Technische Universität Ilmenau
Institut für Mathematik

Preprint No. M 02/09

Zur Numerik gewöhnlicher Differentialgleichungen: Teil 1 Anfangswertprobleme

Vogt, Werner

August 2002

Impressum:
Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69 3621
Fax: +49 3677 69 3270
http://www.tu-ilmenau.de/ifm/

ISSN xxxx-xxxx

ilmedia

Zur Numerik gewöhnlicher Differentialgleichungen

Teil 1
Anfangswertprobleme

Werner Vogt
Technische Universität Ilmenau
Institut für Mathematik
Postfach 100565
98684 Ilmenau

Ilmenau, den 29.08.2002

Zusammenfassung Zwei grundlegende numerische Zugänge zur Lösung von Anfangswertproblemen gewöhnlicher Differentialgleichungssysteme – Einschrittverfahren und lineare Mehrschrittverfahren – werden vorgestellt, theoretisch begründet und algorithmisch aufbereitet. Neben expliziten Formeln werden auch die für steife Systeme effizienteren impliziten Verfahren betrachtet. Grundbegriffe wie Konsistenz, Stabilität und Konvergenz werden eingeführt und veranschaulicht.

1 Einführung

Das dynamische Verhalten von Systemen aus Natur- und Ingenieurwissenschaften kann in vielen Fällen durch gewöhnliche oder partielle Differentialgleichungen modelliert werden. Im Gegensatz zu linearen Gleichungen ist bei nichtlinearen Modellen jedoch nur selten eine geschlossene analytische Lösung möglich. So sind schon die sehr einfach aussehenden gewöhnlichen Differentialgleichungen

$$\dot{x} = t - x^2 \qquad \text{und} \qquad \ddot{x} = 6x^2 + t$$

nicht „analytisch“, d.h. in Termen elementarer Funktionen nach $x(t)$ auflösbar. Bei den ungleich größeren und komplizierteren Systemen der Praxis ist man deshalb fast immer auf Näherungsverfahren angewiesen. Im Gegensatz zu Reihenansätzen und sukzessiven Approximationen, die im analytischen Kontext von hohem Wert sind, haben sich im computerbasierten numerischen Rechnen, dem Scientific Computing, sogenannte Diskretisierungsverfahren durchgesetzt. Ihr Grundprinzip besteht darin, daß das unendlichdimensionale Problem der Bestimmung von Funktionen in eine endlichdimensionale Aufgabe – nämlich die Berechnung von Vektoren reellwertiger Komponenten – transformiert wird. Damit sind Diskretisierungsverfahren fast universell auf

- Anfangswertprobleme gewöhnlicher Differentialgleichungen
- Randwertprobleme gewöhnlicher Differentialgleichungen
- Bifurkationsprobleme gewöhnlicher Differentialgleichungen

und auf weitere Typen von Funktionalgleichungen, insbesondere partielle Differentialgleichungen und Integralgleichungen 1. und 2.Art anwendbar. Die nachfolgend eingeführten Begriffe sind deshalb nicht nur für gewöhnliche Differentialgleichungen bedeutsam; sie werden nur anhand dieser Problemklasse vorgestellt.

Die Simulation des Verhaltens dynamischer Systeme führt häufig auf explizite Differentialgleichungen mit vorgegebenen Anfangswerten. Betrachtet wird deshalb das Anfangswertproblem 1. Ordnung für die gesuchten n Funktionen $x_1(t), x_2(t), \ldots, x_n(t)$ der Form

$$\begin{array}{rclcrcl} \dot{x}_1 &=& f_1(t, x_1, x_2, \ldots, x_n)\,, & & x_1(a) &=& x_{01} \\ \dot{x}_2 &=& f_2(t, x_1, x_2, \ldots, x_n)\,, & & x_2(a) &=& x_{02} \\ & & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & & & & \\ \dot{x}_n &=& f_n(t, x_1, x_2, \ldots, x_n)\,, & & x_n(a) &=& x_{0n} \end{array} \tag{1}$$

auf einem endlichen Zeitintervall $I = [a, b]$. Differentialgleichungen höherer Ordnung lassen sich durch Einführung neuer Variabler auf derartige Systeme 1. Ordnung zurückführen. Ist das Anfangswertproblem für eine explizite Gleichung n-ter Ordnung

$$x^{(n)} = f(t, x, \dot{x}, \ldots, x^{(n-1)}), \quad x(a) = x_{01},\ \dot{x}(a) = x_{02}, \ldots, x^{(n-1)}(a) = x_{0n}$$

gegeben, so erhält man mit den neuen Funktionen

$$x_1(t) = x(t),\ x_2(t) = \dot{x}(t), \ldots, x_n(t) = x^{(n-1)}(t)$$

das spezielle System

$$\begin{array}{lll} \dot{x}_1 = x_2 & , & x_1(a) = x_{01} \\ \dot{x}_2 = x_3 & , & x_2(a) = x_{02} \\ \dot{x}_3 = x_4 & , & x_3(a) = x_{03} \\ \cdots\cdots\cdots & & \\ \dot{x}_{n-1} = x_n & , & x_{n-1}(a) = x_{0,n-1} \\ \dot{x}_n = f(t, x_1, x_2, \ldots, x_n) & , & x_n(a) = x_{0n}\,. \end{array}$$

Ähnlich kann man Differentialgleichungsssysteme höherer Ordnung reduzieren (Übungsaufgabe). Während früher zahlreiche Diskretisierungsverfahren für spezielle Klassen von Anfangswertproblemen entwickelt wurden, geht man nun vom allgemeinen Modell (1) aus und berücksichtigt dessen spezielle Struktur meist nur in großdimensionalen Anwendungsfällen. Führt man die Funktionenvektoren

$$x = (x_1, x_2, \ldots, x_n)^T,\ f = (f_1, f_2, \ldots, f_n)^T, \quad \text{sowie} \quad x_0 = (x_{01}, x_{02}, \ldots, x_{0n})^T$$

ein, so läßt sich System (1) in Vektorschreibweise

$$\frac{dx}{dt} = \dot{x} = f(t, x), \quad x(a) = x_0, \quad f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n \tag{2}$$

mit gegebenem Anfangsvektor x_0 notieren. Im Gegensatz zu exakten (analytischen) Lösungsverfahren verfügen numerische Verfahren über fast universelle Einsatzmöglichkeiten, unabhängig von der konkreten Form der Differentialgleichungen. So setzen wir zur Lösung dieser Anfangswertaufgabe auf einem endlichen Zeitintervall $I = [a, b]$ voraus, daß die Funktion $f(t, x)$ auf dem Bereich $D = I \times \mathbb{R}^n$ stetig ist. Wir wollen folgende grundlegende Annahme treffen:

Voraussetzung 1.1 (Stetigkeiteit und Existenz)
f ist stetig auf $I \times \mathbb{R}^n$ und Problem (2) besitzt genau eine Lösung $x(t)$ auf ganz I.

Offenbar ist diese Lösung x dann stetig differenzierbar auf I. Zudem läßt sich eine hinreichend große Konstante $M > 0$ angeben, so daß die Lösungskurve $(t, x^*(t)),\ a < t < b$ im Innern der kompakten Menge

$$S = \{(t, x) \mid a \le t \le b,\ ||x|| \le M\}$$

verläuft. Hinreichende Bedingungen, die die Existenz- und Eindeutigkeit der Lösung garantieren,z.B. globale Lipschitz-Stetigkeit, findet man in der angegebenen Literatur.

2 Explizite Einschrittverfahren

Die Lösung $x(t)$ des Anfangswertproblems (2) soll nun näherungsweise durch Diskretisierung ermittelt werden, d. h. approximierende Lösungswerte werden nicht auf dem Gesamtintervall $I = [a, b]$, sondern nur auf einem endlichen Gitter

$$I_N = \{\ t_j \mid t_j = t_{j-1} + h_j,\ h_j > 0,\ j = 1(1)N,\ t_0 = a,\ t_N = b\ \}$$

mit den $N+1$ Gitterpunkten t_j und den positiven Schrittweiten h_j bestimmt. Das Gitter soll vorerst als *äquidistant* angenommen werden, d.h. die Schrittweiten h_j seien konstant mit dem Wert $h = h_j = (b-a)/N$. Ein *Diskretisierungsverfahren* ist ein Algorithmus, der zu jedem Gitterpunkt t_j eine diskrete Lösung u_j liefert, die die exakte Lösung $x(t_j)$ approximiert. Die eindeutige Abbildung $\varphi : I_N \to \mathbb{R}^{n(N+1)}$, die jedem Gitter einen Supervektor $u = (u_0, u_1, \ldots, u_N)$ (also einen Vektor aus den Vektoren $u_0, u_1, \ldots, u_N$) zuordnet, nennt man *Gitterfunktion*. Lösungswerte an Zwischenpunkten $t \neq t_j$ können anschließend relativ leicht per Interpolation ermittelt werden.

2.1 Explizites Euler-Verfahren

Für die Herleitung eines Diskretisierungsverfahrens nehmen wir nun einmal zusätzlich zu Voraussetzung 1.1 an, daß die Lösung mindestens zweimal stetig differenzierbar ist. Liegt die Lösung $x(t)$ zum Zeitpunkt t_{j-1} vor, so läßt sich $x(t_j)$ durch Taylorentwicklung an der Stelle $t = t_{j-1}$ darstellen

$$x(t_j) = x(t_{j-1}) + h\dot{x}(t_{j-1}) + \int_0^1 (1-\tau)\ddot{x}(t_{j-1}+\tau h)h^2\, d\tau\,. \tag{3}$$

Nutzt man die Differentialgleichung (2), so lassen sich die Ableitungen $\dot{x}$ und $\ddot{x}$ durch die Funktion $f(t,x)$ und deren erste Ableitungen ausdrücken. Das einfachste Diskretisierungsverfahren erhält man dann, indem man den für kleine Schrittweite h unwesentlichen Restterm in (3) vernachlässigt

$$u_j = u_{j-1} + h\ f(t_{j-1}, u_{j-1})\,, \quad j = 1(1)N. \tag{4}$$

Dieses bereits 1768 von Leonhard Euler[1] benutzte Verfahren wurde durch Augustin Louis Cauchy[2] 1840 theoretisch begründet und wird deshalb als *explizites Euler-Verfahren (Euler-Cauchy-Verfahren, Polygonzugverfahren)* bezeichnet. Anschaulich ersetzt es die Lösungskurve durch einen Polygonzug (vgl. Abb. 1). Im Punkt (t_{j-1}, u_{j-1}) wird der Anstieg $f(t_{j-1}, u_{j-1})$ des Richtungsfeldes der DGL (2) zur Bestimmung des nächsten Näherungswertes u_j benutzt, weshalb man von einem expliziten Einschrittverfahren spricht. Allgemein gilt folgende

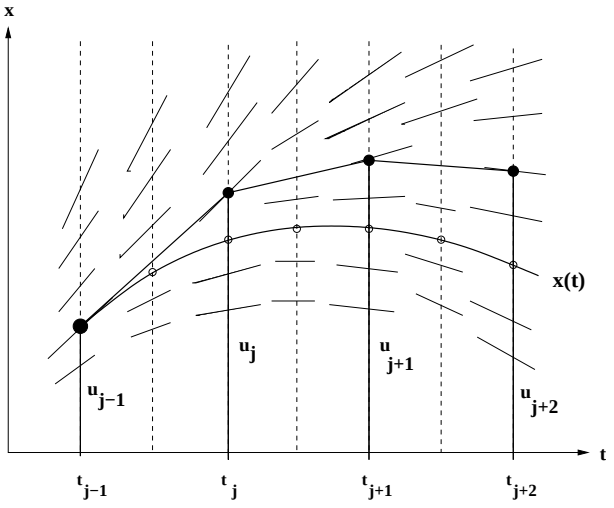

Abbildung 1: Explizites Euler-Verfahren

[1] Leonhard Euler (1707-1783), Schweizer Mathematiker, wirkte in Berlin und St.Petersburg, wo er auf fast allen Gebieten der Mathematik arbeitete

[2] Augustin Louis Cauchy (1789-1857), französischer Mathematiker, Arbeiten zur Trigonometrie, Determinanten- und Reihenlehre sowie Funktionentheorie

Definition 2.1 (Einschrittverfahren)

Ein Einschrittverfahren (ESV) erzeugt eine Gitterfunktion u nach der Vorschrift

$$\begin{aligned} u_j &= u_{j-1} + h\ \Phi(t_{j-1}, u_{j-1}, h), \\ u_0 &= x_0\,, \quad j = 1(1)N, \end{aligned} \tag{5}$$

mit der Verfahrensfunktion $\Phi : I \times \mathbb{R}^n \times H \to \mathbb{R}^n$ und dem Schrittweitenbereich $H = [0, h_0]$, $h_0 \leq b - a$. Ist Φ explizit darstellbar, so heißt das Verfahren explizit, andernfalls ist es implizit.

Beispiele 2.2

1. Das *explizite Euler-Verfahren* hat die von h unabhängige Verfahrensfunktion

$$\Phi(t, x, h) \;=\; f(t, x)\,.$$

2. Das *implizite Euler-Verfahren* erhält man durch Taylorentwicklung von $x(t_{j-1})$ an der Stelle $t = t_j$

$$x(t_{j-1}) = x(t_j) - h\dot{x}(t_j) + \int_0^1 (1-\tau)\ddot{x}(t_j - \tau h)h^2\, d\tau\,,$$

Anwendung der DGL und Vernachlässigung des Restterms:

$$u_j = u_{j-1} + h\ f(t_j, u_j)\,, \quad j = 1(1)N. \tag{6}$$

Seine Verfahrensfunktion kann implizit zu

$$\Phi(t, x, h) \;=\; f(t + h, x + h\Phi(t, x, h)) \tag{7}$$

angegeben werden. ◀

Wegen der einfachen Struktur expliziter Einschrittverfahren lassen sich grundlegende Eigenschaften von Diskretisierungsverfahren leicht daran demonstrieren und anschließend auf weitere Verfahrensklassen übertragen.

2.2 Diskretisierungsfehler

Sei $x(t_j)$ die exakte Lösung und u_j die mit dem expliziten Euler-Verfahren (4) bestimmte Näherungslösung. Dann nennt man die Differenz

$$e_j = u_j - x(t_j), \quad j = 0(1)N, \tag{8}$$

den *globalen Diskretisierungsfehler an der Stelle t_j*. Er setzt sich kumulativ aus den Einzelfehlern der vorherigen Integrationsschritte zusammen, ist jedoch einer direkten Bestimmung meist nicht zugänglich. Deshalb definiert man einen *lokalen Diskretisierungsfehler*, der genau den in (3) vernachlässigten Restterm, dividiert durch die Schrittweite h, angibt:

$$\tau_j = \frac{1}{h}[x(t_j) - x(t_{j-1})] - f(t_{j-1}, x(t_{j-1})), \quad j = 1(1)N. \tag{9}$$

Bemerkung 2.3 Umstellung dieser Formel nach dem Wert $x(t_j)$ ergibt die Darstellung

$$x(t_j) \;=\; x(t_{j-1}) + hf(t_{j-1}, x(t_{j-1})) + h\tau_j \;,$$

mit der eine anschauliche Interpretation dieses Fehlers für das explizite Euler-Verfahren möglich wird. Führt man nämlich mit dem Anfangswert $\eta_{j-1} = x(t_{j-1})$ einen einzelnen Verfahrensschritt aus, so liefert

$$\eta_j \;=\; x(t_{j-1}) + hf(t_{j-1}, x(t_{j-1})$$

genau den Näherungswert des Verfahrens mit der Fehlerdarstellung $x(t_j) = \eta_j + h\tau_j$. Der Wert τ_j stellt folglich in diesem Verfahren den durch die Schrittweite h dividierten Fehler eines einzelnen Integrationsschrittes (engl.: local error per unit step) dar. Man erhält τ_j formal aus der Verfahrensvorschrift (4), wenn man sie in der Form

$$u_j - u_{j-1} - hf(t_{j-1}, u_{j-1}) \;=\; 0$$

darstellt, anschließend durch h dividiert und schließlich die Näherungslösung durch die exakte Lösung ersetzt.

Allgemein definiert man den lokalen Diskretisierungsfehler eines Einschrittverfahrens (5), indem man den Fehler eines einzelnen Integrationsschrittes (local error per unit step) notiert.

Definition 2.4 (Lokaler Diskretisierungsfehler und Konsistenz)

(i) Der lokale Diskretisierungsfehler (Approximationsfehler) des ESV 5 an der Stelle t_j lautet

$$\tau_j = \frac{1}{h}\left[x(t_j) - x(t_{j-1})\right] - \Phi(t_{j-1}, x(t_{j-1}), h),\ j = 1(1)N.$$

(ii) Das ESV heißt konsistent (verträglich), falls gilt

$$\lim_{h\to 0} \tau_j \;=\; 0 \quad j = 1(1)N \;.$$

(iii) Die Konsistenzordnung ist $q \in \mathbb{N}$, falls Konstanten $h_0 > 0$ und $K > 0$ existieren, so daß für alle $h \in (0, h_0]$ gilt

$$||\tau_j|| \;\leq\; K \cdot h^q \;, \quad j = 1(1)N \;.$$

Mit dem Landauschen Ordnungssymbol O läßt sich die Konsistenzordnung q in der abkürzenden Schreibweise $\tau_j = O(h^q)$ notieren. Für ESV erhalten wir den folgenden

Satz 2.5 *Voraussetzung 1.1 gelte und $f \in C^1(S)$ mit obiger Menge*

$$S = \{(t, x) \mid a \leq t \leq b,\ ||x|| \leq M\ \}.$$

Ist die Verfahrensfunktion Φ stetig in $h \in [0, H]$ und genügt der Konsistenzbedingung

$$\Phi(t, x(t), 0) \;=\; f(t, x(t)) \;, \quad t \in I \;, \tag{10}$$

so ist das ESV (5) konsistent.

BEWEIS: Nach Voraussetzung ist

$$\ddot{x}(t) = f_t(t, x(t)) + f_x(t, x(t)) f(t, x(t))$$

stetig, so daß für den lokalen Diskretisierungsfehler

$$\begin{aligned} \tau_j &= \frac{1}{h}\left[x(t_j) - x(t_{j-1})\right] - \Phi(t_{j-1}, x(t_{j-1}), h) \\ &= \dot{x}(t_{j-1}) - \Phi(t_{j-1}, x(t_{j-1}), h) + O(h) \end{aligned}$$

gilt, woraus mit den Bedingungen für Φ unmittelbar $\lim_{h\to 0} \tau_j = 0$ folgt. □

Beispiel 2.6 Für alle Anfangswertprobleme mit $f \in C^1(S)$ ist das explizite Euler-Verfahren und das implizite Euler-Verfahren konsistent, denn ihre Verfahrensfunktionen in Beispiel 2.2 genügen offenbar der Bedingung $\Phi(t, x, 0) = f(t, x)$. ◀

Betrachtet man Anfangswertprobleme mit hinreichend glatten rechten Seiten f, so lassen sich Bedingungen an die Verfahrensfunktion aufstellen, die eine gewünschte Konsistenzordnung q garantieren (Ordnungsbedingungen). Angenommen, alle in τ_j eingehenden Funktionen seien analytisch, so liefert eine Taylorentwicklung aller Teilterme an der Stelle t_{j-1}

$$\begin{aligned} \tau_j &= \frac{1}{h}\left[x(t_j) - x(t_{j-1})\right] - \Phi(t_{j-1}, x(t_{j-1}), h) \\ &= \frac{1}{h}\left[\sum_{k=0}^{\infty} \frac{1}{k!} x^{(k)}(t_{j-1}) h^k - x(t_{j-1})\right] - \sum_{k=0}^{\infty} \frac{1}{k!} \frac{\partial^k \Phi}{\partial h^k}(t_{j-1}, x(t_{j-1}), 0) h^k \\ &= \sum_{k=0}^{\infty} \frac{1}{k!}\left[\frac{1}{k+1} x^{(k+1)}(t_{j-1}) - \frac{\partial^k \Phi}{\partial h^k}(t_{j-1}, x(t_{j-1}), 0)\right] h^k \qquad (11) \\ &= \sum_{k=0}^{q-1} \frac{1}{k!} c_k(t_{j-1}) h^k + \mathcal{O}(h^q) \end{aligned}$$

mit den Koeffizientenfunktionen

$$c_k(t) := \frac{1}{k+1} x^{(k+1)}(t) - \frac{\partial^k \Phi}{\partial h^k}(t, x(t), 0), k = 0, 1, \ldots, q-1 \qquad (12)$$

Damit gewinnt man folgenden

Satz 2.7 *Voraussetzung 1.1 gelte und $f \in C^r(S)$ mit hinreichend großem $r \geq q$. Ist $\Phi(t, x, .) \in C^q(H) \quad \forall (t, x) \in S$, $H = [0, h_0]$ und*

$$c_k(t) = 0 \quad \textit{für} \quad k = 0(1)q-1 \quad \textit{und} \quad t \in I,$$

so ist das ESV (5) konsistent mit Ordnung $q \in \mathbb{N}$. □

Beispiele 2.8

1. Das *explizite Euler-Verfahren* mit der Verfahrensfunktion $\Phi(t,x,h) = f(t,x)$ erfüllt die Voraussetzungen des Satzes, falls $q = r = 1$ ist. Offenbar ist dann

$$c_0(t) = f(t,x(t)) - \Phi(t,x(t),0) = 0\,,$$

wogegen im allgemeinen

$$c_1(t) = \frac{1}{2}\ddot{x}(t) \neq 0$$

gilt. Das Verfahren hat damit im allgemeinen nur die Konsistenzordnung 1.

2. Das *modifizierte (verbesserte) Euler-Verfahren*

$$u_j = u_{j-1} + h\ f(t_{j-1} + \frac{1}{2}h, u_{j-1} + \frac{1}{2}h\ f(t_{j-1},u_{j-1}))\,, \quad j = 1(1)N. \tag{13}$$

besitzt die Verfahrensfunktion

$$\Phi(t,x,h) = f(t + \frac{1}{2}h, x + \frac{1}{2}h\ f(t,x))\,,$$

die 2-mal stetig differenzierbar bezüglich h ist, wenn $q = r = 2$ vorausgesetzt wird. Wegen $\Phi(t,x(t),0) = f(t,x(t))$ ist $c_0(t) = 0$. Differentiation

$$\frac{\partial\Phi}{\partial h}(t,x(t),0) = \frac{1}{2}f_t(t,x(t)) + \frac{1}{2}f_x(t,x(x))f(t,x(t)) = \frac{1}{2}\ddot{x}(t)$$

ergibt zudem $c_1(t) = 0$, so daß dieses Verfahren mindestens die Konsistenzordnung 2 hat. ◀

2.3 Konvergenz

Die Konsistenz stellt gewissermaßen eine minimale Approximationsgüte eines Verfahrens dar. Offenbar besitzt das explizite Euler-Verfahren lediglich die Konsistenzordnung 1. Damit ist allerdings noch nicht geklärt, ob auch der globale Diskretisierungsfehler (8), der für den Verfahrensnutzer vom eigentlichen Interesse ist, für kleine Schrittweite h dasselbe Verhalten zeigt, d. h. ob zumindest

$$\lim_{h\to 0} e_j = 0, \quad j = 0(1)N \tag{14}$$

gilt. Verfahren mit dieser Eigenschaft wird man als konvergent bezeichnen. Existiert darüber hinaus eine natürliche Zahl p mit $e_j = O(h^p)$, $j = 0(1)N$, so wird man p als Konvergenzordnung des Diskretisierungsverfahrens bezeichnen.

Definition 2.9 (Globaler Diskretisierungsfehler und Konvergenz)

(i) *Der globale Diskretisierungsfehler des ESV 5 an der Stelle t_j lautet* $e_j = u_j - x(t_j),\ j = 0(1)N$.

(ii) *Das ESV heißt konvergent, falls gilt*

$$\lim_{h\to 0} e_j = 0 \quad j = 0(1)N\,.$$

(iii) Die Konvergenzordnung ist $p \in \mathbb{N}$ *, falls Konstanten* $h_0 > 0$ *und* $C > 0$ *existieren, so daß für alle* $h \in (0, h_0]$ *gilt*

$$||e_j|| \;\leq\; C \cdot h^p \;, \quad j = 0(1)N \;.$$

Unter geeigneten Voraussetzungen an das Anfangswertproblem (2) folgt aus der Konsistenzordnung 1 des expliziten Euler-Verfahrens bei konsistentem Anfangswert mit $e_0 = 0(h)$ die Konvergenz mit derselben Ordnung 1.

Beispiel 2.10 $\dot{x} = t - x^2, \;\; x(0) = 0, \;\; I = [0, 1]$

Bestimmt man zu gegebener Zahl N und konstanter Schrittweite $h = 1/N$ mit dem expliziten Euler-Verfahren die Näherung u_N für $x(1) = 0,455\,544\,526\,08...$, so erhält man nachfolgende Werte mit den angegebenen globalen Diskretisierungsfehlern $e_N(h)$:

N	h	u_N	$e_N(h)$	Q_N
5	0.2	0.385 559	-6.99E-2	2.14
10	0.1	0.422 802	-3.27E-2	2.07
20	0.05	0.439 720	-1.58E-2	2.04
40	0.025	0.447 767	-7.78E-3	2.02
80	0.0125	0.451 69	-3.86E-3	2.00
160	0.00625	0.453 62	-1.93E-3	–
	$x(1)$	0.455 545		**2**

Die Konvergenzordnung 1 bewirkt annähernd eine Halbierung des globalen Diskretisierungsfehlers e_N bei einer Rechnung mit halber Schrittweite $h/2$. Die in der letzten Spalte angegebenen Fehlerquotienten $Q_N = e_N(h)/e_{2N}(h/2)$ belegen anschaulich die Konvergenzordnung 1 des Verfahrens. ◀

Für allgemeine explizite ESV erhalten wir den folgenden

Satz 2.11 *Voraussetzung 1.1 sei erfüllt sowie folgende Voraussetzungen:*

(i) $f \in C^1(S)$ *mit* $S = \{(t, x) \mid a \leq t \leq b,\; ||x|| \leq M\}$

(ii) $\Phi \in C^1(S \times H),\; H = [0, h_0]$

(iii) Das ESV (5) ist konsistent.

(iv) $\lim_{h \to 0} e_0 = 0$ *(Konsistenz der Anfangswerte).*

Dann ist das ESV (5) konvergent.

Beweis: Nach Definition 2.4(i) gilt mit dem lokalen Diskretisierungsfehler

$$x(t_j) \;=\; x(t_{j-1}) + h\; \Phi(t_{j-1}, x(t_{j-1}), h) + h\; \tau_j,$$

während das ESV die Darstellung

$$u_j \;=\; u_{j-1} + h\;\Phi(t_{j-1}, u_{j-1}, h)$$

besitzt. Wir subtrahieren die beiden Gleichungen und wenden den Mittelwertsatz bezüglich der Variablen x auf die Verfahrensfunktion an

$$e_j \;=\; e_{j-1} + h\;\int_0^1 \frac{\partial \Phi}{\partial x}(t_{j-1}, (1-s)x(t_{j-1}) + s u_{j-1}, h) e_{j-1}\, ds - h\;\tau_j\;.$$

Gehen wir zur Norm beider Seiten über, so erhalten wir mit der Dreiecksungleichung

$$||e_j|| \;\leq\; (1 + h\;L)\;||e_{j-1}|| + h\;T\;,$$

wobei

$$L = \max_{S\times H} \left\| \frac{\partial \Phi}{\partial x}(t,x,h) \right\| \;\geq\; 0 \quad \text{und} \quad T = \max_{j=1(1)N} ||\tau_j||$$

gilt. Setzen wir rekursiv ein, so erhalten wir

$$\begin{aligned}
||e_j|| &\leq (1+hL)^j ||e_0|| + h \sum_{k=0}^{j-1} (1+hL)^k \cdot T \\
&= (1+hL)^j ||e_0|| + h \frac{(1+hL)^j - 1}{(1+hL) - 1} \cdot T \\
&\leq e^{j(hL)} ||e_0|| + \frac{1}{L} \cdot e^{j(hL)} \cdot T, \quad \text{also} \\
&\leq e^{L(b-a)} \left(||e_0|| + \frac{T}{L} \right), \quad j = 1(1)N\;. \qquad (15)
\end{aligned}$$

Dabei wurde von der Eigenschaft der Exponentialfunktion $1 + x \leq e^x\;,\; \forall x \in \mathbb{R},$ Gebrauch gemacht (Übungsaufgabe). Mit den Voraussetzungen (iii) und (iv) konvergiert die rechte Seite gegen 0 für $h \to 0$, womit die Behauptung $||e_j|| \to 0,\; j = 0(1)N$, erfüllt ist. □

Genügen e_0 und T einer Ordnungsbeziehung $O(h^p)$, so überträgt sich diese wegen der Ungleichung (15) auf den globalen Diskretisierungsfehler e_j. Man erhält somit

Satz 2.12 *Voraussetzung 1.1 sei erfüllt sowie zu gegebenem $p \in \mathbb{N}$:*

(i) $f \in C^r(S)$ *mit hinreichend großem $r \geq p$.*

(ii) $\Phi \in C^p(S \times H),\; H = [0, h_0]$

(iii) Das ESV (5) ist konsistent mit Ordnung p.

(iv) Die Anfangswerte sind konsistent mit Ordnung p.

Dann konvergiert das ESV (5) mit der Ordnung p, also $e_j = O(h^p),\; j = 0(1)N$. □

Folgerung 2.13 *Ist $f \in C^1(S)$ und $e_0 = O(h)$, so konvergiert das explizite Euler-Verfahren mit der Ordnung* **1**, *d.h. $e_j = O(h)\;,\; j = 0(1)N$.*

2.4 Einschrittverfahren 2. Ordnung

Um ein Verfahren der Konsistenzordnung 2 zu gewinnen, integrieren wir die DGL (2) über einem Teilintervall $[t_{j-1}, t_j]$

$$x(t_j) = x(t_{j-1}) + \int_{t_{j-1}}^{t_j} f(t, x(t))\, dt = x(t_{j-1}) + h \int_0^1 f(t_{j-1} + hs, x(t_{j-1} + hs))\, ds\,. \tag{16}$$

Approximation des Integrals durch die Rechteckformel mit dem Knoten $\xi_1 = t_{j-1} + hc_1$ ergibt für $c_1 = 0$ das explizite und für $c_1 = 1$ das implizite Euler-Verfahren. Wenden wir eine genauere Quadraturformel mit einem zweiten Knoten $\xi_2 = t_{j-1} + hc_2$ an, so erhalten wir die Näherung

$$u_j = u_{j-1} + h\ [b_1 f(t_{j-1}, u_{j-1}) + b_2 f(t_{j-1} + hc_2, \eta_2)]\ , \tag{17}$$

wobei η_2 eine Approximation des unbekannten Zwischenwertes $x(t_{j-1} + hc_2)$ ist. Diese beschaffen wir uns, indem wir Formel (16) über dem Teilintervall $[t_{j-1}, t_{j-1} + hc_2]$ anwenden

$$x(t_{j-1} + hc_2) = x(t_{j-1}) + \int_{t_{j-1}}^{t_{j-1}+hc_2} f(t, x(t))\, dt \tag{18}$$

und mit einem Knoten $\xi_1 = t_{j-1}$ und einem allgemeinen Gewicht a_{21} die Näherung

$$\eta_2 = u_{j-1} + h\ a_{21}\ f(t_{j-1}, u_{j-1}) \tag{19}$$

gewinnen. Einsetzen von η_2 in (17) liefert explizite ESV mit der 2-stufigen Verfahrensfunktion

$$\Phi(t, x, h)\ =\ b_1 f(t, x) + b_2 f(t + hc_2, x + h\ a_{21}\ f(t, x)) \tag{20}$$

und den 4 Verfahrenskonstanten a_{21}, b_1, b_2, c_2. Setzen wir $f \in C^2(S)$ voraus, so ist auch $\Phi(t, x, .) \in C^2(H) \quad \forall (t, x) \in S,\ H = [0, h_0]$ und wir können die Konsistenzbedingungen des Satzes 2.7 für $k = 0, 1$ auswerten (Übungsaufgabe). Die Konsistenzbedingungen an ein Verfahren 2. Ordnung lauten dann

$$b_1 + b_2 = 1, \quad b_2 c_2 = \frac{1}{2}, \quad b_2 a_{21} = \frac{1}{2}\,. \tag{21}$$

Beispiel 2.14 Die bekanntesten expliziten ESV 2. Ordnung sind das bereits in Beispiel 2.8 genannte *modifizierte (verbesserte) Euler-Verfahren* von C.Runge (1895)

$$u_j = u_{j-1} + h\ f(t_{j-1} + \frac{1}{2}h, u_{j-1} + \frac{1}{2}h\ f(t_{j-1}, u_{j-1}))\,, \quad j = 1(1)N \tag{22}$$

und das *explizite Heun-Verfahren* von K.Heun(1900)

$$u_j = u_{j-1} + \frac{h}{2}\ [f(t_{j-1}, u_{j-1}) + f(t_{j-1} + h, u_{j-1} + h\ f(t_{j-1}, u_{j-1}))]\ , \quad j = 1(1)N. \tag{23}$$

Beide Verfahren sind 2-stufig, da sie pro Integrationsschritt genau 2 Berechnungen der Funktion $f(t, x)$ erfordern. Falls f zweimal stetig differenzierbar auf S ist, so besitzen sie bei konsistenten Anfangswerten der Ordnung 2 nach Satz 2.12 auch die Konvergenzordnung 2, so daß der verdoppelte Rechenaufwand gerechtfertigt ist. ◄

Führt man für die berechneten Funktionswerte die Hilfsvariablen k_1 und k_2 ein, so läßt sich das Heun-Verfahren (23) in algorithmischer Form notieren:

Algorithmus 2.15 (Explizites Heun-Verfahren)

Function HEUN22 $(f, t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t, x)$
   $k_2 = f(t + h,\ x + hk_1)$
3. $x_{neu} = x + \frac{h}{2}(k_1 + k_2)$
   $t_{neu} = t + h$
4. Ausgabe : t_{neu}, x_{neu}

Ähnlich erhält man den Algorithmus für einen Schritt des verbesserten Euler-Verfahrens:

Algorithmus 2.16 (Verbessertes Euler-Verfahren)

Function EULER22 $(f, t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t, x)$
   $k_2 = f(t + \frac{h}{2}h,\ x + \frac{h}{2}hk_1)$
3. $x_{neu} = x + hk_2$
   $t_{neu} = t + h$
4. Ausgabe : t_{neu}, x_{neu}

2.5 Runge-Kutta-Verfahren

C. Runge und K. Heun konstruierten nach demselben Prinzip weitere "modifizierte Euler-Verfahren", die von W. Kutta 1901 zu einem Schema verallgemeinert wurden, das man als *explizite Runge-Kutta-Verfahren* bezeichnet:

$$u_j = u_{j-1} + h\sum_{i=1}^{s} b_i k_i \qquad \text{mit den Steigungswerten} \tag{24}$$

$$k_i = f(t_{j-1} + hc_i, u_{j-1} + h\sum_{j=1}^{i-1} a_{ij}k_j\ ,\ i = 1(1)s. \tag{25}$$

Ein Integrationsschritt von t bis $t_{neu} = t + h$ läßt sich nun leicht nach der allgemeinen algorithmischen Darstellung berechnen:

Algorithmus 2.17 (Explizite Runge-Kutta-Verfahren)

Function RUNGE-KUTTA-TYP $(f, A, b, c, t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t, x)$
   $k_2 = f(t + c_2 h,\ x + a_{21} h k_1)$
   $k_3 = f(t + c_3 h,\ x + a_{31} h k_1 + a_{32} h k_2)$
   $\dots \quad \dots \quad \dots \quad \dots$
   $k_s = f(t + c_s h,\ x + a_{s1} h k_1 + ... + a_{s,s-1} h k_{s-1})$
3. $x_{neu} = x + h(b_1 k_1 + b_2 k_2 + ... + b_s k_s)$
   $t_{neu} = t + h$
4. Ausgabe : $t_{neu},\ x_{neu}$

Da in jeder der s Stufen die Steigungswerte $k_1, k_2, ..., k_s$ explizit berechnet werden können, bezeichnet man die Verfahren (2.17) als s- stufige explizite Runge-Kutta-Verfahren der Konvergenzordnung p oder kurz als (p, s)-RK-Verfahren. Die darin vorkommenden Konstanten werden in dem von J.C.Butcher eingeführten *Parameterschema* übersichtlich zusammengefaßt:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

Die Matrix $A = (a_{i,j})$, $i, j = 1(1)s$, in der die fehlenden Elemente zu Null definiert werden, heißt *RK-Matrix*, während die Vektoren $b = (b_1, b_2, \dots, b_s)^T$ bzw. $c = (c_1, c_2, \dots, c_s)^T$ als *RK-Gewichte* bzw. *RK-Knoten* bezeichnet werden.

Beispiele 2.18 Das explizite Euler-Verfahren (4) besitzt als $(1, 1)$-RK-Verfahren trivialerweise das Parameterschema

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

Das explizite Heun-Verfahren (23) ist vom $(2, 2)$-RK-Typ mit dem Parameterschema

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

Das von W.KUTTA 1901 konstruierte "klassische" Runge-Kutta-Verfahren vom Typ $(4,4)$ besitzt das Parameterschema

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

Bei diesem Verfahren stimmen Konvergenzordnung p und Stufenzahl s überein. Für $p \geq 5$ ist dies jedoch nicht mehr möglich; so benötigen Verfahren 5. Ordnung mindestens 6 Stufen. J.C.BUTCHER entwickelte 1964 ein $(5,6)$-RK-Verfahren mit folgendem Parameterschema:

0						
1/4	1/4					
1/4	1/8	1/8				
1/2	0	-1/2	1			
3/4	3/16	0	0	9/16		
1	-3/7	2/7	12/7	-12/7	8/7	
	7/90	0	32/90	12/90	32/90	7/90

Eine allgemeine Konsistenz- und Konvergenzaussage für RK-Verfahren liefert der

Satz 2.19 *Voraussetzung 1.1 sei erfüllt und $f \in C^1(S)$.*

(i) Das RK-Verfahren (24) ist konsistent genau dann, wenn die Konsistenzbedingung

$$\sum_{i=1}^{s} b_i \;=\; 1 \qquad \textit{erfüllt ist.}$$

(ii) Ist das RK-Verfahren konsistent und gilt $\lim_{h\to 0} e_0 = 0$, so ist es konvergent.

BEWEIS: Mit der Verfahrensfunktion von (24)

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

worin K_i die mit der exakten Lösung $x(t)$ ermittelten Steigungswerte darstellen, erhält man für $h = 0$

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

genau dann, wenn die Konsistenzbedingung erfüllt ist. Mit Satz 2.5 folgt dann Behauptung (i), während Satz 2.11 die Konvergenz (ii) liefert. □

Um spezielle RK-Verfahren abzuleiten, kann man mittels Taylorentwicklung die Konsistenzbedingungen des Satzes 2.7 erfüllen. Allerdings stellt die Konstruktion von (p, s)-Verfahren hoher Ordnung p wegen der extrem ansteigenden Zahl von Konsistenzbedingungen eine sehr komplizierte Aufgabe dar, die in der empfohlenen Literatur für $p \geq 4$ behandelt wird (vgl. Hairer, E.; Norsett, S.P.; Wanner, G. sowie Strehmel, K.; Weiner, R.). So benötigen Verfahren 8. Ordnung mindestens 11 Stufen für 200 Bedingungsgleichungen. Sie lösen das Anfangswertproblem allerdings mit hoher Genauigkeit bei moderaten Schrittweiten, wenn die DGL die geforderte hohe Glattheit besitzt.

Beispiel 2.20 Um ein DGL-System (2) mit dem klassischen (4,4)-RK-Verfahren zu lösen, lautet ein RK-Schritt in algorithmischer Form

Algorithmus 2.21 (Klassisches Runge-Kutta-Verfahren)

Function RUNGE-KUTTA44 $(f, t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t, x)$
   $k_2 = f(t + \frac{1}{2}h,\ x + \frac{1}{2}hk_1)$
   $k_3 = f(t + \frac{1}{2}h,\ x + \frac{1}{2}hk_2)$
   $k_4 = f(t + h,\ x + hk_3)$
3. $x_{neu} = x + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
   $t_{neu} = t + h$
4. Ausgabe : $t_{neu},\ x_{neu}$

Nehmen wir das Anfangswertproblem des Beispiels 2.10

$$\dot{x} = t - x^2, \quad x(0) = 0, \quad I = [0, 1]$$

und bestimmen zu gegebener Zahl N die Näherung u_N für $x(1) = 0,455\ 544\ 526\ 08...$ mit Verfahren (2.21) und Schrittweite $h = 1/N$. Tabelle 1 zeigt den globalen Diskretisierungsfehler $e_N(h)$ und den Quotienten $Q_N = e_N(h)/e_{2N}(h/2)$. Wegen $Q_N \to 2^4$ ist die theoretische Fehlerordnung 4 gut erkennbar; zudem liefert das Verfahren bei gleichem Aufwand wesentlich genauere Näherungen als das explizite Euler-Verfahren. ◄

Am letzten Beispiel wird bei $N = 80$ erkennbar, daß die auftretenden Rundungsfehler die theoretische Entwicklung des Diskretisierungsfehlers verfälschen können, falls dieser bereits nahe der Größe der Rechengenauigkeit (hier bei 11-stelligen Mantissen ca. 10^{-11}) liegt. Empfehlenswert ist deshalb, sämtliche Rechnungen mit 16-stelliger Mantisse in *double precision*

N	h	u_N	$e_N(h)$	Q_N
5	0.2	0.455 533 498 11	-1.10E-5	16.1
10	0.1	0.455 543 842 70	-6.84E-7	16.1
20	0.05	0.455 544 483 62	-4.24E-8	16.0
40	0.025	0.455 544 523 43	-2.65E-9	14.8
80	0.0125	0.455 544 525 90	-1.8E-10	–
	$x(1)$	0.455 544 526 08		**16**

Tabelle 1: Runge-Kutta-Verfahren 4.Ordnung (Ergebnisse u_N bei $x_N = 1$)

auszuführen, um eine Genauigkeit von 10^{-12} - die für praktische Zwecke hinreichend sein dürfte - zu garantieren. Damit erübrigen sich die aufwendigen Rundungsfehler-Analysen.

2.6 Das Runge-Prinzip

Abschätzungen des Diskretisierungsfehlers sollten ohne Kenntnis der exakten Lösung eine Bewertung der erhaltenen Näherungslösungen gestatten. Die in (15) erhaltenen Schranken liefern jedoch meist eine extreme Überschätzung des globalen Diskretisierungsfehlers und sind somit praktisch unbrauchbar. Für den lokalen Diskretisierungsfehler τ_j hingegen sind asymptotische Fehlerschätzungen leicht zu ermitteln, d.h. Zahlenvektoren, die mit kleiner werdender Schrittweite h den Fehler immer genauer approximieren. Das Runge-Prinzip und das Prinzip der eingebetteten Runge-Kutta-Verfahren liefern derartige Schätzunen mit geringem Zusatzaufwand.

Setzt man hinreichende Glattheit der rechten Seiten f der DGL und der Verfahrensfunktion Φ eines ESV der Konsistenzordnung q voraus, so läßt sich die Taylorentwicklung (11) des lokalen Diskretisierungsfehlers τ_j weiterführen zur asymptotischen Entwicklung

$$\tau_j = T_j\, h^q + O(h^{q+1}) \quad \text{mit} \quad T_j = \frac{1}{q!} c_q(t_{j-1})\,. \tag{26}$$

Da für $h \to 0$ die in $O(h^{q+1})$ zusammengefaßten Restterme schneller gegen Null konvergieren als $T_j h^q$, kann dieser *führende Term* als asymptotischer Fehlerschätzer dienen.

1. Mit dem Startwert $x(t_{j-1})$ führt man einen Schritt der Länge $2h$ aus („Grobrechnung“) und erhält anstelle des exakten Wertes $x(t_j)$ den „Grobwert“ u_j^G mit dem Fehler

$$\tau_j^G = T_j\,(2h)^q + O(h^{q+1}) = \frac{x(t_j) - u_j^G}{2h}\,.$$

2. Führt man nun *mit demselben Verfahren* 2 Schritte der Länge h aus („Feinrechnung“), so kann man deren Fehler zusammenfassen und erhält nach Taylorentwicklung (Übungsaufgabe) die Darstellung

$$\tau_j^F = 2T_j\, h^q + O(h^{q+1}) = \frac{x(t_j) - u_j^F}{h}\,, \tag{27}$$

wobei u_j^F den sogenannten „Feinwert“ nach 2 Schritten bezeichnet (vgl. Abb. 2).

Multiplikation der Gleichungen mit $2h$ bzw. h liefert das System

$$\begin{aligned} 2T_j\, 2^q h^{q+1} + O(h^{q+2}) &= x(t_j) - u_j^G \\ 2T_j\, h^{q+1} + O(h^{q+2}) &= x(t_j) - u_j^F\,, \end{aligned}$$

das nach Subtraktion und anschließender Division durch $(2^q - 1)h$ in die Form

$$2T_j\, h^q + O(h^{q+1}) \;=\; \frac{1}{h} \cdot \frac{u_j^F - u_j^G}{2^q - 1}$$

übergeht. Vergleich mit (27) liefert folgenden

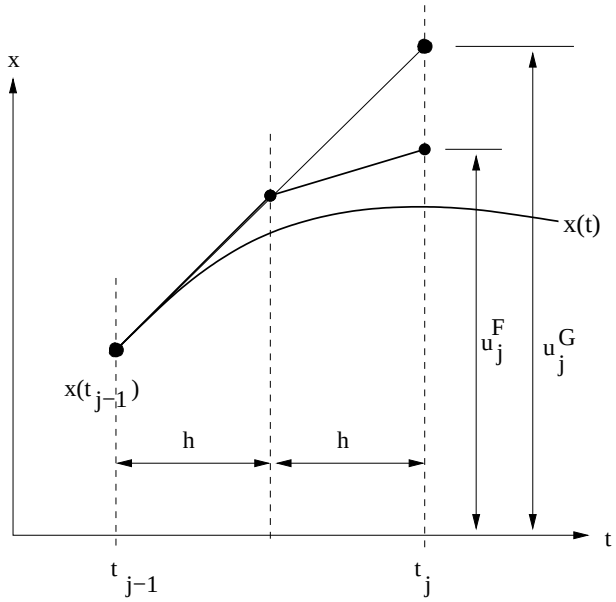

Abbildung 2: Runge-Prinzip

Satz 2.22 *Die Voraussetzungen des Satzes 2.12 seien erfüllt und die Verfahrensfunktion* Φ *sei hinreichend glatt. Dann besitzt der lokale Diskretisierungsfehler der Feinrechnung die Darstellung*

$$\tau_j^F \;=\; \frac{1}{h} \cdot \frac{u_j^F - u_j^G}{2^q - 1} + \mathcal{O}(h^{q+1})\,, \tag{28}$$

und

$$Error \;:=\; \frac{1}{h} \cdot \frac{u_j^F - u_j^G}{2^q - 1} \tag{29}$$

ist ein asymptotischer Fehlerschätzer für τ_j^F *der Ordnung* $q+1$. □

Setzt man nun den lokalen Diskretisierungsfehler τ_j^F aus (28) in die Darstellung (27) ein, so liefert deren Umstellung nach der exakten Lösung

$$x(t_j) \;=\; u_j^F + \frac{u_j^F - u_j^G}{2^q - 1} + O(h^{q+2})\,.$$

Damit läßt sich auf dem Grobgitter $I_{N/2}$ der Näherungswert der Feinrechnung verbessern (extrapolieren) zu

$$u_j^{FG} \;=\; u_j^F + \frac{u_j^F - u_j^G}{2^q - 1}, \qquad t_j \in I_{N/2}\,.$$

Diese verbesserten Näherungen u_j^{FG} ergeben offenbar einen lokalen Diskretisierungsfehler der Ordnung $q+1$.

2.7 Eingebettete Runge-Kutta-Verfahren

Um den lokalen Diskretisierungsfehler τ_j abzuschätzen, kann man auch einen Integrationsschritt von t_{j-1} bis t_j mit zwei Verfahren parallel ausführen (vgl. Abb. 3). Verfahren 1 als (p,s)-RK-Verfahren liefert den Wert u_j, Verfahren 2 als (p^*,s^*)-Verfahren den Wert u_j^* als Näherung für $x(t_j)$. Das "bessere" Verfahren 2 habe die Ordnung $p^* = p+1$ mit $s^* = s+1$ Stufen. Notiert man die lokalen Diskretisierungsfehler dieser Verfahren

$$\begin{aligned} \tau_j &= \frac{x(t_j) - u_j}{h} \\ \tau_j^* &= \frac{x(t_j) - u_j^*}{h} = O(h^{q+1}) , \end{aligned}$$

so erhält man nach Ersetzen des unbekannten Wertes $x(t_j)$ mit Hilfe der zweiten Gleichung folgenden

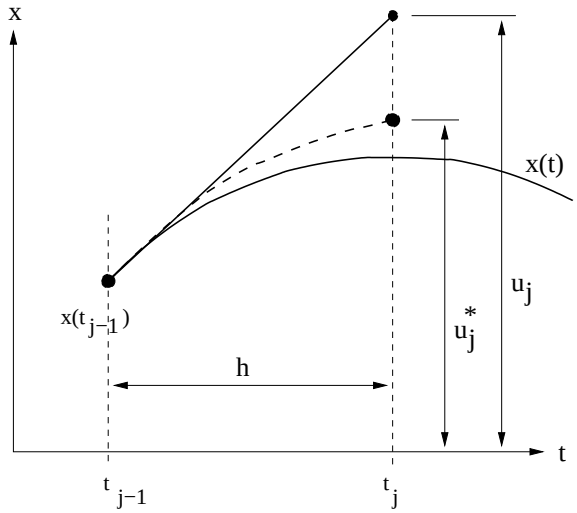

Abbildung 3: Fehlerschätzung mit 2 Verfahren

Satz 2.23 *Die Voraussetzungen des Satzes 2.12 seien erfüllt und die Verfahrensfunktion Φ sei hinreichend glatt. Dann besitzt der lokale Diskretisierungsfehler des (p,s)-RK-Verfahrens 1 die Darstellung*

$$\tau_j = \frac{1}{h}(u_j^* - u_j) + O(h^{q+1}) , \tag{30}$$

und

$$Error := \frac{1}{h}(u_j^* - u_j) \tag{31}$$

ist ein asymptotischer Fehlerschätzer für τ_j der Ordnung $q+1$. □

Den beträchtlichen Aufwand einer Parallelrechnung zweier Verfahren kann man vermeiden, indem man Verfahren 1 so bestimmt, daß seine s Steigungswerte $k_1, k_2, ..., k_s$ mit denen des Verfahrens 2 übereinstimmen. Es entsteht ein sogenanntes eingebettetes $p(p+1)$-RK-Verfahren mit $s+1$ Stufen der Ordnungen p und $p+1$, dessen Parameterschema in folgender Form angegeben wird:

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ c_{s+1} & a_{s+1,1} & a_{s+1,2} & \cdots & a_{s+1,s-1} & a_{s+1,s} \\ \hline u_j & b_1 & b_2 & \cdots & b_{s-1} & b_s \\ u_j^* & b_1^* & b_2^* & \cdots & b_{s-1}^* & b_s^* \end{array} \quad b_{s+1}^*$$

Beispiele 2.24

1. E. Fehlberg entwickelte zahlreiche eingebettete Verfahren. Ein einfaches 2(3)-RK-Verfahren besitzt das Parameterschema

0			
1	1		
1/2	1/4	1/4	
u_j	1/2	1/2	
u_j^*	1/6	1/6	4/6

RKF2(3)

und gestattet damit folgende algorithmische Darstellung:

Algorithmus 2.25 (Runge-Kutta-Fehlberg-Verfahren)

Function RKF23 $(f, t, x, h, t_{neu}, x^*_{neu}, Error)$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t, x)$
   $k_2 = f(t + h,\ x + hk_1)$
   $k_3 = f(t + \frac{1}{2}h,\ x + \frac{1}{4}hk_1 + \frac{1}{4}hk_2)$
3. $x_{neu} = x + \frac{h}{2}(k_1 + k_2)$
   $x^*_{neu} = x + \frac{h}{6}(k_1 + k_2 + 4k_2)$
   $t_{neu} = t + h$
   $Error = \frac{1}{h}(x^*_{neu} - x_{neu})$
4. Ausgabe : $t_{neu},\ x^*_{neu},\ Error$

2. Ein 8-stufiges RK-Verfahren der Ordnungen 5(6) stammt von J.H. Verner, der mit den darin berechneten Steigungswerten $k_1, k_2, \ldots, k_8$ auch Näherungen der Ordnungen 1,2,3 und 4 ermitteln kann und somit eine komplette Menge eingebetteter RK-Verfahren (sogenannte CSIRK-Verfahren) erhält (vgl. nachfolgendes Parameterschema).

0								
$\frac{1}{6}$	$\frac{1}{6}$							
$\frac{4}{15}$	$\frac{4}{75}$	$\frac{16}{75}$						
$\frac{2}{3}$	$\frac{5}{6}$	$-\frac{8}{3}$	$\frac{5}{2}$					
$\frac{3}{4}$	$-\frac{219}{512}$	$\frac{27}{16}$	$-\frac{435}{512}$	$\frac{87}{256}$				
1	$\frac{1}{12}$	0	$\frac{25}{58}$	$-\frac{1}{4}$	$\frac{64}{87}$			
$\frac{1}{15}$	$-\frac{517}{1500}$	$\frac{124}{75}$	$-\frac{212}{145}$	$\frac{291}{500}$	$-\frac{1312}{3625}$	0		
1	$\frac{787}{1812}$	$-\frac{108}{151}$	$\frac{9185}{8758}$	$-\frac{517}{1812}$	$\frac{390368}{538617}$	0	$-\frac{3850}{18573}$	
u_j	$\frac{23}{288}$	0	$\frac{375}{928}$	$\frac{3}{16}$	$\frac{64}{261}$	$\frac{1}{12}$		
u_j^*	$\frac{13}{144}$	0	$\frac{2125}{5104}$	$\frac{1}{6}$	$\frac{2816}{10701}$	0	$-\frac{125}{6888}$	$\frac{151}{1848}$

VERNER 5(6)

3. Während man bei den bisher genannten Verfahren die Fehlerkonstanten des Verfahrens p-ter Ordnung minimiert, haben J.R. Dormand und P.J. Prince ein 7-stufiges Verfahrenspaar der Ordnungen 5(4) konstruiert, bei dem der Fehlerterm des genaueren Verfahrens 5.Ordnung minimiert wird und die Näherung u_j des "schlechteren" Verfahrens der Fehlerschätzung dient. Nachfolgend das Parameterschema dieses häufig benutzten Verfahrens DOPRI5(4), das wegen $a_{7,i} = b_i$, $i = 1(1)6$, besonders efffizient ist. Weitere eingebettete Verfahren, z. B. DOPRI8(7), findet man in der angegebenen Literatur.

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

DOPRI5(4)

2.8 Schrittweitensteuerung

Die asymptotischen Fehlerschätzungen (29) und (31) lassen sich nutzen, um die Integrationsschrittweite h so zu verändern, daß eine vorgegebene Fehlertoleranz $TOL > 0$ in jedem Integrationsschritt eingehalten wird. Dazu benötigt man eine skalare Schätzung für den Gesamtfehler über alle Komponenten von $Error = (Error^1, \ldots, Error^n)$, z.B. als maximaler Absolutfehler

$$EST := \max_{i=1(1)n} |Error^i| .$$

Unterschiedliche Größenordungen der Komponenten werden durch den (komponentenweisen) relativen Fehler

$$EST := \max_{i=1(1)n} \frac{|Error^i|}{|u_j^i|}$$

mit den Lösungsnäherungen $u_j = (u_j^1, \ldots, u_j^n)$ allerdings besser berücksichtigt. Um für sehr kleine Werte $|u_j^i|$ brauchbare Schätzungen zu erhalten, kombiniert man beide Darstellungen zum sogenannten *maximalen Absolut-Relativfehler*

$$EST := \max_{i=1(1)n} \frac{|Error^i|}{|u_j^i| + 1.0} . \tag{32}$$

Um zu garantieren, daß stets $EST \approx TOL$ eingehalten wird, bestimmen wir eine neue Schrittweite $h_{neu} = \varrho h$ durch Multiplikation mit einem Schrittweiten-Veränderungsfaktor ϱ. Wegen der Konsistenzordnung p gilt asymptotisch mit einer Konstanten $C > 0$

$$EST \;=\; C \cdot h^p + O(h^{p+1}) .$$

Gefordert wird aber, daß mit der neu zu wählenden Schrittweite h_{neu} analog für TOL

$$TOL \;=\; C \cdot h_{neu}^p + O(h_{neu}^{p+1}) \;=\; C \cdot \varrho^p h^p + O(h^{p+1})$$

gilt. Division beider Formeln ergibt

$$\frac{TOL}{EST} \;=\; \varrho^p \cdot (1 + O(h)),$$

woraus nach Vernachlässigung des Ordnungsterms und Umstellung nach ϱ die einfache Forderung für den Schrittweiten-Veränderungsfaktor

$$\varrho := \varkappa \sqrt[p]{\frac{TOL}{EST}} \tag{33}$$

folgt. Der Zusatzfaktor $\varkappa < 1$ vermeidet ein häufiges Rechnen an der Schrittweitengrenze; ein Wert $\varkappa = 0.8$ ist dabei angemessen. Eine freie Schrittweitensteuerung wird nun EST nach (32) und p nach (33) berechnen. Für das "bessere" Verfahren ist offenbar p durch $p+1$ in (33) zu ersetzen. Ergibt sich nun $EST > TOL$, so muß der letzte Integrationsschritt mit neuer Schrittweite $h_{neu} = \varrho\, h$ ggf. mehrfach wiederholt werden. Andernfalls wird der Schritt akzeptiert und bei $t := t_{neu}$ weitergerechnet, wobei $h_{neu} = \varrho\ h$ als neuer Schrittweitenvorschlag dient.
Auf diese Weise entstehen adaptive Runge-Kutta-Verfahren mit automatischer Schrittweitensteuerung, die die Integrationsschrittweite dem Lösungsverlauf weitgehend anpassen.

Beispiel 2.26 Wir lösen das Anfangswertproblem des Beispiels 2.10

$$\dot{x} = t - x^2, \quad x(0) = 0, \quad I = [0, 1]$$

mit dem 8-stufigen eingebetteten 5(6)− RK-Verfahren von VERNER zu vorgegebenen Genauigkeiten $TOL = 10^{-1}, 10^{-2}, \ldots, 10^{-10}$. In der Tabelle werden die erhaltenen Näherungen u_N des Endwertes $x(1) = 0.455\ 544\ 526\ 08...$, die dafür benötigte Anzahl INT von Integrationsschritten sowie der letzte geschätzte Fehler EST dargestellt. Im Vergleich zum globalen Fehler $e_N = u_N - x(t_N)$ wird die gute Eignung der asymptotischen Schätzung EST deutlich.

TOL	y_N	INT	EST	$\lvert e_N \rvert$
10^{-1}	0.45462877649	1	2.28E-3	9.16E-4
10^{-2}	0.45462877649	1	2.28E-3	9.16E-4
10^{-3}	0.45553705561	3	5.42E-5	7.47E-6
10^{-4}	0.45553705561	3	5.42E-5	7.47E-6
10^{-5}	0.45554372308	4	6.31E-7	8.03E-7
10^{-6}	0.45554445782	7	1.64E-7	5.03E-8
10^{-8}	0.45554452525	13	4.81E-10	8.33E-10
10^{-10}	0.45554452607	21	1.00E-11	1.18E-11

Als Startschrittweite wurde stets die Intervallänge $h = 1.0$ gewählt, die dann durch das Verfahren automatisch "heruntergeregelt" wurde. ◀

3 Implizite Einschrittverfahren

Alle bisher angegebenen Runge-Kutta-Verfahren sind explizite Einschrittverfahren, da sie nach dem zu bestimmenden Lösungswert u_j aufgelöst vorliegen und lediglich auf die vorherigen Werte t_{j-1} und u_{j-1} zugreifen. Sie benötigen deshalb keine spezielle Anlaufrechnung, und Schrittweitenänderungen sind leicht möglich. In vielen praktischen Anwendungen haben sie sich – insbesondere als eingebettete Verfahren – vorzüglich bewährt. Für bestimmte Problemklassen sind sie allerdings weniger geeignet, besonders wenn eine Langzeit-Integration gewünscht ist.

Beispiel 3.1 Man löse für $0 \leq t \leq 1$ das Anfangswertproblem von L.F.SHAMPINE und M.K.GORDON

$$\begin{aligned} \dot{x_1} &= -29998x_1 - 39996x_2\ , \quad x_1(0) = 1 \\ \dot{x_2} &= 14998.5x_1 + 19997x_2\ , \quad x_2(0) = 1 \end{aligned}$$

mit dem eingebetteten Runge-Kutta-Verfahren der Ordnung 5(6) von *VERNER* mit einer Genauigkeit $TOL = 10^{-8}$. Nachfolgend werden die Anzahl INT der ausgeführten Integrationsschritte, der jeweilige t-Wert sowie die Arbeitsschrittweite h dargestellt.

INT	Aktueller Wert t	Aktuelle Schrittweite h	Phase
7	1.01E-4	5.03E-5	
15	7.75E-4	1.33E-4	1
25	3.89E-3	4.82E-4	
50	1.41E-2	4.19E-4	
100	3.47E-2	4.25E-4	
200	7.53E-2	4.39E-4	2
300	1.17E-1	4.13E-4	
400	1.57E-1	4.26E-4	

Die Rechnung bis zum Endwert $t = 1$ erforderte ca. 2000 Schritte. Nach Überwindung der transienten Phase 1 mit kleiner Schrittweite stagniert das Verfahren in der asymptotischen (glatten) Phase 2 und behält die extrem kleine Schrittweite von ca. 10^{-4} bei. Ausschlaggebend dafür sind jedoch nicht Genauigkeitsforderungen, sondern numerische Stabilitätsgründe. Jeder Versuch, in der asymptotischen Phase mit einer moderaten Schrittweite von ca. 0.01 bis 0.001 weiterzurechnen, führt unweigerlich zu starken Oszillationen und/oder Zahlenüberlauf. ◀

3.1 Absolute Stabilität und Stabilitätsbereich

Das prinzipielle Verhalten eines Diskretisierungsverfahrens bei Problemen des Beispieltyps 3.1 läßt sich anschaulich an der von G.Dahlquist 1963 eingeführten skalaren Testgleichung

$$\dot{x} = \lambda x, \quad x(a) = x_0, \quad \lambda \in \mathbb{C}^- \tag{34}$$

demonstrieren, wobei der Parameter λ aus der Menge $\mathbb{C}^-$ aller komplexen Zahlen mit negativem Realteil ist. Die Lösungen dieses Anfangswertproblems

$$x(t) = x_0 \; e^{\lambda(t-a)} \tag{35}$$

sind dann abklingend, so daß $\lim \; x(t) = 0$ für $t \to \infty$ gilt. Diese grundlegende Eigenschaft muß auch auf die Näherungslösungen u_j vererbt werden, wenn das eingesetzte Verfahren sinnvoll sein soll. Also ist $\lim \; u_j = 0$ für $j \to \infty$ zu fordern, wenn man das Verfahren als „numerisch stabil“ bezeichnen will. Wendet man das explizite Euler-Verfahren auf diese Testgleichung (34) an, so liefert es mit $H = \lambda h$ die Näherungslösung

$$\begin{aligned} u_j &= u_{j-1} + h \; \lambda \; u_{j-1} \\ &= (1 + H) \; u_{j-1} \\ &= (1 + H)^2 \; u_{j-2} \\ u_j &= (1 + H)^j \; x_0. \end{aligned} \tag{36}$$

Dann ist $\lim_{j\to\infty} |u_j| = 0$ nur möglich, wenn $|1+H| < 1$ gilt. Erfüllt der Zahlenwert $H = \lambda h$ diese Bedingung, so kann man das zugrundeliegende Euler-Verfahren als absolut stabil für diesen Wert H bezeichnen. Offenbar beschreibt die Menge aller Parameterwerte

$$\mathbf{H} = \{\ H \in \mathbb{C} \mid |1 + H\ | < 1\ \}$$

das Innere des abgebildeten Kreises in der komplexen Zahlenebene (vgl. Abb.4). An der Abbildung erkennt man leicht, daß für die Zugehörigkeit $H \in \mathbf{H}$ notwendig ist, daß $|H| = |\lambda h| < 2$ gilt, woraus $h < 2/|\lambda|$ als Schrittweitenbeschränkung folgt. Bei schnell abklingenden Lösungen (35) mit $|\lambda| \geq 10^4$ muß deshalb mit extrem kleiner Schrittweite $h < 2 \cdot 10^{-4}$ integriert werden, um eine Lösungsdivergenz $|u_j| \to \infty$ zu vermeiden.

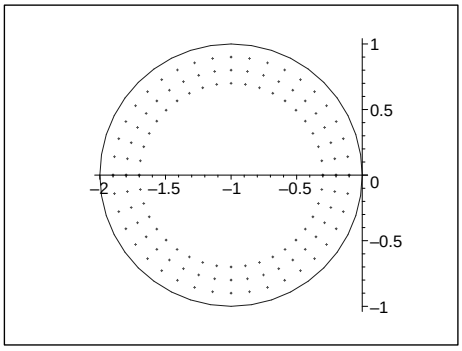

Abbildung 4: Stabilitätsbereich des expliziten Euler-Verfahrens

Wir verallgemeinern nun diese Betrachtung auf die Darstellung 2.17 der Runge-Kutta-Verfahren. Wenden wir die Verfahrensklasse auf die Testgleichung (34) an, so erhalten wir für die mit h multiplizierten Steigungswerte

$$\begin{aligned}
hk_1 &= Hu_{j-1} \\
hk_2 &= H(u_{j-1} + a_{21}hk_1) \\
hk_3 &= H(u_{j-1} + a_{31}hk_1 + a_{32}hk_2) \\
\dots &\quad \dots \\
hk_s &= H(u_{j-1} + a_{sl}hk_1 + ... + a_{s,s-1}hk_{s-1}).
\end{aligned}$$

Führt man die Spaltenvektoren $k = (k_1, k_2, k_3, \dots, k_s)^T$ und $\mathbf{1} = (1, 1, 1, \dots, 1)^T$ der Länge s ein, so lauten diese Beziehungen in vektorieller Form

$$hk \ = \ H(u_{j-1} \cdot \mathbf{1} + Ahk)$$

mit der RK-Matrix A. Umstellung nach hk liefert

$$hk \ = \ H(I - HA)^{-1} \cdot \mathbf{1} \cdot u_{j-1}\ .$$

Für die Summenformel des Runge-Kutta-Verfahrens ergibt sich mit den RK-Gewichten b

$$u_j \ = \ u_{j-1} + b^T(hk)\ .$$

Setzen wir hk darin ein und fassen zusammen, so erhalten wir analog zum expliziten Euler-Verfahren eine Rekursionsdarstellung

$$u_j = \mu(H)u_{j-1}\ , \quad j = 1, 2, 3, \dots .$$

mit der *Stabilitätsfunktion* des Runge-Kutta-Verfahrens

$$\mu(H) = 1 + Hb^T(I - HA)^{-1} \cdot \mathbf{1}, \tag{37}$$

die eine Abbildung $\mu : \mathbb{C} \to \mathbb{C}$ darstellt.

Bemerkung 3.2 Die Potenzen A^r der RK-Matrix expliziter s-stufiger Runge-Kutta-Verfahren verschwinden für alle $r \geq s$, womit sich die inverse Matrix in (37) entwickeln läßt

$$\begin{aligned}\mu(H) &= 1 + Hb^T(I - HA)^{-1} \cdot \mathbf{1} \\ &= 1 + Hb^T(I + HA + H^2A^2 + \ldots + H^{s-1}A^{s-1}) \cdot \mathbf{1} \\ &= 1 + (b^T \cdot \mathbf{1})H + (b^TA \cdot \mathbf{1})H^2 + (b^TA^2 \cdot \mathbf{1})H^3 + \ldots + (b^TA^{s-1} \cdot \mathbf{1})H^s \\ &= 1 + \varepsilon_1\ H + \varepsilon_2\ H^2 + ... + \varepsilon_s H^s\ .\end{aligned}$$

Die Stabilitätsfunktion eines expliziten s-stufigen RK-Verfahrens ist folglich ein Polynom höchstens s-ten Grades, dessen reelle Koeffizienten sich leicht mit den Formeln

$$\varepsilon_0 = 1, \quad \varepsilon_k = b^TA^{k-1}\mathbf{1} \quad \text{für} \quad k = 1(1)s \tag{38}$$

berechnen lassen.

Beispiele 3.3

1. Das *explizite Euler-Verfahren* besitzt wegen $\varepsilon_1 = b^T\mathbf{1} = 1$ die bereits in (36) gewonnene Stabilitätsfunktion $\mu(H) = 1 + H$.

2. Das *modifizierte Euler-Verfahren* (22) mit dem (vollständigen) Parameterschema

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

   liefert die Koeffizienten

$$\varepsilon_1 = b^T\mathbf{1} = 1 \quad \text{und} \quad \varepsilon_2 = b^TA\mathbf{1} = \frac{1}{2}$$

   und damit die Stabilitätsfunktion $\mu(H) = 1 + H + \frac{1}{2}H^2$.

3. Für das *klassische Runge-Kutta-Verfahren* vom Typ $(4, 4)$ berechnet man mit dem Parameterschema

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}$$

   die Koeffizienten

$$\begin{aligned}
\varepsilon_1 &= b^T\mathbf{1} = 1 \\
\varepsilon_2 &= b^T A\mathbf{1} = \frac{1}{2} \\
\varepsilon_3 &= b^T A^2\mathbf{1} = \frac{1}{6} \\
\varepsilon_4 &= b^T A^3\mathbf{1} = \frac{1}{24}\,,
\end{aligned}$$

womit die Stabilitätsfunktion

$$\mu(H) = 1 + H + \frac{1}{2}H^2 + \frac{1}{6}H^3 + \frac{1}{24}H^4$$

lautet. ◀

Wegen der Rekursionsdarstellung $u_j = \mu(H)u_{j-1}$ ist ein betrachtetes RK-Verfahren offenbar genau dann numerisch stabil, wenn seine Stabilitätsfunktion der Bedingung $|\mu(H)| < 1$ genügt.

Definition 3.4 (Absolute Stabilität)

(i) Das ESV 5 heißt absolut stabil für ein $H \in \mathbb{C}$, falls $|\mu(H)| < 1$ gilt.

(ii) Die Menge

$$\mathbf{H} = \{\ H \in \mathbb{C} \mid\ |\mu(H)|\ < 1\ \} \tag{39}$$

heißt Stabilitätsbereich (Bereich absoluter Stabilität) des Einschrittverfahrens.

Die Stabilitätsbereiche **H** der Runge-Kutta-Verfahren der Ordnungen $p = 1, 2, 3, 4$ sind in den Abbildungen 5, 6, 7 und 8 dargestellt.

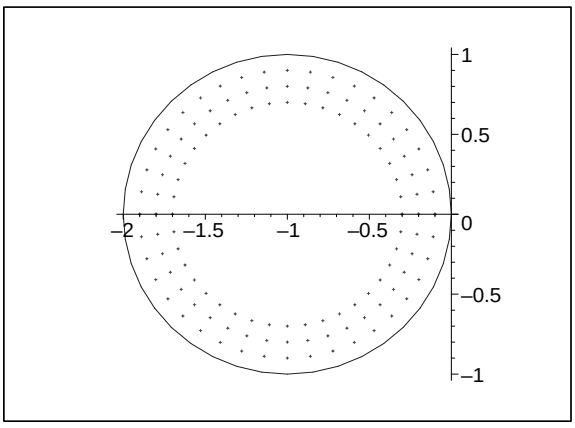

Abbildung 5: Stabilitätsbereich des expliziten Euler-Verfahrens

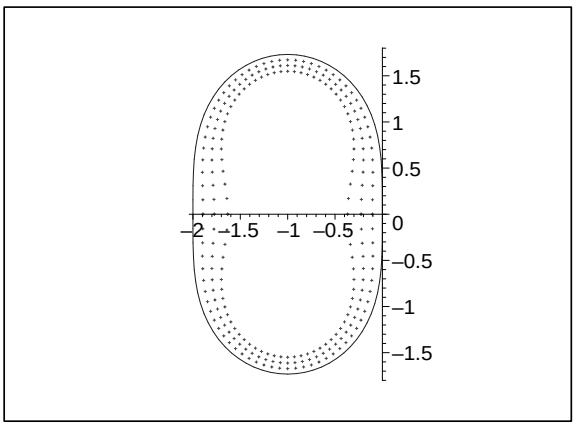

Abbildung 6: Stabilitätsbereich des modifizierten Euler-Verfahrens

Wegen der Beschränktheit der Stabilitätsbereiche expliziter Runge-Kutta-Verfahren muß stets eine Schrittweitenbedingung $h < C/|\lambda|$ mit einer verfahrensabhängigen Konstanten $C > 0$ erfüllt sein, um konvergierende Lösungen zu erhalten. Lautet z. B. die Zeitkonstante $\lambda = -10^4$ in (34), so muß man über 2500 Integrationsschritte ausführen, um die Testgleichung auf dem Zeitintervall [0,1] mit dem (4,4)-RK-Verfahren zu integrieren! Abb.9 enthält die Stabilitätsbereiche der expliziter Runge-Kutta-Verfahren der Ordnungen 1 – 4.

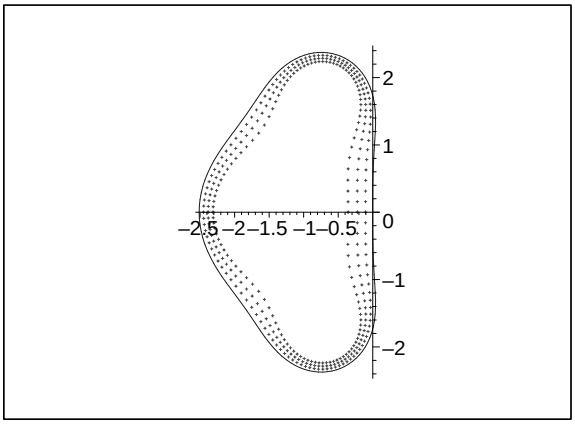

Abbildung 7: Stabilitätsbereich des (3,3)-RK-Verfahrens

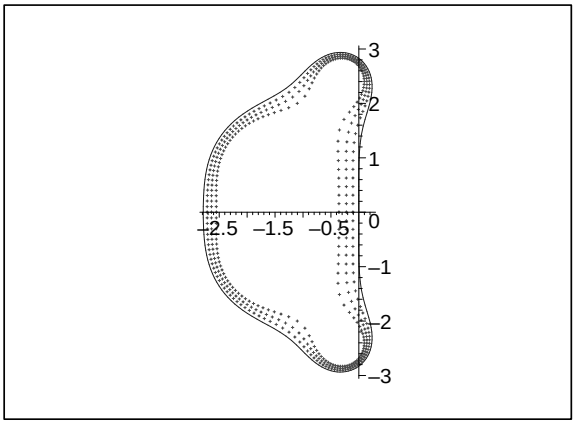

Abbildung 8: Stabilitätsbereich des (4,4)-RK-Verfahrens

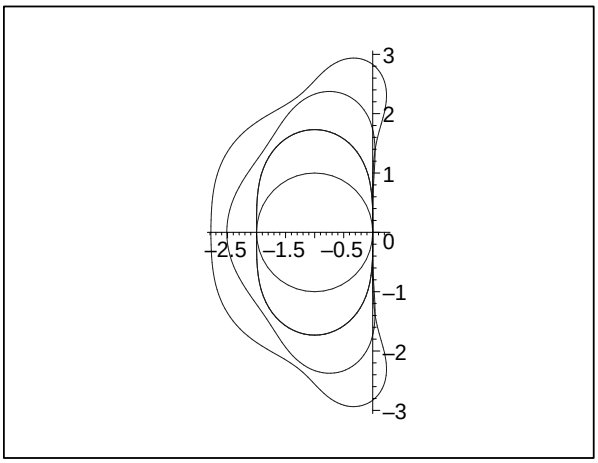

Abbildung 9: Stabilitätsbereiche der expliziten RK-Verfahren der Ordnungen 1-4 (von innen nach außen)

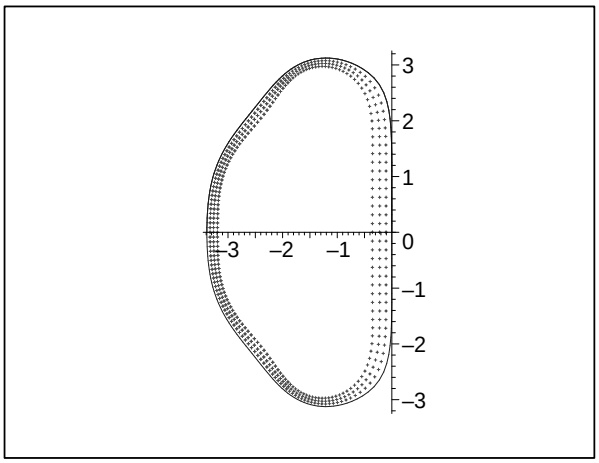

Abbildung 10: Stabilitätsbereich des (5,6)-RK-Verfahrens von Butcher aus Beispiel 2.18

Bemerkung 3.5 Die Bestimmung von **H** ist bei RK-Verfahren höherer Ordnung eine nichttriviale Aufgabe. Numerisch ermittelt man den Rand $\partial\mathbf{H}$ des Stabilitätsbereichs, dessen Punkte H der Gleichung

$$\mu(H) = e^{i\varphi}\,, \quad 0 \le \varphi < 2\pi\,, \quad i = \sqrt{-1} \tag{40}$$

genügen. Dann ist $\partial\mathbf{H}$ das implizit durch (41) gegebene Bild des Einheitskreises $\mu = e^{i\varphi}$ der komplexen $\mu-$Ebene in der $H-$Ebene.

Beispiel 3.6 Das 6-stufige RK-Verfahren von Butcher aus Beispiel 2.18 hat die Stabilitätsfunktion

$$\mu(H) = 1 + H + \frac{1}{2}H^2 + \frac{1}{6}H^3 + \frac{1}{24}H^4 + \frac{1}{120}H^5 + \frac{1}{640}H^6.$$

Sie stimmt mit der Reihenentwicklung der Exponentialfunktion

$$\exp(H) = \sum_{k=0}^{\infty} \frac{1}{k!}H^k \;=\; 1 + H + \frac{1}{2}H^2 + \frac{1}{6}H^3 + \frac{1}{24}H^4 + \frac{1}{120}H^5 + \frac{1}{720}H^6 + \ldots$$

nur bis zum 5.Summanden überein. Ihr Stabilitätsbereich wird in Abb.10 dargestellt. ◄

Bemerkung 3.7 Die Stabilitätsfunktion eines expliziten $(p, s)-$Runge-Kutta-Verfahrens approximiert die Exponentialfunktion $\exp(H)$ bis zur Konsistenzordnung p und besitzt als Polynom $s-$ten Grades dann stets die Gestalt

$$\mu(H) = \sum_{k=0}^{p} \frac{1}{k!} H^k \; + \; \sum_{k=p+1}^{s} \frac{c_k}{k!} H^k$$

mit $c_{p+1} \neq 1$ (vgl. K.Strehmel & R.Weiner).

3.2 Steife DGL-Systeme

Kehren wir zum Ausgangsbeispiel 3.1 zurück, das in Matrixform

$$\dot{x} = Ax, \quad x(0) = x_0 \qquad \text{mit der Matrix} \qquad A \; = \; \begin{pmatrix} -29998 & -39996 \\ 14998.5 & 19997 \end{pmatrix}$$

lautet. Mit den Eigenwerten $\lambda_1 = -1$ und $\lambda_2 = -10000$ von A und der Ähnlichkeitstransformation $\Lambda = TAT^{-1}$ kann es "entkoppelt" und auf die Form der Testgleichung (34)

$$\dot{z}_i = \lambda_i z_1, \quad z_i(a) = z_{i0}, \quad i = 1, 2$$

in $\mathbb{C}$ reduziert werden. Problematisch für die Anwendung expliziter RK-Verfahren ist offenbar das große Verhältnis $|\lambda_2|/|\lambda_1| = 10^4$ der beiden Eigenwerte. Eine Neuskalierung der Zeitvariablen t kann dabei keine Abhilfe schaffen! Derartige „steife“ DGL-Systeme treten jedoch in zahlreichen Anwendungen auf, insbesonder bei Systemen mit unterschiedlichen Zeitkonstanten und bei Semidiskretisierungen partieller Differentialgleichungen.

Definition 3.8 (Steife DGL-Systeme)
Für die Eigenwerte der Matrix A des linearen DGL-Systems $\dot{x} = Ax$ sei $Re\,\lambda_i < 0$ für $i = 1(1)n$. Dann heißt das System steif, falls auf $I = [a, b]$

$$\kappa \; := \; (b-a) \cdot \frac{\max\limits_{i=1(1)n} |\lambda_i|}{\min\limits_{i=1(1)n} |\lambda_i|} \; \gg \; 1 \qquad \textit{gilt.}$$

In praxi bezeichnet man oft lineare DGL-Systeme mit $\kappa \geq 10^4$ als steif. Vorteilhaft sind in jedem – linearen oder nichtlinearen – Fall Diskretisierungsverfahren mit unbeschränkten Stabilitätsbereichen $\mathbf{H}$, die keine Schrittweitenbeschränkungen aus Stabilitätsgründen erfordern.

Definition 3.9 (A-Stabilität, L-Stabilität)
Ein Diskretisierungsverfahren heißt unbeschränkt absolut stabil (A-stabil), falls $\mathbb{C}^- \subseteq \mathbf{H}$ gilt, also $\mathbf{H}$ die gesamte linke Halbebene $\mathbb{C}^-$ umfaßt. Ist zudem

$$\lim_{Re\,H \to -\infty} |\mu(H)| \; < \; 1 \, ,$$

so heißt es stark A-stabil. A-stabile Verfahren mit der zusätzlichen Eigenschaft $\lim_{Re\,H \to -\infty} \mu(H) \; = \; 0$ werden als L-stabil bezeichnet.

Für die skalare Testgleichung $\dot{x} = \lambda x, \ \ \lambda \in \mathbb{C}^-$ bedeutet A-Stabilität offensichtlich, daß keinerlei Schrittweitenbeschränkung aus Stabilitätserwägungen erforderlich wird. Lediglich aus Genauigkeitgründen muß nunmehr die Schrittweite gesteuert werden. Explizite Einschrittverfahren können offenbar niemals A-stabil sein. Denn ihre Stabilitätsfunktion $\mu(H)$ ist als algebraisches Polynom stets unbeschränkt für reelle Werte $H \to -\infty$. Wir konstruieren deshalb implizite Verfahren.

3.3 Implizites Euler-Verfahren

Das einfachste implizites Einschrittverfahren erhält man durch Taylorentwicklung der Lösung $x(t_{j-1})$ an der Stelle $t = t_j$

$$x(t_{j-1}) = x(t_j) - h\dot{x}(t_j) + \int_0^1 (1-\tau)\ddot{x}(t_j - \tau h)h^2 \, d\tau \, , \tag{41}$$

sowie Anwendung der DGL und Vernachlässigung des integralen Restterms

$$u_j = u_{j-1} + h \ f(t_j, u_j) \, , \quad j = 1(1)N. \tag{42}$$

Die Verfahrensfunktion dieses *impliziten Euler-Verfahrens* kann dann zu

$$\Phi(t, x, h) \ = \ f(t + h, x + h\Phi(t, x, h)) \tag{43}$$

angegeben werden.

Bemerkung 3.10 Zur Berechnung des Wertes u_j sollte im nichtlinearen Fall das Newtonsche Näherungsverfahren oder eine Modifikation davon benutzt werden. Definiert man dazu die Nullstellen-Gleichung

$$g(u) \ \equiv \ u - u_{j-1} - h \ f \ (t_j, u) \ = \ 0 \, ,$$

so lautet mit der Ableitung

$$g'(u) = I - h \ \frac{\partial f}{\partial x}(t_j, u), \quad I - \text{Einheitsmatrix}$$

das Newton-Verfahren für die $(k+1)$-te Iterierte

$$\begin{aligned} u_j^{(k+1)} \ &= \ u_j^{(k)} - \left(g'(u_j^{(k)})\right)^{-1} \ g(u_j^{(k)}) \\ &= \ u_j^{(k)} - \left(I - h\frac{\partial f}{\partial x}(t_j, u_j^{(k)})\right) \left(u_j^{(k)} - u_{j-1} - h \ f(t_j, u_j^{(k)})\right) \end{aligned}$$

für $k = 0, 1, 2, \ldots$

Das *implizite Euler-Verfahren* (42) ist konsistent mit Ordnung 1, wenn die Funktion $f(t, x)$ stetig differenzierbar ist. Denn der lokale Diskretisierungsfehler τ_j läßt sich wegen der Taylorentwicklung (41) leicht durch

$$||\tau_j|| \ = \ ||\frac{1}{h}[x(t_j) - x(t_{j-1})] - f(t_j, x(t_j))|| \ \leq \ Kh \, , \quad K > 0$$

abschätzen. Mit der Verfahrensfunktion $\Phi(t, x, h)$ kann man dies auch mittels des Satzes 2.5 zeigen. Die Ausführbarkeit des impliziten Euler-Schrittes ist für alle Schrittweiten $h \in (0, h_0]$ mit hinreichend kleinem h_0 nach dem Satz über die implizite Funktion garantiert. Schließlich liefert Satz 2.12 die Konvergenz des Verfahrens mit Ordnung 1, wenn die Anfangswerte konsistent mit dieser Ordnung sind.

Um den Stabilitätsbereich **H** des Verfahrens zu bestimmen, ermittelt man die Stabilitätsfunktion gemäß (37) wegen $A = (1)$ und $b = (1)$ zu

$$\mu(H) = \frac{1}{1 - H}\,, \tag{44}$$

Der Stabilitätsbereich **H** ist das Äußere des angegebenen Einheitskreises in Abb. 11. Wegen $\mathbb{C}^- \subseteq$ **H** ist das implizite Euler-Verfahren A-stabil und darüber hinaus sogar L-stabil.

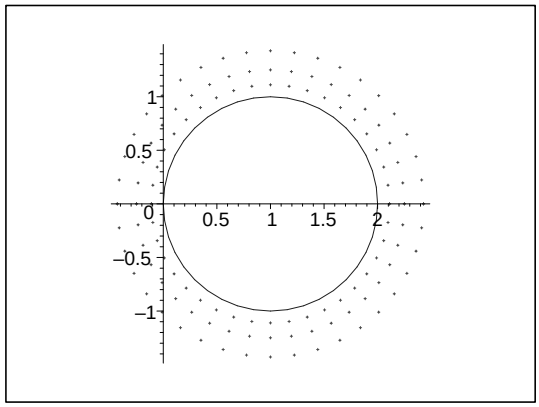

Abbildung 11: Stabilitätsbereich des impliziten Euler-Verfahrens

Folgerung 3.11 *Ist $f \in C^1(S)$ und $e_0 = O(h)$, so existiert ein $h_0 > 0$, mit dem das implizite Euler-Verfahren für alle Schrittweiten $h \in (0, h_0]$ ausführbar ist. Das Verfahren konvergiert mit Ordnung 1, d.h. $e_j = O(h)$, $j = 0(1)N$, und es ist L-stabil.*

3.4 Implizite Runge-Kutta-Verfahren

Die Verallgemeinerung des impliziten Euler-Verfahrens führt auf die impliziten Runge-Kutta-Verfahren mit folgender algorithmischer Darstellung:

Algorithmus 3.12 (Implizite Runge-Kutta-Verfahren)

Function IMPLIZITE-RK-TYP $(f, A, b, c, t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $k_1 = f(t + c_1 h,\ x + h a_{11} k_1 + ... + h a_{1s} k_s)$
   $k_2 = f(t + c_2 h,\ x + h a_{21} k_1 + ... + h a_{2s} k_s)$
   $\dots \qquad \dots$
   $k_s = f(t + c_s h,\ x + h a_{s1} k_1 + ... + h a_{ss} k_s)$
3. $t_{neu} = t + h$
   $x_{neu} = x + h(b_1 k_1 + ... + b_s k_s)$
4. Ausgabe : $t_{neu},\ x_{neu}$

Das Parameterschema besitzt infolgedessen die quadratische Darstellung

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Damit sind formal auch die expliziten RK-Verfahren in diesen Schemata enthalten. Man klassifiziert deshalb mit folgender

Definition 3.13 *Ein $(p,s)-$Runge-Kutta-Verfahren heißt*

(i) *explizit, falls $a_{ij} = 0$ für alle $i \leq j$ ist.*

(ii) *diagonal-implizit, falls $a_{ij} = 0$ für alle $i < j$, aber $a_{ii} \neq 0$ für ein i ist.*

(iii) *voll-implizit, falls ein $a_{ij} \neq 0$ mit $i < j$ existiert.*

Beispiele 3.14

1. Die *Gauß-Legendre-Formel* mit $u_j = u_{j-1} + hk_1$ und

$$k_1 = f(t_{j-1} + \frac{1}{2}h,\ u_{j-1} + \frac{1}{2}h\ k_1)$$

ist ein einstufiges (2,1)-RK-Verfahren der Ordnung $p = 2$ mit der Stabilitätsfunktion

$$\mu(H) = \frac{1 + \frac{1}{2}H}{1 - \frac{1}{2}H},$$

das zudem A-stabil mit $\mathbf{H} = \mathbb{C}^-$ ist.

2. Das *Radau–IA–Verfahren*, das durch das Parameterschema

$$\begin{array}{c|cc} 0 & 1/4 & -1/4 \\ 2/3 & 1/4 & 5/12 \\ \hline & 1/4 & 3/4 \end{array}$$

definiert wird, ist ein implizites (3,2)- Runge-Kutta-Verfahren der Ordnung $p = 3$. Mit der Stabilitätsfunktion

$$\mu(H) = \frac{1 + \frac{1}{3}H}{1 - \frac{2}{3}H + \frac{1}{6}H^2}$$

läßt sich leicht seine L-Stabilität zeigen; der Stabilitätsbereich ist in Abb.12 dargestellt.

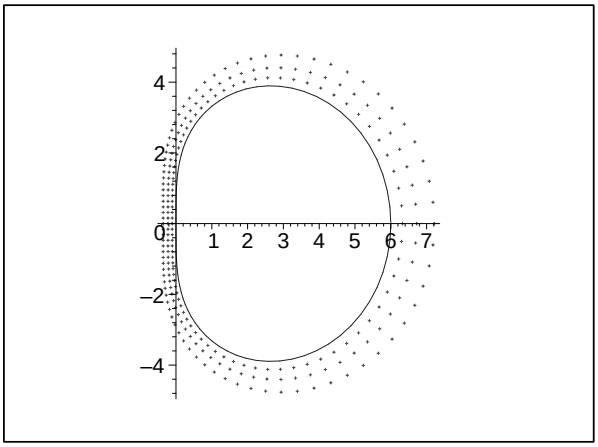

Abbildung 12: Stabilitätsbereich des (3,2)-Radau–IA–Verfahrens

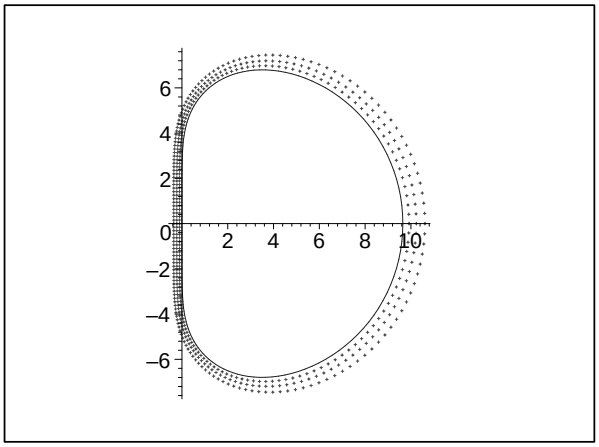

Abbildung 13: Stabilitätsbereich des (6,4)-Lobatto–IIIC–Verfahrens

3. Schließlich besitzt das 4-stufige Lobatto-IIIC-Verfahren die RK-Matrix

$$A = \begin{pmatrix} \frac{1}{12} & -\frac{1}{12}\sqrt{5} & \frac{1}{12}\sqrt{5} & \frac{-1}{12} \\ \frac{1}{12} & \frac{1}{4} & \frac{1}{6} - \frac{7}{60}\sqrt{5} & \frac{1}{60}\sqrt{5} \\ \frac{1}{12} & \frac{1}{6} + \frac{7}{60}\sqrt{5} & \frac{1}{4} & -\frac{1}{60}\sqrt{5} \\ \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{pmatrix}$$

sowie die RK-Gewichte und RK-Knoten

$$b = \left(\frac{1}{12}, \frac{5}{12}, \frac{5}{12}, \frac{1}{12}\right) \qquad c = \left(0, \frac{1}{2} - \frac{1}{10}\sqrt{5}, \frac{1}{2} + \frac{1}{10}\sqrt{5}, 1\right) .$$

Dieses implizites Runge-Kutta-Verfahren hat die Ordnung $p = 6$. Mit der Stabilitätsfunktion

$$\mu(H) = \frac{1 + \frac{1}{3}H + \frac{1}{30}H^2}{1 - \frac{2}{3}H + \frac{1}{5}H^2 - \frac{1}{30}H^3 + \frac{1}{360}H^4}$$

erkennt man die L-Stabilität dieses Verfahrens; der Stabilitätsbereich ist in Abb.13 dargestellt. ◀

Wegen ihrer L-Stabilität sind das implizite Euler-Verfahren und die betrachteten Radau- und Lobatto-Verfahren besonders für die Langzeitintegration dissipativer DGL-Systeme – z.B. gedämpfter Schwingungssysteme – geeignet. Bei der Integration konservativer DGL-Systeme erweisen sich jedoch meist die A-stabilen Gauß-Legendre-Verfahren überlegen.

Implizite Runge-Kutta-Verfahren sind rechenzeitaufwendig, denn die Berechnung der Vektoren k_i erfordert in jedem Schritt des Algorithmus 3.12 die Lösung eines nichtlinearen Gleichungssystems mit $s{\cdot}n$ Unbekannten. Ist dagegen im Parameterschema $a_{ij} = 0$ für $i < j$, also

das Verfahren diagonal - implizit, so können die Steigungsvektoren k_i sukzessive als Lösung jeweils eines Gleichungssystems der Dimension n ermittelt werden. Besonders günstig sind linear-implizite Verfahren, die zur Bestimmung der k_i lediglich lineare Gleichungen auflösen.

3.5 Linear-implizite Einschrittverfahren

Eine Vereinfachung der behandelten Verfahren ergibt sich, wenn man Anfangswertprobleme für autonome DGL-Systeme

$$\frac{dx}{dt} = \dot{x} = f(x)\,, \quad x(a) = x_0\,, \quad f : \mathbb{R}^n \to \mathbb{R}^n \tag{45}$$

auf dem Intervall $I = [a, b]$ betrachtet. Offenbar treten dann die RK-Knoten c nicht explizit in den ESV auf. Nicht-autonome Systeme (2) können leicht durch Einführung der zusätzlichen Funktion $x_0(t) = t$ in die autonome Gestalt

$$\begin{aligned} \dot{x}_0 &= 1 \qquad\quad , \quad x_0(a) = a \\ \dot{x} &= f(x_0, x)\,, \quad x(a) = x_0 \end{aligned}$$

mit $n+1$ Gleichungen und Anfangswerten für die Funktionen $x(t) = (x_0(t), x_1(t), \ldots, x_n(t))^T$ überführt werden.

Wenden wir ein s-stufiges diagonal-implizites RK-Verfahren auf das autonome System (45) an, so können die Steigungsvektoren $k_1, k_2, \ldots, k_s$ sukzessive als Lösungen der n-dimensionalen Gleichungen

$$\begin{aligned} k_1 &= f(u_{j-1} + ha_{11}k_1) \\ k_2 &= f(u_{j-1} + ha_{21}k_1 + ha_{22}k_2) \\ \cdots &\qquad\quad \cdots \\ k_s &= f(u_{j-1} + ha_{s1}k_1 + ha_{s2}k_2 + \ldots + ha_{ss}k_s) \end{aligned}$$

gewonnen werden. Betrachten wir die i-te Gleichung als Nullstellenproblem für den Steigungsvektor k_i, so läßt sich eine Nullstelle der Funktion $g : \mathbb{R}^n \to \mathbb{R}^n$ mit

$$g(k_i) := k_i - f(u_{j-1} + ha_{i1}k_1 + ha_{i2}k_2 + \ldots + ha_{ii}k_i)$$

mittels des Newton-Verfahrens bestimmen. Dessen Iterationsvorschrift lautet dann

$$k_i^{\nu+1} = k_i^\nu - g'(k_i^\nu)^{-1} \cdot g(k_i^\nu)\,, \quad \nu = 0, 1, 2, \ldots,$$

woraus man nach Differentiation und Einsetzen der Funktion g die innere Iteration

$$\begin{aligned} k_i^{\nu+1} &= k_i^\nu - [I - ha_{ii}f'(\eta_i^\nu)]^{-1} \cdot [k_i^\nu - f(\eta_i^\nu)] \qquad \text{mit} \\ \eta_i^\nu &= u_{j-1} + ha_{i1}k_1 + ha_{i2}k_2 + \ldots + ha_{ii}k_i^\nu\,, \quad \nu = 0, 1, 2, \ldots \end{aligned} \tag{46}$$

gewinnt. Nachteilig erweist sich die ständige Neuberechnung der Jacobi-Matrix $f'(\eta_i^\nu)$ für jede Stufe i und jeden Iterationsschritt ν, so daß deren Approximation durch $f'(u_{j-1})$ naheliegt. Führt man lediglich einen einzigen Newtonschritt pro Stufenindex i aus, so ist nur ein lineares Gleichungssystem zu lösen. Mit einem geeigneten Startwert für k_i ergeben sich dann die auf H.H.Rosenbrock (1963) zurückgehenden Verfahren:

Definition 3.15 (Verfahren von Rosenbrock-Typ)

1. *Ein diagonal-implizites RK-Verfahren heißt linear-implizit, wenn zur Bestimmung jedes Steigungsvektors k_i ein lineares Gleichungssystem zu lösen ist.*

2. *Ein linear-implizites RK-Verfahren heißt Rosenbrock-Wanner-Verfahren (ROW-Methode), falls zur Bestimmung aller k_i ein- und dieselbe Koeffizientenmatrix der Form*

$$E \;=\; I - h\gamma f'(u_{j-1}) \tag{47}$$

   mit der Konstanten γ benutzt wird.

3. *Ein linear-implizites RK-Verfahren heißt W-Methode, falls die Koeffizientenmatrix die Form $E = I - h\gamma T$ mit einer beliebigen Matrix T hat.*

Die Verfahrensmodifikationen 2 und 3 gehen auf G.Wanner (1977) sowie T.Steihaug und A.Wolfbrandt (1979) zurück.

3.6 Linear-implizites Euler-Verfahren

Das einfachste linear-implizite Einschrittverfahren gewinnt man aus dem impliziten Euler-Verfahren (42), wenn man es auf autonome DGL-Systeme anwendet:

$$\begin{aligned} k_1 &= f(u_{j-1} + h\; k_1) \\ u_j &= u_{j-1} + h\; k_1\,, \quad j = 1(1)N. \end{aligned} \tag{48}$$

Die obige Darstellung (46) des Newton-Verfahrens für k_1 ergibt mit der trivialen Startnäherung $k_1^0 = 0$ nach einem Newtonschritt die Näherung

$$k_1 \;=\; [I - h\; f'(u_{j-1})]^{-1}\;\; f(u_{j-1})\,, \qquad I \;-\; \text{Einheitsmatrix.}$$

Einsetzen dieses Wertes anstelle des dortigen k_1 macht aus (48) das *linear-implizite Euler-Verfahren (LIEV)*

$$u_j \;=\; u_{j-1} + h\;\; [I - h\; f'(u_{j-1})]^{-1}\;\; f(u_{j-1})\,, \quad j = 1(1)N\,, \tag{49}$$

dessen Verfahrensfunktion offenbar die Darstellung

$$\Phi(t, x, h) \;=\; [I - h\; f'(x)]^{-1}\;\; f(x) \tag{50}$$

besitzt.

Bemerkung 3.16 Bei der praktischen Realisierung des Verfahrens vermeidet man die aufwendige Berechnung der inversen Matrix in der Darstellung (49) und erhält folgende algorithmische Form:

Algorithmus 3.17 (Linear-implizites Euler-Verfahren)

Function LIEV $(f, f', t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $E \ = \ I - hf'(x)$
3. Löse $\quad E \cdot k_1 = hf(x)$
4. $x_{neu} \ = \ x + k_1$
   $t_{neu} \ = \ t + h$
5. Ausgabe : t_{neu}, x_{neu}

Die Konsistenz und Konvergenz dieses Verfahrens läßt sich mit Hilfe der allgemeinen Sätze zeigen; wegen des expliziten Auftretens der Ableitung f' muß jedoch höhere Glattheit vorausgesetzt werden.

Folgerung 3.18 *Ist $f \in C^2(S)$ mit $S = \{x \mid ||x|| \leq M\}$ und $e_0 = O(h)$, so existiert ein $h_0 > 0$, mit dem das implizite Euler-Verfahren für alle Schrittweiten $h \in (0, h_0]$ ausführbar ist. Das Verfahren konvergiert mit Ordnung 1 und ist L-stabil.*

Beweis: Nach Voraussetzung ist die Verfahrensfunktion Φ für hinreichend kleines $h \in (0, h_0]$ stetig differenzierbar mit $\Phi(t, x, 0) = f(x)$. Nach Satz 2.7 ist damit das Verfahren konsistent mit Ordnung $p = 1$. Wegen der Konsistenz der Anfangswerte mit Ordnung 1 liefert Satz 2.12 auch die Konvergenz des Verfahrens mit Ordnung 1. Bei Anwendung des Verfahrens auf die skalare Test-DGL $\dot{x} = \lambda x$ ergibt sich die Stabilitätsfunktion

$$\mu(H) = \frac{1}{1 - H}$$

des impliziten Euler-Verfahrens, womit die L-Stabilität folgt. □

3.7 ROW-Verfahren

Wir verallgemeinern nun diesen Zugang, um zu einem s-stufigen Rosenbrock-Wanner-Verfahren zu gelangen. Wählen wir als Startnäherung der inneren Iteration (46) eine Linearkombination aus den zuvor berechneten Steigungswerten

$$k_i^0 \ = \ c_{i1}k_1 + c_{i2}k_2 + \ldots c_{i,i-1}k_{i-1} \ ,$$

so läßt sich - nach einigen Umformungen und Umbenennungen - das Verfahren in folgender algorithmischer Form angeben:

Algorithmus 3.19 (Rosenbrock-Wanner-Verfahren)

Function ROW $(f, f', t, x, h, t_{neu}, x_{neu})$

1. Eingabe : $t,\ x,\ h$
2. $E \ = \ I - h\gamma f'(x)$
3. Löse mit der LU-Zerlegung von E

   $E \cdot k_1 = hf(x)$

   $E \cdot k_2 = hf(x + a_{21}k_1) + c_{21}k_1$

   $E \cdot k_3 = hf(x + a_{31}k_1 + a_{32}k_2) + c_{31}k_1 + c_{32}k_2$

   $\ldots \qquad \ldots$

   $E \cdot k_s = hf(x + a_{s1}k_1 + \ldots + a_{s,s-1}k_{s-1}) + c_{s1}k_1 + \cdots + c_{s,s-1}k_{s-1}$
4. $x_{neu} \ = \ x + b_1k_1 + b_2k_2 + \ldots + b_sk_s$

   $t_{neu} \ = \ t + h$
5. Ausgabe : t_{neu}, x_{neu}

Parametersätze $(\gamma, a_{ij}, c_{ij}, b_j)$ für spezielle ROW-Verfahren findet man über die angegebene Literatur. Die Prinzipien der lokalen Fehlerschätzung und der Schrittweitensteuerung expliziter Verfahren lassen sich unter entsprechenden Glattheitsvoraussetzungen auch auf ROW-Verfahren übertragen. So liefern *eingebettete* $p(p+1)-$*Verfahren* mit $s+1$ Stufen die zwei Näherungen für den exakten Lösungswert $x(t_j)$:

$$\begin{aligned} u_j &= u_{j-1} + b_1k_1 + b_2k_2 + \ldots + b_sk_s \\ u_j^* &= u_{j-1} + b_1^*k_1 + b_2^*k_2 + \ldots + b_s^*k_s + b_{s+1}^*k_{s+1} \ . \end{aligned}$$

Nach Satz 2.23 stellt $(u_j^* - u_j)/h$ einen asymtotischen Fehlerschätzer der Ordnung $p+1$ dar. Für das "bessere" $(p+1, s+1)-$Verfahren bildet man die Differenz der beiden Darstellungen

$$Error^* \ := \ u_j^* - u_j \ = \ e_1k_1 + e_2k_2 + \ldots + e_sk_s + e_{s+1}k_{s+1} \tag{51}$$

mit den Parametern $e_i = b_i^* - b_i$, $i = 1(1)s$ und $e_{s+1} = b_{s+1}^*$, womit die Gewichte b_i entbehrlich werden.

Beispiel 3.20 Von B.A.Gottwald und G.Wanner (1981) stammt ein eingebettetes 4-stufiges ROW-Verfahren der Ordnungen 3(4) mit folgenden Parameterwerten:

$\gamma = 0.395$	
$a_{21} = 0.438$ $a_{31} = 0.938948678483428$ $a_{32} = 0.0730795420615381$ $a_{41} = a_{31}$ $a_{42} = a_{32}$ $a_{43} = 0$	$c_{21} = -1.94347441894707$ $c_{31} = 0.416957530989189$ $c_{32} = 1.32396782072923$ $c_{41} = 1.51951325778448$ $c_{42} = 1.35370815030093$ $c_{43} = -0.854151495257539$
$e_1 = -0.190858871999474$ $e_2 = 0.255608791716455$ $e_3 = -0.863816280897592$ $e_4 = 0.25$	$b_1 = 0.729044879960308$ $b_2 = 0.0541069773272405$ $b_3 = 0.281599362440017$ $b_4 = 0.25$

Beispiel 3.21 Wir lösen das Anfangswertproblem aus Beispiel 3.1 für $0 \leq t \leq 1$

$$\begin{aligned} \dot{x_1} &= -29998x_1 - 39996x_2\,, \quad x_1(0) = 1 \\ \dot{x_2} &= 14998.5x_1 + 19997x_2\,, \quad x_2(0) = 1 \end{aligned}$$

mit dem eingebetteten ROW-Verfahren der Ordnung 3(4) von Gottwald und Wanner mit einer Genauigkeit $TOL = 10^{-8}$. Nachfolgend werden die Anzahl INT der ausgeführten Integrationsschritte, der jeweilige t-Wert sowie die Arbeitsschrittweite h dargestellt.

INT	Aktueller Wert t	Aktuelle Schrittweite h	Phase
12	7.26E-6	2.33E-6	
21	2.88E-5	2.45E-6	
60	1.38E-4	3.21E-6	1
100	2.94E-4	4.78E-6	
150	6.68E-4	1.22E-5	
183	2.30E-3	3.09E-4	
191	2.51E-2	7.92E-3	
200	1.32E-1	1.60E-2	
220	3.53E-1	1.55E-2	2
240	6.25E-1	2.09E-2	
266	1.00E+0	1.61E-2	

Mit 266 Schritten bis zum Endwert $t = 1$ war das Verfahren wesentlich schneller als das explizite RK-Verfahren aus Beispiel 3.1 trotz seines höheren algebraischen Aufwandes. ◀

Zur Theorie linear-impliziter Verfahren höherer Ordnung sei auf die angegebene Literatur verwiesen. Dort findet man auch Hinweise auf publizierte Parametersätze spezieller ROW-Verfahren.

4 Lineare Mehrschrittverfahren

Die bisher behandelten Einschrittverfahren benutzen zur Bestimmung der Näherungslösung $u_j \approx x(t_j)$ lediglich den vorhergehenden Wert u_{j-1}. Um die geringe Genauigkeit des Euler-Verfahrens zu verbessern, mußten deshalb s Berechnungen der rechten DGL-Seiten $f(t,x)$ an speziellen Zwischenwerten erfolgen. Einen effizienteren Zugang bieten *Mehrschrittverfahren*, die zur Ermittlung von u_j lediglich auf die bereits berechneten Werte $u_{j-1},\ u_{j-2}, \ldots, u_{j-k}$ zurückgreifen. Offenbar stehen damit auch die Werte der rechten DGL-Seiten $f_i := f(t_i, u_i)$ für $i = j-1, j-2, \ldots, j-k,$ zur Verfügung. Ein Mehrschrittverfahren, in dem die k Werte $u_j, u_{j-1}, u_{j-2}, \ldots, u_{j-k}$ und die zugehörigen Ableitungswerte f_i linear auftreten, heißt *lineares Mehrschrittverfahren* (LMSV) oder genauer *lineares k-Schritt-Verfahren.*

4.1 Konstruktion linearer Mehrschrittverfahren

Setzen wir die nach Voraussetzung 1.1 existierende Lösung $x(t)$ des Anfangswertproblems in die DGL (2) ein und integrieren beide Seiten über das Zeitintervall $[t_{j-p}, t_j]$, so läßt sich $x(t_j)$ durch den Ausdruck

$$x(t_j) = x(t_{j-p}) + \int_{t_{j-p}}^{t_j} f(t, x(t))\, dt \tag{52}$$

darstellen. (vgl. Abb. 14). Bei gegebenem $p \geq 1$ bieten sich zahlreiche Quadraturformeln zur Approximation des Integrals an, wobei wir uns wegen der konstanten Schrittweite $h = t_j - t_{j-1}$ auf Newton-Cotes-Formeln (z.B. Rechteckregeln, Trapezregel, Simpsonregel, Milneregel) beschränken. Abgeschlossene Formeln benutzen den Knoten t_j und führen deshalb auf implizite Verfahren, wogegen rechts offene Quadraturformeln explizite Verfahren liefern.

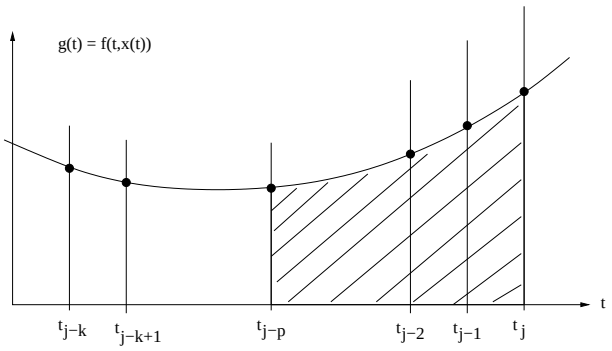

Abbildung 14: Lineare Mehrschrittverfahren

Beispiele 4.1

1. Ersetzt man mit $p = 1$ das Integral durch die Rechteckregel/links, so so entsteht das *explizite Euler-Verfahren*

$$u_j \ = \ u_{j-1} + h\ f(t_j, u_j)\ , \quad j = 1(1)N. \tag{53}$$

2. Approximiert man dagegen das Integral durch die Trapezregel, so so erhält man das *implizite Heun-Verfahren*

$$u_j \ = \ u_{j-1} + \frac{h}{2}\left[f(t_j, u_j) + f(t_{j-1}, u_{j-1})\right]\ , \quad j = 1(1)N. \tag{54}$$

3. Die Simpsonregel auf dem Intervall $[t_{j-2}, t_j]$ der Länge $2h$ liefert ein implizites 2-Schritt-Verfahren

$$u_j \;=\; u_{j-2} + \frac{h}{3}\left[f(t_j, u_j) + 4f(t_{j-1}, u_{j-1}) + f(t_{j-2}, u_{j-2})\right] \;, \quad j = 2(1)N \;, \tag{55}$$

   das unter dem Namen *Adams-Störmer-Verfahren* bekannt ist.

4. Ein explizites 2-Schritt-Verfahren gewinnt man mit der *Mittelpunktregel* für $p = 2$

$$u_j \;=\; u_{j-2} + 2h\; f(t_{j-1}, u_{j-1}) \;, \quad j = 2(1)N. \qquad \blacktriangleleft \tag{56}$$

Im Gegensatz zur Newton-Cotes-Quadratur lassen sich jedoch auch Knoten außerhalb des Integrationsintervalles zur Approximation des Integrals heranziehen. Die bekannteste Verfahrensgruppe stellen die *Adams-Bashforth-Verfahren (AB)* dar, die auf dem Intervall $[t_{j-1}, t_j]$ den Integranden $g(t) = f(t, x(t))$ durch das Interpolationspolynom $N_{k-1}(t)$ an den k zurückliegenden Knoten

$$(t_{j-1}, f_{j-1}),\; (t_{j-2}, f_{j-2}),\; \ldots,\; (t_{j-k}, f_{j-k})$$

ersetzen:

$$u_j \;=\; u_{j-1} + \int_{t_{j-1}}^{t_j} N_{k-1}(t)\, dt \;. \tag{57}$$

Nach Integration des Interpolationspolynoms $N_{k-1}(t)$ ergibt sich eine allgemeine Darstellung dieser Verfahrensklasse in der Form

$$u_j \;=\; u_{j-1} + h\left[c_0 f_{j-1} + c_1 \nabla f_{j-1} + c_2 \nabla^2 f_{j-1} + \ldots + c_{k-1}\nabla^{k-1} f_{j-1}\right] \;, \tag{58}$$

wobei die die sogenannten Rückwärtsdifferenzen ν-ter Ordnung rekursiv durch

$$\nabla^\nu f_i \;:=\; \nabla^{\nu-1} f_i - \nabla^{\nu-1} f_{i-1} \;, \quad \nu = 1, 2, 3, \ldots \quad \text{mit} \quad \nabla^0 f_i \;:=\; f_i$$

definiert werden. Die Konstanten c_ν ergeben sich zu

$$c_\nu \;=\; \int_0^1 (-1)^\nu \binom{-s}{\nu}\, ds \;, \quad \nu = 0(1)k-1 \;,$$

insbesondere sind $c_0 = 1,\; c_1 = 1/2,\; c_2 = 5/12,\; c_3 = 3/8$.

Beispiele 4.2

1. Das *1-Schritt-Verfahren AB1* ist wegen $c_0 = 1$ identisch mit dem expliziten Euler-Verfahren

$$u_j \;=\; u_{j-1} + h\; f_{j-1} \;, \quad j = 1(1)N. \tag{59}$$

2. Das *2-Schritt-Verfahren AB2* erhält man in der Standardform durch Ausrechnen der Rückwärtsdifferenzen zu

$$\begin{aligned} u_j &= u_{j-1} + h\left[f_{j-1} + \frac{1}{2}\nabla f_{j-1}\right] \\ &= u_{j-1} + h\left[f_{j-1} + \frac{1}{2}(f_{j-1} - f_{j-2})\right] \\ &= u_{j-1} + \frac{h}{2}\left[3f_{j-1} - f_{j-2}\right] \;, \quad j = 2(1)N. \end{aligned}$$

3. Das *klassische 4-Schritt-Verfahren AB4* lautet in der Standardform

$$u_j \;=\; u_{j-1} + \frac{h}{24}\left[55f_{j-1} - 59f_{j-2} + 37f_{j-3} - 9f_{j-4}\right] \;, \quad j = 4(1)N \;. \qquad \blacktriangleleft \qquad (60)$$

Beispiel 4.3 Wir lösen das Anfangswertproblem 2.10

$$\dot{x} \;=\; t - x^2 \;, \quad x(0) = 0 \;, \quad I = [0,1]$$

mit dem *klassischen 4-Schritt-Adams-Bashforth*-Verfahren AB4 mit konstanter Schrittweite $h = 0.1$. Die zusätzlichen 3 Startwerte u_1, u_2, u_3 beschaffen wir uns

(1) mittels des klassischen Runge-Kutta-(4,4)-Verfahrens mit Schrittweite $h = 0.1$

(2) mittels Taylorentwicklung der Lösung $x(t)$ am Startwert und Berechnung der u_j auf maximale Genauigkeit.

t_j	u_j gemäß (1)	u_j gemäß (2)
0.4	0.079 511 663	0.079 511 263
0.5	0.123 497 354	0.123 496 959
0.6	0.176 260 847	0.176 260 464
0.7	0.236 991 345	0.236 990 978
0.8	0.304 643 536	0.304 643 188
0.9	0.377 964 615	0.377 964 289
1.0	0.455 548 008	0.455 547 709
x(1)	0.455 544 526 08...	

Im Vergleich mit dem exakten Endwert $x(1) = 0,455\ 544\ 526\ 08...$ stellt man keinen signifikanten Genauigkeitsverlust der Variante (1) gegenüber (2) fest. Da die durchgehende Berechnung mit dem (4,4)-Runge-Kutta-Verfahren den nur geringfügig besseren Endwert $u_{10} = 0.455\ 543\ 843$ lieferte, ist zu vermuten, daß das AB4-Verfahren ebenfalls gute Konvergenzeigenschaften besitzt. ◀

4.2 Konsistenz linearer Mehrschrittverfahren

Alle bisher betrachteten LMSV lassen sich in der *allgemeinen Form linearer k-Schritt-Verfahren*

$$\alpha_k u_j + \alpha_{k-1}u_{j-1} + \ldots + \alpha_0 u_{j-k} \;=\; h\;\left[\beta_k f_j + \beta_{k-1}f_{j-1} + \ldots + \beta_0 f_{j-k}\right] \;, \quad j = k(1)N \;, \quad (61)$$

mit der Abkürzung $f_{j-k+i} = f(t_{j-k+i}, u_{j-k+i})$ darstellen bzw. in der kürzeren Summenschreibweise

$$\sum_{i=0}^{k} \alpha_i u_{j-k+i} \;=\; h \sum_{i=0}^{k} \beta_i f_{j-k+i} \;, \quad j = k(1)N \;. \qquad (62)$$

Bemerkung 4.4 Da der aktuellste Lösungsvektor u_j gesucht ist, sollte sein Koeffizient α_k nicht verschwinden. Dividiert man (62) durch α_k, so werden die Koeffizienten eindeutig, weshalb desweiteren stets $\alpha_k = 1$ vorausgesetzt werden kann. Offenbar entscheidet der Koeffizient β_k darüber, ob das LMSV explizit ($\beta_k = 0$) oder implizit ($\beta_k \neq 0$) ist.

Definition 4.5 (Erzeugende Polynome)
Die aus den Koeffizienten abgeleiteten Polynome maximal k-ten Grades

$$\varrho(\mu) \;=\; \alpha_k\mu^k + \ldots + \alpha_1\mu + \alpha_0 \;=\; \sum_{i=0}^{k} \alpha_i\mu^i \tag{63}$$

$$\sigma(\mu) \;=\; \beta_k\mu^k + \ldots + \beta_1\mu + \beta_0 \;=\; \sum_{i=0}^{k} \beta_i\mu^i \tag{64}$$

heißen 1. und 2. erzeugendes Polynom des LMSV.

Beispiele 4.6

1. Für das *implizite Heun-Verfahren* (54)

   $$u_j - u_{j-1} \;=\; \frac{h}{2}\,[f_j + f_{j-1}]$$

   erhält man die Polynome 1.Grades

   $$\varrho(\mu) = \mu - 1 \qquad \text{und} \qquad \sigma(\mu) = \frac{1}{2}\mu + \frac{1}{2}.$$

2. Die *explizite Mittelpunktregel* (56)

   $$u_j - u_{j-2} \;=\; 2h\; f(t_{j-1}, u_{j-1})$$

   liefert die Polynome

   $$\varrho(\mu) = \mu^2 - 1 \qquad \text{und} \qquad \sigma(\mu) = 2\mu.$$

3. Das *4-Schritt-AB-Verfahren* (60) in der Standardform

   $$u_j - u_{j-1} \;=\; \frac{h}{24}\,[55f_{j-1} - 59f_{j-2} + 37f_{j-3} - 9f_{j-4}]$$

   besitzt die erzeugenden Polynome

   $$\varrho(\mu) = \mu^4 - \mu^3 \quad \text{und} \quad \sigma(\mu) = \frac{55}{24}\mu^3 - \frac{59}{24}\mu^2 + \frac{37}{24}\mu - \frac{9}{24}. \qquad \blacktriangleleft$$

Den lokalen Diskretisierungsfehler τ_j erhält man wie schon bei den Einschrittverfahren, indem man die exakte Lösung $x(t)$ in das Verfahren einsetzt. Auch hier betrachtet man den durch die Schrittweite h dividierten Fehlervektor.

Definition 4.7 (Lokaler Diskretisierungsfehler)
Sei der führende Koeffizient $\alpha_k = 1$. *Der lokale Diskretisierungsfehler* τ_j *des LMSV an der Stelle* t_j *lautet*

$$\tau_j \;=\; \frac{1}{h}\sum_{i=0}^{k} \alpha_i x(t_{j-k+i}) \;-\; \sum_{i=0}^{k} \beta_i f(t_{j-k+i}, x(t_{j-k+i}))\,, \quad j = k(1)N\,. \tag{65}$$

Das LMSV ist konsistent, *falls* $\lim_{h\to 0} \tau_j = 0$ *für* $j = k(1)N$ *gilt.*

Um allgemeine Konsistenzbedingungen herzuleiten, nehmen wir vorerst an, daß die Lösung $x(t)$ analytisch ist und damit an der Stelle $t = t_{j-k}$ in eine Taylorreihe entwickelt werden kann. Ersetzt man in der Definititon des τ_j die rechten Seiten $f(t_{j-k+i}, x(t_{j-k+i}))$ durch die erste Ableitung der Lösung $\dot{x}(t_{j-k+i})$, so kann man nun in der Darstellung

$$\tau_j \;=\; \frac{1}{h}\sum_{i=0}^{k} \alpha_i x(t_{j-k} + ih) \;-\; \sum_{i=0}^{k} \beta_i \dot{x}(t_{j-k} + ih)$$

die Taylorentwicklungen einsetzen

$$\tau_j \;=\; \frac{1}{h}\sum_{i=0}^{k} \alpha_i \sum_{\nu=0}^{\infty} \frac{(ih)^\nu}{\nu!}\, x^{(\nu)}(t_{j-k}) \;-\; \sum_{i=0}^{k} \beta_i \sum_{\nu=0}^{\infty} \frac{(ih)^\nu}{\nu!}\, x^{(\nu+1)}(t_{j-k})\,,$$

woraus nach Umordnung die Entwicklung des lokalen Fehlers nach Potenzen von h

$$\tau_j \;=\; \frac{1}{h} c_0 x(t_{j-k}) + c_1 \dot{x}(t_{j-k}) + c_2 \ddot{x}(t_{j-k}) h + c_3 x^{(3)}(t_{j-k}) h^2 + \cdots \tag{66}$$

folgt. Die Konstanten c_ν sind durch die Formeln

$$\begin{aligned} c_0 &= \sum_{i=0}^{k} \alpha_i \\ c_1 &= \sum_{i=0}^{k} \alpha_i\, i \;-\; \sum_{i=0}^{k} \beta_i \\ c_\nu &= \frac{1}{\nu!}\sum_{i=0}^{k} \alpha_i\, i^\nu \;-\; \frac{1}{(\nu-1)!}\sum_{i=0}^{k} \beta_i\, i^{\nu-1} \\ &\qquad \nu = 2, 3, 4, \ldots \end{aligned} \tag{67}$$

eindeutig bestimmt. Verschwinden die Konstanten c_0 und c_1 in der Entwicklung (66), so kann die Konsistenz mit Ordnung 1 garantiert werden. Damit gilt folgender

Satz 4.8 *Falls* $f \in C^1(S)$ *sowie*

$$\sum_{i=0}^{k} \alpha_i \;=\; 0 \qquad \textit{und} \qquad \sum_{i=0}^{k} \alpha_i\, i \;-\; \sum_{i=0}^{k} \beta_i \;=\; 0 \tag{68}$$

gilt, so ist das LMSV konsistent mit Ordnung 1. □

Bemerkung 4.9
Mit Hilfe der erzeugenden Polynome lassen sich die *Konsistenzbedingungen* (68) in folgender Kurzform notieren:

$$\varrho(1) \;=\; 0 \qquad \text{und} \qquad \varrho'(1) \;=\; \sigma(1)\,. \tag{69}$$

Beispiele 4.10

1. Für das *implizite Heun-Verfahren* (54) liefern die erzeugenden Polynome

   $$\varrho(\mu) = \mu - 1 \qquad \text{und} \qquad \sigma(\mu) = \frac{1}{2}\mu + \frac{1}{2}$$

   offenbar mit $\varrho(1) \;=\; 0,\ \varrho'(1) \;=\; \sigma(1) \;=\; 1$ die Konsistenz.

2. Die *explizite Mittelpunktregel* (56) mit den Polynomen

   $$\varrho(\mu) = \mu^2 - 1 \qquad \text{und} \qquad \sigma(\mu) = 2\mu$$

   ist konsistent wegen $\varrho(1) \;=\; 0,\ \varrho'(1) \;=\; \sigma(1) \;=\; 2.$

3. Das *4-Schritt-AB-Verfahren* (60) mit den erzeugenden Polynomen

   $$\varrho(\mu) = \mu^4 - \mu^3 \quad \text{und} \quad \sigma(\mu) = \frac{55}{24}\mu^3 - \frac{59}{24}\mu^2 + \frac{37}{24}\mu - \frac{9}{24}.$$

   ist wegen $\varrho(1) \;=\; 0,\ \varrho'(1) \;=\; \sigma(1) \;=\; 1$ konsistent. ◀

Kann für die Lösung $x(t)$ sogar $(q+2)-$fache stetige Differenzierbarkeit vorausgesetzt werden, so läßt sich die Konsistenzordnung q mittels der Entwicklung (66) des lokalen Fehlers

$$\tau_j \;=\; \frac{1}{h}c_0 x(t_{j-k}) + c_1 \dot{x}(t_{j-k}) + \ldots + c_q x^{(q)}(t_{j-k}) h^{q-1} + c_{q+1} x^{(q+1)}(t_{j-k}) h^q + O(h^{q+1})$$

gewinnen, falls die Konstanten c_ν für $\nu = 0, 1, 2, \ldots, q$ verschwinden. Mit den Darstellungen (67) gewinnt man unmittelbar den folgenden

Satz 4.11 *Falls $f \in C^{q+1}(S)$ ist und die Konsistenzgleichungen*

$$c_0 = c_1 = \ldots = c_q = 0\,, \quad c_{q+1} \neq 0 \tag{70}$$

mit den Formeln (67) erfüllt sind, so ist das LMSV konsistent mit Ordnung q. Der lokale Diskretisierungsfehler lautet

$$\tau_j \;=\; c_{q+1} x^{(q+1)}(t_{j-k}) h^q + O(h^{q+1})\,;$$

die Fehlerkonstante des LMSV hat den Wert

$$c_{q+1} \;=\; \frac{1}{(q+1)!}\sum_{i=0}^{k} \alpha_i\, i^{q+1} \;-\; \frac{1}{q!}\sum_{i=0}^{k} \beta_i\, i^q\,. \qquad \square \tag{71}$$

Anhand der Konsistenzgleichungen läßt sich nun die Frage beantworten, welche maximale Konsistenzordnung ein lineares k-Schritt-Verfahren besitzen kann. Das lineare Gleichungssystem $c_0 = c_1 = \ldots = c_q = 0$ mit $q+1$ Gleichungen hat wegen $\alpha_k = 1$ für implizite LMSV die $2k+1$ Unbekannten $\alpha_0, \alpha_1, \ldots, \alpha_{k-1}, \beta_0, \beta_1, \ldots, \beta_k$, womit $q \leq 2k$ folgt. Explizite LMSV besitzen wegen $\beta_k = 0$ die Schranke $q \leq 2k-1$.

Beispiele 4.12

1. Das implizite Heun-Verfahren (54) genügt den Konsistenzgleichungen $c_0 = c_1 = c_2 = 0$, wogegen $c_3 = -\frac{1}{12}$ ist. Damit ist sein lokaler Diskretisierungsfehler

$$\tau_j \;=\; -\frac{1}{12}x^{(3)}(t_{j-1})h^2 + O(h^3)$$

   und es besitzt die maximal erreichbare Konsistenzordnung 2.

2. Die explizite Mittelpunktregel (56) erfüllt ebenfalls die Konsistenzgleichungen $c_0 = c_1 = c_2 = 0$, und ist konsistent mit dem lokalen Diskretisierungsfehler

$$\tau_j \;=\; \frac{1}{3}x^{(3)}(t_{j-2})h^2 + O(h^3)\,.$$

   Damit hat es allerdings nicht die maximal erreichbare Konsistenzordnung 3. ◀

Wir wollen deshalb nun ein explizites 2-Schritt-Verfahren mit der maximal erreichbaren Konsistenzordnung 3 konstruieren. Dazu müssen die 4 Konsistenzgleichungen

$$\begin{array}{rcrrrrrrrcl}
c_0 & = & \alpha_0 + & \alpha_1 & + \alpha_2 & & & & & = & 0 \\
c_1 & = & & \alpha_1 & + 2\alpha_2 & - \beta_0 & - \beta_1 & - \beta_2 & & = & 0 \\
2!\, c_2 & = & & \alpha_1 & + 4\alpha_2 & & - 2\beta_1 & - 4\beta_2 & & = & 0 \\
3!\, c_3 & = & & \alpha_1 & + 8\alpha_2 & & - 3\beta_1 & - 12\beta_2 & & = & 0
\end{array}$$

erfüllt sein. Mit den Vorgaben $\alpha_2 = 1,\ \beta_2 = 0$ liefert das Gleichungssystem die eindeutige Lösung

$$\alpha_0 = -5,\ \alpha_1 = 4,\ \alpha_2 = 1,\ \beta_0 = 2,\ \beta_1 = 4,\ \beta_2 = 0,$$

mit der man das explizite 2-Schritt-Verfahren

$$u_j \;=\; -\,4\,u_{j-1} + 5\,u_{j-2} + h\,[4f_{j-1} + 2f_{j-2}]\;, \quad j = 2(1)N \tag{72}$$

erhält. Gemäß Konstruktion besitzt es die maximal erreichbare Konsistenzordnung 3 für explizite 2-Schritt-Verfahren.

Beispiel 4.13 Wir wenden das Verfahren (72) auf das Anfangswertproblem

$$\dot{x} \;=\; -x,\quad x(0) = 1,\quad I = [0,1] \qquad (\text{Lösung } x(t) \;=\; e^{-t})$$

mit konstanter Schrittweite $h = 1/N$ an. Die 2 Startwerte seien exakt vorgegeben, d.h. $u_0 = x(t_0) = 1,\ u_1 = x(t_1) = e^{-h}$. Man stellt fest, daß das Verfahren nur bei sehr kleiner Zahl N von Teilintervallen den Endpunkt $t_N = 1$ erreicht. In allen anderen Fällen tritt - auch bei Rechnung in unterschiedlichen Gleitpunktsystemen - stets ein Zahlenüberlauf ein. In der folgenden Tabelle wird zu gegebenem N stets derjenige t_j-Wert angegeben, bei dem erstmalig die Schranke

- $|u_j| \geq 10^{10}$ bei Rechengenauigkeiten `single` (4 Bytes) und `real` (6 Bytes)
- $|u_j| \geq 10^{100}$ bei Rechengenauigkeit `extended` (10 Bytes)

überschritten wird.

N	t_j mit `single`	t_j mit `real`	t_j mit `extended`
25	1.000E-0	1.000E-0	1.0000E-0
50	5.200E-1	5.400E-1	5.4000E-1
100	2.800E-1	2.800E-1	2.8000E-1
200	1.300E-1	1.500E-1	1.5000E-1
400	6.500E-2	8.000E-2	8.0000E-2
800	3.250E-2	4.125E-2	4.2500E-2
1600	1.625E-2	2.000E-2	2.1875E-2

Das konstruierte Verfahren (72) zeigt auch bei höchster Rechengenauigkeit ein numerisch divergentes Verhalten, das sich mit abnehmender Schrittweite h noch verstärkt. Eine Konvergenz dieses konsistenten Verfahrens mit $\lim_{h\to 0} e_N = 0$ kann trotz exakter Startwerte u_0 und u_1 nicht erkannt werden. ◀

Im Gegensatz zu den expliziten Runge-Kutta-Verfahren folgt offenbar bei linearen Mehrschrittverfahren aus der Konsistenz nicht automatisch auch die Konvergenz des Verfahrens.

4.3 Konvergenz linearer Mehrschrittverfahren

Um eine Näherungslösung $u_j,\ j = 0(1)N,$ auf dem äquidistanten Gitter $t_j,\ j = 0(1)N,$ zu erhalten, sind bei einem linearen k-Schritt-Verfahren die k Startwerte $u_j := \eta_j,\ j = 0(1)k-1,$ vorzugeben. Wie bei Einschrittverfahren wird mit der exakten Lösung $x(t)$ der *globale Diskretisierungsfehler* durch

$$e_j := u_j - x(t_j)\,, \quad j = 0(1)N \tag{73}$$

erklärt. Die Konvergenz des LMSV setzt nun allerdings eine entsprechende Eigenschaft der k Startwerte voraus.

Definition 4.14 (Konvergenz linearer Mehrschrittverfahren)

(i) Ein lineares k-Schritt-Verfahren (62) heißt konvergent, falls aus der Konvergenz der Startwerte

$$\lim_{h\to 0} e_j = \lim_{h\to 0} [\eta_j - x(t_j)] = 0\,, \quad j = 0(1)k-1\,,$$

auch die Konvergenz der berechneten Werte u_j mit

$$\lim_{h\to 0} e_j = \lim_{h\to 0} [u_j - x(t_j)] = 0\,, \quad j = k(1)N\,,$$

folgt.

(ii) Das Verfahren konvergiert mit der Ordnung $p \geq 1$, falls aus $e_j = O(h^p)$, $j = 0(1)k-1$ auch $e_j = O(h^p)$, $j = k(1)N$ folgt.

Wie erhält man nun eine *notwendige* Konvergenzbedingung? Will man nachweisen, daß ein konsistentes LMSV nicht konvergiert, so muß man dies nur für ein Anfangswertproblem mit hinreichend glatter rechter Seite tun, das die Voraussetzung 1.1 erfüllt, z.B. für die triviale skalare Aufgabe

$$\dot{x} \;=\; 0\,, \quad x(a) \;=\; 0\,, \quad a \leq t \leq b \tag{74}$$

mit der Lösung $x(t) = 0,\ t \in I$. Mit der Schrittweite $h = (b-a)/N$ lautet das LMSV dann

$$\alpha_k u_j + \alpha_{k-1} u_{j-1} + \ldots + \alpha_0 u_{j-k} \;=\; 0\,,\ j = k(1)N\,,\ \alpha_k = 1\,. \tag{75}$$

Diese lineare homogene Differenzengleichung kann mit dem Ansatz $u_j = \mu^j,\ \mu \in \mathbb{C}$, gelöst werden. Einsetzen in (75) und Division durch μ^{j-k} liefert die charakteristische Gleichung

$$\varrho(\mu) \;=\; \sum_{i=0}^{k} \alpha_i \mu^i \;=\; \mu^k + \alpha_{k-1}\mu^{k-1} + \ldots + \alpha_1 \mu + \alpha_0 \;=\; 0 \tag{76}$$

mit dem erzeugenden Polynom $\varrho(\mu)$ des LMSV. Betrachten wir den generischen Fall mit k paarweise verschiedenen Lösungen $\mu_1, \mu_2, \ldots, \mu_k$, so lautet die allgemeine Lösung der Differenzengleichung

$$u_j \;=\; C_1\ \mu_1^j \;+\; C_2\ \mu_2^j \;+\; \cdots \;+\; C_k\ \mu_k^j$$

mit Konstanten C_i. Notwendig für die Konvergenz des LMSV ist wegen $\lim_{h\to 0} |e_j| = \lim_{h\to 0} |u_j| = 0$ dann die Bedingung $|\mu_i| \leq 1,\ i = 1(1)k$ an die Wurzeln der charakteristischen Gleichung. Im allgemeinen Fall gilt das *Dahlquist'sche Wurzelkriterium (WK)*:

Definition 4.15 (Wurzelkriterium, Nullstabilität)
Seien $\mu_1, \mu_2, \ldots, \mu_k$ die Wurzeln des erzeugenden Polynoms $\varrho(\mu)$. Das LMSV genügt dem Wurzelkriterium, falls gilt:

(i) $|\mu_i| \leq 1\,, \quad i = 1(1)k$ *und*

(ii) *Ist $|\mu_i| = 1$, so ist μ_i eine einfache Wurzel.*

Ein LMSV mit dieser Eigenschaft heißt nullstabil (zero-stable).

Bemerkung 4.16 Die Nullstabilität eines LMSV ist notwendig für dessen Konvergenz. Den vollständigen Beweis, auch für den Fall mehrfacher Wurzeln, findet man in der genannten Literatur.

Beispiele 4.17

1. Das *implizite Heun-Verfahren* (54) mit $\varrho(\mu) = \mu - 1$ erfüllt wegen $\mu_1 = 1$ trivialerweise das WK.

2. Die *explizite Mittelpunktregel* (56) mit dem erzeugenden Polynom $\varrho(\mu) = \mu^2 - 1$ besitzt die einfachen Wurzeln $\mu_1 = 1\,,\ \mu_2 = -1$ und ist damit nullstabil.

3. Das *4-Schritt-AB-Verfahren* (60) mit $\varrho(\mu) = \mu^4 - \mu^3$ ist ebenfalls nullstabil mit $\mu_1 = 1\,,\ \mu_2 = \mu_3 = \mu_4 = 0\,.$

4. Das konstruierte Verfahren (72)

$$u_j \ = \ -\,4\ u_{j-1} + 5\ u_{j-2} + h\,[4f_{j-1} + 2f_{j-2}]$$

   hat das erzeugende Polynom $\varrho(\mu) = \mu^2 + 4\mu - 5\,.$ Dessen Wurzeln $\mu_1 = 1\,,\ \mu_2 = -5$ genügen nicht dem WK, womit das Verfahren nicht nullstabil ist. Die am Beispiel beobachtete Divergenz ist also eine Verfahrenseigenschaft – (72) konvergiert nicht!

Betrachtet man konsistente LMSV auf hinreichend glatten Anfangswertproblemen, so stellt die Nullstabilität auch eine hinreichende Bedingung für die Konvergenz dar. Den Beweis findet man in der Literatur, z.B. bei Strehmel, K.; Weiner, R., S.130-135.

Satz 4.18 *Voraussetzung 1.1 sei erfüllt sowie folgende weitere Voraussetzungen:*

(i) $f \in C^1(S)$ *mit* $S = \{(t,x) \mid a \le t \le b,\ ||x|| \le M\}$ *(bzw.* $f \in C^{p+1}(S)$ *mit* $p \ge 1$*)*

(ii) Das LMSV (62) ist konsistent (bzw. $\tau_j = O(h^p),\ j = k(1)N$*).*

(iii) Das LMSV (62) ist nullstabil.

Dann ist das LMSV konvergent (bzw. $e_j = O(h^p),\ j = k(1)N$*).* □

Da das konstruierte Verfahren (72) nicht konvergiert, sind die angegebenen Schranken für die maximale Konsistenzordnung eines linearen k-Schritt-Verfahrens nicht relevant für die Konvergenz. G.Dahlquist gab folgende Schranke für die erreichbare Konvergenzordnung an:

Satz 4.19 (Erste Dahlquist-Bariere) *Ein konvergentes lineares k-Schritt-Verfahren hat höchstens die Konvergenzordnung*

$$p \ = \ \begin{cases} k & , \ \textit{falls es explizit} \\ k+1 & , \ \textit{falls es implizit und } k \textit{ ungerade} \\ k+2 & , \ \textit{falls es implizit und } k \textit{ gerade} \end{cases} \tag{77}$$

ist.

Beweis: Vgl. Hairer, E.; Norsett, S.P.; Wanner, G., S.332ff. □

4.4 Absolute Stabilität linearer Mehrschrittverfahren

Wie bei Einschrittverfahren das Stabilitätsverhalten bei fester Schrittweite h unterssucht wurde, ist dies auch für LMSV möglich und erforderlich. Um diese Verfahren auf steife DGL-Systeme anwenden zu können, ist A-Stabilität wünschenswert. Betrachten wir deshalb die in Abschnitt 3 eingeführte skalare Test-DGL

$$\dot{x} = \lambda x, \quad x(a) = x_0, \quad \lambda \in \mathbb{C}^- \tag{78}$$

mit der exakten Lösung $x(t) = x_0\, e^{\lambda(t-a)}$. Anwendung des LMSV (62) auf dieses Anfangswertproblem ergibt die Darstellung

$$\sum_{i=0}^{k} \alpha_i u_{j-k+i} \;-\; H \sum_{i=0}^{k} \beta_i u_{j-k+i} \;=\; 0\,, \quad j = k(1)N\,, \tag{79}$$

mit der Abkürzung $H = \lambda h$. Bei vorgegenenem Satz von k Startwerten $u_0 = \eta_0,\ u_1 = \eta_1, \ldots,\ u_{k-1} = \eta_{k-1}$ erhält man auch hier eine lineare homogene Differenzengleichung k-ter Ordnung mit k Anfangswerten. Mit dem Potenzansatz $u_j = \mu^j,\ \mu \in \mathbb{C}$, liefert Einsetzen in (79) und Division durch μ^{j-k} die Bestimmungsgleichung

$$\sum_{i=0}^{k} \alpha_i \mu^i \;-\; H \sum_{i=0}^{k} \beta_i \mu^i \;=\; 0\,.$$

Nutzt man die erzeugenden Polynome $\varrho(\mu)$ und $\sigma(\mu)$ des LMSV, so erhält man folgende

Definition 4.20 (Charakteristisches Polynom)
Das Polynom k-ten Grades

$$P(\mu; H) \;:=\; \varrho(\mu) \;- H \cdot \sigma(\mu)$$

heißt charakteristisches Polynom zur Testgleichung (78). Die Gleichung

$$P(\mu; H) \;:=\; \varrho(\mu) \;- H \cdot \sigma(\mu) \;=\; 0$$

ist die charakteristische Gleichung.

Betrachten wir zu willkürlich gewähltem $H \in \mathbb{C}$ den generischen Fall mit k paarweise verschiedenen Lösungen $\mu_1(H), \mu_2(H), \ldots, \mu_k(H)$, so lautet die allgemeine Lösung der Differenzengleichung

$$u_j \;=\; C_1\ \mu_1(H)^j \;+\; C_2\ \mu_2(H)^j \;+\; \cdots \;+\; C_k\ \mu_k(H)^j$$

mit Konstanten C_i. Soll sich die Näherungslösung u_j asymptotisch wie die exakte Lösung $x(t)$ verhalten, d.h. $lim_{j\to\infty}\, |u_j| \;=\; 0$ gelten, so ist dafür die hinreichende Bedingung $|\mu_i(H)| < 1\,,\ i = 1(1)k$ an die Wurzeln der charakteristischen Gleichung zu erfüllen.

Definition 4.21 (Absolute Stabilität)

(i) Das LMSV 62 mit dem charakteristischen Polynom $P(\mu; H)$ heißt absolut stabil für $H \in \mathbb{C}$, falls für seine Nullstellen $|\mu_i(H)| < 1\,,\ i = 1(1)k$ gilt.

(ii) Die Menge

$$\mathbf{H} = \{\ H \in \mathbb{C} \mid\ \ |\mu_i(H)| \;< 1\ ,\ i = 1(1)k\ \} \tag{80}$$

heißt Stabilitätsbereich (Bereich absoluter Stabilität) des Mehrschrittverfahrens.

Bemerkung 4.22 Bei der Analyse der Einschrittverfahren wurde deren Stabilitätsfunktion $\mu(H)$ durch die Rekursion $u_j = \mu(H) u_{j-1}$ ermittelt. Einsetzen des Potenzansatzes $u_j = \mu^j,\ \mu \in \mathbb{C}$, liefert den Wert $\mu_1(H) \equiv \mu(H)$, so daß sich die für ESV gegebene Definition der absoluten Stabilität nun als Spezialfall der Definition 4.21 herausstellt.

Beispiele 4.23

1. Das *implizite Heun-Verfahren* (54) hat das charakteristische Polynom

$$P(\mu; H) \;=\; \varrho(\mu) - H\sigma(\mu) \;=\; (1 - \frac{1}{2}H)\mu - (1 + \frac{1}{2}H)\,.$$

   Seine Nullstelle ist damit

$$\mu_1(H) = \frac{1 + \frac{1}{2}\,H}{1 - \frac{1}{2}\,H}$$

   und stimmt mit der Stabilitätsfunktion $\mu(H)$ der impliziten Gauß-Legendre-Formel aus Beispiel 3.14 überein. Damit ist das Verfahren A-stabil mit $\mathbf{H} = \mathbb{C}^-$.

2. Die *explizite Mittelpunktregel* (56) mit dem charakteristischen Polynom

$$P(\mu; H) \;=\; \varrho(\mu) - H\sigma(\mu) \;=\; \mu^2 - 2H\mu - 1$$

   besitzt die 2 Wurzeln $\mu_{1,2}(H) \;=\; H \;\pm\; \sqrt{H^2 + 1}$. Wegen der Eigenschaft

$$|\mu_1(H)| \cdot |\mu_2(H)| \;=\; 1 \quad \forall\, H \in \mathbb{C}$$

   können niemals beide Wurzelbeträge zugleich kleiner als 1 sein; also ist der Stabilitätsbereich **H** dieses Verfahrens leer.

Die Bestimmung des Stabilitätsbereiches **H** ist im Gegensatz zu Einschrittverfahren höherer Ordnung eine relativ leichte Aufgabe. Numerisch ermittelt man den Rand $\partial\mathbf{H}$ des Stabilitätsbereichs, für dessen Punkte H nun

$$\mu(H) = e^{i\varphi}\,, \;\; \varphi \in [0, 2\pi)\,, \quad \text{und} \quad P(\mu; H) \;=\; 0$$

gilt. Stellt man die charakteristische Gleichung nach H um und setzt die Darstellung für μ ein, so ergibt sich nun eine explizite Darstellung $H \;:\; [0, 2\pi) \to \mathbb{C}$ mit

$$H \;=\; H(\varphi) \;=\; \frac{\varrho(e^{i\varphi})}{\sigma(e^{i\varphi})}\,. \tag{81}$$

Die dadurch beschriebene *Wurzelortskurve (engl.: root locus curve)* $\mathbb{K}$ enthält den Rand $\partial\mathbf{H}$ des Stabilitätsbereichs als Teilmenge. Da weitere Wurzeln mit $|\mu_k(H)| > 1$ auf $\mathbb{K}$ liegen können, muß bei Überschneidungen der Kurve der Bereich **H** ausgetestet werden.

Beispiele 4.24

1. Das *klassische Adams-Bashforth-Verfahren AB4* der Ordnung 4

$$u_j \;=\; u_{j-1} + \frac{h}{24}\,[55 f_{j-1} - 59 f_{j-2} + 37 f_{j-3} - 9 f_{j-4}] \tag{82}$$

   besitzt den kleinen Stabilitätsbereich **H** in Abbildung 15. Hier ist offenbar $\partial\mathbf{H}$ eine echte Teilmenge von $\mathbb{K}$.

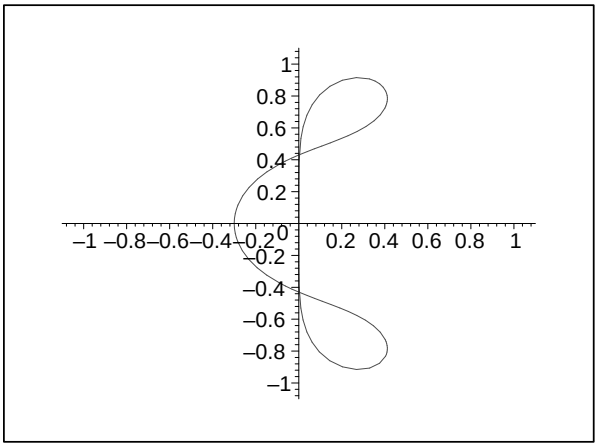

Abbildung 15: Stabilitätsbereich des Adams-Bashforth-Verfahrens AB4

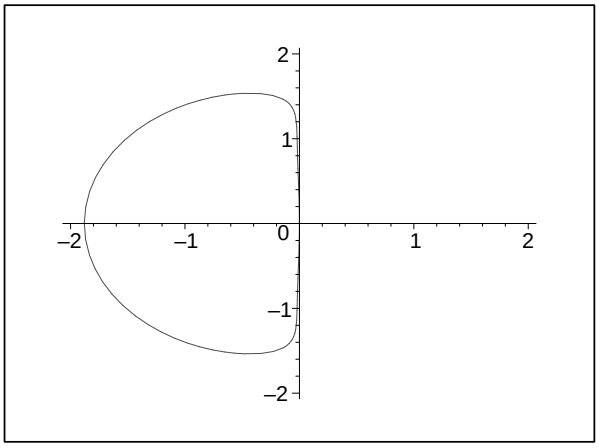

Abbildung 16: Stabilitätsbereich des impliziten Adams-Moulton-Verfahrens AM4

2. Das implizite *Adams-Moulton-Verfahren AM4* der Ordnung 5

$$u_j \;=\; u_{j-1} + \frac{h}{720}\left[251 f_j + 646 f_{j-1} - 264 f_{j-2} + 106 f_{j-3} - 19 f_{j-4}\right] \tag{83}$$

hat ebenfalls einen beschränkten Stabilitätsbereich $\mathbf{H}$, der in Abbildung 16 dargestellt ist. ◀

Bessere Stabilitätseigenschaften besitzen die sogenannten *BDF-Verfahren* (rückwärtige Differentiationsformeln, backward differentiation formulae), die auf C.F.Curtiss & J.O.Hirschfelder zurückgehen und durch Arbeiten von C.W. Gear bekannt wurden (daher auch oft als *Gear-Verfahren* bezeichnet). Diese impliziten Verfahren interpolieren die Lösung durch das Interpolationspolynom $N_k(t)$ an den $k+1$ Knoten

$$(t_j, u_j),\ (t_{j-1}, u_{j-1}),\ (t_{j-2}, u_{j-2}),\ \ldots,\ (t_{j-k}, u_{j-k})\,.$$

Mit der normalisierten Variablen $s \;=\; \frac{1}{h}(t - t_j)$ kann dieses Polynom in der Form

$$N_k(t) \;=\; \sum_{i=0}^{k} (-1)^i \binom{-s}{i} \nabla^i u_j$$

dargestellt werden. Die sogenannten Rückwärtsdifferenzen i-ter Ordnung werden rekursiv durch

$$\nabla^i u_j \;:=\; \nabla^{i-1} u_j - \nabla^{i-1} u_{j-1}\,,\quad i = 1(1)k \quad \text{mit} \quad \nabla^0 u_j \;:=\; u_j$$

definiert (vgl. Adams-Bashforth-Verfahren). Direktes Einsetzen von $N_k(t)$ in die DGL und Auswertung an der Stelle $t = t_j$ liefert die Darstellung

$$\sum_{i=0}^{k} c_i\, \nabla^i u_j \;=\; h f(t_j, u_j)$$

mit den Konstanten $c_0 = 0$ und

$$c_i \;=\; \frac{d}{ds}\left\{(-1)^i \binom{-s}{i}\right\}_{s=0} \;=\; \frac{1}{i}\,,\quad i = 1(1)k\,.$$

Die Klasse der BDF-Verfahren hat damit die allgemeine Form

$$\sum_{i=0}^{k} \frac{1}{i} \nabla^i u_j \;=\; h f(t_j, u_j)\,, \quad j = k(1)N\,. \tag{84}$$

Durch Ausrechnen der Koeffizienten α_i und β_i kann man hieraus die Standardform linearer k-Schritt-Verfahren

$$u_j + \alpha_{k-1} u_{j-1} + \ldots + \alpha_0 u_{j-k} \;=\; h\ \beta_k f_j\,,\ \ j = k(1)N\,, \tag{85}$$

ermitteln. Für Anfangswertprobleme mit $f \in C^{k+1}(S)$ kann man die Konsistenz des k-Schritt-BDF-Verfahrens für beliebiges $k \geq 1$ anhand der Herleitung leicht nachweisen. Nullstabilität und damit Konvergenz liegt allerdings nur für die Verfahren mit $k \leq 6$ vor.

Beispiele 4.25

1. Das *1-Schritt-Verfahren BDF1*

$$u_j \;=\; u_{j-1} + h f_j \tag{86}$$

   ist identisch mit dem impliziten Euler-Verfahren und damit A-stabil.

2. Das *2-Schritt-Verfahren BDF2*

$$u_j - \frac{4}{3} u_{j-1} + \frac{1}{3} u_{j-2} \;=\; \frac{2}{3} h f_j \tag{87}$$

   mit dem charakteristischen Polynom

$$P(\mu; H) \;=\; (1 - \frac{2}{3} H)\mu^2 \;-\; \frac{4}{3}\mu \;+\; \frac{1}{3}$$

   besitzt den unbeschränkten Stabilitätsbereich **H** in Abbildung 17 und ist ebenfalls A-stabil.

3. Der Stabilitätsbereich des *4-Schritt-Verfahrens BDF4*

$$u_j - \frac{48}{25} u_{j-1} + \frac{36}{25} u_{j-2} - \frac{16}{25} u_{j-3} + \frac{3}{25} u_{j-4} \;=\; \frac{12}{25} h f_j \tag{88}$$

   ist in Abbildung 18 dargestellt. Dieses Verfahren ist offnbar nicht A-stabil. ◀

Trotz der fehlenden A-Stabilität sind BDF-Verfahren auch für $k \geq 3$ zur Integration steifer DGL-Systeme geeignet, wenn die Eigenwerte der Matrix A aus Definition 3.8 nicht nahe der imaginären Achse liegen. Für derartige Verfahren existieren zahlreiche Abschwächungen des Begriffes der A-Stabilität, von denen hier nur die wesentlichsten genannt werden sollen:

Definition 4.26 (Stabilitätsbegriffe) *Ein LMSV heißt*

(i) *A-stabil, falls* $\mathbb{C}^- \subseteq \mathbf{H}$ *gilt, also* **H** *die gesamte linke Halbebene* $\mathbb{C}^-$ *umfaßt;*

(ii) *A(α)-stabil, falls für ein* α *mit* $0 < \alpha \leq \pi/2$ *gilt:*

$$\mathbf{H} \;\supset\; \{\ H \in \mathbb{C} \mid |\pi - \arg H| < \alpha,\ H \neq 0\ \}\,;$$

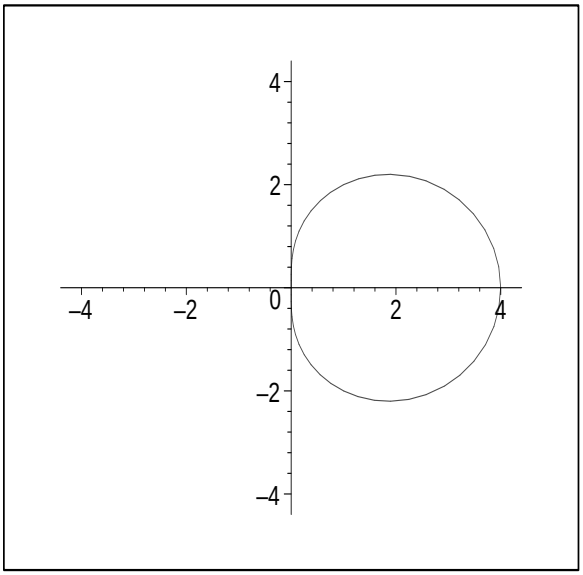

Abbildung 17: Stabilitätsbereich des BDF2-Verfahrens

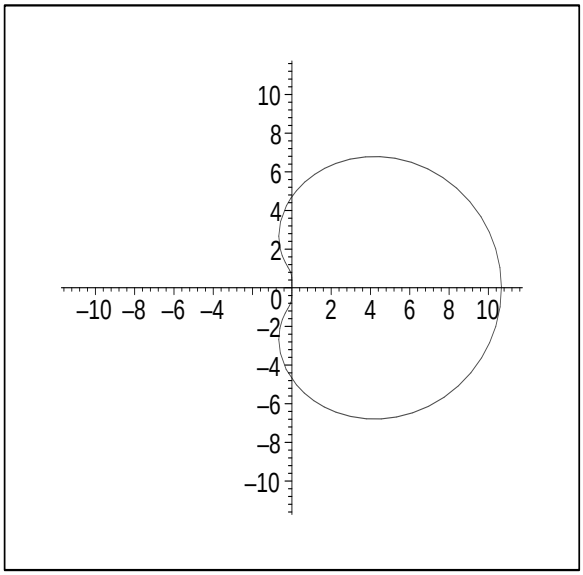

Abbildung 18: Stabilitätsbereich des BDF4-Verfahrens

(iii) A(0)-stabil, falls ein $\alpha_0 > 0$ *existiert, so daß es A(*α_0*)-stabil ist;*

(iv) steif-stabil, falls es A(0)-stabil ist und zudem gilt:

$$\mathbf{H} \supset \{ H \in \mathbb{C} \mid Re(H) \leq \gamma < 0,\ \gamma \text{ const.} \}.$$

Für die BDF-Verfahren lassen sich die in folgender Tabelle angegebenen Stabilitätseigenschaften numerisch nachweisen:

BDF k	A-stabil	A(α)-stabil mit $\alpha[^o]$	steif-stabil mit γ	A(0)-stabil
1	ja	90	0	ja
2	ja	90	0	ja
3	nein	86	-0.1	ja
4	nein	73	-0.7	ja
5	nein	51	-2.3	ja
6	nein	18	-6.1	ja
7	nein	nein	nein	nein

Trotz der mit wachsender Ordnung k abnehmenden Stabilität bilden BDF-Verfahren eine wesentliche Klasse von LMSV zur Behandlung steifer DGL-Systeme. Mit einer eingebauten Ordnungssteuerung kann eine automatische Auswahl der jeweils optimalen BDFk-Formel erfolgen. Zugleich kann das Verfahren auf diese Weise „selbststartend“ arbeiten: Die Integration beginnt mit dem Einschrittverfahren BDF1 und extrem kleiner Schrittweite. Nach

wenigen Schritten schaltet der Algorithmus auf BDF2 um und paßt zugleich die Arbeitsschrittweite an usw. Schließlich werden Schrittweite h und interne Ordnung k nach einem geeigneten Algorithmus in der Laufphase den Gegebeneheiten automatisch angepaßt. Eine komplette Übersicht findet man dazu in der angegebenen Literatur.

Literatur

[1] Ascher, U.M.; Petzold, L.R.: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM Philadelphia 1998.

[2] Deuflhardt, P.; Bornemann, F.: *Numerische Mathematik II – Integration gewöhnlicher Differentialgleichungen.* W. de Gruyter Berlin 1994.

[3] Hairer, E.; Norsett, S.P.; Wanner, G.: *Solving Ordinary Differential Equations.* Band 1. Springer–Verlag Berlin 1987.

[4] Hairer, E.; Wanner, G.: *Solving Ordinary Differential Equations.* Band 2. Springer–Verlag Berlin 1991.

[5] Hanke-Bourgeois, M.: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens.* B.G.Teubner, Stuttgart 2002.

[6] Isaacson, E.; Keller, H. B.: *Analyse numerischer Verfahren.* Verlag Harry Deutsch, Frankfurt 1972.

[7] Maess, G.: *Vorlesungen über numerische Mathematik. Band 1 und 2.* Akademie – Verlag, Berlin 1984.

[8] Strehmel, K.; Weiner, R.: *Numerik gewöhnlicher Differentialgleichungen.* B.G.Teubner Stuttgart 1995.

[9] Törnig, W.; Spellucci, P.: *Numerische Mathematik für Ingenieure und Physiker. Band 1 und 2.* 2. Auflage, Springer-Verlag, Berlin 1988.