Numerische Verfahren für gewöhnliche Differentialgleichungen

Roland Pulch

Vorlesung im Wintersemester 2023/24

Institut für Mathematik und Informatik
Universität Greifswald

Inhalt:

1. Problemstellung und Beispiele
2. Einschrittverfahren
3. Mehrschrittverfahren
4. Methoden für steife Differentialgleichungen

Literatur:

J. Stoer, R. Bulirsch: Numerische Mathematik 2. (5. Aufl.) Springer, 2005. (Kapitel 7)

P. Deuflhard, F. Bornemann: Numerische Mathematik 2. Gewöhnliche Differentialgleichungen. (3. Aufl.) de Gruyter, 2008.

K. Strehmel, R. Weiner, H. Podhaisky: Numerik gewöhnlicher Differentialgleichungen. (2. Aufl.) Vieweg & Teubner, 2012.

H.R. Schwarz, N. Köckler: Numerische Mathematik. (8. Aufl.) Vieweg & Teubner, 2011. (Kapitel 8)

Inhaltsverzeichnis

Kapitel 1

Problemstellung und Beispiele

Diese Veranstaltung behandelt die numerische Lösung von Anfangswertproblemen zu gewöhnlichen Differentialgleichungen. Systeme aus gewöhnlichen Differentialgleichungen (gew. Dgln.) erster Ordnung besitzen die Gestalt

$$y'(x) = f(x, y(x)),$$

oder komponentenweise geschrieben

$$\begin{aligned} y_1'(x) &= f_1(x, y_1(x), \ldots, y_n(x)) \\ y_2'(x) &= f_2(x, y_1(x), \ldots, y_n(x)) \\ &\vdots \\ y_n'(x) &= f_n(x, y_1(x), \ldots, y_n(x)). \end{aligned}$$

Ein solches System besitzt im allgemeinen unendlich viele Lösungen. Daher sind zusätzliche Bedingungen notwendig, um eine eindeutige Lösung zu identifizieren.

Ein Anfangswertproblem (AWP) ergibt sich durch Vorgabe eines Anfangswerts

$$y(x_0) = y_0$$

zu einem bestimmten Anfangspunkt $x_0 \in \mathbb{R}$ zusammen mit einem vorgegebenem Wert $y_0 \in \mathbb{R}^n$. Abb. 1 verdeutlicht diese Problemstellung.

Demgegenüber liegt bei einem Randwertproblem (RWP) eine Bedingung vor, die sowohl einen Anfangszustand als auch einen Endzustand einbezieht,

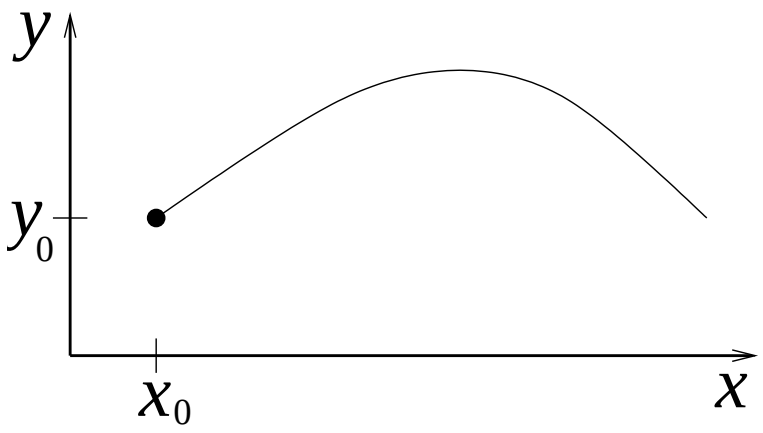


Abbildung 1: Anfangswertproblem einer gewöhnlichen Differentialgleichung.

d.h.

$$r(y(a), y(b)) = 0$$

mit gegebener Funktion $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ zu einem Intervall $[a, b]$. Beispielsweise ergibt sich ein periodisches RWP durch die Forderung $y(a) - y(b) = 0$.

Eine gew. Dgl. n-ter Ordnung lautet

$$z^{(n)}(x) = g(x, z(x), z'(x), z''(x), \ldots, z^{(n-1)}(x)).$$

Wir erhalten ein äquivalentes System erster Ordnung durch die Definition

$$y_1 := z, \quad y_2 := z', \quad y_3 := z'', \quad \ldots \quad , \quad y_n := z^{(n-1)}.$$

Es folgt das System

$$y_1' = y_2, \quad y_2' = y_3, \quad \ldots \quad , \quad y_{n-1}' = y_n, \quad y_n' = g(x, y_1, \ldots, y_n).$$

Daher betrachten wir in dieser Veranstaltung o.E.d.A. nur Systeme erster Ordnung.

Desweiteren gibt es die Klasse der partiellen Differentialgleichungen (part. Dgln.), wobei die Lösung nicht mehr nur von einer unabhängigen Veränderlichen sondern mehreren unabhängigen Veränderlichen abhängt. Hier können AWPe, RWPe oder Kombinationen aus beiden vorliegen.

In den meisten Fällen können AWPe oder RWPe von Dgln. nicht analytisch gelöst werden, d.h. es existiert keine geschlossene Formel für die Lösung.

Auch wenn ein analytischer Lösungsweg möglich ist, so möchte man diesen meist vermeiden wegen des hohen Aufwands. Daher sind numerische Verfahren für diese Probleme erforderlich.

Wir präsentieren später Beispiele von Modellen aus gew. Dgln., die in verschiedenen Anwendungsgebieten auftreten: in der Chemie (Reaktionskinetik), in der Elektrotechnik (elektrische Schaltung) und in der Mechanik (Mehrkörperproblem). In allen Fällen werden diese mathematischen Modelle zur (näherungsweisen) Beschreibung von realen Prozessen eingesetzt. Wegen vereinfachender Modellannahmen stellt die Lösung der Dgln. eine Approximation der realen Zustände dar.

1.1 Existenz und Eindeutigkeit

Gegeben sei das AWP

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \tag{1.1}$$

mit der rechten Seite $f : G \to \mathbb{R}^n$ auf einer offenen Menge $G \subseteq \mathbb{R} \times \mathbb{R}^n$. Es gilt $(x_0, y_0) \in G$ für den Anfangswert. Wir benötigen als Voraussetzung die Existenz und Eindeutigkeit der Lösung y. Der Satz von Peano liefert eine Existenzaussage für eine stetige rechte Seite, jedoch keine Eindeutigkeitsaussage. Für die Eindeutigkeit wird eine stärkere Eigenschaft benötigt.

Definition 1.1 *Die rechte Seite f genügt auf der offenen Menge G einer globalen Lipschitz-Bedingung, wenn eine Konstante $L > 0$ existiert mit*

$$\|f(x, y) - f(x, z)\| \leq L \cdot \|y - z\| \tag{1.2}$$

für alle $(x, y), (x, z) \in G$. Die rechte Seite f erfüllt eine lokale Lipschitz-Bedingung, wenn zu jedem $(\hat{x}, \hat{y}) \in G$ eine Umgebung $U \subset G$ exisitert, so dass f auf U eine Lipschitz-Bedingung mit einer von U abhängigen Konstanten $L > 0$ erfüllt.

In dieser Definition wird eine beliebige Vektornorm $\|\cdot\|$ auf dem $\mathbb{R}^n$ verwendet. Hinreichend für die lokale Lipschitz-Bedingung ist, dass f auf G

bezüglich der Variablen y stetig differenzierbar ist. Der Satz von Picard-Lindelöf gibt nun eine Existenz- und Eindeutigkeitsaussage.

Satz 1.1 (Picard-Lindelöf) *Sei $G \subseteq \mathbb{R} \times \mathbb{R}^n$ offen und $f : G \to \mathbb{R}^n$ eine stetige Funktion, die eine lokale Lipschitz-Bedingung erfüllt. Dann gibt es zu jedem Anfangswert $(x_0, y_0) \in G$ ein $\varepsilon > 0$ und eine eindeutige Lösung $y : [x_0 - \varepsilon, x_0 + \varepsilon] \to \mathbb{R}^n$ des AWPs (1.1).*

Beweis: siehe Satz 4 in Kapitel 12 aus [3].

1.2 Chemische Reaktionskinetik

Chemische Prozesse enthalten typischerweise bimolekulare Reaktionen der Gestalt

$$\mathrm{A} \quad + \quad \mathrm{B} \quad \longrightarrow \quad \mathrm{C} \quad + \quad \mathrm{D}.$$

Sei c_{S} die Konzentration der Substanz S, welche von der Zeit t abhängt. Das zugehörige System gew. Dgln. lautet

$$\begin{aligned} c'_{\mathrm{A}}(t) &= -k\, c_{\mathrm{A}}(t) c_{\mathrm{B}}(t) \\ c'_{\mathrm{B}}(t) &= -k\, c_{\mathrm{A}}(t) c_{\mathrm{B}}(t) \\ c'_{\mathrm{C}}(t) &= +k\, c_{\mathrm{A}}(t) c_{\mathrm{B}}(t) \\ c'_{\mathrm{D}}(t) &= +k\, c_{\mathrm{A}}(t) c_{\mathrm{B}}(t). \end{aligned} \tag{1.3}$$

Die Reaktionsrate $k > 0$ charakterisiert die Wahrscheinlichkeit der chemischen Reaktion im Fall einer Kollision der Moleküle A und B. Der Koeffizient k kann daher Geschwindigkeit der Reaktion interpretiert werden. Die physikalische Einheit des Parameters k ist liter/(s mol). Anfangswerte sind für das System (1.3) vorzugeben.

Nun betrachten wir eine Menge aus m allgemeinen chemischen Reaktionen zwischen n verschiedenen Stoffen $\mathrm{A}_1, \ldots, \mathrm{A}_n$ (Moleküle/Atome), d.h.

$$\alpha_{1j}\mathrm{A}_1 + \alpha_{2j}\mathrm{A}_2 + \cdots + \alpha_{nj}\mathrm{A}_n \quad \xrightarrow{k_j} \quad \gamma_{1j}\mathrm{A}_1 + \gamma_{2j}\mathrm{A}_2 + \cdots + \gamma_{nj}\mathrm{A}_n$$

für $j = 1, \ldots, m$ oder äquivalent

$$\sum_{i=1}^{n} \alpha_{ij} \mathrm{A}_i \quad \xrightarrow{k_j} \quad \sum_{i=1}^{n} \gamma_{ij} \mathrm{A}_i \qquad \text{für } \; j = 1, \ldots, m. \tag{1.4}$$

Die Parameter $\alpha_{ij}, \gamma_{ij} \in \mathbb{N}_0$ stellen die stoichiometrischen Konstanten dar. Die j-te Reaktion besitzt die Geschwindigkeit $k_j \in \mathbb{R}^+$. Es entsteht als mathematisches Modell

$$\frac{\mathrm{d}c_{\mathrm{A}_i}}{\mathrm{d}t} = \sum_{j=1}^{m} (\gamma_{ij} - \alpha_{ij}) k_j \prod_{l=1}^{n} {c_{\mathrm{A}_l}}^{\alpha_{lj}} \qquad \text{für } \; i = 1, \ldots, n \; ,$$

welches ein System aus n gew. Dgln. für die unbekannten Konzentrationen darstellt. Die Auswertung der rechten Seite kann automatisch erfolgen, wenn die Reaktionsgleichungen (1.4) spezifiziert sind.

Die Hydrolyse von Harnstoff ist ein Beispiel für ein chemisches Reaktionssystem. Dabei reagiert Harnstoff (Urea) zusammen mit Wasser zu Ammoniumcarbonat. Für eine genügende Schnelligkeit der Reaktion ist die Hilfe des Enzyms Urease erforderlich, da es die Aktivierungsenergie verringert, d.h. das Enzym fungiert als Katalysator. Das gesamte Reaktion lautet

$$(\mathrm{NH_2})_2\mathrm{CO} + 2\ \mathrm{H_2O} + \text{Urease} \quad \longrightarrow \quad (\mathrm{NH_4})_2\mathrm{CO_3} + \text{Urease}. \tag{1.5}$$

Dieser Prozess ergibt sich aus drei einfacheren Reaktionen. Mit den Abkürzungen U: Urea, E: Urease, UE: Kombination aus Urea and Urease, A: Ammoniumcarbonat besteht die Reaktion (1.5) aus den drei Anteilen

$$\begin{aligned} \mathrm{U} + \mathrm{E} \quad &\xrightarrow{k_1} \quad \mathrm{UE} \\ \mathrm{UE} \quad &\xrightarrow{k_2} \quad \mathrm{U} + \mathrm{E} \\ \mathrm{UE} + 2\ \mathrm{H_2O} \quad &\xrightarrow{k_3} \quad \mathrm{A} + \mathrm{E}. \end{aligned} \tag{1.6}$$

Die Parameter k_1, k_2, k_3 spezifizieren die Reaktionsgeschwindigkeiten.

Wir konstruieren ein mathematisches Modell für dieses Reaktionssystem. Sei wieder c_S die Konzentration der Substanz S in der Einheit mol/liter (mol/l). Die zeitliche Änderung der Konzentrationen soll bestimmt werden. Da alle Reaktionen in Wasser stattfinden und die anderen Konzentrationen

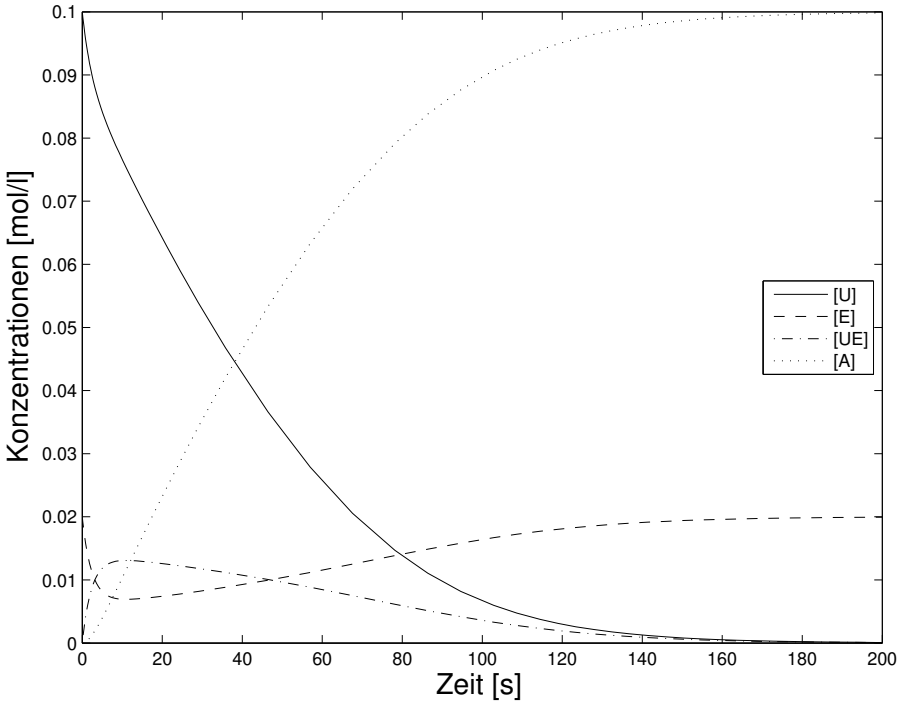


Abbildung 2: Simulation der Hydrolyse von Harnstoff.

demgegenüber relativ klein sind, nehmen wir eine konstante Wasserkonzentration (55.56 mol/l) an. Die Reaktionsgeschwindigkeiten sind

$$k_1 = 3.01 \ \frac{\mathrm{l}}{\mathrm{mol}\cdot\mathrm{s}}, \quad k_2 = 0.02 \ \tfrac{1}{\mathrm{s}}, \quad k_3 = 0.1 \ \tfrac{1}{\mathrm{s}}. \tag{1.7}$$

Es folgt ein System aus vier gew. Dgln. für die unbekannten Konzentrationen

$$\begin{array}{rcrcrcr} c'_{\mathrm{U}} & = & -k_1 c_{\mathrm{U}} c_{\mathrm{E}} & + k_2 c_{\mathrm{UE}} & & \\ c'_{\mathrm{E}} & = & -k_1 c_{\mathrm{U}} c_{\mathrm{E}} & + k_2 c_{\mathrm{UE}} & + k_3 c_{\mathrm{UE}} & \\ c'_{\mathrm{UE}} & = & k_1 c_{\mathrm{U}} c_{\mathrm{E}} & - k_2 c_{\mathrm{UE}} & - k_3 c_{\mathrm{UE}} & \\ c'_{\mathrm{A}} & = & & & k_3 c_{\mathrm{UE}}. & \end{array} \tag{1.8}$$

Dieses System besitzt eine eindeutige Lösung für vorgegebene Anfangswerte. Wir verwenden die Anfangsbedingungen

$$c_{\mathrm{U}} = 0.1 \ \frac{\mathrm{mol}}{\mathrm{l}}, \quad c_{\mathrm{E}} = 0.02 \ \frac{\mathrm{mol}}{\mathrm{l}}, \quad c_{\mathrm{UE}} = c_{\mathrm{A}} = 0. \tag{1.9}$$

Wie bei vielen anderen Anwendungen ist eine analytische Lösung des Systems aus gew. Dgln. nicht möglich, d.h. wir können keine explizite Formel

für die unbekannte Lösung erhalten. Daher verwenden wir eine numerisches Verfahren um eine Näherungslösung zu erhalten. Abb. 2 zeigt das Ergebnis.

Zum einen verringert sich die Konzentration von Harnstoff mit der Zeit bis auf null, da die Substanz in der Hydrolyse abgebaut wird. Zum anderen erhöht sich die Konzentration des Reaktionsprodukts Ammoniumcarbonat bis kein Harnstoff mehr vorhanden ist. Die Konzentration des Enzyms verringert sich zwar anfangs, jedoch liegt am Ende wieder genau soviel Enzym vor wie zu Beginn.

1.3 Elektrische Schaltungen

Als ein einfaches Beispiel einer elektrischen Schaltung betrachten wir den elektromagnetischen Schwingkreis. Dieser besteht aus einer Kapazität C, einer Induktivität L und einem Widerstand R, siehe Abb. 3 (links). Die Kirchhoffsche Knotenregel liefert die Gleichung

$$I_C + I_L + I_R = 0.$$

Die Kirchhoffsche Maschenregel impliziert $U := U_C = U_L = U_R$. Jedes Element der Schaltung ist durch eine Strom-Spannungs-Relation gekennzeichnet, nämlich

$$CU_C' = I_C, \quad LI_L' = U_L, \quad R = \frac{U_R}{I_R}.$$

Es folgt ein lineares System aus zwei gew. Dgln.

$$\begin{array}{rcl} U' & = & -\frac{1}{C} I_L - \frac{1}{RC} U \\ I_L' & = & \frac{1}{L} U \end{array} \tag{1.10}$$

für die unbekannten Funktionen U und I_L. Weitere Umformungen liefern eine gew. Dgl. zweiter Ordnung für die unbekannte Spannung

$$U'' + \tfrac{1}{RC} U' + \tfrac{1}{LC} U = 0.$$

Wenn der Widerstand hinreichend klein ist, dann entsteht als Lösung eine gedämpfte Schwingung

$$U(t) = \mathrm{e}^{-\frac{1}{2RC}t} \left[A \sin(\omega t) + B \cos(\omega t) \right] \quad \text{mit} \quad \omega = \sqrt{\tfrac{1}{LC} - \tfrac{1}{4R^2C^2}}.$$

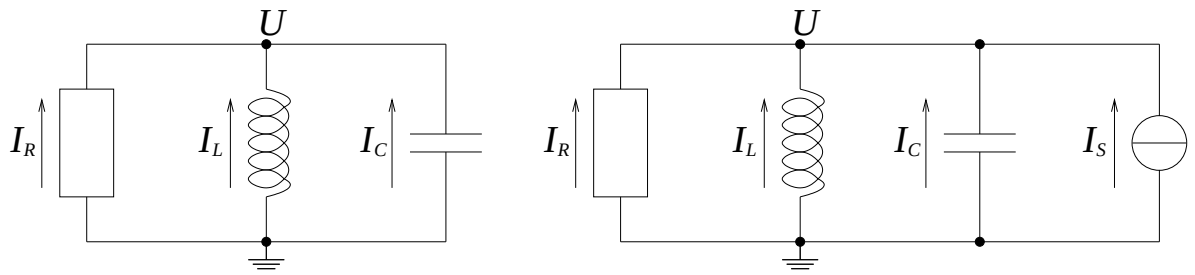


Abbildung 3: Elektromagnetischer Schwingkreis ohne (links) und mit (rechts) Stromquelle.

Die Konstanten A und B bestimmen sich aus Anfangsbedingungen.

Das System (1.10) aus gew. Dgln. ist autonom. Wir erhalten eine zeitabhängiges System durch Hinzufügen einer unabhängigen Stromquelle, siehe Abb. 3 (rechts). Als Eingabe verwenden wir

$$I_{\text{in}}(t) = I_0 \sin(\omega_0 t).$$

Es folgen dann die Dgln.

$$\begin{aligned} U' &= -\tfrac{1}{C} I_L - \tfrac{1}{RC} U - \tfrac{1}{C} I_{\text{in}}(t) \\ I_L' &= \tfrac{1}{L} U. \end{aligned} \tag{1.11}$$

In diesem Fall entstehen als Spannungen und Ströme dann erzwungene Schwingungen. Abb. 4 zeigt Beispiele zu Lösungen von Anfangswertproblemen der Systeme (1.10) und (1.11).

1.4 Mehrkörpersysteme

Wir betrachten ein Zwei-Körper-Problem bestehend aus Partikeln mit den Massen m_1, m_2. Sei $\vec{X}_i = (x_i, y_i, z_i)$ die Position der i-ten Masse. Die Positionen und die Geschwindigkeiten der Körper hängen von der Zeit ab. Die Gravitation bewirkt Kräfte zwischen den Massen. Das Newtonsche Bewegungsgesetz liefert ein System aus gew. Dgln. zweiter Ordnung

$$\begin{aligned} m_1 \vec{X}_1''(t) &= G \tfrac{m_1 m_2}{|\vec{X}_1(t) - \vec{X}_2(t)|^3} (\vec{X}_2(t) - \vec{X}_1(t)) \\ m_2 \vec{X}_2''(t) &= G \tfrac{m_1 m_2}{|\vec{X}_1(t) - \vec{X}_2(t)|^3} (\vec{X}_1(t) - \vec{X}_2(t)) \end{aligned}$$

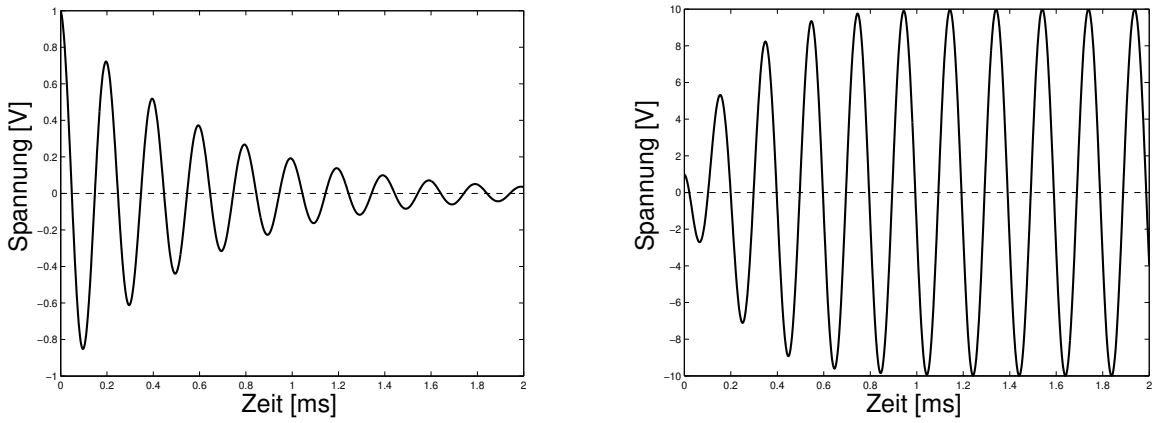


Abbildung 4: Lösung U der Dgl. (1.10) (links) und der Dgl. (1.11) (rechts).

mit der Gravitationskonstanten $G > 0$. Setzen wir $\vec{V}_i := \vec{X}_i'$ für die Geschwindigkeiten, so folgt ein System erster Ordnung

$$\begin{aligned} \vec{X}_1' &= \vec{V}_1 \\ \vec{V}_1' &= G\frac{m_2}{|\vec{X}_1-\vec{X}_2|^3}(\vec{X}_2-\vec{X}_1) \\ \vec{X}_2' &= \vec{V}_2 \\ \vec{V}_2' &= G\frac{m_1}{|\vec{X}_1-\vec{X}_2|^3}(\vec{X}_1-\vec{X}_2) \end{aligned}$$

bestehend aus zwölf Gleichungen. Dieses System ist autonom. Anfangsbedingungen $\vec{X}_i(0), \vec{V}_i(0)$ müssen vorgegeben werden. Abb. 5 zeigt die Trajektorien eines Zwei-Körper-Problems mit unterschiedlichen Massen $m_1 > m_2$. Die Bewegung erfolgt hier typischerweise ungefähr entlang von Ellipsen.

Wir leiten nun das N-Körper-Problem für Massen $m_1, \ldots, m_N$ her. Sei $\vec{F}_{ij}$ die Gravitationskraft auf die i-te Masse verursacht von der j-ten Masse. Das Newtonsche Bewegungsgesetz impliziert

$$m_i\vec{X}_i'' = \sum_{j=1,j\neq i}^{N} \vec{F}_{ij} = \sum_{j=1,j\neq i}^{N} G\frac{m_i m_j}{|\vec{X}_j-\vec{X}_i|^3}(\vec{X}_j-\vec{X}_i)$$

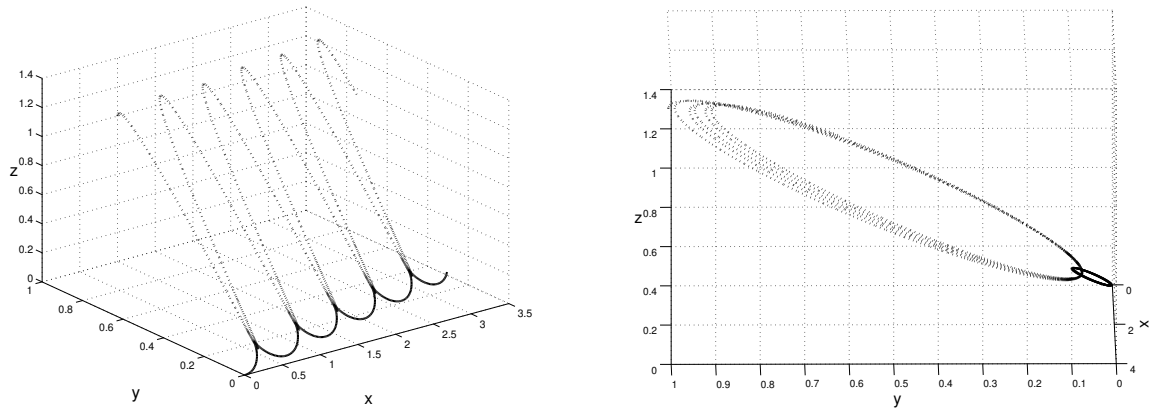


Abbildung 5: Trajektorien (Positionen) eines Zwei-Körper-Problems mit Massen $m_1 > m_2$ aus zwei unterschiedlichen Blickwinkeln (Linie: erster Körper, Punkte: zweiter Körper).

für $i = 1, \ldots, N$. Es folgt ein System erster Ordnung aus $6N$ gew. Dgln.

$$\begin{aligned} \vec{X}_i' &= \vec{V}_i \\ \vec{V}_i' &= G \sum_{j=1, j\neq i}^{N} \frac{m_j}{|\vec{X}_j - \vec{X}_i|^3} (\vec{X}_j - \vec{X}_i) \qquad \text{für } i = 1, \ldots, N. \end{aligned}$$

Das Zwei-Körper-Problem kann noch analytisch gelöst werden, während dies nicht mehr für das N-Körper-Problem mit $N > 2$ gilt. Daher benötigen wir numerische Verfahren für diese Aufgabenstellung.

Weitere Modelle

In den vorangehenden Abschnitten wurden Probleme aus dem Bereich der Chemie, der Elektrotechnik und der Mechanik besprochen. Systeme aus gew. Dgln. treten ebenfalls in den folgenden Anwendungen auf:

- Biologie (z.B. Räuber-Beute-Modelle, Epidemie-Modelle),
- Simulation von Kriegsgefechten (Lanchester Modelle),
- Diskretisierung von partiellen Differentialgleichungen,
- und andere.

Weitere Literatur zu Modellen mit gew. Dgln. ist beispielsweise [1, 2].

Kapitel 2

Einschrittverfahren

Wir besprechen jetzt numerische Methoden für die Anfangswertprobleme, die im vorhergehenden Kapitel eingeführt wurden. Dabei wird mit Einschrittverfahren begonnen, während Mehrschrittverfahren im nächsten Kapitel behandelt werden.

2.1 Vorbereitungen

Wir möchten ein AWP (1.1) eines Systems gew. Dgln. in einem Intervall $[x_0, x_{\text{end}}]$ ($x_0 < x_{\text{end}}$) numerisch lösen. Alle Verfahren für AWPe, die in dieser Veranstaltung betrachtet werden, verwenden eine endliche Anzahl von Gitterpunkten

$$x_0 < x_1 < x_2 < x_3 < \cdots < x_{N-1} < x_N = x_{\text{end}}.$$

Eine mögliche Wahl sind äquidistante Gitterpunkte

$$x_i := x_0 + ih \qquad \text{mit } h := \frac{x_{\text{end}} - x_0}{N} \qquad \text{für } i = 0, 1, \ldots, N.$$

Numerische Lösungen $y_i \approx y(x_i)$ werden sukzessive berechnet. In einem Einschrittverfahren ist die Abhängigkeit der Werte einfach

$$y_0 \longrightarrow y_1 \longrightarrow y_2 \longrightarrow \cdots \longrightarrow y_{N-1} \longrightarrow y_N.$$

Im Gegensatz dazu liegt bei einem Mehrschrittverfahren mit k Schritten eine Abhängigkeit vor der Gestalt

$$y_{i-k}, y_{i-k+1}, \ldots, y_{i-2}, y_{i-1} \quad \longrightarrow \quad y_i \qquad \text{für } i = k, k+1, \ldots, N.$$

Dabei müssen die ersten Werte $y_1, \ldots, y_{k-1}$ durch eine andere Methode bestimmt werden im Fall $k > 1$. Ein Einschrittverfahren liegt auch in einer Mehrschrittmethode im Spezialfall $k = 1$ vor.

Ein allgemeines Einschrittverfahren kann in der Form

$$y_{i+1} = y_i + h_i \Phi(x_i, y_i, h_i), \tag{2.1}$$

geschrieben werden mit einer Inkrementfunktion Φ, die von sowohl dem Verfahren als auch der rechten Seite f abhängt.

2.2 Elementare Integrationsverfahren

Die meisten Methoden für AWP (1.1) basieren auf einer Approximation der äquivalenten Integralgleichung

$$y(x) = y(x_0) + \int_{x_0}^{x} f(s, y(s)) \, \mathrm{d}s. \tag{2.2}$$

Im Intervall $[x_0, x_0 + h]$ erhalten wir

$$\begin{aligned} y(x_0 + h) &= y_0 + \int_{x_0}^{x_0+h} f(s, y(s)) \, \mathrm{d}s \\ &= y_0 + h \int_0^1 f(x_0 + sh, y(x_0 + sh)) \, \mathrm{d}s. \end{aligned} \tag{2.3}$$

Nun wird das Integral auf der rechten Seite durch eine Quadraturformel ersetzt. Die Schwierigkeit besteht darin, dass die Funktion y, welche im Integranden auftritt, a priori unbekannt ist.

Da die Schrittweite h klein ist, verwenden wir einfache Quadraturformeln und keine zusammengesetzten Quadraturformeln. Wir diskutieren die folgenden Beispiele, siehe Abb. 6:

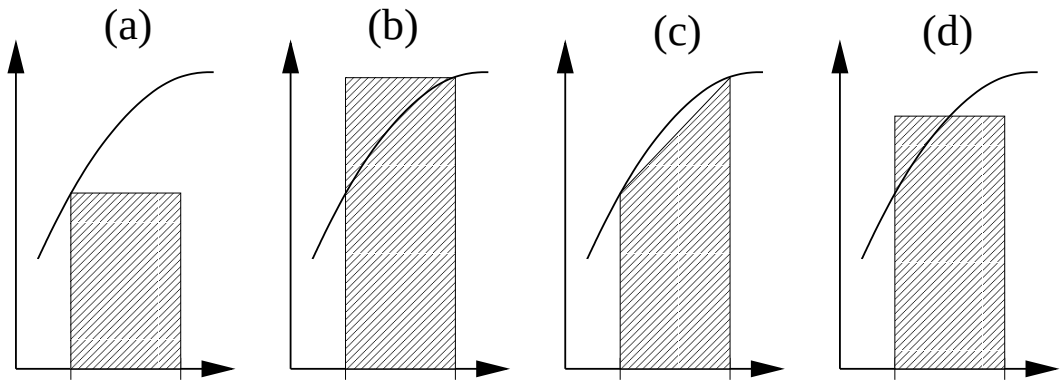


Abbildung 6: Elementare Quadraturregeln: (a) Rechteck (linksseitig), (b) Rechteck (rechtsseitig), (c) Trapezregel, (d) Mittelpunktregel.

(a) Rechteckregel (linksseitig):

Als Näherung ergibt sich

$$y_1 = y_0 + hf(x_0, y_0).$$

Diese Methode wird (explizites) Euler-Verfahren genannt. Es ist die einfachste Methode, die durchführbar ist. Ist der Anfangswert $y(x_0) = y_0$ gegeben, dann kann die Näherung y_1 direkt über eine Funktionsauswertung von f berechnet werden.

(b) Rechteckregel (rechtsseitig):

Nun folgt als Methode

$$y_1 = y_0 + hf(x_0 + h, y_1). \tag{2.4}$$

Diese Technik wird implizites Euler-Verfahren genannt. Der unbekannte Wert y_1 tritt auf beiden Seiten der Gleichung auf. Im allgemeinen kann hier keine explizite Formel für y_1 bestimmt werden. Die Vorschrift (2.4) stellt ein nichtlineares Gleichungssystem (aus algebraischen Gleichungen) für die Unbekannte y_1 dar, d.h. der Wert y_1 ist implizit definiert. Beispielsweise kann mit einer Newton-Iteration eine Näherungslösung erhalten werden. Der Rechenaufwand für einen Integrationsschritt ist somit aber deutlich höher als im expliziten Euler-Verfahren.

(c) Trapezregel:

Wird das Integral mit der Trapezregel approximiert, dann folgt die Vorschrift

$$y_1 = y_0 + h\tfrac{1}{2}\left(f(x_0, y_0) + f(x_0 + h, y_1)\right).$$

Dieser Ansatz führt daher wieder auf eine implizite Methode. Der Rechenaufwand für einen Integrationsschritt ist ungefähr so hoch wir im impliziten Euler-Verfahren. Jedoch kann man eine deutlich besser Näherung erwarten, da die Trapeze besser approximieren als die Rechtecke in der Quadratur.

(d) Mittelpunktregel:

Die Mittelpunktregel verwendet ein bestimmtes Rechteck. Es folgt

$$y_1 = y_0 + hf(x_0 + \tfrac{1}{2}h, y(x_0 + \tfrac{1}{2}h)). \tag{2.5}$$

Diese Vorschrift ist noch nicht durchführbar, denn Unbekannte sind sowohl y_1 als auch $y(x_0 + \frac{1}{2}h)$. Daher benötigen wir eine weitere Gleichung, um den Zwischenwert $y(x_0 + \frac{1}{2}h)$ zu bestimmen. Beispielsweise kann das explizite Euler-Verfahren diesen Wert liefern, wodurch folgt

$$\begin{cases} y_{1/2} &= y_0 + \frac{h}{2} f(x_0, y_0) \\ y_1 &= y_0 + hf(x_0 + \frac{1}{2}h, y_{1/2}) \end{cases}$$

oder äquivalent

$$y_1 = y_0 + hf(x_0 + \tfrac{h}{2}, y_0 + \tfrac{h}{2} f(x_0, y_0)). \tag{2.6}$$

Diese Methode ist explizit, denn man kann sukzessive $y_{1/2}$ und y_1 berechnen ohne nichtlineare Gleichungssysteme zu lösen. Es werden nur zwei Funktionsauswertungen von f benötigt. Die Vorschrift (2.6) wird modifiziertes Euler-Verfahren oder Collatz-Verfahren genannt. Alternativ entsteht eine implizite Methode, wenn der Zwischenwert linear durch y_0 und y_1 interpoliert wird, d.h.

$$y_1 = y_0 + hf(x_0 + \tfrac{h}{2}, \tfrac{1}{2}(y_0 + y_1))$$

ergibt sich als Verfahrensvorschrift. Man nennt dies auch die implizite Mittelpunktregel.

Die Genauigkeit dieser Methoden wird später untersucht.

Explizites Euler-Verfahren

Wir betrachten das explizite Euler-Verfahren jetzt genauer. Diese Formel kann auch durch zwei andere Ansätze erhalten werden. Zum einen ersetzen wir die Ableitung in der Dgl. $y' = f(x, y)$ durch den Differenzenquotienten (erster Ordnung), wodurch folgt

$$\frac{y(x_0 + h) - y(x_0)}{h} \doteq f(x_0, y(x_0)) \quad \Rightarrow \quad y_1 = y_0 + hf(x_0, y_0).$$

Zum anderen verwenden wir die Tangente zu $y(x)$ im Anfangspunkt (x_0, y_0) zur Approximation der Lösung. Die Tangentengleichung lautet

$$t(x) = y(x_0) + (x - x_0)y'(x_0) = y(x_0) + (x - x_0)f(x_0, y(x_0)).$$

Es folgt

$$y_1 := t(x_0 + h) = y_0 + hf(x_0, y_0),$$

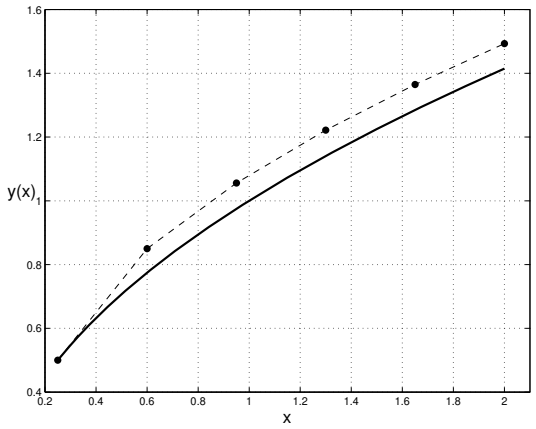
d.h. wir erhalten das explizite Euler-Verfahren. Die sukzessive Anwendung dieser Methode liefert daher Tangentenstücke, wodurch diese Technik auch Polygonzugverfahren genannt wird.

Als Beispiel lösen wir das AWP

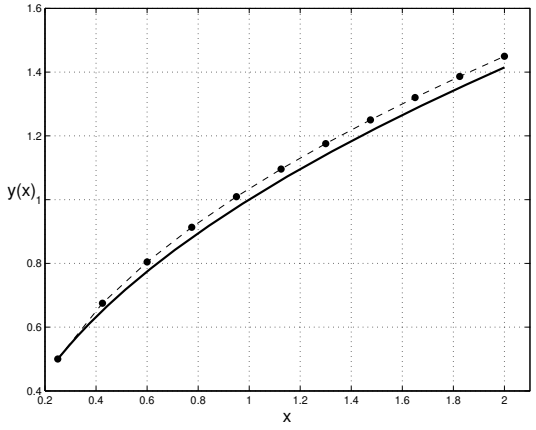
$$y' = \frac{1}{2y}, \qquad y(\tfrac{1}{4}) = \tfrac{1}{2}, \qquad x \in [\tfrac{1}{4}, 2].$$

Die exakte Lösung ist $y(x) = \sqrt{x}$. Abb. 7 zeigt die Näherungslösungen aus dem expliziten Euler-Verfahren. Wir erkennen, dass die Näherungen mit steigender Schrittzahl N bzw. kleinerer Schrittweite h genauer werden.

$N = 5$



$N = 10$



$N = 50$

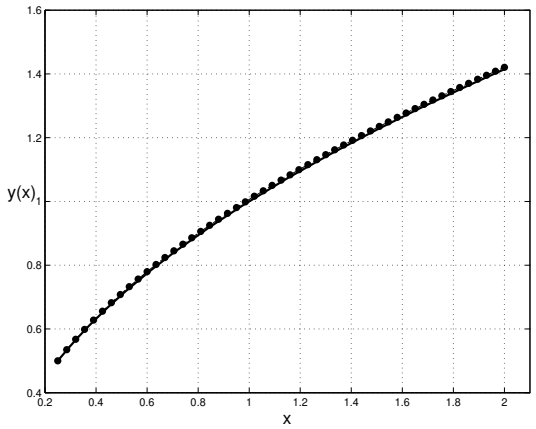


Abbildung 7: Lösung von $y' = \frac{1}{2y}$, $y(\frac{1}{4}) = \frac{1}{2}$ (Linie) und Näherungslösung (Punkte) aus dem expliziten Euler-Verfahren mit N Schritten.

2.3 Konsistenz und Konvergenz

Wir betrachten ein allgemeines Einschrittverfahren der Gestalt (2.1) mit der Inkrementfunktion Φ.

Es existieren unterschiedliche Notationen, um die Genauigkeit der Näherung y_{i+1} mit der exakten Lösung $y(x_{i+1})$ zu vergleichen. Zu einem lokalen Bereich formulieren wir die folgende Definition.

Definition 2.1 (lokaler Diskretisierungsfehler) *Sei $y(x)$ die exakte Lösung des AWPs $y' = f(x, y)$, $y(x_0) = y_0$ und $y_1 = y_0 + h\Phi(x_0, y_0, h)$ die Näherung aus einem Einschrittverfahren mit Schrittweite $h > 0$. Dann lautet der lokale Diskretisierungsfehler*

$$\tau(h) := \frac{y(x_0 + h) - y_1}{h}. \tag{2.7}$$

Der lokale Diskretisierungsfehler hängt auch von der Wahl des Anfangswerts (x_0, y_0) ab, welches in der Notation jedoch nicht extra aufgezeigt wird.

Die Definition (2.7) des lokalen Diskretisierungsfehlers kann auf drei Arten interpretiert werden:

- die Differenz zwischen der exakten Lösung und der Näherungslösung (Diskretisierungsfehler nach einem Schritt ausgehend von der exakten Lösung) skaliert mit der Schrittweite h.
- die Differenz zwischen den Steigungen der entsprechenden Sekanten

  $$\tau(h) = \underbrace{\frac{y(x_0 + h) - y_0}{h}}_{\text{exakte Lösung}} - \underbrace{\frac{y_1 - y_0}{h}}_{\text{Näherungslösung}}.$$

  Die Sekanten sind in Abb. 8 dargestellt. Für $\tau(h) \to 0$ werden beide Sekanten zur Tangente $t(x) = y(x_0) + (x - x_0)y'(x_0)$ im Grenzfall.
- der Defekt

  $$\tau(h) = \frac{y(x_0 + h) - y_0}{h} - \Phi(x_0, y_0, h), \tag{2.8}$$

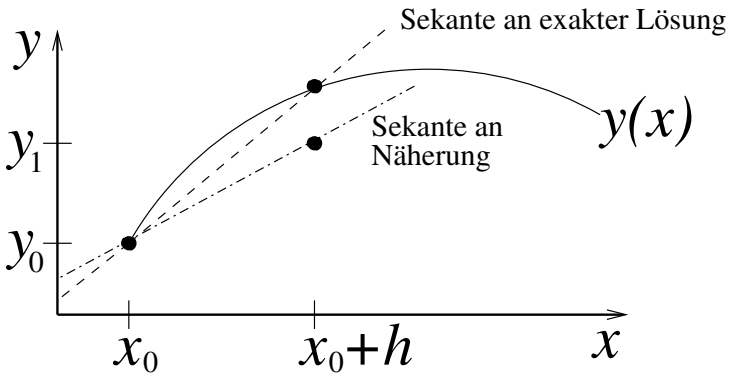


Abbildung 8: Sekanten an die exakte Lösung und an die Näherungslösung.

welcher sich durch Einsetzen der exakten Lösung in die Formel der Näherung ergibt. Mit Definition der stetigen Funktion

$$\Delta(x_0, y_0, h) := \begin{cases} \frac{y(x_0+h)-y_0}{h} & \text{für } h > 0 \\ f(x_0, y_0) & \text{für } h = 0 \end{cases} \tag{2.9}$$

kann man genauer schreiben

$$\tau(x_0, y_0, h) = \Delta(x_0, y_0, h) - \Phi(x_0, y_0, h) \qquad \text{für } h \geq 0.$$

Beispiel 1: Lokaler Diskretisierungsfehler im expliziten Euler-Verfahren

Taylor-Entwicklung liefert unter der Annahme $y \in C^2$

$$\begin{aligned} y(x_0 + h) &= y(x_0) + hy'(x_0) + \tfrac{1}{2}h^2 y''(x_0 + \vartheta(h)h) \\ &= y_0 + hf(x_0, y_0) + \tfrac{1}{2}h^2 y''(x_0 + \vartheta(h)h) \end{aligned}$$

mit $0 < \vartheta(h) < 1$.

Der lokale Diskretisierungsfehler ergibt sich zu

$$\begin{aligned} \tau(h) &= \tfrac{1}{h}(y(x_0 + h) - y_1) \\ &= \tfrac{1}{h}(y(x_0 + h) - y_0 - hf(x_0, y_0)) \\ &= \tfrac{1}{2}hy''(x_0 + \vartheta(h)h). \end{aligned}$$

Es folgt $\tau(h) = \mathcal{O}(h)$.

Beispiel 2: Lokaler Diskretisierungsfehler im impliziten Euler-Verfahren

Zur Vereinfachung setzen wir eine beschränkte rechte Seite voraus, d.h. $|f| \leq M$. Zum einen liefert das implizite Euler-Verfahren

$$y_1 = y_0 + hf(x_0 + h, y_1) = y_0 + hf(x_0 + h, y_0 + hf(x_0 + h, y_1)).$$

Mehrdimensionale Taylor-Entwicklung der Funktion $f \in C^2$ zeigt uns

$$\begin{aligned} y_1 &= y_0 + h\left[f(x_0, y_0) + \tfrac{\partial f}{\partial x}(x_0, y_0)h + \tfrac{\partial f}{\partial y}(x_0, y_0)hf(x_0 + h, y_1) + \mathcal{O}(h^2)\right] \\ &= y_0 + hf(x_0, y_0) + \mathcal{O}(h^2). \end{aligned}$$

Zum anderen liefert eine Taylor-Entwicklung der exakten Lösung

$$\begin{aligned} \tau(h) &= \tfrac{1}{h}(y(x_0 + h) - y_1) \\ &= \tfrac{1}{h}(y_0 + hf(x_0, y_0) + \mathcal{O}(h^2) - (y_0 + hf(x_0, y_0) + \mathcal{O}(h^2))) = \mathcal{O}(h). \end{aligned}$$

Wieder folgt $\tau(h) = \mathcal{O}(h)$.

Aufbauend auf den Eigenschaften des lokalen Diskretisierungsfehlers definieren wir die Konsistenz.

Definition 2.2 (Konsistenz) *Ein Einschrittverfahren (oder dessen Inkrementfunktion Φ) heißt konsistent, wenn der lokale Diskretisierungsfehler bei allen Anfangswerten $(x_0, y_0) \in G$ (G: Definitionsbereich von f) gegen null konvergiert für kleine Schrittweiten:*

$$\|\tau(h)\| \leq \sigma(h) \qquad \textit{mit} \qquad \lim_{h \to 0} \sigma(h) = 0.$$

Das Verfahren heißt konsistent von (mindestens) Ordnung p, wenn

$$\|\tau(h)\| = \mathcal{O}(h^p).$$

Die Konsistenz eines Einschrittverfahrens kann leicht mit der folgenden Eigenschaft charakterisiert werden.

Lemma 2.1 *Sei die rechte Seite f der Dgl. $y' = f(x, y)$ stetig in x und erfülle die Lipschitz-Bedingung (1.2) bezüglich y. Dann gilt die Äquivalenz*

$$\Phi \text{ ist konsistent} \quad \Leftrightarrow \quad \lim_{h \to 0} \Phi(x, y, h) = f(x, y).$$

Beweis:

Sei z die Lösung des AWPs $z'(x) = f(x, z(x))$, $z(x_0) = y_0$ mit $(x_0, y_0) \in G$. Wegen der Definition von τ und dem Mittelwertsatz der Differentialrechnung gilt komponentenweise

$$\tau_j(x_0, y_0, h) = \frac{z_j(x_0 + h) - y_0}{h} - \Phi_j(x_0, y_0, h) = z_j'(x_0 + \theta_j h) - \Phi_j(x_0, y_0, h)$$

mit Zwischenwerten $\theta_j \in (0, 1)$ für $j = 1, \ldots, n$. Mit der Stetigkeit von z' folgt

$$\lim_{h \to 0} z_j'(x_0 + \theta_j h) = z_j'(x_0) = f_j(x_0, y_0)$$

für $j = 1, \ldots, n$. Somit gilt

$$\lim_{h \to 0} \|\tau(x_0, y_0, h)\| = \|f(x_0, y_0) - \lim_{h \to 0} \Phi(x_0, y_0, h)\|$$

falls die Grenzwerte existieren. Diese Gleichung liefert die Behauptung. □

Die Konsistenzordnung beschreibt die Qualität der Näherung nach einem einzelnen Schritt. Jedoch sind wir an der Qualität der Approximation nach N Schritten interessiert, wo der Endpunkt x_{end} erreicht wird. Dies motiviert die nächste Definition.

Definition 2.3 (globaler Diskretisierungsfehler und Konvergenz)
Der globale Diskretisierungsfehler einer Methode auf einem Gitter $x_0 < x_1 < \cdots < x_N$ ist definiert durch die Differenz

$$e_N := y(x_N) - y_N. \tag{2.10}$$

Für $N \to \infty$ nehmen wir $h_{\max} \to 0$ mit $h_{\max} := \max_{i=0,\ldots,N-1} |x_{i+1} - x_i|$ an. Das Verfahren heißt konvergent, wenn für festes $x_{\text{end}} = x_N$ gilt

$$\lim_{N \to \infty} e_N = 0.$$

Das Verfahren ist konvergent von (mindestens) Ordnung p, falls

$$e_N = \mathcal{O}(h_{\max}^p).$$

Bezüglich des Zusammenhangs zwischen Konsistenz und Konvergenz beweisen wir einen Satz. O.E.d.A. verwenden wir dabei die Maximumnorm als Vektornorm. Desweiteren benötigen wir noch eine Hilfsaussage.

Lemma 2.2 *Erfüllt eine Folge* $(r_i)_{i\in\mathbb{N}} \subset \mathbb{R}^n$ *die Abschätzung*

$$\|r_{i+1}\| \leq (1+\delta)\|r_i\| + \beta \qquad \text{für } i = 0, 1, 2, \ldots$$

mit Konstanten $\beta \geq 0$ *und* $\delta > 0$*, dann gilt für* $k \in \mathbb{N}_0$

$$\|r_k\| \leq \mathrm{e}^{k\delta}\|r_0\| + \frac{\mathrm{e}^{k\delta}-1}{\delta}\,\beta\ ,$$

wobei die Vektornorm beliebig ist.

Beweis:

Verwendet wird Induktion über k. Der Induktionsanfang mit $k = 1$ lautet

$$\|r_1\| \leq (1+\delta)\|r_0\| + \beta \leq \mathrm{e}^{\delta\cdot 1}\|r_0\| + \frac{\mathrm{e}^{\delta\cdot 1}-1}{\delta}\,\beta$$

wegen $1+\delta \leq \mathrm{e}^{\delta}$ und somit auch $1 \leq (\mathrm{e}^{\delta}-1)/\delta$. Zum Induktionsschluss sei die Aussage für ein $k \geq 1$ erfüllt. Es folgt wieder mit $1+\delta \leq \mathrm{e}^{\delta}$

$$\begin{aligned}
\|r_{k+1}\| &\leq (1+\delta)\|r_k\| + \beta \leq (1+\delta)\mathrm{e}^{k\delta}\|r_0\| + (1+\delta)\tfrac{\mathrm{e}^{k\delta}-1}{\delta}\,\beta + \beta \\
&\leq \mathrm{e}^{\delta}\mathrm{e}^{k\delta}\|r_0\| + \tfrac{(1+\delta)\mathrm{e}^{k\delta}-(1+\delta)+\delta}{\delta}\,\beta \\
&\leq \mathrm{e}^{(k+1)\delta}\|r_0\| + \tfrac{\mathrm{e}^{(k+1)\delta}-1}{\delta}\,\beta
\end{aligned}$$

und damit ist die Aussage für $k+1$ gezeigt. □

Nun ergibt sich das Hauptresultat dieses Unterabschnitts.

Satz 2.1 (Konvergenz von Einschrittverfahren)
Gegeben sei ein AWP $y' = f(x,y)$, $y(x_0) = y_0$ *mit der exakten Lösung* $y(x)$ *für* $x \in [x_0, x_{\mathrm{end}}]$*. Die Inkrementfunktion* Φ *sei stetig auf*

$$G := \big\{(x,y,h)\ :\ x \in [x_0, x_{\mathrm{end}}],\ \|y - y(x)\| \leq \gamma,\ 0 < h \leq \bar{h}\big\}$$

mit Konstanten $\gamma, \bar{h} > 0$ und es gelte die Lipschitz-Bedingung

$$\|\Phi(x, y_1, h) - \Phi(x, y_2, h)\| \leq K \cdot \|y_1 - y_2\| \tag{2.11}$$

für alle $(x, y_i, h) \in G$, $i = 1, 2$ mit einer Konstanten $K > 0$. Das Einschrittverfahren sei konsistent von Ordnung p im Sinne von

$$\|\tau(x, y(x), h)\| = \|\Delta(x, y(x), h) - \Phi(x, y(x), h)\| \leq C \cdot h^p \tag{2.12}$$

für alle $x \in [x_0, x_{\text{end}}]$ und $0 < h \leq \bar{h}$ mit einer Konstanten $C > 0$. Verwendet wird konstante Schrittweite $h_N := \frac{x_{\text{end}} - x_0}{N}$, d.h. Gitterpunkte sind $x_i := x_0 + i h_N$ für $i = 0, 1, \ldots, N$. Dann gibt es ein $\hat{h} \in (0, \bar{h}]$, so dass die globalen Diskretisierungsfehler beschränkt sind durch

$$\|e_i\| \leq h_N^p \cdot C \cdot \frac{\exp(K(x_{\text{end}} - x_0)) - 1}{K} \qquad \text{für } i = 1, \ldots, N \tag{2.13}$$

und für alle $h_N \leq \hat{h}$.

Beweis:

Zu $y, z \in \mathbb{R}^n$ definieren wir für $\gamma > 0$ die Hilfsgröße $\tilde{y}(y, z) \in \mathbb{R}^n$ komponentenweise für $j = 1, \ldots, n$ ($\tilde{y} = (\tilde{y}_1, \ldots, \tilde{y}_n)^\top$) durch

$$\tilde{y}_j := \begin{cases} z_j + \gamma & \text{falls } \; y_j \geq z_j + \gamma, \\ z_j - \gamma & \text{falls } \; y_j \leq z_j - \gamma, \\ y_j & \text{sonst.} \end{cases}$$

Damit konstruieren wir die Hilfsfunktion

$$\tilde{\Phi}(x, y, h) := \Phi(x, \tilde{y}(y, y(x)), h).$$

Offensichtlich ist $\tilde{\Phi}$ stetig auf

$$\tilde{G} := \left\{(x, y, h) \; : \; x \in [x_0, x_{\text{end}}], \; y \in \mathbb{R}^n, \; h \leq \bar{h}\right\}$$

und erfüllt wegen (2.11) die Bedingung

$$\left\| \tilde{\Phi}(x, y_1, h) - \tilde{\Phi}(x, y_2, h) \right\| \leq K \cdot \|y_1 - y_2\| \tag{2.14}$$

für alle $(x, y_i, h) \in \tilde{G}$, $i = 1, 2$. Wegen $\tilde{\Phi}(x, y(x), h) = \Phi(x, y(x), h)$ für alle $x \in [x_0, x_{\text{end}}]$ sowie $0 < h \leq \bar{h}$ und (2.12) gilt auch

$$\left\| \Delta(x, y(x), h) - \tilde{\Phi}(x, y(x), h) \right\| \leq C \cdot h^p \tag{2.15}$$

für $x \in [x_0, x_{\text{end}}]$ und $0 < h \leq \bar{h}$.

Das zu $\tilde{\Phi}$ gehörige Einschrittverfahren liefert die Näherungslösungen $\tilde{u}_i$ über die Formel

$$\tilde{u}_{i+1} = \tilde{u}_i + h_N \tilde{\Phi}(x_i, \tilde{u}_i, h_N)$$

mit Anfangswert $\tilde{u}_0 = y_0$. Mit der Funktion (2.9) folgt direkt

$$y_{i+1} = y_i + h_N \Delta(x_i, y_i, h_N)$$

für die exakten Lösungswerte $y_i = y(x_i)$. Durch Subtraktion erhalten wir für die Fehler $\tilde{e}_i := \tilde{u}_i - y_i$ die Rekursionsformel

$$\begin{aligned} \tilde{e}_{i+1} &= \tilde{e}_i + h_N \left[\tilde{\Phi}(x_i, \tilde{u}_i, h_N) - \Delta(x_i, y_i, h_N) \right] \\ &= \tilde{e}_i + h_N \left[\tilde{\Phi}(x_i, \tilde{u}_i, h_N) - \tilde{\Phi}(x_i, y_i, h_N) \right] \\ &\quad + h_N \left[\tilde{\Phi}(x_i, y_i, h_N) - \Delta(x_i, y_i, h_N) \right]. \end{aligned}$$

Es ergibt sich aus (2.14) und (2.15)

$$\begin{aligned} \left\| \tilde{\Phi}(x_i, \tilde{u}_i, h_N) - \tilde{\Phi}(x_i, y_i, h_N) \right\| &\leq K \cdot \|\tilde{u}_i - y_i\| = K \cdot \|\tilde{e}_i\|, \\ \left\| \Delta(x_i, y_i, h_N) - \tilde{\Phi}(x_i, y_i, h_N) \right\| &\leq C \cdot h_N^p. \end{aligned}$$

Wir erhalten die Abschätzung

$$\|\tilde{e}_{i+1}\| \leq (1 + h_N K)\|\tilde{e}_i\| + C \cdot h_N^{p+1}$$

für $i = 0, 1, \ldots, N-1$. Lemma 2.2 liefert nun mit $\tilde{e}_0 = \tilde{u}_0 - y_0 = 0$ die Abschätzung

$$\|\tilde{e}_i\| \leq C \cdot h_N^p \cdot \frac{\exp(K i h_N) - 1}{K} \qquad \text{für alle } i = 0, 1, \ldots, N.$$

Damit folgt auch

$$\|\tilde{e}_i\| \leq C \cdot h_N^p \cdot \frac{\exp(K(x_{\text{end}} - x_0)) - 1}{K} \qquad \text{für alle } i = 0, 1, \ldots, N$$

wegen $ih_N \leq x_{\text{end}} - x_0$. Wir erkennen, dass ein $\hat{h} \leq \bar{h}$ existiert, so dass $\|\tilde{e}_i\| \leq \gamma$ für alle $i = 0, 1, \ldots, N$ gilt, falls $h_N \leq \hat{h}$. Damit erhalten wir $\tilde{\Phi}_i(x_i, \tilde{u}_i, h_N) = \Phi(x_i, \tilde{u}_i, h_N)$ für $h_N \leq \hat{h}$ aus der Definition der Hilfsfunktion $\tilde{\Phi}$. In diesem Fall sind die Näherungen $\tilde{u}_i$ aus dem von $\tilde{\Phi}$ erzeugten Einschrittverfahren identisch mit den Näherungen u_i aus dem von Φ festgelegten Einschrittverfahren. Somit gilt $\tilde{e}_i = e_i$ für $i = 0, 1, \ldots, N$ mit $e_i := u_i - y_i$. Dadurch folgt die Abschätzung (2.13). □

Bemerkungen:

- Die Lipschitz-Bedingung (2.11) ist bei den üblichen Verfahren erfüllt, falls die rechte Seite f der Dgl. lokal Lipschitz-stetig in y auf G ist.
- Die Konsistenzbedingung (2.12) stellt eine leichte Verschärfung der Bedingung aus Def. 2.2 dar und ist bei allen gängigen Verfahren gegeben.
- Die Konvergenzaussage (2.13) in Satz 2.1 betrifft nicht nur den Endwert bei $x = x_{\text{end}}$ sondern alle Gitterpunkte $x_1, \ldots, x_N$.
- Die Konvergenzaussage gilt auch bei nichtkonstanten Schrittweiten, d.h. einem Gitter $x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = x_{\text{end}}$ und $h_i := x_{i+1} - x_i$. Im Grenzfall muss dann $h_{\max} \to 0$ gelten mit der maximalen Schrittweite $h_{\max} := \max\{h_0, h_1, \ldots, h_{N-1}\}$.
- In die Konvergenzanalyse können auch Fehler $e_0 = u_0 - y_0 \neq 0$ in den Anfangswerten sowie Rundungsfehler einbezogen werden.

Satz 2.1 zeigt, dass die Konsistenz hinreichend für die Konvergenz ist. Zudem stimmt die Konsistenzordnung mit der Konvergenzordnung überein. Die Konsistenz kann durch eine Untersuchung der Inkrementfunktion Φ des Einschrittverfahrens nachgewiesen werden. Umgekehrt gibt es konvergente Methoden, die nicht konsistent sind. Konsistenz ist somit nicht notwendig für Konvergenz. Jedoch werden inkonsistente Verfahren nicht in der Praxis eingesetzt.

2.4 Runge-Kutta-Verfahren

Der wichtigste Typ von Einschrittverfahren sind die Runge-Kutta-Verfahren. Die Idee besteht darin, das Integral in (2.3) durch eine Quadraturformel mit den Knoten $c_1, \ldots, c_s \in [0,1]$ und (äußeren) Gewichten $b_1, \ldots, b_s \in \mathbb{R}$ zu ersetzen. O.E.d.A. sei $c_1 \leq c_2 \leq \cdots \leq c_s$. Es folgt eine endliche Summe

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(x_0 + c_i h, y(x_0 + c_i h)).$$

Das Problem dabei ist, dass die Zwischenwerte $y(x_0 + c_i h)$ a priori unbekannt sind. Wir erhalten Näherungen für die Zwischenwerte wieder aus der Integralgleichung (2.2), d.h.

$$y(x_0 + c_i h) = y_0 + h \int_0^{c_i} f(x_0 + sh, y(x_0 + sh)) \, \mathrm{d}s.$$

Die beteiligten Integrale werden durch Quadraturformeln ersetzt. Um die Einführung neuer Unbekannten zu vermeiden, dürfen nur die gleichen Knoten $c_1, \ldots, c_s$ wie zuvor verwendet werden. Jedoch entstehen neue (innere) Gewichte. Es folgen die Näherungen

$$z_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, z_j) \tag{2.16}$$

für $i = 1, \ldots, s$. Die letztlich gesuchte Näherung wird zu

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(x_0 + c_i h, z_i).$$

Die Gleichungen (2.16) stellen ein nichtlineares Gleichungssystem (aus algebraischen Gleichungen) für die Unbekannten $z_1, \ldots, z_s$ dar. Wenn die Zwischenwerte bestimmt wurden, dann können wir die Näherung y_1 direkt aus s Auswertungen der Funktion f erhalten. Man nennt s auch die Stufenzahl des Verfahrens.

Betreffend (2.16) ist eine natürliche Forderung, dass eine konstante Funktion $f \equiv 1$ ($y(x_0 + c_i h) = y_0 + c_i h$) exakt reproduziert. Wir erhalten die

Bedingungen

$$c_i = \sum_{j=1}^{s} a_{ij} \qquad \text{für jedes } i = 1, \ldots, s. \tag{2.17}$$

Diese Gleichung bedeutet, dass die Summe der Gewichte gleich der (relativen) Länge des Teilintervalls sein muss.

Eine Runge-Kutta-Methode ist eindeutig durch seine Koeffizienten festgelegt. Die Koeffizienten können in einem sogenannten Butcher-Tableau angeordnet werden:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \qquad \text{oder kurz} \qquad \begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

mit $c \in \mathbb{R}^s$, $b \in \mathbb{R}^s$, $A \in \mathbb{R}^{s \times s}$.

Beispiele: Verfahren aus Abschnitt 2.2

(a): expl. Euler-Verfahren, (b): impl. Euler-Verfahren, (c): Trapezregel, (d): Collatz-Verfahren:

(a) $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$ (b) $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$ (c) $\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$ (d) $\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$

Beispiel: Gauß-Runge-Kutta-Verfahren

Wir verwenden die Gauß-Legendre-Quadratur für die Knoten c_i und die Gewichte b_i. Diese Quadraturformel besitzt die Ordnung $2s$, d.h. es gilt

$$\sum_{i=1}^{s} b_i p(c_i) = \int_0^1 p(x)\,\mathrm{d}x \qquad \text{für alle } \; p \in \mathbb{P}_{2s-1}$$

($\mathbb{P}_m$: Polynome bis Grad m). Die Gewichte a_{ij} bestimmen wir für jedes $i = 1, \ldots, s$ derart, dass

$$\sum_{j=1}^{s} a_{ij} p(c_j) = \int_0^{c_i} p(x)\,\mathrm{d}x \qquad \text{für alle } \; p \in \mathbb{P}_{s-1}.$$

Im einfachen Fall $s = 1$ folgt direkt $c_1 = \frac{1}{2}$, $b_1 = 1$ und $a_{11} = \frac{1}{2}$. Es entsteht das Runge-Kutta-Verfahren

$$\begin{aligned} z_1 &= y_0 + \tfrac{h}{2} f(x_0 + \tfrac{h}{2}, z_1), \\ y_1 &= y_0 + h f(x_0 + \tfrac{h}{2}, z_1). \end{aligned} \tag{2.18}$$

Dieser Ansatz entspricht der Mittelpunktregel (2.5), wobei die Näherung $z_1 \doteq y(x_0 + \frac{1}{2}h)$ durch das implizite Euler-Verfahren bestimmt wird.

Das Butcher-Tableau schreibt sich im Fall $s = 2$:

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Wenn die Matrix $A = (a_{ij})$ vollbesetzt ist, dann ist das Runge-Kutta-Verfahren implizit. Ein nichtlineares Gleichungssystem (2.16) aus $s \cdot n$ algebraischen Gleichungen muss dann gelöst werden. Im Gegensatz dazu möchten wir nun explizite Methoden erhalten. Die entsprechende Bedingung lautet $a_{ij} = 0$ für $i \leq j$. Dadurch wir A zu einer strikten unteren Dreiecksmatrix. Das Butcher-Tableau besitz dann die Gestalt:

$$\begin{array}{c|ccccc} 0 & 0 & 0 & \cdots & \cdots & 0 \\ c_2 & a_{21} & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 & 0 \\ c_s & a_{s1} & \cdots & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

Insbesondere folgt $c_1 = 0$ aus der Bedingung (2.17) und damit $z_1 = y_0$. Nun ergeben sich die Zwischenwerte sukzessive aus

$$z_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} f(x_0 + c_j h, z_j) \quad \text{für} \;\; i = 1, \ldots, s.$$

Der Rechenaufwand einer expliziten Runge-Kutta-Methode besteht somit nur in s Auswertungen der rechten Seite f. Man kann ein explizites Verfahren daher als eine sukzessive Extrapolation mit den gegebenen Zwischenwerten interpretieren. Implizite Verfahren entsprechen dann einer Interpolation mit den Zwischenwerten.

Beispiele: Einige bekannte explizite Runge-Kutta-Verfahren

Heun-Verfahren (links), Kutta-Simpson-Verfahren (mitte) und klassisches Runge-Kutta-Verfahren (rechts):

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

Eine äquivalente Notation für Runge-Kutta-Schemata entsteht durch die Definition von Inkrementen k_i mittels

$$k_i = f(x_0 + c_i h, z_i) = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right) \tag{2.19}$$

für $i = 1, \ldots, s$. Ein Runge-Kutta-Verfahren besitzt dann die Gestalt

$$\begin{aligned} k_i &= f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij} k_j\right), \quad i = 1, \ldots, s\,, \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{2.20}$$

Die Inkremente k_i sind nun a priori unbekannt.

Ordnungsbedingungen

Ein Runge-Kutta-Methode ist durch ihre Koeffizienten c_i, b_i, a_{ij} eindeutig bestimmt. Wir leiten Bedingungen an diese Koeffizienten her, welche die Konsistenz des Einschrittverfahrens mit Ordnung p liefern. Wir betrachten eine autonome skalare Dgl. $y' = f(y)$. Es folgt

$$\begin{aligned} y'' &= f'y' = f'f, \\ y''' &= f''y'f + f'f'y' = f''f^2 + (f')^2 f. \end{aligned}$$

Taylor-Entwicklung der exakten Lösung führt auf

$$\begin{aligned} y(x_0 + h) &= y(x_0) + hy'(x_0) + \tfrac{h^2}{2} y''(x_0) + \tfrac{h^3}{6} y'''(x_0) + \mathcal{O}(h^4) \\ &= y_0 + hf(y_0) + \tfrac{h^2}{2} f'(y_0) f(y_0) \\ &\quad + \tfrac{h^3}{6}\left[f''(y_0) f(y_0)^2 + f'(y_0)^2 f(y_0)\right] + \mathcal{O}(h^4). \end{aligned}$$

Im folgenden benutzen wir die Abkürzungen $f = f(y_0)$, $f' = f'(y_0)$, etc. Die Inkremente k_i aus (2.20) hängen von der Wahl der Schrittweite h ab. Wir nehmen an, dass diese Inkremente beschränkt sind in einer Umgebung von $h = 0$. Dies ist beispielsweise gesichert, wenn eine beschränkte rechte Seite f auftritt.

Die Runge-Kutta-Methode erfülle die fundamentale Bedingung (2.17). Eine Taylor-Entwicklung der Funktion f bezüglich der Inkremente (2.19) liefert

für $i = 1, \ldots, s$

$$\begin{aligned}
k_i &= f + f'h\left(\sum_{j=1}^{s} a_{ij}k_j\right) + \tfrac{1}{2}f''h^2\left(\sum_{j=1}^{s} a_{ij}k_j\right)^2 + \mathcal{O}(h^3) \\
&= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'h\left(\sum_{\ell=1}^{s} a_{j\ell}k_\ell\right) + \mathcal{O}(h^2)\right)\right) \\
&\quad + \tfrac{1}{2}f''h^2\left(\sum_{j=1}^{s} a_{ij}\left(f + \mathcal{O}(h)\right)\right)^2 + \mathcal{O}(h^3) \\
&= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'h\left(\sum_{\ell=1}^{s} a_{j\ell}\left(f + \mathcal{O}(h)\right)\right) + \mathcal{O}(h^2)\right)\right) \\
&\quad + \tfrac{1}{2}f''h^2\left(fc_i + \mathcal{O}(h)\right)^2 + \mathcal{O}(h^3) \\
&= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'fhc_j + \mathcal{O}(h^2)\right)\right) + \tfrac{1}{2}f''f^2h^2c_i^2 + \mathcal{O}(h^3) \\
&= f + hf'fc_i + h^2(f')^2 f\left(\sum_{j=1}^{s} a_{ij}c_j\right) + \tfrac{1}{2}h^2f''f^2c_i^2 + \mathcal{O}(h^3).
\end{aligned}$$

Die Näherung aus dem Runge-Kutta-Verfahren resultiert zu

$$\begin{aligned}
y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i \\
&= y_0 + hf\left(\sum_{i=1}^{s} b_i\right) + h^2 f'f\left(\sum_{i=1}^{s} b_i c_i\right) + h^3(f')^2 f\left(\sum_{i,j=1}^{s} b_i a_{ij} c_j\right) \\
&\quad + \tfrac{1}{2}h^3 f''f^2\left(\sum_{i=1}^{s} b_i c_i^2\right) + \mathcal{O}(h^4).
\end{aligned}$$

Ein Vergleich mit der Taylor-Entwicklung der exakten Lösung zeigt die Bedingungen für Konsistenz bis Ordnung $p = 3$. Die Konsistenzbedingungen bis Ordnung $p = 4$ sind ebenfalls dargestellt:

$p=1:$	$\sum_{i=1}^{s} b_i = 1$
$p=2:$	$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$
$p=3:$	$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}$

$p=4:$	$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_i c_j = \frac{1}{8}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12}$
	$\sum_{i,j,\ell=1}^{s} b_i a_{ij} a_{j\ell} c_\ell = \frac{1}{24}$

Die Konsistenzbedingungen können mit dem Ansatz über Taylor-Entwicklungen bis zu einer beliebigen Ordnung p hergeleitet werden. Im Fall von expliziten Runge-Kutta-Verfahren brauchen in den Summen nur die Koeffizienten ungleich null aufgeführt zu werden. Zu einer gewünschten Konsistenzordnung p möchten wir ein Runge-Kutta-Verfahren mit möglichst kleiner Stufenzahl s erhalten. Bei impliziten Methoden kann mit s Stufen die maximale Ordnung $p = 2s$ erreicht werden, welche dann bei den Gauß-Runge-Kutta-Schemata auftritt. Bei den explizten Methoden gibt Tabelle 1 eine Information.

Stufenzahl s	1	2	3	4	5	6	7	8	9	10	11	$\cdots$	17
maximale Ordnung p	1	2	3	4	4	5	6	6	7	7	8	$\cdots$	10

Ordnung p	1	2	3	4	5	6	7	8
minimale Stufenzahl s	1	2	3	4	6	7	9	11
Anzahl Ordnungsbedingungen	1	2	4	8	17	37	85	200

Tabelle 1: Ordnung und Stufenzahl bei expliziten Runge-Kutta-Verfahren.

2.5 Schrittweitensteuerung

In einer numerischen Integration werden die Näherungen $y_k \doteq y(x_k)$ sukzessive durch eine numerische Methode berechnet. Wir möchten, dass die

Schrittweiten $h_k := x_{k+1} - x_k$ automatisch bestimmt werden, so dass der entstehende Fehler in der Methode hinreichend klein bleibt.

Seien $y = (y_1, \ldots, y_n)^\top$ die Komponenten der Lösung. Wir nehmen an, dass das numerische Verfahren die Konsistenzordnung p besitzt, d.h. die Näherung $y^h \doteq y(x_0 + h)$ erfüllt die Bedingung

$$y_i^h - y_i(x_0 + h) = \mathcal{O}(h^{p+1}) = C_i h^{p+1} + \mathcal{O}(h^{p+2}) \tag{2.21}$$

mit Konstanten $C_i \neq 0$ für jede Komponente. Ein ähnliches numerisches Verfahren wird zur Berechnung der Näherung $\hat{y}^h$ mit einer Ordnung höher eingesetzt, d.h.

$$\hat{y}_i^h - y_i(x_0 + h) = \mathcal{O}(h^{p+2}). \tag{2.22}$$

Bei Runge-Kutta-Methoden werden typischerweise eingebettete Verfahren eingesetzt. Bei Mehrschrittverfahren existieren verschiedene Möglichkeiten. Desweiteren kann Richardson-Extrapolation sowohl bei Einschritt- als auch Mehrschrittverfahren eingesetzt werden.

Wir möchten den Fehler $y^h - y(x_0+h)$ im Verfahren der niedrigeren Ordnung schätzen. Die Bedingungen (2.21) und (2.22) liefern

$$y_i^h - y_i(x_0 + h) = y_i^h - \hat{y}_i^h - (y_i(x_0 + h) - \hat{y}_i^h) = y_i^h - \hat{y}_i^h + \mathcal{O}(h^{p+2}). \tag{2.23}$$

Daher stellt $\hat{y}^h - y^h$ einen Schätzer für den lokalen Fehler, welcher Größenordnung $p + 1$ hat, dar. Aus (2.21) und (2.23) folgt

$$y_i^h - \hat{y}_i^h = C_i h^{p+1} + \mathcal{O}(h^{p+2}). \tag{2.24}$$

Wir nehmen an, dass wir bereits einen Integrationsschritt mit der Schrittweite h_{used} durchgeführt haben. Nun möchten wir eine geeignete Schrittweite h_{opt} schätzen, um den Integrationsschritt zu wiederholen. Die Eigenschaften (2.21) und (2.24) implizieren ungefähr

$$\begin{aligned} y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}} &\doteq C_i h_{\text{used}}^{p+1}, \\ y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}}) &\doteq C_i h_{\text{opt}}^{p+1}. \end{aligned}$$

Elimination der Konstanten C_i liefert

$$\frac{|y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}})|}{|y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}}|} = \left(\frac{h_{\text{opt}}}{h_{\text{used}}}\right)^{p+1}. \tag{2.25}$$

Der Fehlerschätzer zum durchgeführten Schritt lautet

$$\eta_i := |y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}}| \tag{2.26}$$

für $i = 1, \ldots, n$. Der Fehler zum neuen Schritt soll

$$|y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}})| = \text{TOL} \tag{2.27}$$

erfüllen mit einer absoluten Toleranz TOL > 0 in allen Komponenten. Wir möchten nicht, dass der Fehler kleiner als TOL ist, denn ein kleinerer Fehler bedeutet eine kleinere Schrittweite und dadurch einen höheren Rechenaufwand aufgrund eine größeren Schrittanzahl. Einsetzen von (2.26),(2.27) in Gleichung (2.25) führt auf

$$h_{\text{opt},i} = h_{\text{used}} \cdot \sqrt[p+1]{\frac{\text{TOL}}{\eta_i}},$$

wobei jede Kompomente eine eigene Schrittweite erzeugt. Die Länge des nächsten Schritts wird daher gewählt als

$$h_{\text{new}} = \delta \cdot \min_{i=1,\ldots,n} h_{\text{opt},i}$$

mit einem Sicherheitsfaktor δ, z.B. $\delta = 0.9$. Um oszillierende Schrittweiten zu verhindern, wird desweiteren gefordert

$$\sigma \; h_{\text{used}} \leq h_{\text{new}} \leq \theta \; h_{\text{used}}$$

mit $0 < \sigma < 1 < \theta$, z.B. $\sigma = \frac{1}{5}, \theta = 5$.

Falls $h_{\text{new}} < h_{\text{used}}$ gilt, dann wurde unsere Genauigkeitsforderung (2.27) verfehlt, da der Fehler größer ist. Somit wiederholen wir den Schritt mit h_{new} anstelle von h_{used}. Daraufhin wird jedoch wieder der Fehler geschätzt. Falls $h_{\text{new}} \geq h_{\text{used}}$ erfüllt ist, dann akzeptieren wir den Schritt, da der Fehler kleiner oder gleich der Genauigkeitsforderung (2.27) ist. Der nächste Schritt wird dann mit h_{new} als vorgeschlagene Schrittweite durchgeführt.

Oft wird die Toleranz relativ bezüglich der Größenordnung der Lösung vorgegeben. Sei RTOL > 0 eine relative Toleranz und ATOL > 0 eine absolute Toleranz, dann definieren wir

$$\text{TOL} = \text{ATOL} + \text{RTOL} \cdot |y_i^{h_{\text{used}}}|.$$

Der absolute Teil ATOL wird benötigt für den Fall $|y_i^{h_{\text{used}}}| \approx 0$. Typische Werte sind z.B. RTOL $= 10^{-3}$ und ATOL $= 10^{-6}$.

Die obige Verwendung des Betrags $|\cdot|$ entspricht der Maximumnorm als Vektornorm. Jedoch besitzt die Maximumnorm ein Defizit an Glattheit und verursacht dadurch manchmal Probleme in der Integration. Daher wir in der Praxis häufig die skalierte Norm

$$\text{ERR} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i^{h_{\text{used}}} - y_i^{h_{\text{used}}}}{\text{ATOL} + \text{RTOL}\cdot|y_i^{h_{\text{used}}}|}\right)^2} \tag{2.28}$$

eingesetzt, die man als gewichtete Euklidische Norm interpretieren kann. Man beachte, dass der Nenner in (2.28) immer positiv ist. Die Bedingung (2.27) entspricht nun ERR $= 1$. Die neue Schrittweite wird definiert als

$$h_{\text{new}} = \delta \cdot h_{\text{used}} \cdot \frac{1}{\sqrt[p+1]{\text{ERR}}}$$

mit einem Sicherheitsfaktor δ.

Die Schätzung des lokalen Fehlers erfolgt für das Verfahren mit Ordnung p, während das Ergebnis aus dem Verfahren der Ordnung $p+1$ nur zur Berechnung des Fehlerschätzers eingesetzt wird. Jedoch ist die Näherung aus der Methode mit Ordnung $p+1$ in den meisten Fällen genauer. Daher wird häufig die Näherung höherer Ordnung als Ausgabe des Integrationsschritts festgesetzt.

Der obige Ansatz kontrolliert den lokalen Fehler in jedem Integrationsschritt. Jedoch hätten wir gerne eine Schrittweitenbestimmung derart, dass der globale Fehler (2.10) eine Genauigkeitsschranke erfüllt. Leider existieren keine erfolgreichen Strategien zur Kontrolle des globalen Fehlers. Daher verwenden numerische Integrationsverfahren aus üblichen Softwarepaketen (z.B. MATLAB) nur Schrittweitensteuerungen auf Basis der lokalen Fehler.

Eingebettete Verfahren

Es verbleibt zwei numerische Verfahren zur Schätzung des lokalen Fehlers festzulegen. Im Fall von Runge-Kutta-Methoden werden eingebettete

Verfahren angewendet, da der zusätzliche Rechenaufwand für die zweite Näherung relativ klein ausfällt.

Das Butcher-Tableau eines eingebetteten Verfahrens lautet

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \\ \hline & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_s \end{array}$$

mit zwei Mengen b_i und $\hat{b}_i$ von Gewichten. Die entstehenden Näherungen sind

$$\begin{aligned} y^h &= y_0 + h(b_1 k_1 + \cdots + b_s k_s), \\ \hat{y}^h &= y_0 + h(\hat{b}_1 k_1 + \cdots + \hat{b}_s k_s). \end{aligned}$$

Wenn die Inkremente $k_1, \ldots, k_s$ zur Berechnung der Näherung y^h verfügbar sind, dann kann die zweite Näherung $\hat{y}^h$ ohne wesentlichen Mehraufwand bestimmt werden.

Im Fall von expliziten Runge-Kutta-Verfahren stellen die Runge-Kutta-Fehlberg Methoden eine Klasse eingebetteter Verfahren dar.

Beispiel: Runge-Kutta-Fehlberg 2(3)

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{4} & \frac{1}{4} & & & \\ \frac{27}{40} & -\frac{189}{800} & \frac{729}{800} & & \\ 1 & \frac{214}{891} & \frac{1}{33} & \frac{650}{891} & \\ \hline & \frac{214}{891} & \frac{1}{33} & \frac{650}{891} & 0 \\ \hline & \frac{533}{2106} & 0 & \frac{800}{1053} & -\frac{1}{78} \end{array}$$

Kapitel 3

Mehrschrittverfahren

In diesem Kapitel untersuchen wir Mehrschrittmethoden, d.h. mehrere alte Näherungen werden eingesetzt um eine neue Näherung zu konstruieren. Betrachtet wird wieder ein AWP $y' = f(x, y),\ y(x_0) = y_0$, siehe (1.1). Sind die Approximationen

$$(x_{i-k+1}, y_{i-k+1}), (x_{i-k+2}, y_{i-k+2}), \ldots, (x_{i-1}, y_{i-1}), (x_i, y_i) \tag{3.1}$$

für ein $k \geq 1$ gegeben, dann wird daraus eine neue Näherung (x_{i+1}, y_{i+1}) bestimmt. Im Gegensatz zu Einschrittverfahren reicht hier die Konsistenz alleine für die Konvergenz der Methoden nicht aus.

3.1 Methoden über numerischer Quadratur

Wir führen eine wichtige Klasse von Mehrschrittverfahren ein, deren Konstruktionsprinzip auf der Integralgleichung (2.2) beruht. Eine Polyninterpolation wird aufgestellt und das exakte Integral wird mit dem Integral des Polynoms approximiert. Wir wählen eine ganze Zahl $\ell \geq 1$ und erhalten für die exakte Lösung des AWPs (1.1) die Integralgleichung

$$\begin{aligned} y(x_{i+1}) &= y(x_{i-\ell+1}) + \int_{x_{i-\ell+1}}^{x_{i+1}} y'(s)\, \mathrm{d}s \\ &= y(x_{i-\ell+1}) + \int_{x_{i-\ell+1}}^{x_{i+1}} f(s, y(s))\, \mathrm{d}s. \end{aligned} \tag{3.2}$$

Nun approximieren wir den Integranden $f(x, y(x))$. Wir stellen das Polynom $p_{k,i} \in \mathbb{P}_{k-1}$ auf, welches die Stützpunkte

$$(x_j, f(x_j, y_j)) \qquad \text{für } \; j = i-k+1, i-k+2, \ldots, i-1, i$$

interpoliert. Dementsprechend gilt

$$p_{k,i}(x_j) = f(x_j, y_j) \qquad \text{für } \; j = i-k+1, i-k+2, \ldots, i-1, i.$$

Dieses Interpolationspolynom existiert und ist eindeutig. Mit der Lagrange-Basis

$$L_{i,j}(x) = \prod_{\nu=1, \nu \neq j}^{k} \frac{x - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}} \qquad \text{für } \; j = 1, \ldots, k \; ,$$

und $f_i := f(x_i, y_i)$ lautet das Polynom

$$p_{k,i}(x) = \sum_{j=1}^{k} f_{i-j+1} L_{i,j}(x).$$

Die Erwartung ist $p_{k,i}(x) \approx f(x, y(x))$ im betrachteten Gebiet. Daher ergibt sich als neue Näherung wegen (3.2)

$$y_{i+1} = y_{i-\ell+1} + \sum_{j=1}^{k} f_{i-j+1} \int_{x_{i-\ell+1}}^{x_{i+1}} L_{i,j}(s) \, \mathrm{d}s.$$

Da die Lagrange-Polynome gegeben sind, kann das Integral exakt ausgewertet werden.

In den meisten Fällen gilt $\ell \leq k$, d.h. das Intervall der Interpolation enthält das Intervall der Integration (zur linken Seite hin). Abb. 9 verdeutlicht diese Konstruktion. Wir erhalten ein explizites k-Schritt-Verfahren.

Im Fall von äquidistanten Gitterpunkten $x_i = x_0 + ih$ sind die Lagrange-

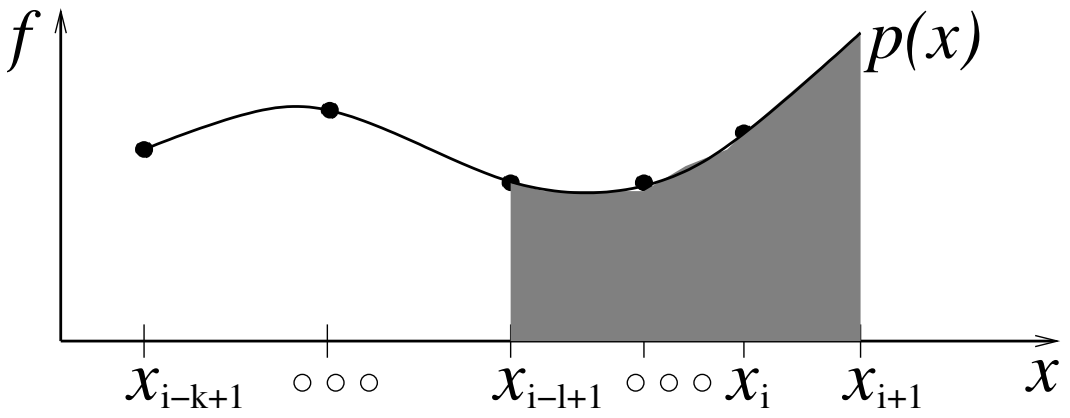


Abbildung 9: Konstruktion von Mehrschrittverfahren mittels Quadratur.

Polynome unabhängig vom Index i

$$\begin{aligned}\int_{x_{i-\ell+1}}^{x_{i+1}} L_{i,j}(s)\,\mathrm{d}s &= \int_{x_{i-\ell+1}}^{x_{i+1}} \prod_{\nu\neq j} \frac{s - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}}\,\mathrm{d}s \\ &= h\int_{1-\ell}^{1} \prod_{\nu\neq j} \frac{x_0 + (i+u)h - (x_0 + (i-\nu+1)h)}{x_0 + (i-j+1)h - (x_0 + (i-\nu+1)h)}\,\mathrm{d}u \\ &= h\int_{1-\ell}^{1} \prod_{\nu\neq j} \frac{u+\nu-1}{\nu-j}\,\mathrm{d}u.\end{aligned}$$

Es folgt die Methode

$$y_{i+1} = y_{i-\ell+1} + h\sum_{j=1}^{k} \beta_j f(x_{i-j+1}, y_{i-j+1})$$

mit den konstanten Koeffizienten

$$\beta_j := \int_{1-\ell}^{1} \prod_{\nu=1,\nu\neq j}^{k} \frac{u+\nu-1}{\nu-j}\,\mathrm{d}u \qquad \text{für } j = 1,\ldots,k.$$

Ein implizites Mehrschrittverfahren entsteht, wenn die unbekannte neue Näherung (x_{i+1}, y_{i+1}) in die Interpolation einbezogen wird. Sei $q_{k,i} \in \mathbb{P}_k$ das Interpolationspolynom zu den Stützpunkten

$$(x_j, f(x_j, y_j)) \qquad \text{für } j = i-k+1, i-k+2, \ldots, i-1, i, i+1.$$

Es folgt

$$q_{k,i}(x_j) = f(x_j, y_j) \qquad \text{fürr } \; j = i-k+1, i-k+2, \ldots, i-1, i, i+1.$$

Die zugehörigen Lagrange-Polynome lauten

$$L^*_{i,j}(x) = \prod_{\nu=0, \nu\neq j}^{k} \frac{x - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}} \qquad \text{für } \; j = 0, 1, \ldots, k$$

und somit

$$q_{k,i}(x) = \sum_{j=0}^{k} f_{i-j+1} L^*_{i,j}(x).$$

Wir schreiben $q_{k,i}(x; y_{i+1})$ um zu betonen, dass dieses Polynom noch von der neuen Näherung abhängt, welche a priori unbekannt ist. Es ergibt sich

$$y_{i+1} = y_{i-\ell+1} + \int_{x_{i-\ell+1}}^{x_{i+1}} q_{k,i}(s; y_{i+1}) \, \mathrm{d}s.$$

Diese Formel stellt ein nichtlineares Gleichungssystem für die Unbekannten y_{i+1} dar. Daher liefert dieser Ansatz eine implizite Methode mit k Schritten.

Im Fall von äquidistanten Schrittweiten lautet das Verfahren

$$y_{i+1} = y_{i-\ell+1} + h \sum_{j=0}^{k} \beta^*_j f(x_{i-j+1}, y_{i-j+1}) \tag{3.3}$$

mit den konstanten Koeffizienten

$$\beta^*_j := \int_{1-\ell}^{1} \prod_{\nu=0, \nu\neq j}^{k} \frac{u+\nu-1}{\nu - j} \, \mathrm{d}u \qquad \text{für } \; j = 0, 1, \ldots, k.$$

Äquivalent können wir schreiben

$$y_{i+1} - h\beta^*_0 f(x_{i+1}, y_{i+1}) = y_{i-\ell+1} + h \sum_{j=1}^{k} \beta^*_j f(x_{i-j+1}, y_{i-j+1}),$$

	β_1	β_2	β_3	β_4
$k=1$	1			
$k=2$	$\frac{3}{2}$	$-\frac{1}{2}$		
$k=3$	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
$k=4$	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

	β_0^*	β_1^*	β_2^*	β_3^*	β_4^*
$k=1$	$\frac{1}{2}$	$\frac{1}{2}$			
$k=2$	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$		
$k=3$	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$	
$k=4$	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$

Tabelle 2: Koeffizienten in Adams-Bashforth (links) und Adams-Moulton (rechts).

wobei die rechte Seite die bekannten Daten und die linke Seite die unbekannte neue Näherung enthält.

Adams-Methoden

Eine beliebte Klasse von Mehrschrittverfahren sind die Adams-Methoden, die aus der Wahl $\ell = 1$ in (3.2) entstehen. Daher wird die Integration nur im Teilintervall $[x_i, x_{i+1}]$ durchgeführt.

Die expliziten Methoden heißen Adams-Bashforth-Verfahren. Das k-Schritt-Verfahren lautet

$$y_{i+1} = y_i + h \sum_{j=1}^{k} \beta_j f(x_{i-j+1}, y_{i-j+1}) \tag{3.4}$$

im Fall von äquidistanten Schrittweiten. Die impliziten Methoden heißen Adams-Moulton-Verfahren. Das k-Schritt-Verfahren besitzt die Formel

$$y_{i+1} = y_i + h \sum_{j=0}^{k} \beta_j^* f(x_{i-j+1}, y_{i-j+1}). \tag{3.5}$$

Tabelle 2 zeigt die Koeffizienten dieser Methode für $k = 1, 2, 3, 4$. Die Einschritt-Adams-Bashforth-Methode ist gerade das explizite Euler-Verfahren, während die Einschritt-Adams-Moulton-Methode die Trapezregel ergibt.

Nyström-Verfahren und Milne-Verfahren

Wir erhalten eine weitere bedeutende Klasse von Mehrschrittverfahren aus der Wahl $\ell = 2$ in (3.2). Die zugehörigen expliziten Verfahren heißen dann

Nyström-Methoden. Beispielsweise führt die Wahl $k = 1$ (jetzt ausnahmsweise $k < \ell$) auf die explizite Mittelpunktregel

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i), \tag{3.6}$$

welche ein Zweischrittverfahren darstellt. Die entsprechenden impliziten Verfahren heißen Milne-Methoden. Für äquidistante Schrittweiten liefert der Fall $k = 1$ wieder die explizite Mittelpunktregel, da der Term mit f_{i+1} herausfällt. Die Wahl $k = 2$ ergibt die Milne-Simpson-Regel

$$y_{i+1} = y_{i-1} + h\tfrac{1}{3}\left(f(x_{i-1}, y_{i-1}) + 4f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right),$$

d.h. ein implizites Verfahren. Diese Methode entspricht der Simpson-Regel in der numerischen Quadratur.

Die Fälle $\ell \geq 3$ in (3.2) sind für die Praxis irrelevant. Zudem ist die Anzahl der Schritte (d.h. $\max\{k, \ell\}$) üblicherweise kleiner 15 und häufig nicht größer als 5 in Softwarebibliotheken.

3.2 Methoden über numerische Differentiation

Wir führen einen anderen Typ von impliziten Mehrschrittverfahren ein, welcher durch numerische Differentiation entsteht. Ist eine Differentialgleichung $y' = f(x, y)$ gegeben, dann können wir die Ableitung auf der linken Seite durch eine Differenzenformel ersetzen, was einer numerischen Differentiation entspricht. Der übliche Differenzenquotient lautet

$$y'(x_0 + h) = \tfrac{1}{h}\left[y(x_0 + h) - y(x_0)\right] + \mathcal{O}(h).$$

Zusammen mit $y'(x_0 + h) = f(x_0 + h, y(x_0 + h))$ erhalten wir als numerische Methode

$$y_1 = y_0 + hf(x_0 + h, y_1),$$

d.h. das implizite Euler-Verfahren

Dieser Ansatz kann zu einem k-Schritt-Verfahren verallgemeinert werden wie folgt: Mit den gegebenen Daten $(x_{i-k+\ell}, y_{i-k+\ell})$ für $\ell = 1, \ldots, k$ stellen wir das Interpolationspolynom $p \in \mathbb{P}_k$ mit

$$p(x_{i-k+\ell}) = y_{i-k+\ell} \qquad \text{für } \ \ell = 1, \ldots, k, k+1$$

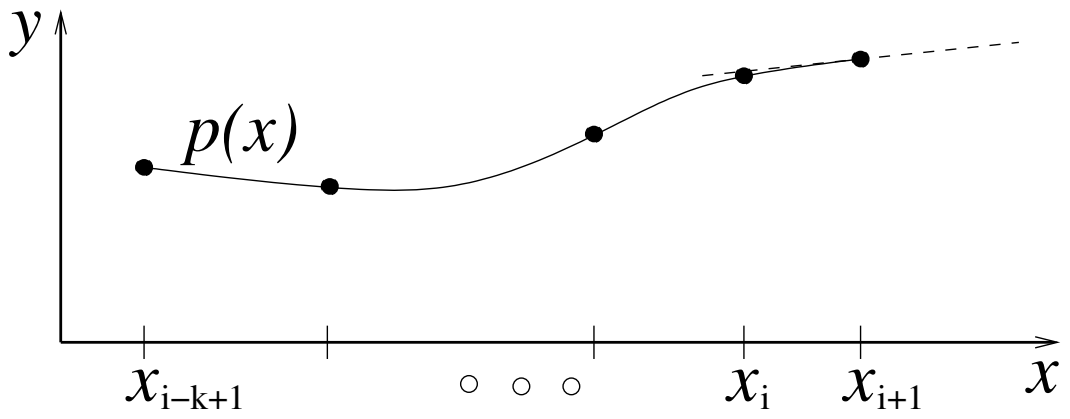


Abbildung 10: Konstruktion von Mehrschrittverfahren durch numerische Differentiation.

auf. Darin ist der unbekannte Wert y_{i+1} eingeschlossen, wodurch die Methode implizit ist. Die Strategie ist in Abb. 10 dargestellt. Die Ableitung p' kann man als Approximation der Ableitung y' interpretieren. Der unbekannte Wert wird bestimmt durch die Bedingung

$$p'(x_{i+1}) = f(x_{i+1}, y_{i+1}), \tag{3.7}$$

d.h. man verlangt, dass das Interpolationspolynom nur an der Stelle x_{i+1} die Differentialgleichung erfüllt. Die entstehenden Schemata nennt man BDF-Methoden (backward differentiation formulas).

Das Interpolationspolynom besitzt die Darstellung

$$p(x) = \sum_{j=0}^{k} y_{i+1-j} L_j(x)$$

mit den Lagrange-Polynomen

$$L_j(x) = \prod_{\nu=0, \nu \neq j}^{k} \frac{x - x_{i+1-\nu}}{x_{i+1-j} - x_{i+1-\nu}}.$$

Wir erhalten

$$p'(x_{i+1}) = \sum_{j=0}^{k} y_{i+1-j} L_j'(x_{i+1}) = f(x_{i+1}, y_{i+1}).$$

	α_0	α_1	α_2	α_3	α_4
$k=1$	1	-1			
$k=2$	$\frac{3}{2}$	-2	$\frac{1}{2}$		
$k=3$	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$	
$k=4$	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$

Tabelle 3: Koeffizienten in den BDF-Verfahren.

Im Fall von konstanten Schrittweiten ($x_\ell = x_0 + \ell h$) können die Lagrange-Polynome transformiert werden zu

$$\tilde{L}_j(u) = \prod_{\nu=0,\nu\neq j}^{k} \frac{u+\nu-1}{\nu-j} \qquad \text{mit } \; x = x_i + uh.$$

Die neuen Polynome sind unabhängig vom Index i. Das entstehende k-Schritt-Verfahren lautet

$$\alpha_0 y_{i+1} + \alpha_1 y_i + \cdots + \alpha_{k-1} y_{i-k+2} + \alpha_k y_{i-k+1} = h f(x_{i+1}, y_{i+1}) \tag{3.8}$$

mit den konstanten Koeffizienten

$$\alpha_j = \tilde{L}_j'(1) \qquad \text{für } \; j = 0, \ldots, k.$$

(Denn es gilt $\mathrm{d}x = h\mathrm{d}u$.) Tabelle 3 zeigt die Koeffizienten der ersten vier BDF Methoden.

In diesem Ansatz sind alle Koeffizienten bereits durch die Polynominterpolation und die Bedingung (3.7) festgelegt. Dadurch sind keine weiteren Freiheitsgrade enthalten.

Es existiert noch eine Modifikation der BDF-Verfahren zu den sogenannten NDF-Verfahren (numerical differentiation formulas), siehe z.B. [6].

3.3 Konsistenz, Stabilität und Konvergenz

Wir definieren zunächst die Form von Mehrschrittverfahren (MSV), die in diesem Abschnitt analysiert werden soll.

Definition 3.1 *Ein lineares k-Schritt-Verfahren mit konstanter Schrittweite h zur Dgl. $y' = f(x, y)$ lautet*

$$\sum_{\ell=0}^{k} \alpha_\ell \, y_{i+\ell} = h \sum_{\ell=0}^{k} \beta_\ell f(x_{i+\ell}, y_{i+\ell}) \tag{3.9}$$

mit reellwertigen Koeffizienten $\alpha_0, \ldots, \alpha_k$ sowie $\beta_0, \ldots, \beta_k$ für die $\alpha_k \neq 0$ und $\alpha_0 \beta_0 \neq 0$ gilt.

Das Mehrschrittverfahren ist explizit für $\beta_k = 0$ und implizit für $\beta_k \neq 0$. Alle Methoden aus Abschnitt 3.1 und Abschnitt 3.2 sind lineare Mehrschrittverfahren. Nichtlineare Mehrschrittverfahren dagegen werden in der Praxis nur selten eingesetzt.

Definition 3.2 (lokaler Diskretisierungsfehler eines MSVs)
Sei $y(x)$ die exakte Lösung des AWPs $y' = f(x, y)$, $y(x_0) = y_0$. Der lokale Diskretisierungsfehler eines linearen Mehrschrittverfahrens (3.9) ist definiert durch den Defekt

$$\tau(h) := \frac{1}{h} \left(\sum_{\ell=0}^{k} \alpha_\ell \, y(x_0 + \ell h) - h \sum_{\ell=0}^{k} \beta_\ell f(x_0 + \ell h, y(x_0 + \ell h)) \right). \tag{3.10}$$

Diese Definition stimmt für ein explizites Verfahren im Fall $k = 1$ mit dem lokalen Diskretisierungsfehler (2.7) aus Def. 2.1 überein. Für implizite Verfahren lässt sich im Fall $k = 1$ eine gleiche Größenordnung des Fehlers begründen.

Bei einem expliziten linearen Mehrschrittverfahren (3.9) ergibt sich die Näherung mit $i = 0$ und $x_\ell = x_0 + \ell h$ aus ($\beta_k = 0$)

$$\alpha_k y_k + \sum_{\ell=0}^{k-1} \alpha_\ell y_\ell = h \sum_{\ell=0}^{k-1} \beta_\ell f(x_\ell, y_\ell).$$

Wir nehmen an, dass die Anfangswerte exakt gegeben sind ($y_\ell = y(x_\ell)$ für $\ell = 0, \ldots, k-1$). Es folgt

$$\alpha_k y_k + \sum_{\ell=0}^{k-1} \alpha_\ell y(x_\ell) = h \sum_{\ell=0}^{k-1} \beta_\ell f(x_\ell, y(x_\ell)).$$

Die exakte Lösung erfüllt wegen (3.10)

$$\alpha_k y(x_k) + \sum_{\ell=0}^{k-1} \alpha_\ell y(x_\ell) = h \sum_{\ell=0}^{k-1} \beta_\ell f(x_\ell, y(x_\ell)) + h \cdot \tau(h).$$

Der lokale Diskretisierungsfehler besitzt daher die Gestalt

$$\tau(h) = \frac{\alpha_k}{h}(y(x_k) - y_k).$$

Ein lineares MSV (3.9) kann noch normiert werden zu $\alpha_k := 1$.

Analog zu Def. 2.2 charakterisieren wir die Konsistenz des Verfahrens.

Definition 3.3 (Konsistenz eines MSVs)
Das lineare Mehrschrittverfahren (3.9) ist konsistent, wenn der lokale Diskretisierungsfehler aus (3.10) die Eigenschaft

$$\|\tau(h)\| \leq \sigma(h) \quad mit \quad \lim_{h \to 0} \sigma(h) = 0$$

für jeden Anfangswert $(x_0, y_0) \in G$ (G: Definitionsbereich von f) besitzt. Die Methode ist konsistent von Ordnung (mindestens) p, falls $\tau(h) = \mathcal{O}(h^p)$.

Für den globalen Diskretisierungsfehler $e_N := y(x_N) - u_N$ an einem festen Endpunkt $x_{\text{end}} = x_N$ gilt die Def. 2.3 wie bei Einschrittverfahren. Ebenso verwenden wir den Konvergenzbegriff aus Def. 2.3. Desweiteren setzen wir zur Vereinfachung in der theoretischen Untersuchung voraus, dass die in einem k-Schritt-Verfahren benötigten Anfangswerte $y_0, y_1, \ldots, y_{k-1}$ als exakte Lösungswerte von $y(x)$ vorliegen.

Es ergibt sich jedoch, dass die Konsistenz allein nicht hinreichend für die Konvergenz eines MSVs ist. Zusätzlich wird noch die Stabilität des Verfahrens benötigt. Dabei zeigt sich, dass es ausreicht die Dgl. $y' \equiv 0$ zu betrachten (d.h. $f \equiv 0$). Ein lineares MSV (3.9) reduziert sich dann zu der homogenen linearen Differenzengleichung

$$\sum_{\ell=0}^{k} \alpha_\ell y_{i+\ell} = \alpha_0 y_i + \alpha_1 y_{i+1} + \cdots + \alpha_{k-1} y_{i+k-1} + \alpha_k y_{i+k} = 0.$$

Zu Differenzengleichungen kann ein charakteristisches Polynom definiert werden.

Definition 3.4 *Das charakteristische Polynom eines linearen MSVs (3.9) bzw. der zugehörigen Differenzengleichung lautet*

$$p(\lambda) := \sum_{\ell=0}^{k} \alpha_\ell \lambda^\ell = \alpha_0 + \alpha_1\lambda + \alpha_2\lambda^2 + \cdots + \alpha_{k-1}\lambda^{k-1} + \alpha_k\lambda^k.$$

Das charakteristische Polynom besitzt also reelle Koeffizienten. Damit kann das Stabilitätskriterium angegeben werden.

Definition 3.5 (Stabilität eines MSVs)
Ein lineares MSV (3.9) heißt stabil, wenn die Nullstellen $\lambda_1, \ldots, \lambda_k \in \mathbb{C}$ des zugehörigen charakteristischen Polynoms die folgende Bedingung erfüllen:
i) $|\lambda_j| \leq 1$ falls λ_j einfache Nullstelle,
ii) $|\lambda_j| < 1$ falls λ_j mehrfache Nullstelle.

Man nennt das Stabilitätskriterium aus Def. 3.5 auch die Dahlquistsche Wurzelbedingung. Das Kriterium aus Def. 3.5 gilt für ein $p(\lambda)$ genau dann, wenn es auch für $\gamma p(\lambda)$ mit einer Konstanten $\gamma \neq 0$ erfüllt ist. Gilt $\alpha_0 = \alpha_1 = \cdots = \alpha_{r-1} = 0$ und $\alpha_r \neq 0$, dann kann statt $p(\lambda)$ das Polynom

$$\tilde{p}(\lambda) = \alpha_r + \alpha_{r+1}\lambda + \cdots + \alpha_{k-1}\lambda^{k-r-1} + \alpha_k\lambda^{k-r}$$

diskutiert werden, denn es folgt $p(\lambda) = \lambda^r \tilde{p}(\lambda)$ mit der (mindestens) r-fachen Nullstelle $\lambda = 0$.

Die Stabilität kann noch wie folgt interpretiert werden. Seien $y_0, y_1, \ldots, y_{k-1}$ und $z_0, z_1, \ldots, z_{k-1}$ zwei Wahlen von Anfangswerten für das MSV (3.9) zur Dgl. $y' \equiv 0$. Gilt die Stabilität aus Def. 3.5, dann gibt es eine Konstante $C > 0$ mit

$$|y_i - z_i| \leq C \sum_{j=0}^{k-1} |y_j - z_j| \quad \text{für alle } i.$$

Die Näherungslösungen hängen somit alle Lipschitz-stetig von den Eingabedaten ab. Zudem existiert eine feste Lipschitz-Konstante für alle i.

Damit kann das Hauptergebnis dieses Abschnitts angegeben werden. Dabei wird noch an das AWP $y' = f(x, y)$ vorausgesetzt, dass f auf $[x_0, x_{\text{end}}] \times \mathbb{R}^n$ stetig in x, p-mal stetig differenzierbar in y und alle Ableitungen beschränkt sind. Desweiteren seien die im MSV benötigten Anfangswerte $y_0, \ldots, y_{k-1}$ als exakte Lösungswerte vorgegeben.

Satz 3.1 (Konvergenz von MSV)
Ein lineares MSV (3.9) ist genau dann konvergent von Ordnung p, wenn das Verfahren konsistent von Ordnung p und stabil ist.

Dahlquist 1956 zeigte, dass für ein allgemeines MSV (d.h. lineares oder nichtlineares MSV) die Konsistenz und die Stabilität hinreichend für die Konvergenz sind. Bei linearen MSV gilt auch die Umkehrung. Bei nichtlinearen MSV folgt aus der Konvergenz nur die Stabilität. Den Beweis der Aussagen von Satz 3.1 kann man aus 7.2.10.1, 7.2.10.2 und 7.2.11.4 in [7] entnehmen. Es sei noch erwähnt, dass diese Aussagen nur für konstante Schrittweiten gelten. Für beliebige Schrittweitenwahlen gelten die Resultate nicht immer bzw. bedürfen intensiver weiterer Untersuchungen.

Ordnungsbedingungen

Wir leiten nun die Konsistenzbedingungen für ein lineares k-Schritt-Verfahren (3.9) zu beliebiger Ordnung $p \geq 1$ her. Die exakte Lösung eines AWPs sei hinreichend glatt. Der lokale Diskretisierungsfehler (3.10) kann geschrieben werden als

$$\tau(h) = \frac{1}{h} \left(\sum_{\ell=0}^{k} \alpha_\ell y(x + \ell h) - h \sum_{\ell=0}^{k} \beta_\ell y'(x + \ell h) \right). \tag{3.11}$$

Taylor-Entwicklungen führen auf

$$\begin{aligned} y(x + \ell h) &= \sum_{q=0}^{p} y^{(q)}(x) \cdot \frac{(\ell h)^q}{q!} + \mathcal{O}(h^{p+1}) \\ &= y(x) + \sum_{q=1}^{p} y^{(q)}(x) \cdot \frac{(\ell h)^q}{q!} + \mathcal{O}(h^{p+1}), \end{aligned}$$

$$
\begin{aligned}
y'(x+\ell h) &= \sum_{q=0}^{p-1} y^{(q+1)}(x)\cdot\frac{(\ell h)^q}{q!} + \mathcal{O}(h^p) \\
&= \sum_{q=1}^{p} y^{(q)}(x)\cdot\frac{(\ell h)^{q-1}}{(q-1)!} + \mathcal{O}(h^p).
\end{aligned}
$$

Einsetzen dieser Entwicklungen in den lokalen Fehler (3.11) ergibt

$$
\begin{aligned}
\tau(h) &= \frac{1}{h}\Bigg(\sum_{\ell=0}^{k}\alpha_\ell\left[y(x)+\sum_{q=1}^{p} y^{(q)}(x)\frac{(\ell h)^q}{q!}+\mathcal{O}(h^{p+1})\right] \\
&\qquad - h\sum_{\ell=0}^{k}\beta_\ell\left[\sum_{q=1}^{p} y^{(q)}(x)\frac{(\ell h)^{q-1}}{(q-1)!}+\mathcal{O}(h^p)\right]\Bigg) \\
&= \frac{y(x)}{h}\sum_{\ell=0}^{k}\alpha_\ell + \frac{1}{h}\sum_{\ell=0}^{k}\left[\sum_{q=1}^{p}\frac{y^{(q)}(x)}{q!}(\alpha_\ell\ell^q h^q - q\beta_\ell\ell^{q-1}h^q)\right]+\mathcal{O}(h^p) \\
&= \frac{y(x)}{h}\sum_{\ell=0}^{k}\alpha_\ell + \sum_{q=1}^{p}\frac{y^{(q)}(x)}{q!}h^{q-1}\left[\sum_{\ell=0}^{k}(\alpha_\ell\ell^q - q\beta_\ell\ell^{q-1})\right]+\mathcal{O}(h^p).
\end{aligned}
$$

Hier können wir die Ordnungsbedingungen ablesen. Für Konsistenz von Ordnung $p=1$ brauchen wir nur $\tau(h)=\mathcal{O}(h)$. Es folgen die Konsistenzbedingungen für Ordnung 1

$$
\sum_{\ell=0}^{k}\alpha_\ell = 0 \qquad \text{und} \qquad \sum_{\ell=0}^{k}(\alpha_\ell\ell - \beta_\ell) = 0. \tag{3.12}
$$

Die zusätzlichen Bedingungen für eine Ordnung $p>1$ ergeben sich zu

$$
\sum_{\ell=1}^{k}\alpha_\ell\ell^q = q\sum_{\ell=1}^{k}\beta_\ell\ell^{q-1} \qquad \text{für } q=2,\ldots,p.
$$

Man beachte, dass die erste Bedingung in (3.12) für die Konsistenz mit Ordnung 1 äquivalent zu $p(1)=0$ mit dem charakteristischen Polynom $p(\lambda)$ aus Def. 3.4 ist. Daher besitzt das charakteristische Polynom eines konsistenten linearen MSVs stets eine Nullstelle bei $\lambda=1$.

Ist ein MSV konsistent mit Ordnung genau p (d.h. es gilt $\tau(h)=\mathcal{O}(h^p)$,

$\tau(h) \neq \mathcal{O}(h^{p+1})$), dann besitzt der lokale Diskretisierungsfehler die Gestalt

$$\tau(h) = h^p y^{(p+1)}(x) \frac{1}{(p+1)!} \left[\sum_{\ell=1}^{k} (\alpha_\ell \ell^{p+1} - (p+1)\beta_\ell \ell^p) \right] + \mathcal{O}(h^{p+1}).$$

Daher hängt dieser Fehler von den höheren Ableitungen der exakten Lösung ab, welches relevant für eine Schrittweitensteuerung ist.

Beispiel: Adams-Moulton-Verfahren

Wir bestimmen die Konsistenzordnung der ersten beiden Adams-Moulton-Methoden. Die Koeffizienten sind in Tabelle 2 enthalten.

Das erste Schema ist die Trapezregel

$$-y_i + y_{i+1} = h \left[\tfrac{1}{2} f_i + \tfrac{1}{2} f_{i+1} \right].$$

Die Koeffizienten sind $\alpha_0 = -1$, $\alpha_1 = 1$, $\beta_0 = \beta_1 = \frac{1}{2}$. Es folgt

$$\sum_{\ell=0}^{1} \alpha_\ell = -1 + 1 = 0$$

und

$$\sum_{\ell=0}^{1} (\alpha_\ell \ell - \beta_\ell) = (-1) \cdot 0 - \tfrac{1}{2} + 1 \cdot 1 - \tfrac{1}{2} = 0.$$

Damit gilt eine Konsistenzordnung $p \geq 1$. Die Bedingung für $p = 2$ lautet

$$\sum_{\ell=1}^{1} (\alpha_\ell \ell^2 - 2\beta_\ell \ell^1) = 1 \cdot 1^2 - 2 \cdot \tfrac{1}{2} \cdot 1 = 0.$$

Es folgt $p \geq 2$. Die Bedingung für $p = 3$ ist verletzt wegen

$$\sum_{\ell=1}^{1} (\alpha_\ell \ell^3 - 3\beta_\ell \ell^2) = 1 \cdot 1^3 - 3 \cdot \tfrac{1}{2} \cdot 1^2 = -\tfrac{1}{2} \neq 0.$$

Die Trapezregel ist somit konsistent von genau Ordnung $p = 2$.

Das zweite Schema ergibt sich zu

$$-y_{i+1} + y_{i+2} = h \left[-\tfrac{1}{12} f_i + \tfrac{8}{12} f_{i+1} + \tfrac{5}{12} f_{i+2} \right].$$

Die Koeffizienten sind $\alpha_0 = 0$, $\alpha_1 = -1$, $\alpha_2 = 1$, $\beta_0 = -\frac{1}{12}$, $\beta_1 = \frac{8}{12}$, $\beta_2 = \frac{5}{12}$. Die Konsistenzbedingungen mit Ordnung $p = 1$

$$\sum_{\ell=0}^{2} \alpha_\ell = 0 + (-1) + 1 = 0$$

und

$$\sum_{\ell=0}^{2} (\alpha_\ell \ell - \beta_\ell) = 0 \cdot 0 - (-\tfrac{1}{12}) + (-1) \cdot 1 - \tfrac{8}{12} + 1 \cdot 2 - \tfrac{5}{12} = 0$$

sind erfüllt, d.h. es gilt die Ordnung $p \geq 1$. Die Bedingung für Ordnung $p = 2$ bestätigt sich aus

$$\sum_{\ell=1}^{2} (\alpha_\ell \ell^2 - 2\beta_\ell \ell^1) = (-1) \cdot 1^2 - 2 \cdot \tfrac{8}{12} \cdot 1^1 + 1 \cdot 2^2 - 2\tfrac{5}{12} \cdot 2^1 = 0.$$

Es folgt die Ordnung $p \geq 2$. Die Bedingung für Ordnung $p = 3$ ist nun

$$\sum_{\ell=1}^{2} (\alpha_\ell \ell^3 - 3\beta_\ell \ell^2) = (-1) \cdot 1^3 - 3 \cdot \tfrac{8}{12} \cdot 1^2 + 1 \cdot 2^3 - 3 \cdot \tfrac{5}{12} \cdot 2^2 = 0.$$

Somit haben wir $p \geq 3$. Die Forderung für Ordnung $p = 4$ ist verletzt:

$$\sum_{\ell=1}^{2} (\alpha_\ell \ell^4 - 4\beta_\ell \ell^3) = (-1) \cdot 1^4 - 4 \cdot \tfrac{8}{12} \cdot 1^3 + 1 \cdot 2^4 - 4 \cdot \tfrac{5}{12} \cdot 2^3 = -1 \neq 0.$$

Dadurch besitzt diese Methode die genaue Ordnung $p = 3$. Allgemein kann man zeigen, dass das k-Schritt Adams-Moulton-Vefahren die genaue Ordnung $p = k + 1$ aufweist.

Beispiel von Dahlquist

Das explizite Zweischrittverfahren definiert durch

$$y_{i+2} + 4y_{i+1} - 5y_i = h(4f(x_{i+1}, y_{i+1}) + 2f(x_i, y_i))$$

besitzt die Konsistenzordnung 3, siehe Abschnitt III.3 in [4]. Das charakteristische Polynom lautet

$$p(\lambda) = \lambda^2 + 4\lambda - 5.$$

Die Nullstellen sind daher $\lambda_1 = 1$ und $\lambda_2 = -5$. Somit ist das Verfahren instabil.

Stabilitätsanalyse

Wir untersuchen nun die Stabilität und damit Konvergenz einiger Typen von linearen MSV.

Einschrittverfahren:

Im Spezialfall $k = 1$ lautet das charakteristische Polynom eines linearen Einschrittverfahrens

$$p(\lambda) = \alpha_0 + \alpha_1 \lambda.$$

Da für die Stabilitätsuntersuchung die Koeffizienten β_ℓ keine Rolle spielen (Fall $f \equiv 0$) sind hier sowohl explizite als auch implizite Verfahren einbezogen. Ist das Verfahren konsistent, dann folgt $\alpha_0 + \alpha_1 = 0$. Immer gilt $\alpha_1 \neq 0$. Somit ist $\lambda = 1$ die einzige Nullstelle und das Verfahren ist stets stabil. Dieses Verhalten motiviert auch, dass die Stabilität bei Einschrittverfahren wie Runge-Kutta-Methoden immer gegeben ist und daher nicht eigens gefordert werden muss.

Verfahren aus numerischer Quadratur:

Die Methoden von Adams-Bashforth (3.4) und Adams-Moulton (3.5) besitzen beide wegen $\alpha_0 = \cdots = \alpha_{k-2} = 0$, $\alpha_{k-1} = -1$, $\alpha_k = 1$ das charakteristische Polynom

$$p(\lambda) = \lambda^k - \lambda^{k-1} = (\lambda - 1)\lambda^{k-1}$$

in der k-Schritt-Variante. Die Dahlquistsche Wurzelbedingung ist somit erfüllt und die Verfahren sind stets stabil. Über die Konvergenz wird folgende Aussage zitiert.

Satz 3.2 (Konvergenz von Adams-Methoden)
Die k-Schritt-Verfahren von Adams-Bashforth und Adams-Moulton sind jeweils konsistent von Ordnung k bzw. $k+1$ und stabil für alle k.

Für ein allgemeines Verfahren (3.3) aus dem Ansatz über Quadratur mit $l > 1$ sind nur die Koeffizienten $\alpha_k = 1$, $\alpha_{k-\ell} = -1$ ungleich null. Es folgt das charakteristische Polynom

$$p(\lambda) = \lambda^k - \lambda^{k-\ell} = (\lambda^\ell - 1)\lambda^{k-\ell}.$$

Eventuell tritt hier die Nullstelle $\lambda = 0$ auf. Immer sind als Nullstellen von p die Einheitswurzeln

$$\lambda_j = \mathrm{e}^{\mathrm{i}2\pi\frac{j-1}{\ell}} \qquad \text{für } \; j = 1, 2, \ldots, \ell$$

gegeben. Dadurch erhalten wir l einfache Nullstellen mit $|\lambda_j| = 1$. Die Dahlquistsche Wurzelbedingung ist erfüllt und die Verfahren sind immer stabil.

Verfahren aus numerischer Differentiation:

Die BDF-Verfahren (3.8) besitzen für verschiedene Schrittzahl k jeweils unterschiedliche Koeffizienten $\alpha_0, \ldots, \alpha_k$. Über die Konvergenz wird folgende Aussage zitiert.

Satz 3.3 (Konvergenz von BDF-Methoden)
Das BDF-Verfahren mit k-Schritten besitzt die Konsistenzordnung k. Die Methoden sind stabil für alle $k \leq 6$ und instabil für alle $k \geq 7$.

Da das Konstruktionsprinzip der BDF-Verfahren bereits alle Koeffizienten eindeutig festlegt, existieren keine Freiheitsgrade mit denen die Konvergenz für hohes k noch erreicht werden kann. Jedoch sind die Konvergenzordnungen bis $p \leq 6$ für die Praxis auch ausreichend.

Optimale Konvergenzordnung

Wir kehren nun zu allgemeinen Betrachtungen zurück. Es ist naheliegend zu fragen, welche Konvergenzordnung in einem linearen k-Schritt-Verfahren (3.9) für festes k höchstens erreicht werden kann. O.E.d.A. sei $\alpha_k = 1$. Wir erhalten somit $2k + 1$ Freiheitsgrade in Form der Koeffizienten $\alpha_0, \ldots, \alpha_{k-1}$ und $\beta_0, \ldots, \beta_k$. Wir können eine Methode konstruieren, die Konsistent von Ordnung $p = 2k$, da $p + 1$ Konsistenzbedingungen erfüllt werden müssen. Jedoch muss das Verfahren auch stabil sein um die Konvergenz zu erreichen. Das Dahlquistsche Wurzelkriterium liefert k Bedingungen für die Nullstellen des charakteristischen Polynoms. Ein konsistentes Schema besitzt die Nullstelle $\lambda = 1$, welche das Wurzelkriterium erfüllt. Daher verbleiben $k-1$ Einschränkungen. Wir erwarten daher als maximale Konvergenzordnung $p \approx 2k - (k-1) = k + 1$. Der folgende Satz von Dahlquist (1956/59) zeigt die exakten Aussage.

Satz 3.4 (erste Dahlquist-Schranke)
Ein lineares k-Schritt-Verfahren (3.9), welches die Stabilitätsbedingung erfüllt, besitzt die maximale Konsistenzordnung

$$\begin{array}{ll} k+2 & \text{für } k \text{ gerade,} \\ k+1 & \text{für } k \text{ ungerade,} \\ k & \text{für } \beta_k/\alpha_k \leq 0 \text{ (insbesondere für explizite Verfahren).} \end{array}$$

Im Vergleich hierzu hat ein implizites Runge-Kutta-Verfahren mit s Stufen als Freiheitsgrade $s^2 + s$ Koeffizienten. (Die Knoten folgen aus den inneren Gewichten über (2.17).) Ein explizites Runge-Kutta-Verfahren besitzt etwa $\frac{s^2}{2}$ Freiheitsgrade. Zusätzliche Bedingungen für die Stabilität existieren bei Einschrittverfahren nicht. Die optimale Konvergenzordnung bei fester Stufenzahl s lautet $p = 2s$ für implizite Methoden (Gauss-Runge-Kutta) und $p \leq s$ für explizite Methoden. Man beachte, dass die maximale Ordnung linear bzw. sublinear mit der Stufenzahl anwächst, während die Anzahl der Koeffizienten quadratisch ansteigt.

3.4 Prädiktor-Korrektor-Verfahren

Wir betrachten ein AWP eines Systems gew. Dgln. $y' = f(x, y)$, $y(x_0) = y_0$. In diesem Abschnitt diskutieren wir die Lösung des nichtlinearen Gleichungssystems aus algebraischen Gleichungen, das bei impliziten MSV auftritt. Ein lineares k-Schritt-Verfahren mit konstanter Schrittweite kann geschrieben werden als

$$y_{i+1} - h\beta_0 f(x_{i+1}, y_{i+1}) = h \sum_{\ell=1}^{k} \beta_\ell f(x_{i+1-\ell}, y_{i+1-\ell}) - \sum_{\ell=1}^{k} \alpha_\ell y_{i+1-\ell}. \quad (3.13)$$

Die Formel (3.13) stellt ein System aus n algebraischen Gleichungen für die Unbekannten $y_{i+1} \in \mathbb{R}^n$ dar. Die rechte Seite

$$w_i := h \sum_{\ell=1}^{k} \beta_\ell f(x_{i+1-\ell}, y_{i+1-\ell}) - \sum_{\ell=1}^{k} \alpha_\ell y_{i+1-\ell}$$

ist bereits gegeben durch die alten Näherungswerte.

Das nichtlineare Gleichungssystem

$$y_{i+1} - h\beta_0 f(x_{i+1}, y_{i+1}) - w_i = 0$$

kann numerisch mittels des Newton-Verfahrens gelöst werden. Wir definieren die Matrizen $A^{(\nu)} \in \mathbb{R}^{n\times n}$

$$A^{(\nu)} := I - h\beta_0 (\mathrm{D}f)(x_{i+1}, y_{i+1}^{(\nu)})$$

mit der Einheitsmatrix $I \in \mathbb{R}^{n\times n}$ und der Jacobi-Matrix $\mathrm{D}f \in \mathbb{R}^{n\times n}$. Die Iteration aus dem Newton-Verfahren lautet

$$\begin{aligned} A^{(\nu)} \Delta y_{i+1}^{(\nu)} &= y_{i+1}^{(\nu)} - h\beta_0 f(x_{i+1}, y_{i+1}^{(\nu)}) - w_i \\ y_{i+1}^{(\nu+1)} &= y_{i+1}^{(\nu)} - \Delta y_{i+1}^{(\nu)} \end{aligned}$$

für $\nu = 0, 1, 2, \ldots$ mit einem Startwert $y_{i+1}^{(0)} \in \mathbb{R}^n$. Somit erhalten wir eine Folge aus linearen Gleichungssystemen. In dieser Situation sind geeignete Startwerte gegeben durch $y_{i+1}^{(0)} = y_i$. Die Konvergenzgeschwindigkeit der Iteration ist quadratisch.

Wir verwenden das vereinfachte Newton-Verfahren um Rechenaufwand einzusparen. Die Iteration ändert sich zu

$$\begin{aligned} A^{(0)} \Delta y_{i+1}^{(\nu)} &= y_{i+1}^{(\nu)} - h\beta_0 f(x_{i+1}, y_{i+1}^{(\nu)}) - w_i \\ y_{i+1}^{(\nu+1)} &= y_{i+1}^{(\nu)} - \Delta y_{i+1}^{(\nu)} \end{aligned} \tag{3.14}$$

für $\nu = 0, 1, 2, \ldots$. Die Konvergenzgeschwindigkeit der Iteration ist linear. Der Rechenaufwand diese vereinfachten Newton-Iteration besteht aus folgenden Anteilen:

Start-Phase:

1. Berechne die Jacobi-Matrix $\mathrm{D}f$ bei $x = x_{i+1}$, $y = y_{i+1}^{(0)}$. Falls numerische Differentiation verwendent wird, so sind n zusätzliche Auswertungen von f erforderlich.

2. Zerlege $A^{(0)} = L \cdot R$ in eine linke untere Dreiecksmatrix L und eine rechte obere Dreiecksmatrix R. Der Rechenaufwand ist proportional zu n^3.

In jedem Schritt:

1. Werte f bei $x = x_{i+1}$, $y = y_{i+1}^{(\nu)}$ aus.
2. Löse das lineare Gleichungssystem in (3.14) mit der vorhandenen LR-Zerlegung. Der Rechenaufwand für jede Vorwärts- und Rückwärtssubstitution ist proportional zu n^2.

Falls eine Schrittweitensteuerung verwendet wird und die Newton-Iteration nicht konvergiert, dann wird die Schrittweite $h_i = x_{i+1} - x_i$ reduziert. Beispielsweise wird die Iteration neu gestartet mit dem veränderten Gitterpunkt $x_{i+1} = x_i + \frac{h_i}{2}$, weil der verfügbare Startwert $y_{i+1}^{(0)} = y_i$ dazu eine bessere Approximation darstellt wegen der Stetigkeit der exakten Lösung.

Wir betrachten eine alternative Technik, welche noch deutlich an Rechenaufwand einspart. Das nichtlineare Gleichungssystem (3.13) kann als Fixpunktproblem

$$y_{i+1} = \Phi(y_{i+1})$$

mit der Funktion

$$\Phi(y_{i+1}) := h\beta_0 f(x_{i+1}, y_{i+1}) + w_i$$

geschrieben werden. Nach dem Banachschen Fixpunktsatz konvergiert die Fixpunktiteration

$$y_{i+1}^{(\nu+1)} = \Phi(y_{i+1}^{(\nu)}) \qquad \text{für } \nu = 0, 1, 2, \ldots \tag{3.15}$$

falls die Abbildung Φ kontraktiv ist. In einer beliebigen Vektornorm folgt

$$\begin{aligned} \|\Phi(y) - \Phi(z)\| &= \|h\beta_0 f(x_{i+1}, y) + w_i - (h\beta_0 f(x_{i+1}, z) + w_i)\| \\ &= h \cdot |\beta_0| \cdot \|f(x_{i+1}, y) - f(x_{i+1}, z)\| \\ &\leq h \cdot |\beta_0| \cdot L \cdot \|y - z\| \end{aligned}$$

unter der Voraussetzung der Lipschitz-Bedingung (1.2) an die rechte Seite mit Konstante $L > 0$. Folglich ist die Abbildung Φ kontraktiv falls

$$h \cdot |\beta_0| \cdot L < 1 \qquad \Leftrightarrow \qquad h < \frac{1}{|\beta_0| \cdot L} \, . \tag{3.16}$$

Daher erhalten wir eine konvergente Fixpunktiteration für hinreichend kleine Schrittweite. Die Konvergenzgeschwindigkeit ist linear mit der Konstanten $h|\beta_0|L$. Der Rechenaufwand pro Iterationsschritt (3.15) besteht nur aus einer einzelnen Auswertung der rechten Seite f. Insbesondere müssen hier keine linearen Gleichungssysteme gelöst werden.

Jedoch schränkt die Kontraktivitätsbedingung (3.16) die Schrittweite h stark ein im Falle von hohen Konstanten L. Große Lipschitz-Konstanten L treten bei steifen Differentialgleichung auf, die als mathematisches Modell in vielen Anwendungen vorliegen. In diesen Fällen wird die Fixpunktiteration (3.15) nutzlos, da eine extrem hohe Anzahl von Integrationsschritten erforderlich ist. Im Gegensatz dazu liefert das Newton-Verfahren immer noch geeignete Näherungen auch für große Schrittweiten h.

Nun betrachten wir implizite MSV (3.13) für moderate Konstanten L. Die Bestimmung der Unbekannten y_{i+1} kann durch ein *Prädiktor-Korrektor-Verfahren* erfolgen. Die Technik besteht aus zwei Teilen:

- *Prädiktor-Methode*: Ein Verfahren, das einen guten Startwert liefert.
- *Korrektor-Methode*: Ein Iterationsverfahren, das gegen den a priori unbekannten Wert konvergiert, wobei eine feste Anzahl an Iterationsschritten durchgeführt wird.

Als Beispiel betrachten wir die Adams-Moulton-Verfahren. Das k-Schritt (implizite) Adams-Moulton-Verfahren (3.5) besitzt die Ordnung $k+1$, während das k-Schritt (explizite) Adams-Bashforth-Verfahren (3.4) die Ordnung k hat. Die Fixpunktiteration (3.15) in der k-Schritt Adams-Moulton-Methode wird nun als Korrektor-Schritt gewählt. Die k-Schritt Adams-Bashforth-Methode wird im Prädiktor-Schritt verwendet.

Algorithmus: $\mathbf{P}(\mathbf{EC})^m\mathbf{E}$ Verfahren

P: $y_{i+1}^{(0)} := y_i + h(\beta_1 f_i + \beta_2 f_{i-1} + \cdots + \beta_k f_{i-k+1})$ (Adams-Bashforth)

für $\nu = 0, 1, \ldots, m-1$

E: $f_{i+1}^{(\nu)} := f(x_{i+1}, y_{i+1}^{(\nu)})$

C: $y_{i+1}^{(\nu+1)} := y_i + h(\beta_0^* f_{i+1}^{(\nu)} + \beta_1^* f_i + \beta_2^* f_{i-1} + \cdots + \beta_k^* f_{i-k+1})$
(Fixpunktiteration für Adams-Moulton)

E: $f_{i+1} := f(x_{i+1}, y_{i+1}^{(m)})$ (Auswertung für nächsten Integrationsschritt)

Tabelle 4: Algorithmus des Prädiktor-Korrektor-Verfahrens für einen Integrationsschritt.

Wir bezeichnen die Anwendung des Prädiktors mit P, einen Korrektor-Schritt mit C und eine benötigte Funktionsauswertung der rechten Seite f mit E (da der Rechenaufwand durch die Anzahl dieser Funktionsauswertungen charakterisiert ist). Sei $f_i := f(x_i, y_i)$. Es folgt die $\mathrm{P(EC)}^m\mathrm{E}$-Methode zu einer konstanten ganzen Zahl m. Tabelle 4 spezifiziert den Algorithmus. Üblicherweise wird nur $m = 1$ oder $m = 2$ verwendet, da mehr Korrektor-Schritte die Genauigkeit im Ergebnis nicht wesentlich erhöhen.

In der Praxis wird die $\mathrm{P(EC)}^m\mathrm{E}$-Methode mit lokaler Fehlerkontrolle (d.h. Schrittweitensteuerung) verwendet. Dabei müssen in jedem Integrationsschritt die Koeffizienten erneut berechnet werden mittels dividierter Differenzen (Newton-Interpolation). Die Differenz

$$y_{i+1}^{(m)} - y_{i+1}^{(0)} = \mathcal{O}(h^{k+1})$$

ergibt einen Fehlerschätzer für die Schrittweitensteuerung, da $y_{i+1}^{(0)}$ eine Approximation der Ordnung k und $y_{i+1}^{(m)}$ eine Approximation der Ordnung $k+1$ darstellt, vergleiche Abschnitt 2.5. Zudem kann eine variable Ordnung durch eine Ordnungssteuerung verwendet werden.

3.5 Ordnungssteuerung

Die Schrittweitensteuerung schätzt die größtmögliche Schrittweite, so dass der lokale Fehler unterhalb einer gegebenen Schranke verbleibt, siehe Abschnitt 2.5. Das Ziel ist die Anzahl der benötigten Schritte in der Integration niedrig zu halten. Die Anzahl der Schritte kann weiter reduziert werden durch Hinzunahme einer Ordnungssteuerung. Hierzu setzen wir voraus, dass mehrere Verfahren mit den Konvergenzordnungen $p = 1, 2, \ldots, p_{\max}$ verfügbar sind ($p_{\max} = 5 - 15$ in der Praxis). Die Idee ist nun dasjenige Verfahren auszuwählen, bei dem die Schrittweitenkontrolle die größte Schrittweitenschätzung im nächsten Schritt ergibt.

Es sei die Schrittweite h bereits gewählt und die Ordnung p vorgeschlagen. Wir berechnen dann drei Näherungen mittels der Verfahren für Ordnung $p-1, p, p+1$. Zu jeder Methode wird eine Schätzung der optimalen Schrittweite h_{p-1}, h_p, h_{p+1} bestimmt. Falls eine dieser Schrittweiten größergleich h ist, so wird der Schritt mit der entsprechenden Näherung akzeptiert.

Desweiteren benötigen wir eine Zahl w_p, welche den Rechenaufwand im Verfahren der Ordnung p quantifiziert. (Beispielsweise kann dies die Anzahl der Funktionsauswertungen der rechten Seite der Dgl. sein.) Nun folgt aus jedem Verfahren eine Schätzung

$$\sigma_{p-1} := \frac{w_{p-1}}{h_{p-1}}, \qquad \sigma_p := \frac{w_p}{h_p}, \qquad \sigma_{p+1} := \frac{w_{p+1}}{h_{p+1}}$$

des Rechenaufwands pro Einheitsschrittweite ($h = 1$). Wir verwenden im nächsten Integrationsschritt die Ordnung $\hat{p}$ mit dem kleinsten Wert $\sigma_{\hat{p}}$ als Vorschlag für eine optimale Ordnung. Die Schrittweite $h_{\hat{p}}$ wird im nächsten Schritt wieder in allen drei Methoden für $\hat{p}-1, \hat{p}, \hat{p}+1$ verwendet.

Algorithmen zu linearen MSV verwenden üblicherweise Ordnungssteuerung, beispielsweise die Adams-Methoden oder die BDF-Methoden. Der Grund ist, dass der Rechenaufwand w_p nahezu unabhängig vom Wert p in diesen Verfahren ist. Man beachte, dass nur $m+1$ zusätzliche Funktionsaufswertungen in jedem Schritt der P(EC)^{m}E-Method bei beliebiger Ordnung erforderlich sind, da die anderen Funktionsauswertungen bereits aus den

vorangegangenen Schritten vorliegen. Im Gegensatz dazu ist der Aufwand bei expliziten Runge-Kutta-Verfahren ungefähr $w_p \approx Cp$ mit einer Konstanten C, weil $p \approx s$ mit der Stufenzahl s gilt und die Anzahl der Funktionsauswertungen identisch mit s ist.

Eine weitere Klasse von Verfahren, bei der sich eine Ordnungssteuerung in natürlicher Weise anbietet, sind die Extrapolationsmethoden. Diese Techniken können auf der Grundlage von sowohl Einschrittverfahren als auch Mehrschrittverfahren konstruiert werden.

Es sei betont, dass die Implementierung einer Ordnungssteuerung noch viele hochentwickelte Einzelheiten in Abhängigkeit von den jeweiligen Verfahren enthält, auf die in diesem Abschnitt nicht näher eingegangen wurde.

Kapitel 4

Methoden für steife Differentialgleichungen

Steife Systeme von gewöhnlichen Differentialgleichungen treten in vielen Anwendungen auf wie beispielsweise in der chemischen Reaktionskinetik, in der Mechanik und bei der Simulation elektrischer Schaltungen. Theoretisch können diese Systeme mit jedem konvergenten Verfahren aus den vorhergehenden beiden Kapiteln numerisch gelöst werden. Jedoch sind explizite Methoden vollkommen ineffizient bei steifen Differentialgleichungen. Dies motiviert die Notwendigkeit von impliziten Methoden.

4.1 Beispiele

Um das Phänomen der Steifheit zu verdeutlichen betrachten wir zwei Beispiele: den Van-der-Pol Oszillator und ein bestimmtes lineares System.

Van-der-Pol Oszillator

Der Van-der-Pol Oszillator wird beschrieben durch eine gew. Dgl. zweiter Ordnung

$$z''(t) + \mu(z(t)^2 - 1)z'(t) + z(t) = 0$$

mit dem Parameter $\mu > 0$. Die Lösung ist jeweils periodisch, wobei die Periode von μ abhängt. Damit die Periode (nahezu) unabhängig von μ wird,

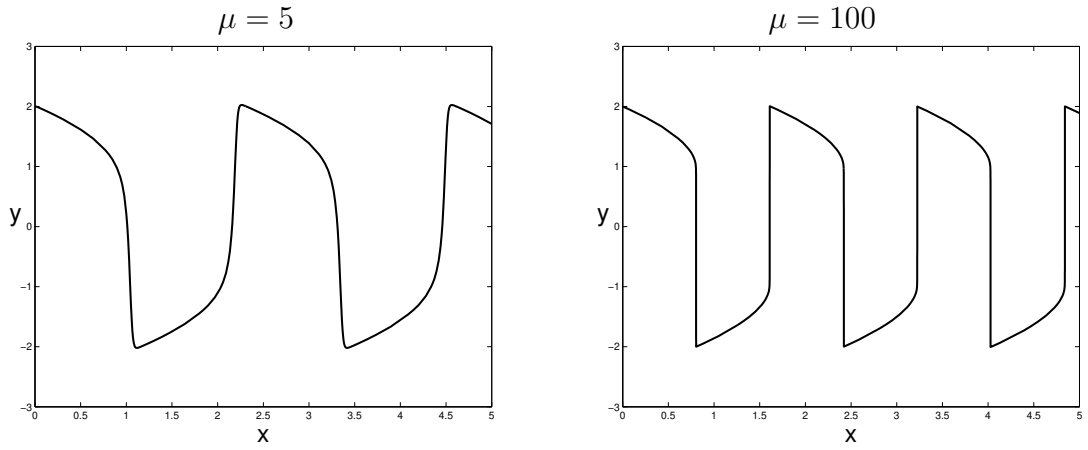


Abbildung 11: Lösungen des Van-der-Pol Oszillators.

verwenden wir die Transformation $x = \frac{t}{\mu}$. Es folgt mit $y(x) = z(\mu x)$

$$\frac{1}{\mu^2} y''(x) + (y(x)^2 - 1)y'(x) + y(x) = 0.$$

Die Anfangswerte seien $y(0) = 2$ und $y'(0) = 0$. Wir lösen das äquivalente System erster Ordnung

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= -\mu^2((y_1(x)^2 - 1)y_2(x) + y_1(x)). \end{aligned}$$

Abb. 11 zeigt Lösungen für unterschiedliche Parameter μ.

Wir lösen das System mit zwei Methoden: ein explizites Runge-Kutta-Verfahren der Ordnung 2(3) und die implizite Trapezregel (Ordnung 2). In beiden Varianten wird eine lokale Fehlerkontrolle durchgeführt mit den Genauigkeitsforderungen rtol $= 10^{-2}$ und atol $= 10^{-4}$. Die Integration erfolgt im Intervall $x \in [0, 5]$. Tabelle 5 enthält die Anzahl der benötigten Schritte für verschiedene Parameter μ. Der Rechenaufwand ist proprotional zur Anzahl der Schritte in jeder Methode. Wir bemerken, dass das explizite Verfahren mehr Schritte benötigt je größer der Parameter μ ist. Falls die Schrittweite im expliziten Verfahren erhöht wird, dann werden die Ergebnisse deutlich falsch. Im Gegensatz dazu steigt die Anzahl der Schritte im impliziten Verfahren nur geringfügig an. Damit ist die implizite Variante überlegen. Das Verhalten des Systems aus gew. Dgln. für hohe Parameter μ nennt man steif.

	explizites Verfahren	implizites Verfahren
$\mu = 5$	145	201
$\mu = 10$	434	294
$\mu = 50$	9017	483
$\mu = 100$	36.067	542
$\mu = 200$	144.453	616
$\mu = 1000$	3.616.397	624

Tabelle 5: Anzahl der Schritte in der Simulation des Van-der-Pol Oszillators.

Lineares System gew. Dgln.

Wir untersuchen ein bestimmtes lineares System aus gew. Dgln., nämlich

$$\begin{pmatrix} y_1'(x) \\ y_2'(x) \end{pmatrix} = \begin{pmatrix} -298 & 99 \\ -594 & 197 \end{pmatrix} \begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix}. \tag{4.1}$$

Die Matrix besitzt die Eigenwerte $\lambda_1 = -1$ und $\lambda_2 = -100$ mit den Eigenvektoren $v_1 = (1, 3)^\top$ und $v_2 = (1, 2)^\top$. Die allgemeine Lösung des Systems (4.1) lautet

$$y(x) = C_1 \mathrm{e}^{-x} \begin{pmatrix} 1 \\ 3 \end{pmatrix} + C_2 \mathrm{e}^{-100x} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

mit beliebigen Konstanten $C_1, C_2 \in \mathbb{R}$. Alle Lösungen besitzen die Eigenschaft

$$\lim_{x \to \infty} y(x) = 0.$$

Jedoch fällt einer der beiden Terme (der zu λ_2) sehr schnell ab, während der andere Term (der zu λ_1) sich relativ langsam verändert.

Wir betrachten das Anfangswertproblem $y_1(0) = -\frac{1}{2}$, $y_2(0) = \frac{1}{2}$. Abb. 12 (links) zeigt die zugehörige Lösung. Wir verwenden wieder ein explizites Runge-Kutta-Verfahren der Ordnung 2(3) und die implizite Trapezregel jeweils mit Schrittweitensteuerung (rtol $= 10^{-3}$, atol $= 10^{-6}$). Im Intervall $x \in [0, 10]$ benötigt das explizite Verfahren 413 Schritte und das implizite Verfahren führt 94 Schritte durch. Abb. 12 (rechts) zeigt, dass die explizite Methode auch kleine Schrittweiten gegen Ende des Intervalls wählt, während die Lösung dort nahezu konstant ist. Wenn die Schrittwei-

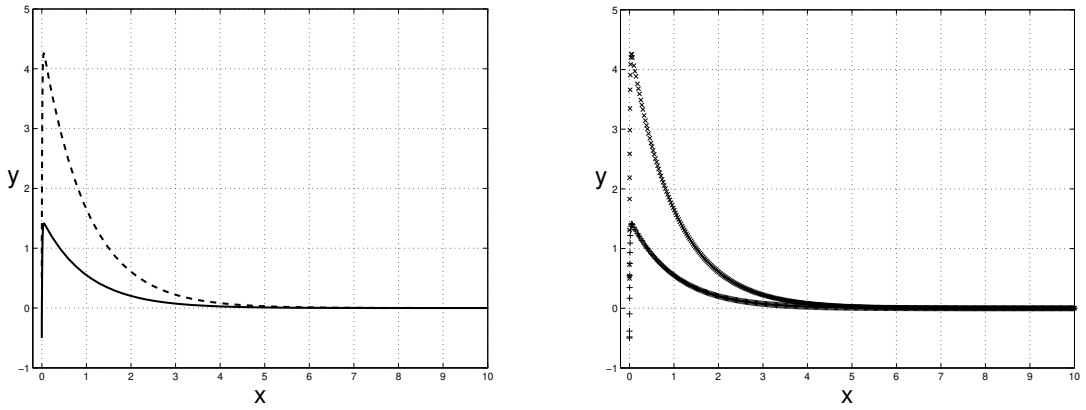


Abbildung 12: Steifes lineares System: exakte Lösung (links) – y_1 (—) und y_2 (- - -) – und Näherungen aus der expliziten Methode mit Schrittweitenkontrolle (rechts).

ten in der expliziten Methode vergrößert werden, dann entstehen vollkommen falsche Näherungen. Wir möchten dieses unterschiedliche Verhalten der Integrationsverfahren verstehen.

In diesem Beispiel kann das steife Verhalten wie folgt charakterisiert werden: Die Lösungen von Anfangswertproblemen nähern sich schnell einer Lösung, welche sich nur langsam verändert. Jedoch gibt es sich schnell verändernde Lösungen in einer Umgebung der sich langsam verändernden Lösung.

4.2 Testgleichungen

Wir analysieren das obige lineare Beispiel nun im allgemeinen Fall. Gegeben sei ein lineares System von gew. Dgln.

$$y'(x) = Ay(x), \quad y : \mathbb{R} \to \mathbb{R}^n, \quad A \in \mathbb{R}^{n\times n}. \tag{4.2}$$

Wir nehmen an, dass die Koeffizientenmatrix diagonalisierbar ist, d.h.

$$A = T^{-1}DT, \quad T \in \mathbb{C}^{n\times n}, \quad D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n).$$

Die Eigenwerte $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ können auch für reellwertige Matrix A komplexe Zahlen sein. Die Transformation $z(x) = Ty(x)$ entkoppelt das System in unabhängige skalare lineare gew. Dgln.

$$z_j'(x) = \lambda_j z_j(x) \qquad \text{für } \; j = 1, \ldots, n. \tag{4.3}$$

Entsprechend transformieren sich die Anfangswerte zu $z(x_0) = Ty(x_0)$.

Dahlquist'sche Testgleichung

Motiviert durch die entkoppelten Dgln. (4.3) diskutieren wir die skalare lineare Dgl.

$$y'(x) = \lambda y(x), \quad y : \mathbb{R} \to \mathbb{C}, \quad \lambda \in \mathbb{C}. \tag{4.4}$$

Die Dgl. (4.4) nennt man Dahlquist'sche Testgleichung (1963). Zu einem Anfangswert $y(0) = y_0$ lautet die exakte Lösung

$$y(x) = y_0 \mathrm{e}^{\lambda x} = y_0 \mathrm{e}^{\mathrm{Re}(\lambda)x} \cdot \mathrm{e}^{\mathrm{i}\cdot \mathrm{Im}(\lambda)x}.$$

Es folgt

$$|y(x)| = |y_0| \cdot \mathrm{e}^{\mathrm{Re}(\lambda)x}.$$

Falls $\mathrm{Re}(\lambda) < 0$ gilt, dann fällt die Lösung streng monoton.

Wir wenden das explizite und implizite Euler-Verfahren auf dieses Testproblem an. Abb. 13 verdeutlicht die numerischen Lösungen für $\lambda = -10$ und $y_0 = 1$. Wir erkennen, dass das implizite Verfahren das qualitative Verhalten der exakten Lösung für alle Schrittweiten reproduziert. Andererseits liefert das explizite Verfahren nur für kleine Schrittweiten das qualitativ korrekte Verhalten.

Es ist einfach dieses Verhalten der Euler-Methoden zu erklären:

(i) *explizites Euler-Verfahren*
Anwendung auf die Dahlquist-Testgleichung (4.4) ergibt

$$y_1 = y_0 + h\lambda y_0 = (1 + h\lambda)y_0.$$

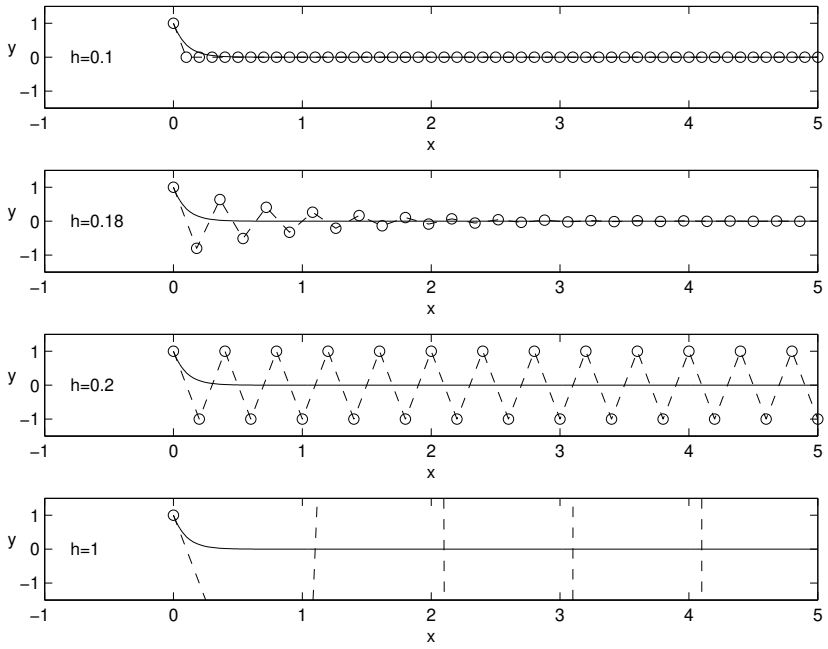
Es folgt sukzessive (y_j ist Näherung zu $y(jh)$)

$$y_j = (1 + h\lambda)^j \, y_0.$$

Somit gilt $|y_j| \leq |y_{j-1}|$ genau dann, wenn

$$|1 + h\lambda| \leq 1.$$

explizites Euler-Verfahren :



implizites Euler-Verfahren :

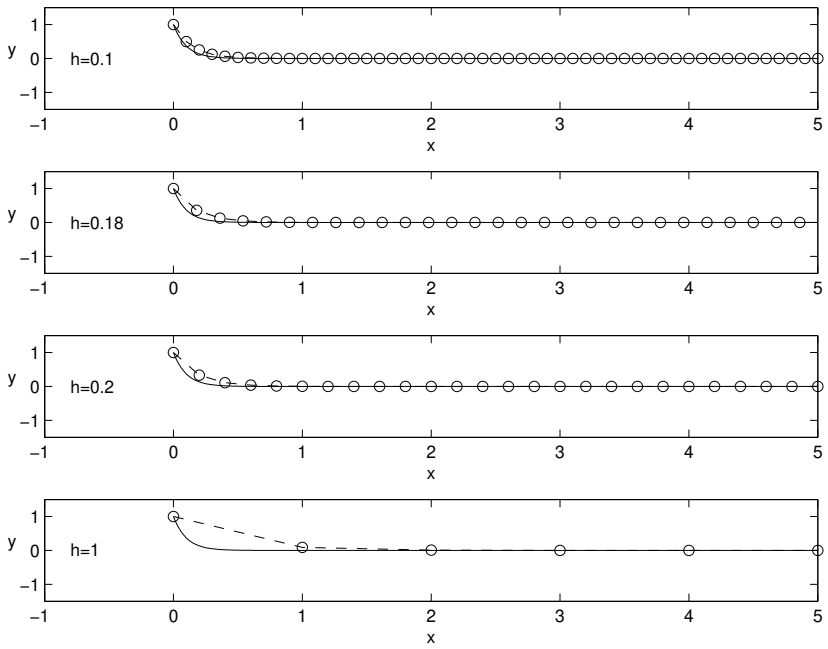


Abbildung 13: Lösungen der Dahlquist'schen Testgleichung mit $\lambda = -10$: exakte Lösung (—) und Näherungen (∘).

Für $\lambda \in \mathbb{R}$ und $\lambda < 0$ (und natürlich $h > 0$), erhalten wir eine Schrittweitenrestriktion

$$h \leq \tfrac{2}{|\lambda|}.$$

Nur Schrittweiten, die diese Bedingung erfüllen, ergeben Approximation, die nicht ansteigen. Für hohe $|\lambda|$ muss die Schrittweite h klein sein.

(ii) *Implizites Euler-Verfahren*
Nun führt die Dahlquist-Testgleichung (4.4) auf die Formel

$$y_1 = y_0 + h\lambda y_1 \qquad \Rightarrow \qquad y_1 = \frac{1}{1 - h\lambda} y_0.$$

Wir erhalten die Näherungen

$$y_j = \left(\frac{1}{1 - h\lambda}\right)^j y_0.$$

Die Eigenschaft $|y_j| \leq |y_{j-1}|$ ist erfüllt genau dann, wenn

$$\left|\frac{1}{1 - h\lambda}\right| \leq 1 \qquad \Leftrightarrow \qquad 1 \leq |1 - h\lambda|$$

gilt. Für $\lambda \in \mathbb{R}$ und $\lambda < 0$ ist diese Eigenschaft bei beliebiger Schrittweite $h > 0$ gegeben. Daher tritt keine Schrittweitenrestriktion auf.

Wir untersuchen die Dahlquist'sche Testgleichung im Fall von Parametern λ mit hohem negativen Realteil. Man beachte, dass die zugehörige Lipschitz-Konstante der rechten Seite dann auch groß wird, da mit $f(x, y) = \lambda y$ folgt

$$|f(x, y) - f(x, z)| = |\lambda y - \lambda z| = |\lambda| \cdot |y - z|.$$

Bei einer Fixpunktiteration in einer Prädiktor-Korrektor-Methode, siehe Abschnitt 3.4, wäre eine deutliche Schrittweitenrestriktion erforderlich um die Konvergenz der Iteration sicherzustellen.

Prothero-Robinson Testgleichung

Ein anderes skalares Problem, welches das steife Verhalten verdeutlicht, ist die Prothero-Robinson Testgleichung (1973)

$$y'(x) = \lambda(y(x) - \varphi(x)) + \varphi'(x), \qquad y(x_0) = y_0 \tag{4.5}$$

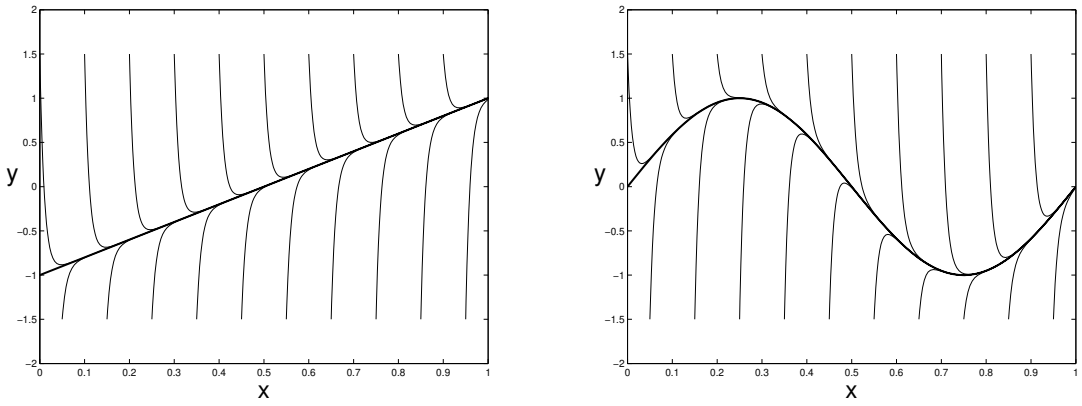


Abbildung 14: Lösungen von verschiedenen Anfangswertproblemen zur Prothero-Robinson Testgleichung mit Parameter $\lambda = -100$ und den Funktionen $\varphi(x) = 2x - 1$ (links) sowie $\varphi(x) = \sin(2\pi x)$ (rechts).

mit der Lösung $y : \mathbb{R} \to \mathbb{R}$ und einem reellen Parameter $\lambda < 0$. Die glatte Funktion $\varphi : \mathbb{R} \to \mathbb{R}$ sei vorgegeben. Die Lösungen von Anfangswertproblemen zu (4.5) lauten

$$y(x) = (y_0 - \varphi(x_0))\mathrm{e}^{\lambda(x-x_0)} + \varphi(x).$$

Die spezielle Lösung $y \equiv \varphi$ stellt die asymptotische Phase dar, d.h. die anderen Lösungen nähern sich dieser Funktion schnell an im Fall von hohen negativen Werten λ. Abb. 14 zeigt zwei Beispiele. Desweiteren ergibt der Spezialfall $\varphi \equiv 0$ die Dahlquist'sche Testgleichung (4.4).

Definition von steifen linearen Systemen

Wir definieren nun das Phänomen der Steifheit für lineare Differentialgleichungssysteme. Man beachte, dass es keine exakte Definition von Steifheit (für lineare oder nichtlineare Systeme) gibt. Ein Grund dafür ist, dass Steifheit nicht nur eine qualitatives Eigenschaft sondern auch ein quantitatives Verhalten bedeutet. Wir geben zwei Definitionen an:

- Wir nehmen an, dass im linearen System $y' = Ax$ die Eigenwerte λ_j alle negativen Realteil besitzen. Das System ist steif, wenn sowohl Eigenwerte mit kleinem negativen Realteil als auch Eigenwerte mit hohem

negativen Realteil existieren, d.h. das Verhältnis

$$\frac{\max\limits_{j=1,\ldots,n} |\mathrm{Re}(\lambda_j)|}{\min\limits_{j=1,\ldots,n} |\mathrm{Re}(\lambda_j)|} \tag{4.6}$$

ist sehr groß. (Falls alle Eigenwerte einen hohen negativen Realteil in der gleichen Größenordnung besitzen, d.h. das Verhältnis (4.6) ist klein, dann kann das steife Verhalten aus dem System heraustransformiert werden.)

- Die folgende Charakterisierung von Curtis und Hirschfelder (1952) geht auf ihre Beobachtungen bei der Simulation von chemischer Reaktionskinetik zurück (und gilt auch für nichtlineare Systeme): „Stiff equations are equations, where certain implicit methods perform better – usually tremendously better – than explicit ones.“ In Kurzform: Implizit ist besser als explizit.

4.3 A-Stabilität für Einschrittverfahren

Die Eigenschaften der Euler-Verfahren bei Anwendung auf die Dahlquist'-sche Testgleichung (4.4) motiviert die Definition eines Stabilitätskonzepts. Stabilität bedeutet hier eine notwendige (nicht hinreichende) Bedingung um geeignete Näherungen zu erhalten. In diesem Abschnitt betrachten wir nur Einschrittverfahren.

Definition 4.1 (A-Stabilität für Einschrittverfahren)
Ein Einschrittverfahren heißt A-stabil, wenn die zugehörige Folge von Näherungen $(y_j)_{j\in\mathbb{N}}$ zur Dahlquist-Gleichung (4.4) mit $Re(\lambda) \leq 0$ für alle Schrittweiten $h > 0$ nicht ansteigt, d.h. $|y_{j+1}| \leq |y_j|$ gilt für alle j.

Wenn ein Einschrittverfahren A-stabil ist, dann ist es geeignet zur numerischen Lösung von steifen linearen Differentialgleichungssystemen. Umgekehrt sollte eine Methode, die nicht A-stabil ist, auch nicht bei (linearen oder nichtlinearen) steifen Problemen verwendet werden.

Wir möchten eine Technik erhalten, mit der nachgewiesen werden kann, ob ein Verfahren A-stabil ist oder nicht. Als Abkürzung sei $z := h\lambda \in \mathbb{C}$. Auf einem äquidistanten Gitter $x_j = x_0 + jh$ erfüllt die exakte Lösung der Dahlquist-Gleichung (4.4)

$$y(x_{j+1}) = \mathrm{e}^{h\lambda} y(x_j) = \mathrm{e}^{z} y(x_j).$$

Dadurch gilt $|y(x_{j+1})| \leq |y(x_j)|$ genau dann, wenn $\mathrm{Re}(\lambda) \leq 0$, welches äquivalent ist zu $\mathrm{Re}(z) \leq 0$.

Definition 4.2 (Stabilitätsfunktion eines Einschrittverfahrens)
Falls ein Einschrittverfahren bei Anwendung auf die Dahlquist-Gleichung in der Form $y_{j+1} = R(z)y_j$ mit $z = h\lambda$ geschrieben werden kann, dann heißt $R : \mathbb{C} \to \mathbb{C}$ die Stabilitätsfunktion des Verfahrens.

Die Euler-Verfahren angewendet auf die Dahlquist'sche Testgleichung lauten

$$y_{j+1} = R(z)y_j$$

mit

$$\text{expl. Euler}: \quad R(z) = 1 + z, \qquad \text{impl. Euler}: \quad R(z) = \frac{1}{1-z}.$$

Wir möchten, dass $|R(z)| \leq 1$ für alle z mit $\mathrm{Re}(z) \leq 0$ erfüllt ist. Jedes Einschrittverfahren besitzt eine Darstellung $y_1 = R(z)y_0$. Die Abbildung $R : \mathbb{C} \to \mathbb{C}$ nennt man die Stabilitätsfunktion des Verfahrens.

Definition 4.3 (Stabilitätsgebiet von Einschrittverfahren)
Das Stabilitätsgebiet $S \subset \mathbb{C}$ eines Einschrittverfahrens $y_1 = R(z)y_0$ ist die Menge

$$S := \{z \in \mathbb{C} : |R(z)| \leq 1\}.$$

Desweiteren sei $\mathbb{C}^- := \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. Dadurch lässt sich die A-Stabilität charakterisieren durch

$$\text{A-stabil} \quad \Leftrightarrow \quad |R(z)| \leq 1 \ \text{ für alle } \ z \in \mathbb{C}^- \quad \Leftrightarrow \quad \mathbb{C}^- \subseteq S.$$

Für die Euler-Verfahren erhalten wir die Stabilitätsgebiete

$$\begin{aligned} &\text{expl. Euler:} && S = \{z \in \mathbb{C} : |1+z| \leq 1\}, \\ &\text{impl. Euler:} && S = \left\{z \in \mathbb{C} : \left|\tfrac{1}{1-z}\right| \leq 1\right\} = \{z \in \mathbb{C} : 1 \leq |1-z|\}. \end{aligned}$$

Diese Stabilitätsgebiete sind das Innere eines Kreises um $z = -1$ mit Radius 1 bzw. das Äußere eines Kreises um $z = 1$ mit Radius 1, siehe Abb. 15. Dadurch gilt $\mathbb{C}^- \subseteq S$ für das implizite Euler-Verfahren, jedoch nicht für das explizite Euler-Verfahren.

Beispiel: Trapezregel

Die Trapezregel angewendet auf die Dahlquist'sche Testgleichung (4.4) liefert

$$y_1 = y_0 + \tfrac{h}{2}\left[\lambda y_0 + \lambda y_1\right].$$

Es folgt

$$y_1 = \frac{1 + \frac{1}{2}h\lambda}{1 - \frac{1}{2}h\lambda}\, y_0.$$

Die Stabilitätsfunktion ergibt sich zu

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}.$$

Eine genauere Untersuchung ergibt, dass hier $S = \mathbb{C}^-$ erfüllt ist. Somit ist die Trapezregel A-stabil.

Beispiel: Explizite Mittelpunktregel (Collatz-Verfahren)

Die explizite Mittelpunktregel (2.6) führt auf

$$y_1 = y_0 + h\lambda\left(y_0 + \tfrac{h}{2}\lambda y_0\right) = \left(1 + h\lambda + \tfrac{1}{2}h^2\lambda^2\right) y_0$$

bei der Dahlquist-Gleichung (4.4). Es folgt die Stabilitätsfunktion

$$R(z) = 1 + z + \tfrac{1}{2}z^2.$$

Die explizite Mittelpunktregel ist nicht A-stabil, da das Stabilitätsgebiet beschränkt ist.

Abb. 15 demonstriert die Stabilitätsgebiete dieser vier grundlegenden Einschrittverfahren, vergleiche Abschnitt 2.2.

Allgemeines Runge-Kutta-Verfahren

Ein allgemeines Runge-Kutta-Verfahren mit s Stufen zum Anfangswertproblem $y' = f(x, y)$, $y(x_0) = y_0$ lautet

$$\begin{aligned} k_i &= f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij} k_j\right) \qquad \text{für } i = 1, \ldots, s\ , \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned}$$

Die Methode ist eindeutig festgelegt durch ihre Koeffizienten

$$c = (c_i) \in \mathbb{R}^s, \qquad b = (b_i) \in \mathbb{R}^s, \qquad A = (a_{ij}) \in \mathbb{R}^{s\times s}.$$

Im Fall der Dahlquist'schen Testgleichung $y' = \lambda y$ kann eine Formel für die Stabilitätsfunktion des Verfahrens hergeleitet werden. Diese Formel gilt für sowohl explizite als auch implizite Verfahren.

Satz 4.1 (Stabilitätsfunktion eines Runge-Kutta-Verfahren)
Die Stabilitätsfunktion eines Runge-Kutta-Verfahrens ist gegeben durch

$$R(z) = 1 + zb^\top (I - zA)^{-1}\mathbb{1} \tag{4.7}$$

mit $\mathbb{1} := (1, \ldots, 1)^\top \in \mathbb{R}^s$ *und der Einheitsmatrix* $I \in \mathbb{R}^{s\times s}$ *oder äquivalent*

$$R(z) = \frac{\det(I - zA + z\mathbb{1}b^\top)}{\det(I - zA)}. \tag{4.8}$$

Satz 4.1 zeigt, dass die Stabilitätsfunktion eines Runge-Kutta-Verahrens eine rationale Funktion in der unabhängigen Veränderlichen z darstellt. Die Stabilitätsfunktion ist nicht definiert an Stellen mit $\det(I - zA) = 0$. Daher kann eine endliche Anzahl von Polen auftreten.

Für A-stabile Runge-Kutta Verfahren zeigt sich folgende Einschränkung.

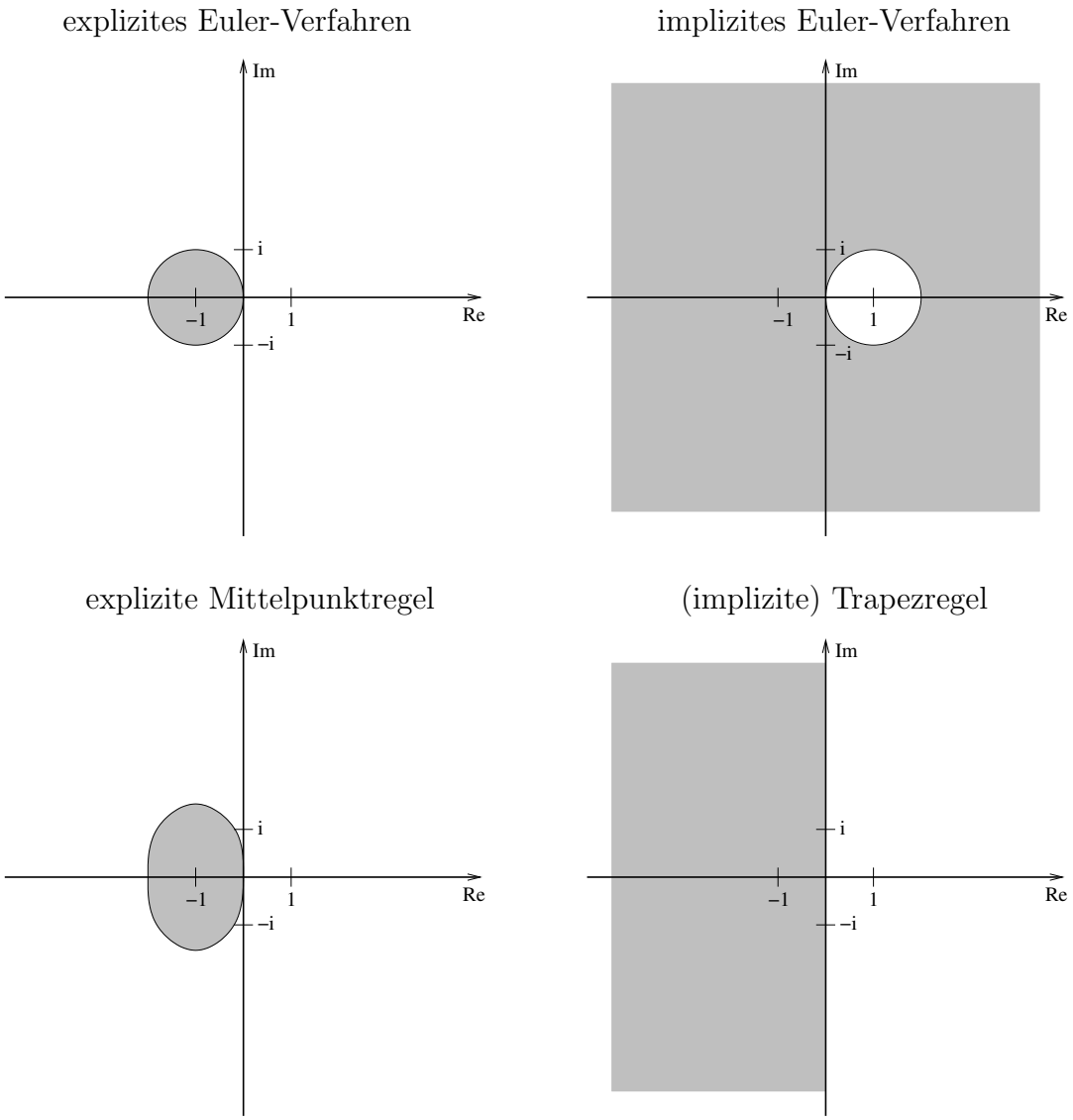


Abbildung 15: Stabilitätsgebiete für einige wichtige Einschrittverfahren.

Satz 4.2 *Ein konvergentes explizites Runge-Kutta-Verfahren ist niemals A-stabil.*

Beweis:

Ein explizites Runge-Kutta-Verfahren besitzt eine strikte untere Dreiecksmatrix A. Es folgt $\det(I - zA) = 1$ für alle $z \in \mathbb{C}$. Die Stabilitätsfunktion eines expliziten Runge-Kutta-Verfahrens ist laut Formel (4.8) somit ein Polynom

$$R(z) = \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \cdots + \alpha_{s-1} z^{s-1} + \alpha_s z^s.$$

Damit gilt

$$|R(z)| \overset{\mathrm{Re}(z)\to-\infty}{\longrightarrow} +\infty$$

sofern das Polynom nicht konstant ist.

Wäre das Polynom konstant, d.h. $R(z) = C$, dann hätte man bei der Dahlquist'schen Testgleichung die Näherungen $y_{j+1} = Cy_j$ und insbesondere $|y_j| = |C|^j |y_0|$. Die Folge der Näherungen wäre damit entweder monoton fallend oder streng monoton steigend je nach Konstante C. Mit der Konvergenz des Verfahrens muss jedoch die Folge streng monoton steigen für $\mathrm{Re}(\lambda) > 0$ und streng monoton fallen für $\mathrm{Re}(\lambda) < 0$. Dies ergibt einen Widerspruch und das Polynom muss konstant sein.

Ein explizites Runge-Kutta-Methode kann daher nicht A-stabil sein. □

Nur implizite Runge-Kutta-Verfahren können somit A-stabil sein. Jedoch ist nicht jedes implizite Runge-Kutta-Verfahren A-stabil.

L-Stabilität

Das Konzept der L-Stabilität stellt eine Verschärfung der A-Stabilität dar. Wieder beruht diese Bedingung auf der Dahlquist'schen Testgleichung (4.4). Die exakte Lösung erfüllt die Gleichung

$$y(h) = \mathrm{e}^z y(0) \qquad \text{mit } \; z = h\lambda.$$

Im Grenzfall von Parametern λ mit riesigem negativen Realteil folgt

$$\lim_{\mathrm{Re}(z)\to-\infty} y(h) = y(0) \lim_{\mathrm{Re}(z)\to-\infty} \mathrm{e}^z = 0.$$

Wir möchten, dass die Näherungen

$$y_1 = R(z)y_0$$

aus einem Einschrittverfahren diese Eigenschaft erben.

Definition 4.4 (L-Stabilität für Einschrittverfahren)
Ein Einschrittverfahren heißt L-stabil, wenn es A-stabil ist und zusätzlich gilt

$$\lim_{z\to\infty} R(z) = 0.$$

L-stabile Verfahren eignen sich zur numerischen Lösung von extrem steifen Differentialgleichungen. Man beachte, dass $R(z)$ eine rationale Funktion bei Runge-Kutta-Verfahren ist. Daher gilt

$$\lim_{\mathrm{Re}(z)\to-\infty} R(z) = \lim_{|z|\to\infty} R(z) = \lim_{z\to\infty} R(z)$$

vorausgesetzt die Grenzwerte existieren. Eine rationale Funktion $R(z)$ besitzt die Gestalt

$$R(z) = \frac{a_0 + a_1 z + \cdots + a_{n-1}z^{n-1} + a_n z^n}{b_0 + b_1 z + \cdots + b_{m-1}z^{m-1} + b_m z^m}$$

mit $a_n, b_m \neq 0$. Somit folgt

$$\lim_{z\to\infty} |R(z)| \begin{cases} = 0 & \text{für } n < m, \\ = \left|\frac{a_n}{b_n}\right| & \text{für } n = m, \\ \to \infty & \text{für } n > m. \end{cases}$$

Das implizite Euler-Verfahren ist auch L-stabil, weil

$$\lim_{z\to\infty} R(z) = \lim_{z\to\infty} \frac{1}{1-z} = 0.$$

Jedoch folgt bei der Trapezregel für $\omega \in \mathbb{R}$

$$|R(\mathrm{i}\omega)|^2 = \frac{|1 + \frac{1}{2}\mathrm{i}\omega|^2}{|1 - \frac{1}{2}\mathrm{i}\omega|^2} = \frac{1 + \frac{1}{4}\omega^2}{1 + \frac{1}{4}\omega^2} = 1.$$

Da $R(z)$ eine rationale Funktion ist, ergibt sich

$$\lim_{z\to\infty} R(z) = 1$$

und daher ist die Trapezregel nicht L-stabil. Somit ist die Trapezregel ungeeignet für extrem steife lineare Probleme.

Lösung der nichtlinearen Gleichungssysteme

Es wird noch die effiziente Lösung der nichtlinearen Gleichungssysteme, die bei impliziten Runge-Kutta-Verfahren entstehen, angesprochen. In einem einzelnen Integrationsschritt zu einem nichtlinearen Differentialgleichungssystem $y' = f(x, y)$ mit $y : \mathbb{R} \to \mathbb{R}^n$ entsteht das Verfahren

$$\begin{aligned} k_i &= f\left(x_i, y_0 + h\sum_{j=1}^{s} a_{ij}k_j\right) \qquad \text{für } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i \end{aligned}$$

mit $x_i := x_0 + c_i h$. Darin enthalten ist das nichtlineare Gleichungssystem $G(K) = 0$ mit $K \in \mathbb{R}^{sn}$ und $G : \mathbb{R}^{sn} \to \mathbb{R}^{sn}$

$$K := \begin{pmatrix} k_1 \\ \vdots \\ k_s \end{pmatrix}, \qquad G(K) := \begin{pmatrix} k_1 - f\left(x_1, y_0 + h\sum_{j=1}^{s} a_{1j}k_j\right) \\ \vdots \\ k_s - f\left(x_s, y_0 + h\sum_{j=1}^{s} a_{sj}k_j\right) \end{pmatrix}.$$

Dieses nichtlineare Gleichungssystem wird mit dem vereinfachten Newton-Verfahren iterativ gelöst. Da $k_i = \mathcal{O}(h)$ gilt, sind als Startwerte $k_i^{(0)} = 0$ für $i = 1, \ldots, s$ geeignet. Die Iterationsvorschrift lautet

$$\begin{aligned} \mathrm{D}G(K^{(0)})\Delta K^{(\nu)} &= -G(K^{(\nu)}) \\ K^{(\nu+1)} &= K^{(\nu)} + \Delta K^{(\nu)} \qquad \text{für } \nu = 0, 1, 2, \ldots\ . \end{aligned}$$

Die darin auftretende Iterationsmatrix besitzt wegen $K^{(0)} = 0$ die Gestalt

$$\mathrm{D}G(K^{(0)}) := I_{sn} - h \begin{pmatrix} a_{11}\mathrm{D}f\,(x_1, y_0) & \cdots & a_{1s}\mathrm{D}f\,(x_1, y_0) \\ \vdots & & \vdots \\ a_{s1}\mathrm{D}f\,(x_s, y_0) & \cdots & a_{ss}\mathrm{D}f\,(x_s, y_0) \end{pmatrix}.$$

Da $x_i = x_0 = c_i h = x_0 + \mathcal{O}(h)$ gilt, kann man die Matrix weiter vereinfachen, indem $\mathrm{D}f\,(x_i, y_0)$ durch $\mathrm{D}f\,(x_0, y_0)$ für alle $i = 1, \ldots, s$ ersetzt wird. Dadurch ist zur Berechnung der Iterationsmatrix $\mathrm{D}G(K^{(0)})$ nur eine Jacobi-Matrix der Funktion f auszuwerten.

Der Rechenaufwand für eine LR-Zerlegung der Matrix $\mathrm{D}G$ beträgt etwa $\frac{2}{3}(sn)^3$ Operationen, d.h. eine Proportionalität zu s^3n^3. Ist die Matrix A^{-1} jedoch reell diagonalisierbar, dann kann die Iterationsmatrix $\mathrm{D}G$ mit einem Rechenaufwand von nur ca. sn Operationen auf eine Block-Diagonalform mit s Blöcken der Dimension n transformiert werden. Somit sind nur s separate LR-Zerlegungen erforderlich und der Rechenaufwand ca. $\frac{2}{3}n^3$ pro Teilsystem.

Zudem kann ein besonders günstiger Rechenaufwand in den folgenden beiden Spezialfällen von Runge-Kutta-Verfahren erhalten werden.

Definition 4.5 (Diagonal-implizite Verfahren)
Ein Runge-Kutta-Verfahren (2.20) mit der Matrix $A \in \mathbb{R}^{s\times s}$ der inneren Gewichte heißt

- *diagonal-implizites R.-K.-V. (DIRK), wenn A eine untere Dreiecksmatrix ist (d.h. $a_{ij} = 0$ für $i < j$) und $a_{ii} \neq 0$ für ein i gilt,*
- *einfach diagonal-implizites R.-K.-V. (SDIRK von engl. singly DIRK), falls zudem noch $a_{11} = a_{22} = \cdots = a_{ss}$ gilt.*

Bei einer DIRK Methode müssen nur s separate LR-Zerlegungen der Dimension n durchgeführt werden und bei einer SDIRK Methode sogar nur eine einzige LR-Zerlegung der Dimension n, falls die vereinfachte Newton-Iteration eingesetzt wird.

4.4 A-Stabilität für Mehrschrittverfahren

Nun erfolgt die Untersuchung von linearen Mehrschrittverfahren (3.9) bei Anwendung auf steife Differentialgleichungen. Ein lineares Mehrschrittverfahren ist (numerisch) stabil genau dann, wenn sein charakteristisches Polynom die Wurzelbedingung aus Def. 3.5 erfüllt.

Die Anwendung eines linearen k-Schritt-Verfahrens (3.9) auf die Dahlquist-Testgleichung (4.4) führt auf die homogene lineare Differenzengleichung

$$\sum_{\ell=0}^{k} \alpha_\ell y_{i+\ell} = h \sum_{\ell=0}^{k} \beta_\ell \lambda y_{i+\ell} \qquad \Rightarrow \qquad \sum_{\ell=0}^{k} (\alpha_\ell - h\lambda\beta_\ell)\, y_{i+\ell} = 0.$$

Mit $z := h\lambda$ lautet das zugehörige charakteristische Polynom

$$q_z : \mathbb{C} \to \mathbb{C}, \qquad q_z(\xi) = \sum_{\ell=0}^{k} (\alpha_\ell - z\beta_\ell)\xi^\ell. \tag{4.9}$$

Man kann zeigen, dass alle Lösungen dieser Differenzengleichung beschränkt sind genau dann, wenn das charakteristische Polynom die Wurzelbedingung erfüllt. Die Nullstellen $\xi_1, \ldots, \xi_k$ von q_z hängen von z ab.

Sei $\mathrm{Re}(\lambda) \leq 0$ in der Dahlquist'schen Testgleichung (4.4). Die exakten Lösungen sind vom Betrag her dann monoton fallend. Insbesondere sind sie dadurch beschränkt. Die numerische Lösung aus einem linearen Mehrschrittverfahren kann zunächst leicht ansteigen. Daher wird nur gefordert, dass die numerische Lösung für alle Schrittweiten beschränkt ist. Die Wurzelbedingung führt auf die folgende Definition.

Definition 4.6 (Stabilitätsgebiet eines Mehrschrittverfahrens)
Das Stabilitätsgebiet $S \subset \mathbb{C}$ eines linearen Mehrschrittverfahrens ist

$$S := \{z \in \mathbb{C} : \textit{für alle Nullstellen } \xi \textit{ von } q_z \textit{ gilt } |\xi| \leq 1 \textit{ und } |\xi| < 1 \textit{ für mehrfache Nullstellen}\}.$$

Jetzt kann die A-Stabilität von Mehrschrittverfahren wie bei Einschrittverfahren festgelegt werden.

Definition 4.7 (A-Stabilität von Mehrschrittverfahren)
Ein lineares Mehrschrittverfahren heißt A-stabil, wenn sein Stabilitätsgebiet die Bedingung $\mathbb{C}^- \subseteq S$ erfüllt.

Man kann zeigen, dass die A-Stabilität von Einschrittverfahren aus Def. 4.1 äquivalent ist zur A-Stabilität von linearen Einschrittverfahren aus Def. 4.7. Ein lineares Einschrittverfahren ($k = 1$) besitzt die Gestalt

$$\alpha_1 y_1 + \alpha_0 y_0 = h\left[\beta_1 f(x_1, y_1) + \beta_0 f(x_0, y_0)\right].$$

Bei Anwendung auf die Testgleichung (4.4) folgt

$$\alpha_1 y_1 + \alpha_0 y_0 = h\left[\beta_1 \lambda y_1 + \beta_0 \lambda y_0\right] \quad \Rightarrow \quad y_1 = \underbrace{\frac{-\alpha_0 + z\beta_0}{\alpha_1 - z\beta_1}}_{R(z)} y_0.$$

Das zugehörige charakteristisches Polynom lautet

$$q_z(\xi) = (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0).$$

Dieses Polynom besitzt nur die einfache Nullstelle

$$\xi_1(z) = \frac{-\alpha_0 + z\beta_0}{\alpha_1 - z\beta_1}.$$

Andererseits stimmt die Stabilitätsfunktion $R(z)$ dieses Einschrittverfahrens mit der Nullstelle $\xi_1(z)$ des Polynoms überein. Folglich sind die Bedingungen $|R(z)| \leq 1$ und $|\xi_1(z)| \leq 1$ für alle z mit $\mathrm{Re}(z) \leq 0$ äquivalent.

Das Konzept der A-Stabilität für lineare k-Schritt-Verfahren ist im Fall $k > 1$ etwas schwächer als bei Einschrittverfahren. Der Spezialfall $z = 0$ entspricht der numerischen Stabiltiät des Mehrschrittverfahrens, siehe Def. 3.5.

Wieder ergibt sich, dass explizite Verfahren nicht geeignet für steife Differentialgleichungen sind.

Satz 4.3 *Ein konvergentes explizites lineares Mehrschrittverfahren hat ein beschränktes Stabilitätsgebiet und ist somit nie A-stabil.*

Beweis:

Ein explizites lineares Mehrschrittverfahren (3.9) besitzt die Eigenschaft $\beta_k = 0$. O.E.d.A. sei $\alpha_k = 1$. Das charakteristische Polynom bei Anwendung auf die Dahlquist'sche Testgleichung (4.4) lautet

$$q_z(\xi) = \xi^k + (\alpha_{k-1} - z\beta_{k-1})\xi^{k-1} + \cdots + (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0).$$

Das Polynom kann in der Gestalt

$$\begin{aligned} q_z(\xi) &= \xi^k + \gamma_{k-1}(z)\xi^{k-1} + \cdots + \gamma_1(z)\xi + \gamma_0(z) \\ &= (\xi - \xi_1(z))(\xi - \xi_2(z)) \cdots (\xi - \xi_k(z)) \end{aligned}$$

geschrieben werden mit den Nullstellen $\xi_1, \ldots, \xi_k \in \mathbb{C}$ abhängig von z. Der Satz von Vieta, siehe S. 171 in [8], liefert die Formel

$$\gamma_{k-i}(z) = (-1)^i \sum_{1 \le j_1 < j_2 < \cdots < j_i \le k} \xi_{j_1}(z)\xi_{j_2}(z) \cdots \xi_{j_i}(z).$$

Angenommen das Stabilitätsgebiet S wäre unbeschränkt. Dann gibt es eine Folge $(z_i)_{i\in\mathbb{N}} \subset S$ mit $|z_i| \to \infty$. Da das Verfahren konvergent ist, muss mindestens ein Koeffizient $\beta_\ell \neq 0$ auftreten. Es folgt $|\alpha_\ell - z_i\beta_\ell| \to \infty$. Somit wird ein Koeffizient $\gamma_\ell(z)$ von q_z unbeschränkt entlang der Folge. Wären alle Nullstellen beschränkt entlang der Folge, dann wären durch die Formel von Vieta alle Koeffizienten beschränkt. Also muss mindestens eine Nullstelle $\xi_j(z)$ unbeschränkt entlang der Folge sein. Dadurch gilt $|\xi_j(z_i)| > 1$ für unendlich viele i. Aus Def. 4.6 des Stabilitätsgebiets S folgt dann $z_i \notin S$ für unendlich viele i. Dies steht im Widerspruch zu $z_i \in S$ für alle i. Also ist S beschränkt. Die Bedingung $\mathbb{C}^- \subseteq S$ aus Def. 4.7 kann nicht gelten. □

Für implizite lineare Mehrschrittverfahren gilt $\beta_k \neq 0$ und das charakteristische Polynom lautet

$$q_z(\xi) = (\alpha_k - z\beta_k)\xi^k + (\alpha_{k-1} - z\beta_{k-1})\xi^{k-1} + \cdots + (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0).$$

die Nullstellen dieses Polynoms sind die gleichen wie von

$$\tilde{q}_z(\xi) = \xi^k + \frac{\alpha_{k-1} - z\beta_{k-1}}{\alpha_k - z\beta_k}\,\xi^{k-1} + \cdots + \frac{\alpha_1 - z\beta_1}{\alpha_k - z\beta_k}\,\xi + \frac{\alpha_0 - z\beta_0}{\alpha_k - z\beta_k}$$

unter der Voraussetzung $\alpha_k - z\beta_k \neq 0$. Nun sind die Koeffizienten rationale Funktionen in der Variablen z. Die Koeffizienten sind beschränkt für $|z| \to \infty$. A-stabile Mehrschrittverfahren sind eine Teilmenge der impliziten Verfahren. Jedoch gilt für die A-Stabilität von Mehrschrittverfahren noch eine wesentliche Einschränkung.

Satz 4.4 (zweite Dahlquist-Schranke)
Ein lineares Mehrschrittverfahren, das konvergent von Ordnung $p > 2$ ist, kann nicht A-stabil sein.

Für ein k-Schritt-Verfahren möchte man eine Konvergenzordnung $p \geq k$ erhalten (z.B. Adams-Verfahren, BDF-Methoden). Daher gibt es keine A-stabilen Verfahren mit $k > 2$ Schritten und Konvergenzordnung $p \geq k$.

A(α)-Stabilität

Die BDF-Verfahren mit $k = 1$ und $k = 2$ Schritten sind A-stabil, während die BDF-Verfahren für $k \geq 3$ nicht A-stabil sind. Trotzdem zeigen die BDF-Methoden für $k = 3, 4, 5$ ein gutes Verhalten bei der Lösung von steifen Problemen. Die Gestalt ihrer Stabilitätsgebiete legt eine Modifikation des Konzepts der A-Stabilität nahe. Für $0 \leq \alpha \leq \frac{\pi}{2}$ sei $\mathbb{C}_\alpha \subset \mathbb{C}$ mit

$$\mathbb{C}_\alpha := \left\{ z = |z| \cdot \mathrm{e}^{\mathrm{i}\varphi} \in \mathbb{C} : |\pi - \varphi| \leq \alpha \right\}.$$

Abb. 16 verdeutlicht dieses Gebiet.

Definition 4.8 (A(α)-Stabilität)
Ein (Einschritt- oder Mehrschritt-) Verfahren heißt A(α)-stabil mit einem $\alpha \in [0, \frac{\pi}{2}]$, wenn sein Stabilitätsgebiet S die Bedingung $\mathbb{C}_\alpha \subseteq S$ erfüllt.

Ein Verfahren wird natürlicherweise durch das maximale α gekennzeichnet, für das die A(α)-Stabilität noch gilt. Der Spezialfall $\alpha = \frac{\pi}{2}$ entspricht der gewöhnlichen A-Stabilität wegen $\mathbb{C}_{\pi/2} = \mathbb{C}^-$. Ist α nahe $\frac{\pi}{2}$, dann ist die Methode für steife Differentialgleichungen noch geeignet.

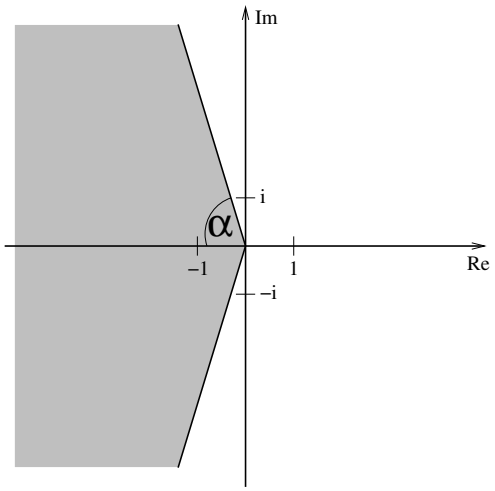


Abbildung 16: Gebiet $\mathbb{C}_\alpha$ zur A(α)-Stabilität.

Die k-Schritt BDF-Verfahren besitzen die maximalen Winkel:

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°

Die BDF-Methoden für $k \geq 7$ sind nicht A(α)-stabil für beliebiges $\alpha \geq 0$. Diese Methoden sind auch nicht mehr numerisch stabil und daher uninteressant.

L-Stabilität

Eine Übertragung des Konzepts der L-Stabilität von Einschrittverfahren auf Mehrschrittverfahren kann wie folgt geschehen.

Definition 4.9 (L-Stabilität für Mehrschrittverfahren)
Ein Mehrschrittverfahren heißt L-stabil, wenn es A-stabil ist und zusätzlich für die Nullstellen $\xi_1(z), \ldots, \xi_k(z)$ des Polynoms $q_z(\xi)$ aus (4.9) gilt

$$\lim_{z\to\infty} \left(\max_{j=1,\ldots,k} |\xi_j(z)| \right) = 0.$$

Die Def. 4.9 ist im Einklang mit Def. 4.4 bei linearen Einschrittverfahren. Für lineare Einschrittverfahren liegt nur eine Nullstelle $\xi_1(z)$ vor und diese

erfüllt $\xi_1(z) = R(z)$. Somit gilt die Äquivalenz

$$\lim_{z\to\infty} |\xi_1(z)| = 0 \quad \Leftrightarrow \quad \lim_{z\to\infty} |R(z)| = 0.$$

Nur wenige lineare Mehrschrittverfahren sind L-stabil. Ein L-stabiles Verfahren ist die BDF2-Methode.

4.5 Vergleich der Verfahrensklassen

In diesem Abschnitt werden die allgemeinen Eigenschaften von Einschrittverfahren und Mehrschrittverfahren diskutiert und verglichen. Jeder Typ hat seine eigenen Vor- und Nachteile.

Zuerst erfolgt eine Charakterisierung des Rechenaufwands bei einem einzelnen Integrationsschritt in der nachfolgenden Tabelle. Dabei wird davon ausgegangen, dass nichtlineare Gleichungssysteme mit dem vereinfachten Newton-Verfahren iterativ gelöst werden. Zudem wird im Runge-Kutta-Verfahren vorausgesetzt, dass die linearen Gleichungssysteme auf Block-Diagonalform transformiert werden können.

	Runge-Kutta-Verfahren s Stufen	lineares Mehrschrittverfahren k Schritte
expl.	s Aufwertungen von f	eine Auswertung von f
impl.	eine Jacobi-Matrix von f LR-Zerl.: $\geq s \cdot \frac{2}{3}n^3$ Operationen (ca. $\frac{2}{3}n^3$ Op. für SDIRK) pro Newton-Schritt: s Auswertungen von f s lineare Gl.sys. der Dim. n	eine Jacobi-Matrix von f LR-Zerl.: ca. $\frac{2}{3}n^3$ Operationen pro Newton-Schritt: eine Auswertung von f ein lineares Gl.sys. der Dim. n

Zusammenfassend ergibt sich bei einem einzelnen Integrationsschritt mit einem Runge-Kutta-Verfahren ein höherer Rechenaufwand als bei einem linearen Mehrschrittverfahren. Jedoch ist für eine Beurteilung der Effizienz der Verfahren noch die Genauigkeit einzubeziehen, d.h. die Anzahl der benötigten Integrationsschritte für eine vorgegebene Genauigkeit.

Die nachfolgende Tabelle zeigt einige Vorteile und Nachteile der Einschrittverfahren gegenüber den Mehrschrittverfahren.

	Runge-Kutta-Verfahren		lineare Mehrschrittverfahren
$\ominus$	relativ hoher Rechenaufwand pro Schritt (abhängig von s)	$\oplus$	relativ geringer Rechenaufwand pro Schritt (unabhängig von k)
$\oplus$	viele Koeffizienten ($s^2 + s$) (zusätzliche Bedingungen erfüllbar)	$\ominus$	nur $2k + 1$ Koeffizienten (niedrige Anzahl an Freiheitsgraden)
$\oplus$	immer (numerisch) stabil (keine Reduzierung der Freiheitsgrade)	$\ominus$	Wurzelbedingung für Stabilität erforderlich (Reduzierung der Freiheitsgrade, erste Dahlquist-Schranke)
$\oplus$	Verfahren hoher Ordnung für steife Probleme (A-Stabilität)	$\ominus$	nur Verfahren niedriger Ordnung sind A-stabil (zweite Dahlquist-Schranke), nur A(α)-stabile Verfahren höherer Ordnung
$\oplus$	robuste Schrittweitensteuerung	$\ominus$	Stabilitätsbedingung erfordert kleine Änderungen in der Schrittweite (z.B. bei BDF-Verfahren)
$\ominus$	keine effiziente Ordnungssteuerung	$\oplus$	effiziente Ordnungssteuerung

Man kann nicht folgern, dass Einschrittverfahren oder Mehrschrittverfahren im allgemeinen besser sind. Es hängt stets vom System der Differentialgleichungen ab, ob ein Verfahren besser geeignet als eine andere Methode ist.

Verfahren in MATLAB

In der Software MATLAB (MATrix LABoratory), Version 9.8.0 (R2020a) sind sieben Funktionen zur numerischen Lösung von Anfangswertproblemen zu gewöhnlichen Differentialgleichungen $y' = f(x, y),\ y(x_0) = y_0$ verfügbar. Die nachfolgende Tabelle listet diese Algorithmen auf. Die meisten dieser Methoden wurden in den vorangegangenen Kapiteln besprochen. Der Problemtyp, für das ein Verfahren geeignet ist, wird angegeben. Alle Verfahren verwenden eine Schrittweitensteuerung zur Kontrolle des lokalen Diskretisierungsfehlers. Die Tabelle zeigt, welche Technik zur Schätzung des lokalen Fehlers verwendet wird. Desweiteren benutzen zwei der Methoden eine Ordnungssteuerung. Für weitere Einzelheiten zu diesen Verfahren siehe [6].

Tabelle 6: Verfahren für Anfangswertprobleme in MATLAB.

Code	Methode	Problemtyp	Schrittweitensteuerung	Ordnungssteuerung
`ode23`	explizites Runge-Kutta-Verfahren	nicht steif	eingebettetes Verfahren	nein
`ode45`	explizites Runge-Kutta-Verfahren	nicht steif	eingebettetes Verfahren	nein
`ode113`	Prädiktor-Korrektor-Verfahren Adams-Methoden	nicht steif	Differenz zwischen Prädiktor und Korrektor	ja Ordnungen 1-13
`ode23t`	Trapezregel	moderat steif	kubischer Interpolant für dritte Ableitung	nein
`ode23s`	Rosenbrock-Wanner-Verfahren	steif	eingebettetes Verfahren	nein
`ode15s`	Numerical Differentiation F. (NDF) optional: BDF	steif	Interpolations-polynome	ja Ordnungen 1-5
`ode23tb`	Trapezregel und BDF2 (abwechselnd)	steif	eingebettetes Verfahren	nein

Literatur

[1] M. Braun: Differentialgleichungen und ihre Anwendungen. (2. Aufl.) Springer, 1991.

[2] P. Deuflhard, F. Bornemann: Scientific Computing with Ordinary Differential Equations. Springer, 2002.

[3] O. Forster: Analysis 2. (10. Aufl.) Springer Spektrum, 2013.

[4] E. Hairer, S.P. Nørsett, G. Wanner: Solving Ordinary Differential Equations. Vol. 1: Nonstiff Problems. (2nd ed.) Springer, Berlin 1993.

[5] E. Hairer, G. Wanner: Solving Ordinary Differential Equations. Vol. 2: Stiff and Differential-Algebraic Equations. (2nd ed.) Springer, Berlin 1996.

[6] L.F. Shampine, M.W. Reichelt: The MATLAB ODE suite. SIAM J. Sci. Comput. 18:1 (1997), 1–22.

[7] J. Stoer, R. Bulirsch: Numerische Mathematik 2. (5. Aufl.) Springer, 2005.

[8] W. Walter: Analysis I. (6. Aufl.) Springer, 2001.