runge-Kutta-Verfahren wie in Abbildung 3.2 beschrieben werden.*

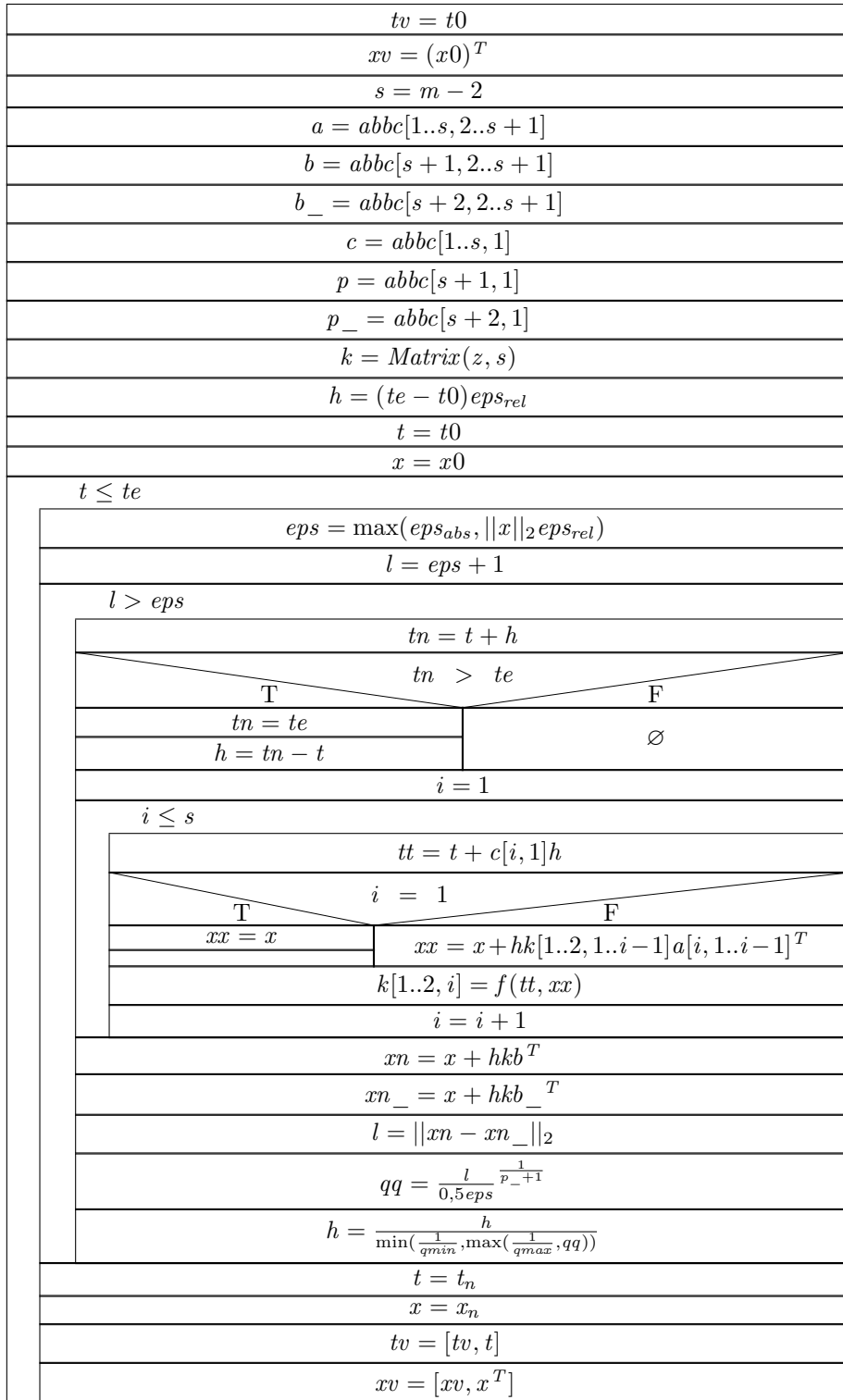


Abbildung 3.2: Algorithmus des schrittweitengesteuerten Runge-Kutta-Verfahrens

Kapitel 4

Graphische Darstellungsmethoden

Die mit Hilfe der Runge-Kutta-Verfahren erhaltene Lösung der Differentialgleichung kann auf verschiedene Weise dargestellt werden. Man kann die Lösung zum Beispiel als Ort-Zeit-Verlauf, als Phasendiagramm, als Leistungsspektrum oder mit Hilfe der Waveletanalyse darstellen. Welche Darstellung man bevorzugen sollte, hängt davon ab, für welche Eigenschaften der Lösung der Differentialgleichung man sich interessiert.

Beim Erzeugen des Leistungsspektrums und der Untersuchung mit Hilfe der Waveletanalyse ist darauf zu achten, dass die Lösung zu äquidistanten Zeitpunkten vorliegt. Löst man die Differentialgleichung mit schrittweitengesteuerten Verfahren, ist dies nicht der Fall. Die erhaltene Lösung muss deshalb zunächst bearbeitet werden. Durch die Lösungspunkte wird ein natürlicher, kubischer Spline gelegt und dieser anschließend an äquidistanten Punkten ausgewertet. Von diesen gesampleten Daten kann das Leistungsspektrum erzeugt und mit ihnen die Waveletanalyse durchgeführt werden.

4.1 Ort-Zeit-Verlauf

Beim Ort-Zeit-Verlauf $x(t)$ werden die berechneten Orte x , die zum Beispiel den Ausschlag eines Pendels beschreiben, über die Zeit dargestellt. Anhand dieser Graphik kann man verschiedene Eigenschaften ablesen. Weist das Signal ein immer wiederkehrendes Muster auf, d.h. wiederholen sich die Schwingungen nach einer bestimmten Zeit, lässt sich zum Beispiel die Periodendauer bestimmen. Die Periodendauer beschreibt die Zeit, die für eine vollständige Schwingung benötigt wird. Sie ist die kürzeste Zeit, nach der sich die Schwingung

wiederholt (vergleiche Abbildung 4.1). Die Periodendauer, oder auch Schwingungsdauer genannt, wird in der Physik meist mit dem Formelbuchstaben T abgekürzt und in Sekunden angegeben. Aus ihr lässt sich die Frequenz der Schwingung bestimmen:

$$f = \frac{1}{T}$$

Bei einer Pendelbewegung von links nach rechts, ist die Periodendauer die Zeit, die das Pendel benötigt, von rechts nach links und wieder zurück zu schwingen.

Eine andere Eigenschaft, die man am Ort-Zeit-Verlauf gut ablesen kann, ist die Amplitude. Die Amplitude beschreibt die maximale Auslenkung einer Schwingung aus der Mittellage (vergleiche Abbildung 4.1). Bei einer Pendelbewegung ist die Amplitude die Distanz zwischen dem Punkt, in dem sich die Bewegungsrichtung umkehrt, und dem Punkt, bei dem der Pendel in Ruhe ist. [10]

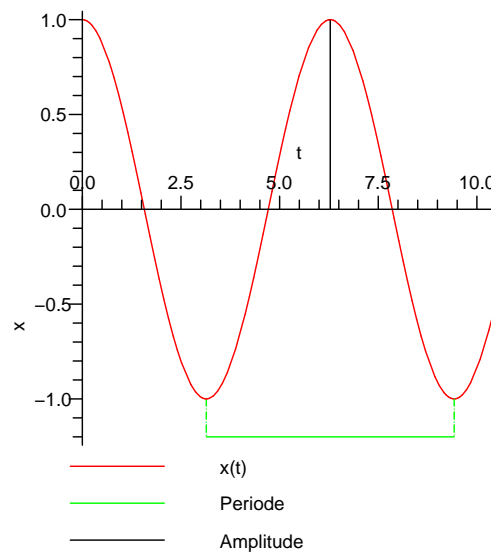


Abbildung 4.1: Ort-Zeit-Verlauf

4.2 Phasendiagramm

Eine andere Darstellungsart ist das Phasendiagramm, bei dem die Ableitung des Ortes \dot{x} in Bezug zum Ort x für jeden Zeitschritt untersucht wird. Es wird demzufolge das Phasendiagramm bzw. die Phasenkurve $\dot{x}(x)$ näher betrachtet. Anhand des Phasendiagramms

kann man Aussagen über die Stabilität und Gleichgewichtspunkte der Differentialgleichung treffen.

Die Phasenkurve kann in der unteren Halbebene nur von rechts nach links und in der oberen Halbebene nur von links nach rechts laufen, da bei positiver Geschwindigkeit \dot{x} x ansteigen muss und bei negativer Geschwindigkeit x abnehmen muss.

Bei einer Geschwindigkeit $v = \dot{x} = 0$ sind die Extremwerte für x zu finden. Bei einer Pendelbewegung ist dies zum Beispiel der größte Ausschlag nach links bzw. nach rechts.

Zeichnet man mehrere Phasenkurven der Differentialgleichung zu unterschiedlichen Anfangsbedingungen, erhält man das Phasenportrait. Anhand des Phasenportraits kann man die Gleichgewichtslagen der Differentialgleichung ablesen. Gleichgewichtslagen sind singuläre Punkte, die bei $\dot{x} = 0 = f(t, x)$ zu finden sind. In diesen Punkten befindet sich der Schwinger in Ruhe. Man unterscheidet dabei je nach Verlauf der Phasenkurven unterschiedliche Gleichgewichtspunkte. Verlaufen die Phasenkurve elliptisch wie in Abbildung

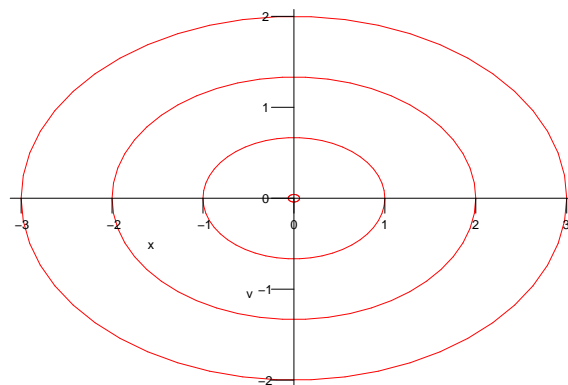


Abbildung 4.2: Phasenportrait einer ungedämpften Schwingung

4.2, so spricht man von einem Wirbelpunkt. Dieses Verhalten ist typisch für eine ungedämpfte Schwingung um eine stabile Gleichgewichtslage. Die Gleichgewichtslage befindet sich in Abbildung 4.2 im Nullpunkt.

Ist eine Dämpfung vorhanden, erhält man das in Abbildung 4.3 dargestellte Phasenportrait. Die elliptischen Phasenkurven werden zu Spiralen, die in einem Strudelpunkt enden. Ist die Dämpfung

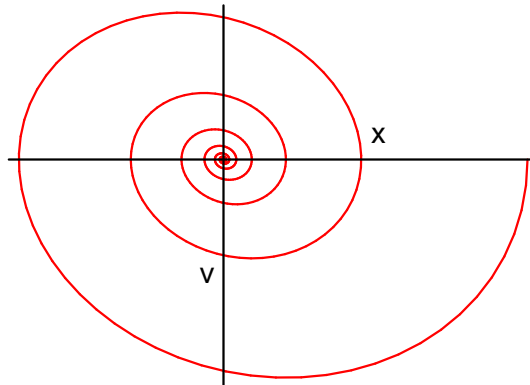


Abbildung 4.3: Phasenportrait einer gedämpften Schwingung

sehr hoch, haben die Spiralen immer weniger Windungen im gleichem Zeitbereich und das Phasenportrait erhält die Gestalt von Abbildung 4.4. Hier wird der Nullpunkt zum Knotenpunkt.

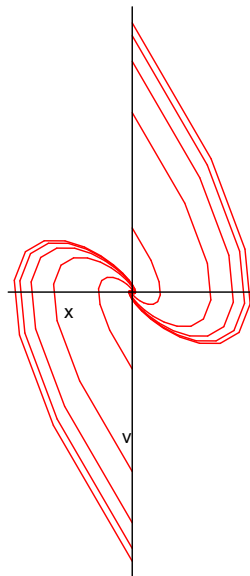


Abbildung 4.4: Phasenportrait einer stark gedämpften Schwingung

Das Phasenportrait in Abbildung 4.5 ist charakteristisch für einen Sattelpunkt. Es gibt zwei Phasenkurven, die durch die Sattelpunkte verlaufen. Alle anderen Phasenkurven haben hyperbelähnliche Gestalt. Das Phasenportrait ist im Gegensatz zu den anderen Phasenportraits typisch für eine instabile Gleichgewichtslage. Aus diesen vier Phasenportraits lassen sich alle Phasenportraits von Schwingern aufbauen. [10]

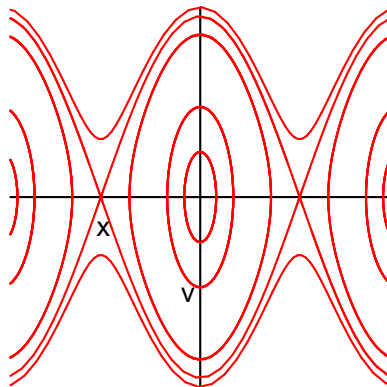


Abbildung 4.5: Phasenportrait einer Schwingung mit Sattelpunkt

4.3 Leistungsspektrum

Interessiert man sich für die enthaltenen Frequenzen der Schwingung, die durch die Differentialgleichung beschrieben wird, können diese mit dem Leistungsspektrum bestimmt werden. Das Leistungsspektrum wird mit Hilfe der Fouriertransformation (siehe Kapitel 2.4) ermittelt und ist durch $|\hat{f}|^2$ definiert. Es ist demzufolge das Quadrat des Amplitudenspektrums.

Die Fouriertransformation ist in Maple schon realisiert und kann mit der Funktion „FourierTransform“ durchgeführt werden. Diese Funktion benutzt den schnellen Algorithmus und kann auch für Datenlängen, die nicht einer Zweierpotenz entsprechen, verwendet werden. Die erhaltenen Fourierkoeffizienten werden quadriert. Da wie in Kapitel 2.4 beschrieben nur die erste Hälfte der Fouriertransformierten die Frequenzen richtig wiedergibt, werden auch nur diese graphisch dargestellt. Der Algorithmus kann folgendermaßen angegeben werden.

Algorithmus 3 Sei $x \in \mathbb{R}^n$ die Lösung der Differentialgleichung zu den äquidistanten Zeitpunkten $t \in \mathbb{R}^n$, dann wird das Leistungsspektrum folgendermaßen berechnet:

| |
|------------------------------------|
| $f = \hat{x}$ |
| $f_conjugiert = \bar{f}$ |
| $spektrum = f \cdot f_conjugiert$ |
| Plot erzeugen |

Abbildung 4.6: Algorithmus des Leistungsspektrums

[11]

4.4 Waveletanalyse

Mit Hilfe der Waveletanalyse kann man die Frequenzen abhängig von der Zeit bestimmen. Es wurden zwei verschiedene Verfahren implementiert, zum einen die diskrete Waveletanalyse und zum anderen eine spezielle Form der diskreten Waveletanalyse basierend auf der Multiskalenanalyse. Beide geben über die enthaltenen Frequenzen zu den jeweiligen Zeiten Auskunft.

Bei der diskreten Waveletanalyse wird eine 3D-Graphik erzeugt, mit den Achsen Zeit, Skalierung und Amplitude. Die Skalierung ist dabei in etwa der reziproke Wert der Frequenz. Ein Beispiel für diese Graphik ist in Abbildung 4.7 dargestellt.

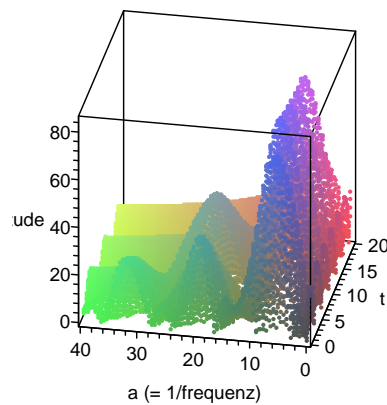


Abbildung 4.7: Diskrete Waveletanalyse

Die diskrete Wavelettransformation kann mit Hilfe der Definition 11 durchgeführt werden. Für die vom Benutzer vorgegebenen Skalierungs- und Verschiebungsparameter werden die Wavelettransformierten berechnet. Die Transformation kann zu einem Informationsverlust führen, da die vorgegebenen Skalierungs- und Verschiebungsparameter in der Regel keinen Frame bilden.

Bei der Implementierung wurde die Integration

$$\int_{-\infty}^{\infty} \bar{\psi}\left(\frac{t-b}{a}\right) f(t) dt$$

durch die endliche Summe

$$\sum_{t=1}^N \bar{\psi}\left(\frac{t-b}{a}\right) f(t)$$

ersetzt, da die Lösung der Differentialgleichung $f(t)$ nur an N diskreten Punkten vorliegt.

Algorithmus 4 Sei $x \in \mathbb{R}^n$ der Lösungsvektor der Differentialgleichung zu den äquidistanten Zeitpunkten $t \in \mathbb{R}^n$, $a \in \mathbb{R}^m$ der Vektor der Skalierungen und $b \in \mathbb{R}^p$ der Vektor der Verschiebungen, dann kann die diskrete Wavelettransformation mit ψ als Wavelet folgendermaßen implementiert werden:

| | |
|---|--|
| $L = \text{Matrix}(m, p)$ | |
| $i = 1$ | |
| $i \leq m$ | |
| $j = 1$ | |
| $j \leq p$ | |
| $L[i, j] = \sum_{z=1}^n \bar{\psi}\left(\frac{t[z]-b[j]}{a[i]}\right) \frac{x[z]}{\sqrt{a[i]}}$ | |
| $j = j + 1$ | |
| $i = i + 1$ | |
| Plot erzeugen | |

Abbildung 4.8: Algorithmus der diskreten Wavelettransformation

Die Matrix L enthält anschließend die Wavelettransformierten zu den angegebenen Skalierungen und Verschiebungen.

Bei der schnellen Wavelettransformation mit Hilfe der Multiskalenanalyse werden 2D-Graphiken erzeugt. Für jeden Skalierungswert werden die Wavelettransformierten, die zu dieser Skalierungsstufe gehören, angezeigt. Der Skalierungsparameter durchläuft dabei die Werte

$$h2^m, \quad m = 1, \dots, M.$$

Die Skalierungsanzahl M wird vom Benutzer vorgegeben. Da der Skalierungsparameter proportional zur Frequenz ist, werden in etwa die Frequenzen

$$\frac{1}{h2^m}, \quad m = 1, \dots, M$$

untersucht. Die Verschiebungswerte sind dabei ganzzahlige Vielfache des Skalierungswertes:

$$h2^m k$$

Anschließend werden die geglättete Schwingung und alle Wavelet-transformierten wie bei der diskreten Wavelettransformation in einer 3D-Graphik angezeigt. Ein Beispiel für die erzeugten Graphiken zeigt Abbildung 5.4 auf Seite 62.

Für die Implementierung benötigt man zunächst, wie in Kapitel 2.5.4 beschrieben, eine Folge c_k^0 , die die Gleichung

$$f(x) = \sum_{k \in \mathbb{Z}} c_k^0 \phi(x - k), \quad f \in V_0$$

erfüllt. Man kann zeigen, dass die diskreten Lösungswerte der Differentialgleichung eine gute Näherung für c^0 sind. Ausgehend von c^0 ist es nun einfach die Wavelettransformierten mit Hilfe der Gleichung (2.20):

$$d_k^m = \sum_{l \in \mathbb{Z}} g_{l-2k} c_l^{m-1}$$

und Gleichung (2.21):

$$c_k^m = \sum_{l \in \mathbb{Z}} h_{l-2k} c_l^{m-1}$$

zu bestimmen. Da nur wenige Werte der g und h Koeffizienten ungleich 0 sind, reicht es aus, über diese zu summieren. Die Koeffizienten für das Daubechies 2 - Wavelets lauten zum Beispiel:

$$\{g_k\}_{k=-2}^1 = \left\{ \frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{-1 + \sqrt{3}}{4\sqrt{2}} \right\}$$

$$\{h_k\}_{k=0}^3 = \left\{ \frac{1 - \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{1 + \sqrt{3}}{4\sqrt{2}} \right\}$$

Bei der Berechnung der d_k^m müssen dadurch nur die Indizes $l = 2k - 2$ bis $l = 2k + 1$ und bei der Berechnung der c_k^m nur die Indizes $l = 2k$ bis $l = 2k + 3$ berücksichtigt werden. Bei der Implementierung ist außerdem darauf zu achten, dass in Maple die Vektorindizes bei 1 beginnen. Eine Implementierung könnte folgendermaßen realisiert werden:

Algorithmus 5 Sei $c \in \mathbb{R}^{2^n}$ der Lösungsvektor der Differentialgleichung zu den äquidistanten Zeitpunkten $t \in \mathbb{R}^{2^n}$. Weiterhin sei $h \in \mathbb{R}^m$ der Vektor der Skalierungskoeffizienten mit den Indexgrenzen h_{ug} und h_{og} sowie $g \in \mathbb{R}^p$ der Vektor der Waveletkoeffizienten

mit den Indexgrenzen g_{ug} und g_{og} . Außerdem sei $M \in \mathbb{N}$ die gewünschte Skalierungsanzahl. Dann kann die diskrete Wavelettransformation mit ψ als Wavelet mit Hilfe der Multiskalenanalyse folgendermaßen implementiert werden:

| |
|---|
| $n = 2^n$ |
| $c_{alt} = c$ |
| $z = 1$ |
| $z \leq M$ |
| $i = 0$ |
| $i < \frac{n}{2}$ |
| $c[i + 1] = \sum_{l=2i}^{\min(h_{og}+2i, n-1)} h[l - 2i + 1 - h_{ug}] c_{alt}[l + 1]$ |
| $c[i + 1 + \frac{n}{2}] = \sum_{l=2i}^{\min(g_{og}+2i, n-1)} g[l - 2i + 1 - g_{ug}] c_{alt}[l + 1]$ |
| $i = i + 1$ |
| Ergebnis speichern |
| $n = n/2$ |
| $c_{alt} = c$ |
| $z = z + 1$ |
| Plots erzeugen |

Abbildung 4.9: Algorithmus der schnellen Wavelettransformation

Die ersten $\frac{n}{2}$ Elemente des Vektor c beinhalten in jeder Stufe die c_k und die nächsten $\frac{n}{2}$ Elemente die d_k .

4.5 Vergleich der Darstellungsmethoden

Anhand der Differentialgleichung

$$\ddot{x} - (1 - x^2)\dot{x} + x = \cos(t)$$

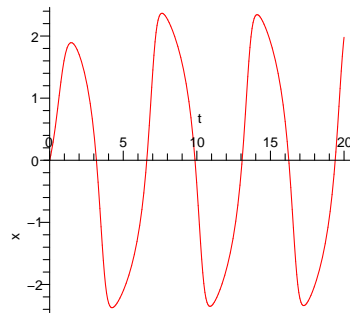
sollen nun die verschiedenen Darstellungsmethoden verglichen werden. Für den Zeitbereich von 0 bis 20 und die Anfangsbedingungen $x(0) = 0$ und $\dot{x}(0) = 1$ erhält man die in Abbildung 4.10 dargestellten Lösungen. Im Ort-Zeit-Verlauf kann man erkennen, dass das Signal nach einer kurzen Zeit periodisch verläuft. Die Periode beträgt etwa 6,5 Sekunden, was einer Frequenz von etwa 0,15 Hz entspricht. Außerdem kann man im Ort-Zeit-Verlauf eine Amplitude von ca. 2,2 ablesen.

Das periodische Verhalten des Signals ist auch im Phasendiagramm ersichtlich. Die Phasenkurve beschreibt nach kurzer Zeit immer wieder dieselbe Bahn. Außerdem kann man keine stabilen Gleichgewichtslagen erkennen. Die Schwingung wird demzufolge nicht in einen stabilen Zustand hinein laufen.

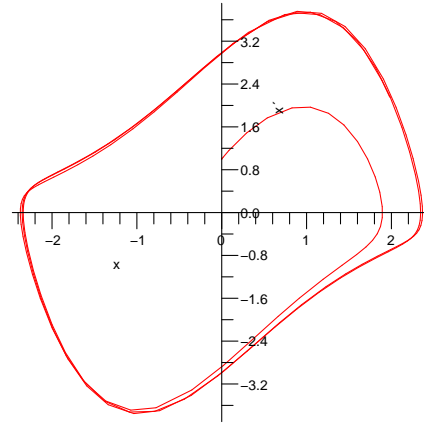
Im Leistungsspektrum kann man wie schon im Ort-Zeit-Verlauf die Frequenz von etwa 0,15 ermitteln. Diese muss jedoch nicht aus der Periodendauer bestimmt werden, sondern kann direkt abgelesen werden. Die Frequenz wurde auch mit Hilfe der diskreten Waveletanalyse ermittelt. Die größten Amplituden sind bei $a = 6$ anzutreffen. Dies entspricht einer Frequenz von etwa 0,17 Hz.

Bei der diskreten Waveletanalyse mit Hilfe der Multiskalenanalyse ist eine Interpretation der Ergebnisse, bedingt durch die wenigen Auswertungspunkte, nur schwer möglich.

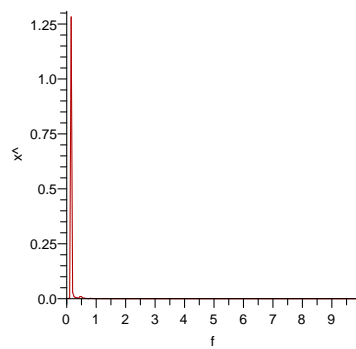
Ort-Zeit-Verlauf



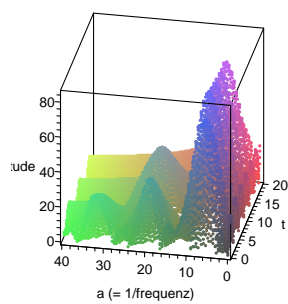
Phasendiagramm



Leistungsspektrum



diskrete Wavelettransformation



schnelle Wavelettransformation

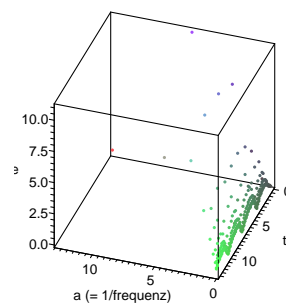


Abbildung 4.10: Vergleich der Darstellungsmethoden

Kapitel 5

Benutzeranleitung

5.1 Die graphische Oberfläche

Das Hauptfenster der graphischen Oberfläche befindet sich im Modul „Schwingungen“ und kann durch das Maplet „Start_Schwingungen“ gestartet werden. Es ist beim Start darauf zu achten, dass die Bibliothek „Schwingungen.lib“ sowie die zugehörige IND-Datei „Schwingungen“ in dem Verzeichnis stehen, aus dem das Maplet gestartet wurde. Alternativ können die Dateien auch im Bibliotheksverzeichnis von Maple abgelegt werden. Nach dem Start des Maplets erhält man das in Abbildung 5.1 dargestellte Eingabefenster.

In der oberen Hälfte befinden sich die Plotflächen, in denen die erzeugten Graphiken nach dem Lösen der Differentialgleichung angezeigt werden. In der linken Plotebene erscheint der Ort-Zeit-Verlauf, in der Mitte das Phasendiagramm und in der rechten Fläche das Leistungsspektrum.

In dem Abschnitt unter den Graphiken kann man die Parameter für die Runge-Kutta-Verfahren einstellen. Die Anfangszeit t_0 gibt an, ab welchem Zeitpunkt die Differentialgleichung gelöst werden soll. Die Endzeit t_e beschreibt den Zeitpunkt, bis zu dem eine Lösung

ermittelt wird. Die Anfangsbedingungen $x_0 = \begin{pmatrix} x(t_0) \\ \dot{x}(t_0) \end{pmatrix}$ müssen in

Listenform angegeben werden, d.h. x_0 wird als $[x(t_0), \dot{x}(t_0)]$ geschrieben. Für Runge-Kutta-Verfahren ohne Schrittweitensteuerung muss außerdem noch die Schrittweite, die für den Abstand zwischen zwei Lösungszeiten steht, angegeben werden.

Im darunter liegenden Bereich kann man die Parameter für die Differentialgleichungen angeben. Alle Variablen in der Differentialgleichung werden durch die eingegebenen Werte ersetzt.

In der nächsten Zeile kann man zunächst die Frequenzauflösung des

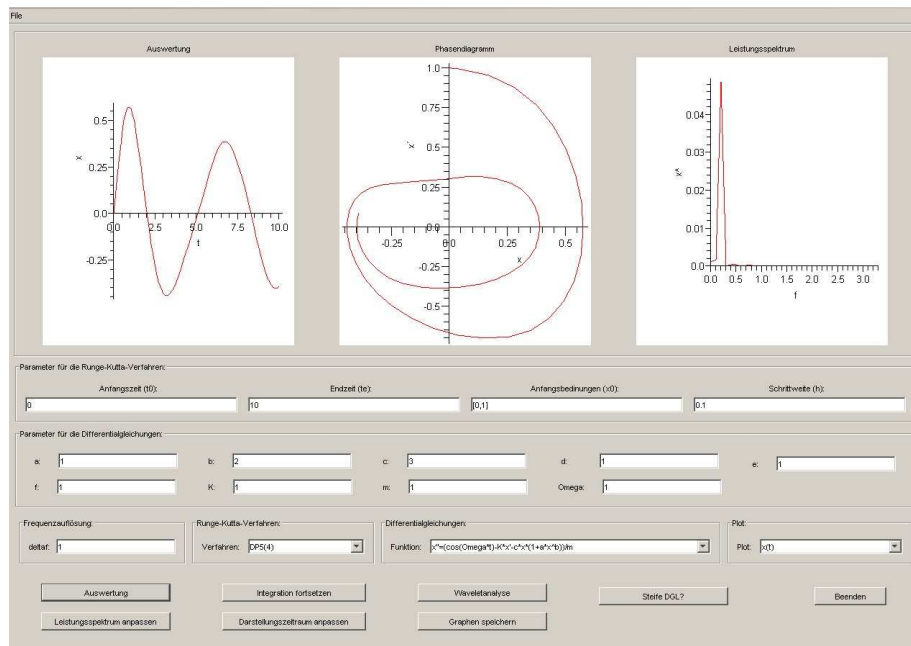


Abbildung 5.1: Hauptmaplet

Leistungsspektrums angeben. Die Frequenzauflösung δf gibt die kleinste Frequenz ungleich null an, die untersucht wird. Sie ist umgekehrt proportional zu der Zeit, für die die Lösung bestimmt wurde.

$$\delta f = \frac{1}{t_e - t_0}$$

Weiterhin kann man das Runge-Kutta-Verfahren wählen, mit dem die Lösung berechnet werden soll. Man kann hier zwischen folgenden Verfahren wählen:

1. Euler
2. Collatz
3. klassischer Runge-Kutta
4. Kutta 3/8
5. Runge-Kutta-Fehlerberg 2(3)
6. Runge-Kutta 4(3)
7. DP5(4)

Die Verfahren 1 bis 4 sind Verfahren ohne Schrittweitensteuerung. Die Schrittweite muss in diesem Fall angegeben werden. Die Verfahren 5 bis 7 sind schrittweitengesteuert. Die Schrittweite wird, wie in Kapitel 3.3.2 beschrieben, automatisch angepasst. Die im Maplet

angegebene Schrittweite wird ignoriert.

Die Differentialgleichung kann ebenfalls ausgewählt werden. Hier stehen folgende zur Auswahl:

- $m\ddot{x} + K\dot{x}(1 + ax^b) = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x} + a \operatorname{sgn}(\dot{x} - b)(f + \frac{d-f}{1+e|\dot{x}-b|}) + cx = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x} + (b + c + \sin(at))x = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x}(a + b\dot{x}^d) + cx = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x} + cx - a \sin(x) = \cos(\Omega t)$
- $m\ddot{x} - K(1 - ax^2)\dot{x} + cx = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x} + \frac{c}{2}x(1 - \operatorname{sgn}(x) \operatorname{sgn}(\operatorname{sgn}(x)a - x)) + dx = \cos(\Omega t)$
- $m\ddot{x} + K\dot{x} + cx + a \operatorname{sgn}(x) = \cos(\Omega t)$
- $\ddot{x} + \dot{x} = 0$
- $m\ddot{x} + K\dot{x} + c \sin(x) = \cos(\Omega t)$

In der Auswahlliste sind die Differentialgleichungen schon nach der höchsten Ableitung \ddot{x} aufgelöst. Da für die Umformung durch „m“ dividiert werden muss, darf „m“ nicht gleich Null sein.

Ist man nicht an den Standardplots interessiert, kann man den gewünschten Plot in der letzten Auswahlliste angeben. Anstatt des Ort-Zeit-Verlauf kann man sich auch $\dot{x}(t)$ (erste Ableitung des Ortes über die Zeit) oder $\ddot{x}(t)$ (zweite Ableitung des Ortes über die Zeit) ansehen. Beim Phasendiagramm ist es möglich, die zweite Ableitung des Ortes über die erste Ableitung des Ortes ($\ddot{x}(\dot{x})$) zeichnen zu lassen. Möchte man im Leistungsspektrum nicht das Quadrat der Amplitude des Ortes über die Frequenz untersuchen, sondern das Quadrat der Amplitude der zweiten Ableitung des Ortes über die Frequenz, kann man $\hat{\ddot{x}}(f)$ durch zwei verschiedene Verfahren berechnen lassen. Das erste Verfahren, bei dem die Amplitude über die Fouriertransformation bestimmt wird, ist jedoch genauer und sollte deshalb bevorzugt werden.

Alle Aktionen werden über die Buttons im unteren Bereich der Mapletoberfläche gesteuert. Zunächst sind nur die Buttons „Auswertung“ und „Beenden“ aktiviert. Für alle anderen Aktionen muss zuerst durch den Button „Auswertung“ eine Lösung des Anfangwertproblems bestimmt werden. Diese wird anschließend automatisch angezeigt, sofern dies möglich ist. Ein Problem tritt beim Zeichnen

auf, wenn die x- bzw y-Werte zu groß sind. Maple sendet in diesem Fall die Fehlermeldung: „com.maplesoft.plot.util.PlotStruktureError: X axis range is too large.“. Ist dieser Fehler aufgetreten muss, die Lösung der Differentialgleichung noch einmal für einen kleineren Zeitbereich oder mit veränderten Parametern bestimmt werden.

Wählt man ein schrittweitengesteuertes Verfahren zur Lösung der Differentialgleichung aus, wird man aufgefordert, die „sample time“ anzugeben. Diese beschreibt den Abstand zwischen zwei äquidistanten Zeitpunkten, für die die Lösung angenähert wird. Das Resampling ist nötig, da das Leistungsspektrum, das automatisch angezeigt wird, nur für Daten, die zu äquidistanten Zeiten ermittelt wurden, erzeugt werden kann.

Mit Hilfe des Buttons "Integration fortsetzen" kann die Differentialgleichung anschließend für einen weiteren Zeitbereich gelöst werden. Es erscheint ein Untermaplet (siehe Abbildung 5.2) mit der Aufforderung, diesen einzugeben. Wurde die Differentialgleichung zum Beispiel schon von $t_0 = 0$ bis $t_e = 10$ gelöst und man gibt den Wert 2 ein, erhält man eine Lösung für den Bereich von $t_0 = 0$ bis $t_e = 12$, ohne dass der schon berechnete Zeitbereich von 0 bis 10 noch einmal ausgewertet wird.



Abbildung 5.2: Maplet: Integration fortsetzen

Die Waveletanalyse wird durch den gleichnamigen Button gestartet. Man kann zwischen der diskreten Waveletanalyse und der Multiskalenanalyse wählen. Bei der diskreten Waveletanalyse erscheint das in Abbildung 5.3 dargestellte Untermaplet. Hier kann man die gewünschten Skalierungs- und Verschiebungsparameter, für die die Wavelettransformation durchgeführt werden soll, angeben. Es wird jeweils der Anfangs- und Endwert sowie die Schrittgröße als Eingabe erwartet. Die Schrittgröße beschreibt den Abstand zwischen zwei auf einander folgenden Werten. Würde man zum Beispiel für den Skalierungsparameter die Werte von 1 bis 10 mit der Schrittweite (Step) 2 wählen, wird für die Skalierung die Werte 1, 3, 5,

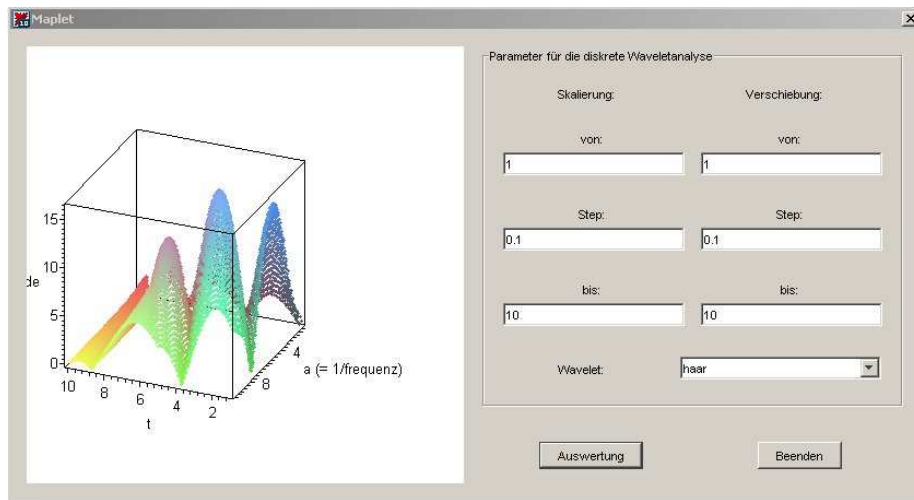


Abbildung 5.3: Maplet: diskrete Waveletanalyse

7, 9 verwendet. Da die Frequenz in etwa dem reziproken Wert der Skalierung entspricht, wird in diesem Fall die Frequenzen $1, \frac{1}{3}, \dots, \frac{1}{9}$ untersucht.

Der Verschiebungsparameter kann analog angegeben werden. Die Verschiebung kann jedoch nur bis zum Ende der Lösungszeit te erfolgen. Ist der Endverschiebungswert größer als te wird dieser auf te gesetzt und der Benutzer erhält eine entsprechende Benachrichtigung.

Weiterhin kann man das gewünschte Wavelet, mit dem die Transformation durchgeführt werden soll, angeben. Der Benutzer hat die Wahl zwischen dem Haar-Wavelet, dem Morlet-Wavelet und dem Mexican-Hat-Wavelet.

Hat man die gewünschten Werte und das Wavelet angegeben, kann mit Hilfe des Button „Auswertung“ die Transformation gestartet werden. Die Wavelettransformierte wird nach der Berechnung in der Plotebene auf der linken Seite angezeigt. Mit Hilfe der Maus kann die 3D-Graphik gedreht werden. Hierzu muss die Maus in die gewünschte Drehrichtung bewegt werden, während die linke Maustaste gedrückt wird.

Im Gegensatz zum Maplet in Abbildung 5.3 können bei dem Maplet für die schnelle Wavelettransformation mit Hilfe der Multiskalenanalyse (siehe Abbildung 5.4) die Skalierungs- und Verschiebungsparameter nicht explizit vorgegeben werden. Diese durchlaufen, wie in Kapitel 2.5.4 und 4.4 beschrieben, die Werte $h2^m$, $m = 1, \dots, M$ bzw. $h2^m k$, $m = 1, \dots, M$. Der Benutzer steuert über die Angabe von *level*

(= M) nur die Anzahl der Skalierungsparameter. Zu jedem Skalierungswert werden die Wavelettransformierten in den sechs Plot-ebenen auf der linken Seite des Maplets angegeben. Reichen diese sechs Plotflächen nicht aus, werden die ersten Plots ersetzt. Der Benutzer muss die Ersetzung durch den Button „weiter“ einleiten. Zum Schluss wird in der rechten oberen Plotfläche die geglättete Schwingung und in der rechten unteren Plotfläche ein 3D-Plot wie bei der diskreten Wavelettransformation, in der die Werte für alle berechneten Skalierungs- und Verschiebungswerte berücksichtigt werden, angezeigt.

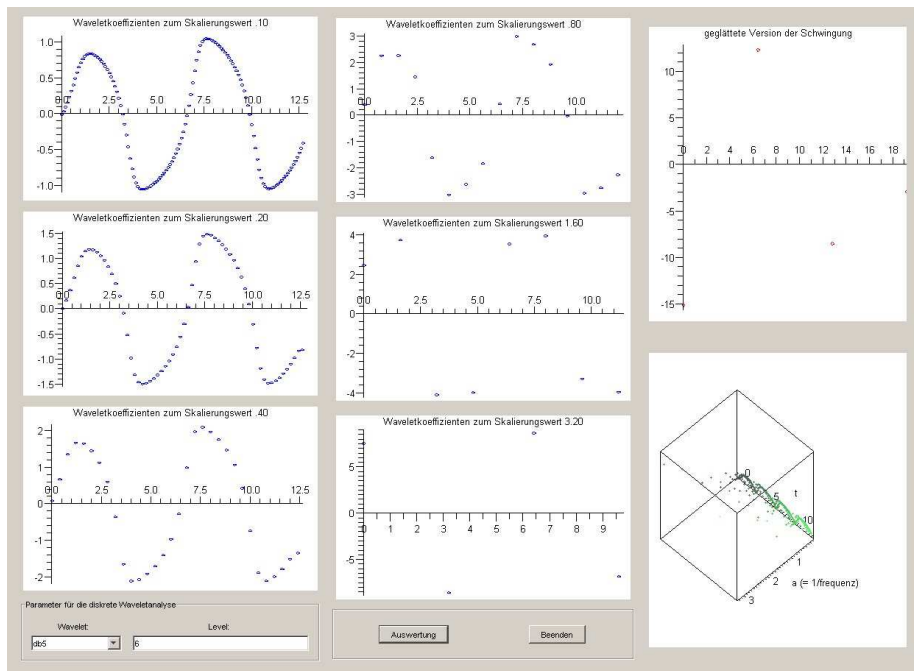


Abbildung 5.4: Maplet: schnelle Waveletanalyse

Durch den Button „Steife DGL“ kann man herausfinden, ob die Differentialgleichung steif ist. Bei steifen Differentialgleichungen verändern sich die einzelnen Komponenten auf sehr unterschiedlichen Zeitskalen. Die Differenz von Real- und Imaginärteil der Eigenwerte der Jacobimatrix von f ist in diesem Fall betragsmäßig sehr groß. Die aufgeführten expliziten Runge-Kutta-Verfahren sind zur Lösung steifer Differentialgleichungen nicht geeignet. Bei zu hoher Schrittweite sind die Ergebnisse extrem ungenau. Bei der Schrittweitensteuerung werden viel zu viele Schritte durchgeführt.

Mit Hilfe des Buttons „Leistungsspektrum anpassen“ wird die Frequenzauflösung angepasst. Die kleinste Frequenz ungleich null, die untersucht wird, ist wie oben beschrieben umgekehrt proportional zur Integrationszeit. Falls der Zeitbereich der Lösung für die gewünschte Frequenz zu klein ist, wird die Lösung weiter berechnet. Der Benutzer kann entscheiden, ob die Lösung durch weitere Anwendung der Runge-Kutta-Verfahren bestimmt werden soll oder durch Spiegelung der Daten. Bei der Spiegelung wird der letzte Hoch- bzw. Tiefpunkt im Ort-Zeit-Verlauf bestimmt und an dieser Stelle der Graph gespiegelt. Falls nach der Spiegelung der Zeitbereich immer noch nicht ausreicht, wird noch einmal gespiegelt.

Interessiert man sich bei den Lösungen nur für einen bestimmten Zeitbereich, können die Graphen mit Hilfe des Buttons „Darstellungszeitraum anpassen“ verändert werden. Es erscheint ein Untermaplet, in dem man die gewünschte Anfangs- und Endzeit des Darstellungszeitraums angeben kann. Es werden anschließend der Ort-Zeit-Verlauf, das Phasendiagramm und das Leistungsspektrum für den angegebenen Bereich dargestellt. Für die anderen Darstellungsmethoden ist das Auswählen des Zeitbereichs nicht möglich.

Weiterhin ist es möglich, die erzeugten Graphiken mit Hilfe des Button „Graphen speichern“ zu speichern. Für die in diesem Moment angezeigten Plots wird automatisch ein Dateiname, der sich aus der Art des Plots (Zeit, Phase, Frequenz), dem aktuellen Datum und der Uhrzeit zusammensetzt, im aktuellen Verzeichnis als JPG-Datei gespeichert. Wurde der Ort-Zeit-Verlauf am Montag, dem 01.02.2006 um 13:14 Uhr und 15 Sekunden gespeichert, wird der Dateiname `Zeit__Mo_Feb_01__13-14-15__2006` verwendet. Möchte man nicht alle Plots speichern oder den Dateinamen selber angeben, kann dies über die Menüleiste \rightarrow File \rightarrow SaveAs erfolgen. Es erscheint ein Untermaplet, in dem der Dateiname angegeben werden muss. Anschließend kann der Benutzer wählen, welcher Plot gespeichert werden soll.

5.2 Verwendung im Worksheet

Es ist möglich, die einzelnen Funktionen nicht nur über das Hauptmaplet „Start_Schwingungen“, sondern auch aus einem Worksheet heraus aufzurufen. Dadurch können die Funktionen auch in anderen Programmen verwendet werden.

Bindet man das Paket „Schwingungen“ ein, stehen einem die Prozeduren „Schwingungen_Maplet“ und „Standard_DGL“ zur Verfügung. Die Prozedur „Schwingungen_Maplet“ startet das Haupt-

maplet. Ruft man die Prozedur ohne Parameter auf, werden die Differentialgleichungen, die mit Hilfe der Prozedur „Standard_DGL“ erzeugt werden, verwendet. Es ist jedoch auch möglich, beim Aufruf eigene Differentialgleichungen anzugeben. Die Differentialgleichungen müssen einem System erster Ordnung der Dimension 2 entsprechen. Entsprechend Satz 1 sollten die Differentialgleichungen folgende Form besitzen:

$$\dot{x} = f(t, x) = (x_2, h(t, x_1, x_2))^T$$

Die Differentialgleichungen müssen als Matrixstruktur übergeben werden. Jede Zeile entspricht dabei einer Differentialgleichung. In der ersten Spalte steht jeweils $x[2]$ und in der zweiten Spalte

$$h(t, x[1], x[2]).$$

Auch die Runge-Kutta-Verfahren können direkt aus dem Worksheet gestartet werden. Hierzu muss das Paket „Runge_Kutta“ eingebunden werden. Anschließend sind die Prozeduren „RK_feste_Schrittweite“ und „RK_variable_Schrittweite“ bekannt und können im Worksheet verwendet werden. „RK_feste_Schrittweite“ löst wie im Kapitel 3 beschrieben die Differentialgleichung mit einer konstanten Schrittweite. Der Prozedur müssen die Parameter aus der Tabelle 5.1 in der angegebenen Reihenfolge übergeben werden.

| Parameter | Bedeutung | Typ |
|------------------|---|---------|
| <i>funktion</i> | zu lösende Differentialgleichung | proc |
| <i>t0</i> | Anfangszeit | numeric |
| <i>te</i> | Endzeit | numeric |
| <i>x0</i> | Anfangsbedingungen | list |
| <i>abc</i> | Butcher-Schema | Matrix |
| <i>h</i> | Schrittweite | numeric |
| <i>xv_return</i> | Ausgabeparameter: berechnete Ortspunkte und deren Ableitung | Matrix |
| <i>tv_return</i> | Ausgabeparameter: berechnete Zeitpunkte | Matrix |

Tabelle 5.1: Parameter der Funktion „RK_feste_Schrittweite“

Der Aufruf der Prozedur „RK_variable_Schrittweite“ ähnelt dem Aufruf der nicht schrittweitengesteuerten Prozedur sehr. Anstatt dem Butcher-Schema *abc* muss nun das eingebettete Butcher-Schema *abbc* angegeben werden. Der Parameter *h*, der die Schrittweite angibt, wird nicht mehr benötigt. Der Aufruf sieht dementsprechend

insgesamt folgendermaßen aus:

$$RK_variable_Schrittweite(funktion, t0, te, x0, abbc, \\ xv_return, tv_return)$$

Man erhält bei beiden Funktionen die Lösung der Differentialgleichung in den Matrizen *xv_return* und *tv_return*. Diese kann durch die Funktionen aus dem Paket „graphen“ auf verschiedene Weise graphisch dargestellt werden. Es stehen dem Benutzer die Plotfunktionen „zeitplot“, „phasenplot“, „leistungsspektrum“, „x_punkt_von_t“, „x_2_punkt_von_t“ und „x_2_punkt_von_x_punkt“ zur Verfügung. Die Tabelle 5.2 gibt die Aufgaben und benötigten Parameter der jeweiligen Funktionen wieder.

| Funktion | Parameter | Aufgabe |
|------------------------------|---|--|
| <i>zeitplot</i> | x, t, pl | Plottet den Ort über die Zeit ($x[1](t[1])$) |
| <i>phasenplot</i> | x, pl | Plottet den Ort über die erste Ableitung des Ortes ($x[1](x[2])$) |
| <i>leistungsspektrum</i> | <i>x_sampled</i> , <i>t_sampled</i> , pl | Ermittelt mit Hilfe der Fouriertransformation das Leistungsspektrum von x und stellt dieses graphisch dar. |
| <i>x_punkt_von_t</i> | x, t, pl | Plottet die erste Ableitung des Ortes über die Zeit ($x[2](t[1])$) |
| <i>x_2_punkt_von_t</i> | x, t, pl, funktion | Plottet die zweite Ableitung des Ortes über die Zeit |
| <i>x_2_punkt_von_x_punkt</i> | x, t, pl, funktion | Plottet die zweite Ableitung des Ortes über die erste Ableitung des Ortes |

Tabelle 5.2: Aufgaben der Plotfunktionen

Die Parameter, die den Funktionen übergeben werden, entsprechen den in Tabelle 5.3 angegebenen Typen.

Für das Erzeugen des Leistungsspektrum ist es nötig, dass die Näherungslösung der Differentialgleichung zu äquidistanten Zeiten vorliegt. Löst man die Differentialgleichungen mit den schrittweitengesteuerten Runge-Kutta-Verfahren, ist dies nicht der Fall. Um das

| Parameter | Bedeutung | Typ |
|--------------|---|--------|
| x | Näherungslösung der Differentialgleichung (berechnete Orte und deren Ableitung) | Matrix |
| t | Lösungszeiten | Matrix |
| $x_sampled$ | Lösungswerte zu äquidistanten Zeiten | Vector |
| $t_sampled$ | äquidistante Lösungszeiten | Vector |
| pl | Ausgabeparameter: erzeugte Graphik | plot |
| $funktion$ | zu lösende Differentialgleichung | proc |

Tabelle 5.3: Parameter der Plotfunktionen

Leistungsspektrum dennoch erzeugen zu können, muss zunächst die Funktion

$$resample(x, t, t0, te)$$

aus dem Paket „resampling“ ausgeführt werden. Diese ermittelt aus der ermittelten Näherungslösung eine Lösung zu äquidistanten Zeiten.

Dasselbe gilt für die Waveletanalyse. Diese kann ebenfalls aus dem Worksheet heraus gestartet werden. Für diesen Zweck benötigt man das Paket „wavelet“. Es enthält die Funktionen „dwt“, „fwt“ und „wavelet_steuerung“. Die Funktion „dwt“ realisiert die diskrete Wavelettransformation, die in Kapitel 2.5.3 beschrieben ist. Für die angegebenen Skalierungs- und Verschiebungsparameter werden die Wavelettransformierten berechnet. Als Rückgabewert erhält man einen 3D-Plot, der diese enthält. Bei dem Funktionsaufruf

$$dwt(x_sampled, t_sampled, te, a_start, a_step, a_stopp, \\ b_start, b_step, b_stopp, wavelet)$$

stehen die Parameter für die in Tabelle 5.4 angegebenen Werte. Durch die Funktion „fwt“ werden die Wavelettransformierten mit Hilfe der schnellen Wavelettransformation bestimmt. Der Funktionsaufruf

$$fwt(x_sampled, t_sampled, level, t0, wavelet, \\ L, L_dim_a, L_dim_b, c_return)$$

beinhaltet die in Tabelle 5.5 angegebenen Parameter. Als Rückgabewert erhält man wiederum eine 3D-Graphik, die die Wavelettransformierten beinhaltet.

Möchte man die Waveletanalyse über Maplets gesteuert ausführen, so kann man die Funktion „wavelet_steuerung“ verwenden. Der Be-

| Parameter | Bedeutung | Typ |
|--------------|---|---------|
| $x_sampled$ | Lösungswerte zu äquidistanten Zeiten | Vector |
| $t_sampled$ | äquidistante Lösungszeiten | Vector |
| te | Endzeit | numeric |
| a_start | Anfangsskalierungswert (ungleich 0) | numeric |
| a_step | Skalierungsschrittweite | numeric |
| a_stopp | Endskalierungswert | numeric |
| b_start | Anfangsverschiebungswert | numeric |
| b_step | Verschiebungsschrittweite | numeric |
| b_stopp | Endverschiebungswert | numeric |
| $wavelet$ | Wavelet (haar,mexican__hat oder morlet) | string |

Tabelle 5.4: Parameter der Funktion „dwt“

nutzer kann in einem Maplet angeben, ob er die diskrete Wavelettransformation oder die diskrete Wavelettransformation mit Hilfe der Multiskalenanalyse verwenden möchte. In einem entsprechenden weiteren Maplet können anschließend die gewünschten Einstellungen vorgenommen und die Berechnung durchgeführt werden. Eine genauere Beschreibung der Benutzung der Maplets ist in Kapitel 5.1 gegeben.

Die Parameter $x_sampled$, $t_sampled$, $t0$ und te , die der Funktion „wavelet_steuerung“ übergeben werden müssen, entsprechen den in Tabelle 5.4 und 5.5 beschriebenen Bedeutungen und Typen.

| Parameter | Bedeutung | Typ |
|------------------|---|------------|
| <i>x_sampled</i> | Lösungswerte zu äquidistanten Zeiten | Vector |
| <i>t_sampled</i> | äquidistante Lösungszeiten | Vector |
| <i>level</i> | Anzahl der gewünschten Skalierungsparameter | numeric |
| <i>t0</i> | Anfangszeit | numeric |
| <i>wavelet</i> | Daubechies-Wavelet (db2, db4, db6, db8 oder db10) | string |
| <i>L</i> | Ausgabeparameter: berechnete Wavelettransformierte | Matrix |
| <i>L_dim_a</i> | Ausgabeparameter: Anzahl der Zeilen der Matrix L (entspricht der tatsächlichen Anzahl der Skalierungsparameter) | numeric |
| <i>L_dim_b</i> | Ausgabeparameter: Anzahl der Wavelettransformierten für jeden Skalierungswert | Vector |
| <i>c_return</i> | Ausgabeparameter: geglättete Version des Signals | plot |

Tabelle 5.5: Parameter der Funktion „fwt“

Kapitel 6

Vergleich der Runge-Kutta-Verfahren für ausgewählte nicht lineare Differentialgleichungen

In diesem Kapitel werden die verschiedenen Runge-Kutta-Verfahren, die in dieser Arbeit verwendet wurden (siehe Tabelle 6.1 und Tabelle 6.2), miteinander verglichen. Dabei wird insbesondere auf die Stabilität, die Rechenzeit und die Genauigkeit eingegangen.

| | | | | | |
|-------------------------------|-----|------|-----|-----|-----|
| Euler | 0 | 0 | | | |
| | 0 | 1 | | | |
| Collatz | 0 | 0 | 0 | | |
| | 1/2 | 1/2 | 0 | | |
| | 0 | 0 | 1 | | |
| klassischer Runge-Kutta (KRK) | 0 | 0 | 0 | 0 | 0 |
| | 1/2 | 1/2 | 0 | 0 | 0 |
| | 1/2 | 0 | 1/2 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 |
| | 0 | 1/6 | 1/3 | 1/3 | 1/6 |
| Runge-Kutta 3/8 (RK38) | 0 | 0 | 0 | 0 | 0 |
| | 1/3 | 1/3 | 0 | 0 | 0 |
| | 2/3 | -1/3 | 1 | 0 | 0 |
| | 1 | 1 | -1 | 1 | 0 |
| | 0 | 1/8 | 3/8 | 3/8 | 1/8 |

Tabelle 6.1: Butcher-Schemata der verwendeten Runge-Kutta-Verfahren mit konstanter Schrittweite

| | | | | | | | | |
|--|----------------|----------------------|-----------------------|----------------------|--------------------|-------------------------|--------------------|----------------|
| Runge-Kutta- Fehlerberg 2(3) (RKF23) | 0 | 0 | 0 | 0 | | | | |
| | 1 | 1 | 0 | 0 | | | | |
| | 1/2 | 1/4 | 1/4 | 0 | | | | |
| | 2 | 1/2 | 1/2 | 0 | | | | |
| | 3 | 1/6 | 1/6 | 2/3 | | | | |
| Runge- Kutta 4(3) (RK43) | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | 1/2 | 1/2 | 0 | 0 | 0 | 0 | | |
| | 1/2 | 0 | 1/2 | 0 | 0 | 0 | | |
| | 1 | 0 | 0 | 1 | 0 | 0 | | |
| | 1 | 1/6 | 1/3 | 1/3 | 1/6 | 0 | | |
| | 4 | 1/6 | 1/3 | 1/3 | 1/6 | 0 | | |
| | 3 | 1/6 | 1/3 | 1/3 | 0 | 1/6 | | |
| Dormand- Prince 5(4) (DP54) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $\frac{1}{5}$ | $\frac{1}{5}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | $\frac{3}{10}$ | $\frac{3}{40}$ | $\frac{9}{40}$ | 0 | 0 | 0 | 0 | 0 |
| | $\frac{4}{5}$ | $\frac{44}{45}$ | $\frac{-25360}{2187}$ | $\frac{64448}{6561}$ | $\frac{-212}{729}$ | 0 | 0 | 0 |
| | 1 | $\frac{9017}{3168}$ | $\frac{-355}{33}$ | $\frac{46732}{5247}$ | $\frac{49}{176}$ | $\frac{-5103}{18656}$ | 0 | 0 |
| | 1 | $\frac{35}{384}$ | 0 | $\frac{500}{1113}$ | $\frac{125}{192}$ | $\frac{-2187}{6784}$ | $\frac{11}{84}$ | 0 |
| | 5 | $\frac{35}{384}$ | 0 | $\frac{500}{1113}$ | $\frac{125}{192}$ | $\frac{-2187}{6784}$ | $\frac{11}{84}$ | 0 |
| | 4 | $\frac{5179}{57600}$ | 0 | $\frac{7571}{16695}$ | $\frac{393}{640}$ | $\frac{-92097}{339200}$ | $\frac{187}{2100}$ | $\frac{1}{40}$ |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Tabelle 6.2: Butcher-Schemata der verwendeten Runge-Kutta-Verfahren mit variabler Schrittweite

6.1 Stabilität

Bei den Runge-Kutta-Verfahren unterscheidet man zwischen zwei verschiedenen Stabilitätsbegriffen: 0-stabil und A-stabil. Ein Verfahren ist 0-stabil, wenn geringe Abweichungen in den Eingangsdaten nicht zu einem explodierenden Fehler und damit zu völlig falschen Ergebnissen führen. Man untersucht dazu die Fehler für Schrittweiten, die gegen Null laufen. Vergleiche hierzu auch die Definition 16 auf Seite 36.

Bei der Untersuchung auf A-Stabilität betrachtet man hingegen nicht den Fehler in jedem Schritt, sondern das Verhalten der Lösung für $t \rightarrow \infty$. Hierzu verwendet man die Testgleichung

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

mit der exakten Lösung

$$x(t) = e^{\lambda t}.$$

Für komplexe λ mit $\operatorname{Re}(\lambda) < 0$ geht die exakte Lösung für $t \rightarrow \infty$ gegen Null. Die approximierten Werte x_k sollten das gleiche Verhalten zeigen und deshalb die Eigenschaft

$$\lim_{k \rightarrow \infty} x_k = 0, \quad \text{falls } \operatorname{Re}(\lambda) < 0$$

besitzen. Gilt dies bei festen Schrittweiten für alle λ , dann nennt man das Verfahren A-stabil. Die exakte Beschreibung geben die folgenden zwei Definitionen aus [3] wieder.

Definition 18 Die Menge $A = \{\lambda h \in \mathbb{C} \mid \lim_{k \rightarrow \infty} x_k = 0, x_k \text{ approximierte Werte des Verfahrens } \varphi \text{ mit fester Schrittweite } h \text{ für die Differentialgleichung } \dot{x} = \lambda x, x(0) = 1\}$ heißt Stabilitätsgebiet von φ .

Definition 19 Ist $A \supset \mathbb{C}^- = \{z \mid z \in \mathbb{C}, \operatorname{Re}(z) < 0\}$ dann heißt φ A-stabil.

Abbildung 6.1 zeigt die Grenzen des Stabilitätsgebiet der Runge-Kutta-Verfahren mit fester Schrittweite. Das Stabilitätsgebiet befindet sich jeweils im inneren der Grenzen. Je höher die Verfahrensordnung ist, desto größer sind die Stabilitätsgebiete, jedoch füllt keines die komplette linke Halbebene aus. Die Verfahren sind somit nur bedingt A-stabil.

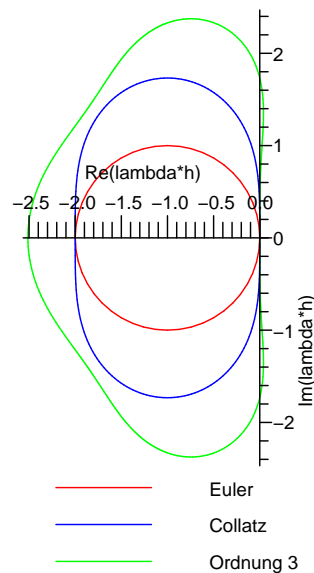


Abbildung 6.1: Stabilitätsgebiete der Runge-Kutta-Verfahren

6.2 Rechenzeit

Im Folgenden soll die benötigte Rechenzeit der Verfahren mit einander verglichen werden. Hierzu wurde zuerst die Rechenzeit (CPU-Zeit) der Runge-Kutta-Verfahren mit fester Schrittweite untersucht. Für die Untersuchung wurde die Differentialgleichung

$$\ddot{x} + \dot{x} + x - \sin(x) = \cos(t)$$

für die Anfangswerte $x(0) = 0, \dot{x}(0) = 1$ für den Zeitbereich von 0 bis 10 Sekunden betrachtet. Um Abweichungen der Rechenzeit durch andere Systemprozesse zu minimieren, wurde die Differentialgleichung für jede Schrittweite 10 Mal gelöst und aus diesen Werten jeweils das arithmetische Mittel gebildet. Für die Rechenzeiten der Runge-Kutta-Verfahren mit fester Schrittweite ergaben sich dabei die in Abbildung 6.2 und Tabelle 6.3 dargestellten Werte.

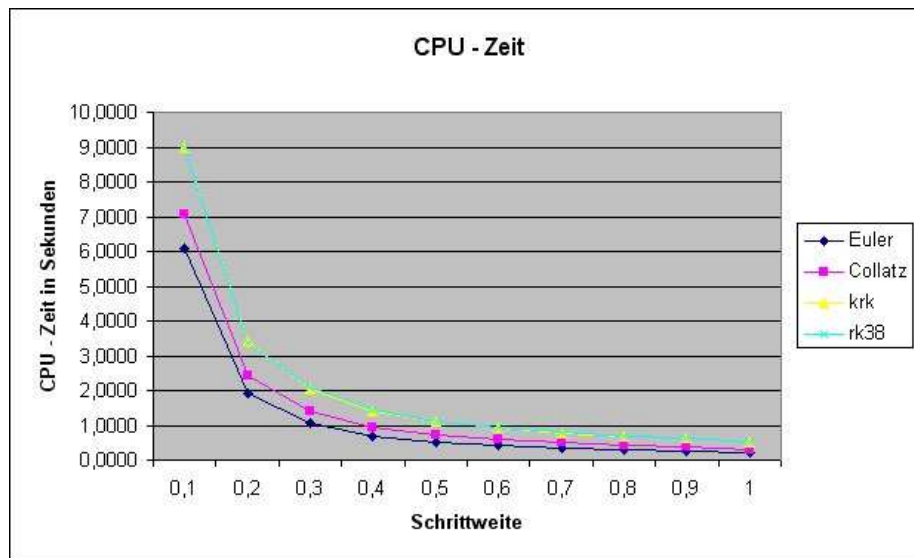


Abbildung 6.2: Rechenzeit der Runge-Kutta-Verfahren mit konstanter Schrittweite

Man kann erkennen, dass die Rechenzeit mit abnehmender Schrittweite zunimmt. Je kleiner die Schrittweite ist, desto mehr Schritte müssen durchgeführt werden, um die Endzeit zu erreichen. Es ist leicht einzusehen, dass sich dadurch der Rechenaufwand erhöht. Weiterhin kann man erkennen, dass die Stufenzahl des Verfahrens die Rechenzeit beeinflusst. Mit steigender Stufenzahl nehmen die f -Auswertungen in jedem Schritt zu. Hat ein Verfahren die Stufenzahl s , werden pro Schritt s Funktionsauswertungen durchgeführt. Dadurch steigt der Rechenaufwand in jedem Schritt.

| | Euler | | Collatz | | klassischer Runge- Kutta | | Runge- Kutta 3/8 | |
|-----|--------------|-----|----------------|-----|---|-----|---------------------------------|-----|
| h | RZ in sec | f | RZ in sec | f | RZ in sec | f | RZ in sec | f |
| 0,1 | 6,0855 | 100 | 7,0502 | 200 | 8,9769 | 300 | 8,9909 | 300 |
| 0,2 | 1,9228 | 50 | 2,4085 | 100 | 3,3839 | 150 | 3,3769 | 150 |
| 0,3 | 1,0746 | 34 | 1,4160 | 68 | 2,0549 | 102 | 2,0900 | 102 |
| 0,4 | 0,7020 | 25 | 0,9335 | 50 | 1,4210 | 75 | 1,4271 | 75 |
| 0,5 | 0,5147 | 20 | 0,7060 | 40 | 1,1045 | 60 | 1,1076 | 60 |
| 0,6 | 0,4177 | 17 | 0,5828 | 34 | 0,9183 | 51 | 0,9163 | 51 |
| 0,7 | 0,3525 | 15 | 0,4968 | 30 | 0,7872 | 45 | 0,7911 | 45 |
| 0,8 | 0,2814 | 13 | 0,4236 | 26 | 0,6820 | 39 | 0,6740 | 39 |
| 0,9 | 0,2693 | 12 | 0,3845 | 24 | 0,6198 | 36 | 0,6279 | 36 |
| 1 | 0,2083 | 10 | 0,3075 | 20 | 0,5067 | 30 | 0,5178 | 30 |

Tabelle 6.3: Anzahl der Funktionsauswertungen der Runge-Kutta-Verfahren mit konstanter Schrittweite.

h = Schrittweite, RZ = Rechenzeit, f = Anzahl der Funktionsauswertungen

Verwendet man eine Schrittweite von 0,1 müssen für den zu untersuchenden Zeitbereich von 0 bis 10 100 Schritte durchgeführt werden. Bei dem Euler-Verfahren (Stufenzahl 1) werden demzufolge auch 100 Funktionsauswertungen benötigt. Beim Collatz-Verfahren sind es bedingt durch die Stufenzahl zwei 200 Funktionsauswertungen und beim klassischen Runge-Kutta und Runge-Kutta 3/8 300 Funktionsauswertungen. Für 300 Funktionsauswertungen wurde mit einer Rechenzeit von fast 9 Sekunden in etwa 3 Sekunden mehr als im Vergleich zu 100 Funktionsauswertungen benötigt.

Würde die Rechenzeit nur von der Anzahl der Funktionsauswertungen abhängen, müsste die Rechenzeit linear mit zunehmender Schrittweite abnehmen. Dies ist aber nicht der Fall, da die Rechenzeit zusätzlich vom Speicheraufwand abhängt. In jedem Schritt wird die ermittelte Lösung und die Zeit in Matrizen gespeichert. Je mehr Schritte durchgeführt werden, desto mehr Werte müssen gespeichert werden. Vergleicht man zum Beispiel das Euler-Verfahren mit Schrittweite 0,2 mit dem Collatz-Verfahren mit Schrittweite 0,4, müssen bei beiden Verfahren 50 Funktionsauswertungen durchgeführt werden, was in etwa gleiche Rechenzeiten vermuten lässt. Beim Collatz-Verfahren müssen jedoch nur 25 ermittelte Lösungen und Zeiten gespeichert werden im Gegensatz zu 50 beim Euler-Verfahren. Aus diesem Grund ist die benötigte Rechenzeit beim Collatz-Verfahren mit 0,9335 Sekunden geringer als beim Euler-

Verfahren, das mit 1,9228 Sekunden fast eine Sekunde länger benötigt.

Die schrittweitengesteuerten Verfahren wurden mit Hilfe der selben Differentialgleichung bei einer Fehlertoleranz von $\varepsilon = 10^{-6}$ (vgl. Gleichung (3.2)) untersucht. Für jedes Verfahren wurde die benötigte Rechenzeit, die Anzahl der Funktionsauswertungen und benötigten Schritte, die minimale, maximale und durchschnittliche Schrittweite sowie die Anzahl der verworfenen Lösungen bestimmt. Die Ergebnisse wurden in Tabelle 6.4 zusammengefasst. Man kann

| Ver- fahren | Rechen- zeit | f | S | min | max | average | v |
|----------------|-----------------|------|-----|----------|-------------|-------------|---|
| RKF23 | 218,0327 | 1938 | 645 | 0,000001 | 0,042879588 | 0,015489039 | 1 |
| RK43 | 16,7161 | 670 | 131 | 0,000001 | 0,115362994 | 0,075278134 | 3 |
| DP54 | 5,3347 | 357 | 45 | 0,000001 | 0,327741512 | 0,213077685 | 6 |

Tabelle 6.4: Rechenzeit der Runge-Kutta-Verfahren mit variabler Schrittweite. RZ = Rechenzeit, f = Anzahl der Funktionsauswertungen, S = Anzahl der Schritte, min = minimale Schrittweite, max = maximale Schrittweite, average = durchschnittliche Schrittweite, v = Anzahl verworfener Lösungen

wiederum erkennen, dass die Rechenzeit entscheidend von der Anzahl der Funktionsauswertungen und dem Speicheraufwand beeinflusst wird. Löst man die Differentialgleichung mit Hilfe des RKF23-Verfahren benötigt man 1938 Funktionsauswertungen und es müssen 645 ermittelte Lösungen und Zeiten gespeichert werden. Beim RK43-Verfahren reduziert sich die Anzahl der Funktionsauswertungen auf ein Drittel und der Speicheraufwand auf ein Fünftel. Dadurch erzielt man eine deutliche Verringerung der Rechenzeit von etwa 200 Sekunden. Beim DP54-Verfahren ist der Rechenaufwand noch geringer, da wiederum weniger Funktionsauswertungen benötigt und weniger Ergebnisse gespeichert werden müssen.

In vielen Fällen kann man beobachten, dass beim DP54-Verfahren weniger Schritte als beim RK43 und RKF23-Verfahren benötigt werden. Aus diesem Grund ist das DP54-Verfahren in der Regel schneller als die andern beiden Verfahren, obwohl in jedem Schritt mindestens sieben Funktionsauswertungen im Gegensatz zu fünf beim RK43 und drei beim RKF23-Verfahren durchgeführt werden müssen.

6.3 Genauigkeit

Die Genauigkeit der Lösung hängt von der Ordnung des Verfahrens und von der gewählten Schrittweite ab. Je höher die Ordnung bzw. je kleiner die Schrittweite, desto genauer approximieren die Runge-Kutta-Verfahren die exakte Lösung. Dies liegt daran, dass bei einer Schrittweite $h \rightarrow 0$ und Verfahrensordnung p der lokale Fehler in jedem Schritt nach Definition 15 die Ordnung $O(h^{p+1})$ hat.

Zur Untersuchung der Genauigkeit wurde zunächst dieselbe Differentialgleichung wie im Kapitel 6.2 verwendet. Da diese Gleichung nicht analytisch lösbar ist, kann für diese Differentialgleichung keine exakte Lösung angegeben werden. Die Lösungen wurden deshalb nur untereinander verglichen.

In der Abbildung 6.3 wurden die Verfahren mit konstanter Schrittweite für verschiedene Schrittweiten untersucht. In den ersten vier Graphiken sind die Lösungskurven der Differentialgleichung für die Schrittweiten 0,1, 0,2 bis 1,0 für die verschiedenen Verfahren dargestellt. Wegen der Übersichtlichkeit wurde jeweils nur eine Legende für die Schrittweite 0,1 und 1,0 hinzugefügt. In den letzten zwei Abbildungen wurden die Lösungen der Verfahren zur Schrittweite 0,1 bzw. 1,0 gezeichnet.

Man kann erkennen, dass die Ergebnisse beim Euler-Verfahren für verschiedene Schrittweiten sehr unterschiedlich und damit sehr ungenau sind. Dies liegt an der niedrigen Ordnung des Verfahrens. Das Verfahren hat nur die Ordnung eins. Der Fehler in jeden Schritt (Ordnung h^2) ist damit sehr hoch. Wird eine große Schrittweite von 1,0 gewählt ist das Ergebnis fast unbrauchbar. Selbst bei einer kleinen Schrittweite von 0,1, bei der die anderen drei Verfahren nahezu identische Lösungen liefern, weicht die Lösung des Euler-Verfahrens noch ab.

Das Collatz-Verfahren hat Ordnung 2 und liefert konstantere Ergebnisse als das Euler-Verfahren. Die Hoch- und Tiefpunkte der Lösungskurve werden jedoch noch sehr unterschiedlich approximiert. Bei einer Schrittweite von 0,1 wird nahezu dieselbe Lösungskurve erreicht, wie bei den beiden Verfahren mit Ordnung drei. Bei der Schrittweite von 1,0 erhält man jedoch noch ein abweichendes Ergebnis.

Das klassische Runge-Kutta-Verfahren und das Runge-Kutta3/8-Verfahren führen bei jeder Schrittweite zu fast identischen Lösungen. Man kann somit vermuten, dass die gesuchte Lösungskurve durch diese Verfahren gut approximiert wird. Beim Euler- und beim Collatz-Verfahren führen nur kleinere Schrittweiten zu einem genaueren Ergebnis.

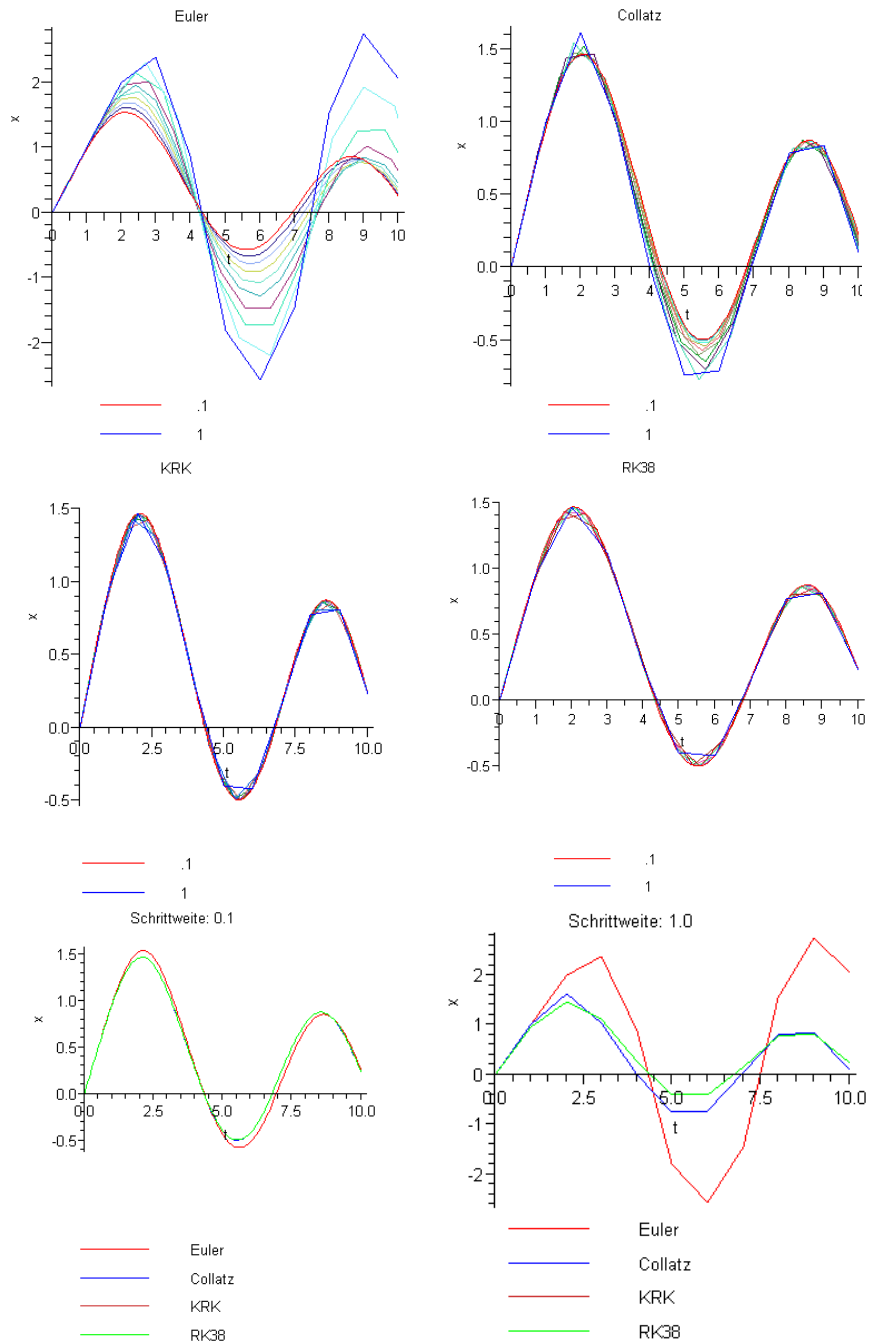


Abbildung 6.3: Vergleich der Genauigkeit der Lösungen für verschiedene Schrittweiten

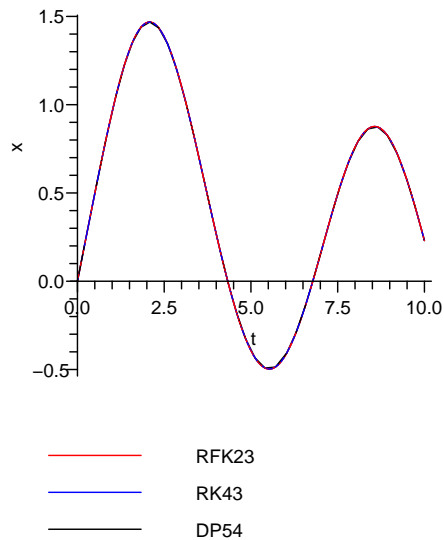


Abbildung 6.4: Vergleich der Genauigkeit der schrittweitengesteuerten Verfahren

Die Lösungskurven, die mit Hilfe der schrittweitengesteuerten Verfahren ermittelt wurden, sind nahezu identisch (Abbildung 6.4). Sie approximieren die gesuchte Lösung deshalb wahrscheinlich sehr gut. Bei der Berechnung der Lösung wurde jeweils ein lokaler Fehler ε von 10^{-6} zugelassen.

Um die Genauigkeit der Verfahren noch besser untersuchen zu können, wurden sie auch mit Hilfe einer linearen Differentialgleichung untersucht. Bei linearen Differentialgleichungen ist es möglich, die Lösung analytisch zu bestimmen. Man kennt dadurch die exakte Lösung. Zur Untersuchung wurde die Differentialgleichung

$$\ddot{x}(t) + \dot{x}(t) + x(t) = \cos(t)$$

mit der exakten Lösung $x(t) = \sin(t)$ verwendet. Dabei ergab sich das in Abbildung 6.5 und Tabelle 6.5 dargestellte Fehlerverhalten. Man kann wie schon bei dem vorherigen Beispiel erkennen, dass die Genauigkeit von der Schrittweite und der Ordnung des Verfahrens abhängig ist. Beim Euler-Verfahren, das nur eine Ordnung von eins hat, beläuft sich der lokale Fehler auch für sehr kleine Schrittweiten h immer noch auf $O(h^2)$. Löst man die Differentialgleichung mit dem Euler-Verfahren mit einer Schrittweite von 0,1, erhält man ein Ergebnis, bei dem nur die Vorkommastelle exakt approximiert wird.

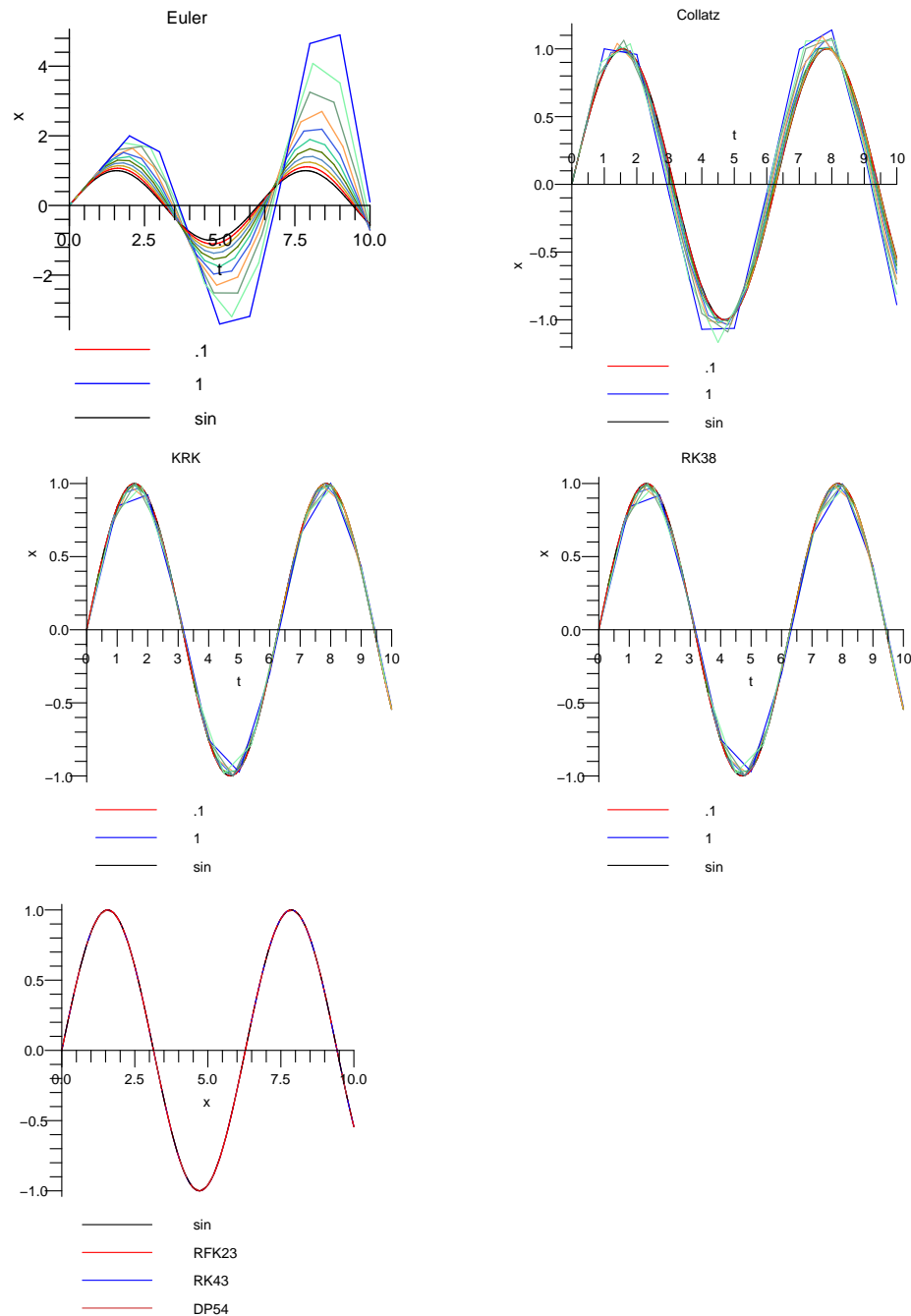


Abbildung 6.5: Vergleich der Genauigkeit der Lösungen für verschiedene Schrittweiten (lineare Differentialgleichung)

Der durchschnittliche Fehler beläuft sich in diesem Fall auf 0,06318. Man kann außerdem erkennen, dass die approximierten Lösungen mit zunehmender Zeit immer ungenauer werden. Dies liegt daran, dass der Verstärkungsfehler durch die immer ungenaueren Werte anwächst. Das Euler-Verfahren liefert für dieses Beispiel insgesamt nur sehr ungenaue Näherungen. Möchte man es trotzdem verwenden, sollte man auf kleine Schrittweiten achten.

Das Collatz-Verfahren hat die Ordnung zwei und liefert deshalb schon etwas bessere Ergebnisse als das Euler-Verfahren. Mit einem maximalen Fehler von 0,003434 und einen durchschnittlichen Fehler von 0,001812 bei einer Schrittweite von 0,1, wird die erste Nachkommastelle exakt und die zweite Nachkommastelle sehr genau approximiert. Eine Schrittweite von 0,6 reicht sogar aus, um in etwa denselben durchschnittlichen Fehler wie beim Euler-Verfahren mit Schrittweite 0,1 zu erreichen.

Noch deutlicher ist der Unterschied zwischen dem KRK-Verfahren und dem Euler-Verfahren. Da die Verfahrensordnung um zwei größer ist, gelangt man sogar mit einer Schrittweite von 1,0 beim KRK-Verfahren zu besseren Ergebnissen beim maximalen und beim durchschnittlichen Fehler als beim Euler-Verfahren mit Schrittweite 0,1. Ab einer Schrittweite von 0,4 werden die ersten zwei Nachkommastellen genau approximiert. Der maximale Fehler bei dieser Schrittweite beträgt nur 0,0005069. Verwendet man die Schrittweite 0,8 beläuft sich der maximale Fehler für dieses Beispiel auf 0,008640. Ab dieser Schrittweite erhält man somit ein Ergebnis, das bis zur ersten Dezimalstelle exakt ist. Bei einer Schrittweite von 0,1 erhält man sogar ein auf 4 Nachkommastellen genaues Ergebnis. Das KRK-Verfahren nähert die exakte Lösung insgesamt bei geeigneter Schrittweite sehr gut an.

Ähnliche Ergebnisse kann man beim RK38-Verfahren feststellen. Dies hat die selbe Ordnung wie das KRK-Verfahren, wodurch sich die Übereinstimmungen erklären lassen.

Die schrittweitengesteuerten Verfahren liefern ebenfalls sehr gute Ergebnisse. Der gewünschte maximale lokale Fehler ε wurde wiederum auf 10^{-6} festgelegt. Beim RK43 und DP54-Verfahren übertritt der absolute Fehler in jedem Schritt diese Schranke nicht. Der maximale Fehler liegt bei diesen Verfahren bei $4,679 \cdot 10^{-7}$ bzw. $3,777 \cdot 10^{-7}$. Das Ergebnis wurde somit auf fünf Dezimalstellen genau approximiert. Beim RKF23-Verfahren sind es, bedingt durch die geringere Ordnung der Verfahren zur Lösungsbestimmung und Fehlerschätzung, nur 3 Dezimalstellen.

Insgesamt kann man feststellen, dass die schrittweitengesteuerten

Verfahren zu guten Ergebnissen führen. Bei den Verfahren mit konstanter Schrittweite sollte man nicht zu große Schrittweiten und möglichst hohe Ordnungen verwenden.

| Verfahren | Schrittweite | minimaler Fehler | maximaler Fehler | durchschnittlicher Fehler |
|----------------|--------------|-----------------------|-----------------------|---------------------------|
| Euler | 0,1 | 0,0001666 | 0,1214 | 0,06318 |
| | 0,2 | 0,001331 | 0,2663 | 0,1354 |
| | 0,3 | 0,004480 | 0,4445 | 0,2151 |
| | 0,4 | 0,005678 | 0,6681 | 0,3171 |
| | 0,5 | 0,02057 | 0,9435 | 0,4402 |
| | 0,6 | 0,03536 | 1,335 | 0,5736 |
| | 0,7 | 0,05578 | 1,846 | 0,7325 |
| | 0,8 | 0,08264 | 2,378 | 0,9734 |
| | 0,9 | 0,06572 | 3,109 | 1,192 |
| | 1,0 | 0,1190 | 4,486 | 1,551 |
| Collatz | 0,1 | 0,00003383 | 0,003434 | 0,001812 |
| | 0,2 | 0,0004305 | 0,01394 | 0,007360 |
| | 0,3 | 0,001283 | 0,03149 | 0,01695 |
| | 0,4 | 0,006408 | 0,05585 | 0,03001 |
| | 0,5 | 0,007155 | 0,08819 | 0,04691 |
| | 0,6 | 0,008998 | 0,1245 | 0,06769 |
| | 0,7 | 0,005075 | 0,1715 | 0,09334 |
| | 0,8 | 0,05491 | 0,2089 | 0,1231 |
| | 0,9 | 0,03045 | 0,2805 | 0,1593 |
| | 1,0 | 0,04965 | 0,3448 | 0,1887 |
| KRK | 0,1 | $2,170 \cdot 10^{-8}$ | 0,000001896 | 0,000001014 |
| | 0,2 | $7,220 \cdot 10^{-7}$ | 0,00003077 | 0,00001643 |
| | 0,3 | $2,025 \cdot 10^{-7}$ | 0,0001585 | 0,00008214 |
| | 0,4 | 0,00002485 | 0,0005069 | 0,0002679 |
| | 0,5 | 0,00003656 | 0,001258 | 0,0006607 |
| | 0,6 | 0,0002018 | 0,002581 | 0,001395 |
| | 0,7 | 0,0004503 | 0,004824 | 0,002586 |
| | 0,8 | 0,00001076 | 0,008640 | 0,004364 |
| | 0,9 | 0,001102 | 0,01395 | 0,007245 |
| | 1,0 | 0,0002892 | 0,02182 | 0,01135 |
| R38K | 0,1 | $9,0 \cdot 10^{-9}$ | 0,000001806 | $9,657 \cdot 10^{-7}$ |
| | 0,2 | $2,450 \cdot 10^{-7}$ | 0,00002957 | 0,000001573 |
| | 0,3 | 0,000002503 | 0,0001531 | 0,00008067 |
| | 0,4 | 0,000009273 | 0,0004943 | 0,0002614 |
| | 0,5 | 0,00002223 | 0,0001215 | 0,0006509 |
| | 0,6 | 0,00005053 | 0,002608 | 0,001374 |
| | 0,7 | 0,00008722 | 0,004932 | 0,002599 |
| | 0,8 | 0,0001265 | 0,008456 | 0,004595 |
| | 0,9 | 0,0001486 | 0,01420 | 0,0075384 |
| | 1,0 | 0,00011546 | 0,02196 | 0,01168 |
| RKF23 | | 0 | 0,00006340 | 0,00003183 |
| RK43 | | 0 | $4,679 \cdot 10^{-7}$ | $2,294 \cdot 10^{-7}$ |
| DP54 | | 0 | $3,777 \cdot 10^{-7}$ | $1,746 \cdot 10^{-7}$ |

Tabelle 6.5: Fehlerverhalten

Kapitel 7

Zusammenfassung

In dieser Arbeit wurde eine Entwicklungsumgebung zur Analyse von Differentialgleichungen erstellt. Die Differentialgleichungen werden mit Hilfe von Runge-Kutta-Verfahren numerisch gelöst und anschließend graphisch dargestellt. Es ist möglich, verschiedene Eigenschaften der Lösung zu untersuchen. Die Lösung wird als Ort-Zeit-Verlauf, Phasendiagramm und als Leistungsspektrum dargestellt, so dass Eigenschaften wie die Schwingungsdauer, Amplitude, Gleichgewichtslagen und Frequenzen ersichtlich werden. Mit Hilfe der Waveletanalyse kann man außerdem, die enthaltenen Frequenzen genauer untersuchen.

Für den Verein Deutscher Ingenieure wird mit Hilfe dieser Oberfläche ein Bericht erstellt. In diesem Bericht werden die Eigenschaften von grundlegenden Schwingungstypen beschrieben, so dass der Leser in der Lage ist, anhand der Eigenschaften auf die zu Grunde liegende Differentialgleichung zu schließen.

Literaturverzeichnis

- [1] *Schwingung*, <http://de.wikipedia.org/wiki/Schwingung>
- [2] M. Reißel, *Numerik I*, Fachhochschule Aachen, Vorlesungsskript (2004)
- [3] M. Reißel, *Numerik II*, Fachhochschule Aachen, Vorlesungsskript (2005)
- [4] Uni Dortmund, *Funktionen, Stetigkeit und Differentialrechnung* http://www.mathematik.uni-dortmund.de/hm/hm1petii0203/-Folien_Kapitel_4_2.pdf
- [5] Louis/Maaß/Rieder, *Wavelets*, Teubner Studienbücher: Mathematik (1998)
- [6] H.-G. Stark, *Wavelets and Signal Processing*, Springer-Verlag Berlin Heidelberg (2005)
- [7] S. Wärow, *Schnelle Wavelet basierte Berechnung von langreichweitigen Wechselwirkungen in Vielteilchensystemen*, FZJ-ZAM-IB-2005-04, Forschungszentrum Jülich GmbH in der Helmholtz-Gemeinschaft (2005)
- [8] M. Johns *Anwendung von Wavelets für die biometrische Authentikation*, <http://polyboy.net/akademisches/diplomarbeit/html> (2003)
- [9] *Daubechies-Wavelets* <http://de.wikipedia.org/wiki/Daubechies-Wavelets>
- [10] K. Magnus/K. Popp, *Schwingungen*, Teubner Studienbücher: Mechanik (1997)
- [11] Maple 10 *Maple 10 Help*
- [12] Westermann, *Mathematik für Ingenieure mit Maple*, Springer-Verlag Berlin Heidelberg (2001)

- [13] H. J. Weaver, *Theory of Discrete and Continuous Fourier Analysis*, John Wiley & Sons (1989)

Anhang A

Quellcode