9. MEHRSCHRITTVERFAHREN

Wir betrachten wieder das Anfangswertproblem (8.1).

9.1 Adams-Verfahren

Ähnlich wie Runge-Kutta-Verfahren kann man auch *Adams-Verfahren* mit Hilfe numerischer Integration motivieren. Wir beginnen mit *expliziten Adams-Verfahren* (Adams, 1855).

Angenommen, wir kennen bereits $y_0, \ldots, y_n$ als Näherungen für $y(t_0), \ldots, y(t_n)$ und möchten y_{n+1} berechnen. Dann ersetzen wir in der Formel für die exakte Lösung zur Zeit t_{n+1}:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt \tag{9.1}$$

$f(t, y(t))$ durch das Interpolationspolynom p durch die Punkte

$$(t_{n-k+1}, f_{n-k+1}), \ldots, (t_n, f_n), \qquad f_j := f(t_j, y_j).$$

Die Newton'sche Interpolationsformel liefert

$$p(t) = p(t_n + \tau h) = \sum_{j=0}^{k-1} (-1)^j \binom{-\tau}{j} \nabla^j f_n.$$

Hierbei sind Rückwärtsdifferenzen $\nabla^j f_n$ rekursiv durch

$$\nabla^0 f_n = f_n, \qquad \nabla^{j+1} f_n = \nabla^j f_n - \nabla^j f_{n-1}$$

definiert. Die numerische Approximation von (9.1) lautet

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} p(t)dt = y_n + h \int_0^1 p(t_n + \tau h)d\tau,$$

bzw. nach Einsetzen der Newton'schen Formel

$$y_{n+1} = y_n + h \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n$$

mit

$$\gamma_j = (-1)^j \int_0^1 \binom{-\tau}{j} d\tau.$$

Man rechnet leicht die folgende Tabelle nach:

j	0	1	2	3	4	5	6
γ_j	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$

Die expliziten Adams-Verfahren für $k \leq 4$ lauten

$$\begin{aligned}
k=1: &\quad y_{n+1} = y_n + hf_n \\
k=2: &\quad y_{n+1} = y_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right) \\
k=3: &\quad y_{n+1} = y_n + h\left(\frac{23}{12}f_n - \frac{16}{12}f_{n-1} + \frac{5}{12}f_{n-2}\right) \\
k=4: &\quad y_{n+1} = y_n + h\left(\frac{55}{24}f_n - \frac{59}{24}f_{n-1} + \frac{37}{24}f_{n-2} - \frac{9}{24}f_{n-3}\right)
\end{aligned}$$

Für $k = 1$ erhalten wir das bekannte explizite Euler-Verfahren.

Bemerkung. Der lokale Fehler ist im Allgemeinen groß, da das Polynom $p(t)$ außerhalb des Interpolationsintervalls $[t_{n-k+1}, t_n]$ benutzt wird.

Zur Konstruktion impliziter Adams-Verfahren bestimmt man ein Interpolationspolynom p durch die Punkte

$$(t_{n-k+1}, f_{n-k+1}), \ldots, (t_n, f_n), (t_{n+1}, f_{n+1}),$$

man nimmt also den noch unbekannten Wert f_{n+1} hinzu. Die Newton'sche Interpolationsformel lautet in diesem Fall

$$p(t) = p(t_n + \tau h) = \sum_{j=0}^{k} (-1)^j \binom{-\tau+1}{j} \nabla^j f_{n+1}.$$

Analog zum expliziten Adams-Verfahren ergibt sich

$$y_{n+1} = y_n + h \sum_{j=0}^{k} \gamma_j^* \nabla^j f_{n+1}$$

mit

$$\gamma_j^* = (-1)^j \int_0^1 \binom{-\tau+1}{j} d\tau.$$

Nachrechnen ergibt

j	0	1	2	3	4	5	6
γ_j^*	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$

Die impliziten Adams-Verfahren für $k = 0, 1, 2, 3$ lauten

$$\begin{aligned}
k=0: &\quad y_{n+1} = y_n + hf_{n+1} \\
k=1: &\quad y_{n+1} = y_n + h\left(\frac{1}{2}f_{n+1} + \frac{1}{2}f_n\right) \\
k=2: &\quad y_{n+1} = y_n + h\left(\frac{5}{12}f_{n+1} + \frac{8}{12}f_n - \frac{1}{12}f_{n-1}\right) \\
k=3: &\quad y_{n+1} = y_n + h\left(\frac{9}{24}f_{n+1} + \frac{19}{24}f_n - \frac{5}{24}f_{n-1} + \frac{1}{24}f_{n-2}\right)
\end{aligned}$$

Für $k = 0$ erhalten wir das implizite Euler-Verfahren, für $k = 1$ die Trapezregel.

Allgemein haben Adams-Verfahren die Form

$$y_{n+1} = y_n + h \sum_{j=0}^{k} \beta_{k-j} f_{n+1-j}. \tag{9.2}$$

Bei expliziten Adams-Verfahren ist $\beta_k = 0$.

Nach dem Banach'schen Fixpunktsatz sind die Gleichungen des impliziten Adams-Verfahrens eindeutig nach y_{n+1} auflösbar, falls

$$hL|\sum_{j=0}^{k} \gamma_j^*| < 1,$$

wobei L die Lipschitz-Konstante von f ist. Die Lösung kann mit Hilfe von Fixpunktiteration erfolgen. Als Startpunkt bietet sich die durch ein explizites Adams-Verfahren erhaltene Näherung an. Dies führt auf *Predictor-Corrector-Verfahren*, bei denen folgende Schritte auftreten:

P: Berechne einen **P**redictor

$$\widehat{y}_{n+1} = y_n + h \sum_{j=0}^{k-1} \gamma_j \nabla^j f_n$$

E: Berechne (**E**valuate) den Funktionswert für diese Näherung:

$$\widehat{f}_{n+1} = f(t_{n+1}, \widehat{y}_{n+1}).$$

C: Wende die **C**orrector-Formel

$$y_{n+1} = y_n + h\beta_k \widehat{f}_{n+1} + h \sum_{j=1}^{k} \beta_{k-j} f_{n+1-j}$$

an, um y_{n+1} zu berechnen.

E: Berechne (**E**valuate) erneut den Funktionswert für diese Näherung:

$$f_{n+1} = f(t_{n+1}, y_{n+1})$$

Diese, mit $PECE$ bezeichnete Variante, ist die am häufigsten angewendete. Andere Möglichkeiten sind $P(EC)^2E$ (zwei Fixpunktiterationen pro Schritt) oder PEC (verwende $\widehat{f}_{n+1}$ statt f_{n+1} im nächsten Schritt).

9.2 BDF-Verfahren

BDF-Verfahren (**b**ackward **d**ifferentiation **f**ormula) beruhen auf numerischer Differentiation. Sei dazu p das Interpolationspolynom durch die Punkte

$$(t_{n-k+1}, y_{n-k+1}), \ldots, (t_{n+1}, y_{n+1}).$$

Die Newton'sche Interpolationformel liefert

$$p(t) = p(t_n + \tau h) = \sum_{j=0}^{k} (-1)^j \binom{-\tau + 1}{j} \nabla^j y_{n+1}.$$

Die unbekannte Näherung y_{n+1} wird jetzt so bestimmt, dass das Polynom p die Differentialgleichung zur Zeit t_{n+1-r} erfüllt:

$$p'(t_{n+1-r}) = f(t_{n+1-r}, y_{n+1-r}).$$

Für $r = 1$ erhalten wir bekannte explizite Verfahren, nämlich das explizite Euler-Verfahren für $k = 1$ und die explizite Mittelpunktsregel für $k = 2$. Für $k \geq 3$ sind explizite BDF-Verfahren instabil.

Interessanter ist der Fall $r = 0$, bei dem sich die implizite Formel

$$\sum_{j=0}^{k} \delta_j^* \nabla^j y_{n+1} = h f_{n+1}$$

mit den Koeffizienten

$$\delta_j^* = (-1)^j \frac{d}{d\tau} \binom{-\tau + 1}{j} |_{\tau=1}$$

ergibt. Die δ_j^* können leicht explizit berechnet werden, denn wegen

$$(-1)^j \binom{-\tau + 1}{j} = \frac{1}{j!}(\tau - 1)\tau(\tau + 1) \cdots (\tau + j - 2)$$

gilt

$$\delta_0^* = 0, \qquad \delta_j^* = \frac{1}{j}, \qquad j \geq 1.$$

BDF-Verfahren sind also von der Form

$$\sum_{j=1}^{k} \frac{1}{j} \nabla^j y_{n+1} = h f_{n+1}.$$

Auch hier wollen wir die ersten BDF-Verfahren explizit angeben:

$$\begin{aligned}
k = 1: &\quad y_{n+1} - y_n = h f_{n+1} \\
k = 2: &\quad \frac{3}{2} y_{n+1} - 2 y_n + \frac{1}{2} y_{n-1} = h f_{n+1} \\
k = 3: &\quad \frac{11}{6} y_{n+1} - 3 y_n + \frac{3}{2} y_{n-1} - \frac{1}{3} y_{n-2} = h f_{n+1} \\
k = 4: &\quad \frac{25}{12} y_{n+1} - 4 y_n + 3 y_{n-1} - \frac{4}{3} y_{n-2} + \frac{1}{4} y_{n-3} = h f_{n+1} \\
k = 5: &\quad \frac{137}{60} y_{n+1} - 5 y_n + 5 y_{n-1} - \frac{10}{3} y_{n-2} + \frac{5}{4} y_{n-3} - \frac{1}{5} y_{n-4} = h f_{n+1}.
\end{aligned}$$

BDF(1) entspricht wieder dem impliziten Euler-Verfahren.

9.3 Ordnung von Mehrschrittverfahren

Ein allgemeines lineares Mehrschrittverfahren hat die Form

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \qquad f_j = f(t_j, y_j), \qquad t_j = t_0 + jh, \tag{9.3}$$

mit reellen Parametern α_j, β_j und der Schrittweite h. Wir werden im Folgenden voraussetzen, dass

$$\alpha_k \neq 0, \qquad |\alpha_0| + |\beta_0| > 0.$$

Die erste Annahme garantiert, dass sich (9.3) zumindest für hinreichend kleines h nach y_{n+k} auflösen lässt. Die zweite Annahme ist keine Einschränkung, da sie gegebenenfalls durch Reduktion von k erfüllt werden kann. Sind beide Annahmen erfüllt, so sprechen wir von einem linearen k-Schritt-Verfahren. Bei diesen unterscheiden wir explizite ($\beta_k = 0$) und implizite ($\beta_k \neq 0$) Mehrschrittverfahren.

Definition 9.1. *Der* **lokale Fehler** *eines Mehrschrittverfahrens ist definiert durch* $y(t_k) - y_k$, *wobei* $y(t)$ *die exakte Lösung von* (8.1) *ist und die Näherung* y_k *aus* (9.3) *mit exakten Startwerten* $y_i = y(t_i)$, $i = 0, \dots, k-1$ *berechnet wurde.*

Für $k = 1$ entspricht das der früheren Definition für Einschrittverfahren.

Um den lokalen Fehler zu analysieren benötigen wir den linearen Differenzenoperator $L : C^1(\mathbb{R}, \mathbb{R}^d) \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^d$ definiert durch

$$L(y, t, h) = \sum_{j=0}^{k} \big(\alpha_j y(t + jh) - h\beta_j y'(t + jh)\big). \tag{9.4}$$

Lemma 9.2. *Für* f *stetig differenzierbar ist*

$$y(t_k) - y_k = \left(\alpha_k I - h\beta_k \frac{\partial f}{\partial y}(t_k, \eta)\right)^{-1} L(y, t_0, h)$$

mit einer Zwischenstelle η, *die von Zeile zu Zeile verschieden sein kann.*

Beweis. Da wir von exakten Startwerten ausgehen, ist y_k implizit definiert durch

$$\sum_{j=0}^{k-1} \big(\alpha_j y(t_j) - h\beta_j f(t_j, y(t_j))\big) + \alpha_k y_k - h\beta_k f(t_k, y_k) = 0.$$

Damit gilt

$$L(y, t_0, h) = \alpha_k \big(y(t_k) - y_k\big) - h\beta_k \big(f(t_k, y(t_k)) - f(t_k, y_k)\big),$$

woraus die Behauptung aus dem Mittelwertsatz folgt. □

Das Lemma zeigt, dass $\alpha_k^{-1} L(y, t_0, h)$ im Wesentlichen dem lokalen Fehler entspricht. Damit können wir jetzt die Ordnung eines Mehrschrittverfahrens definieren:

Definition 9.3. *Ein Mehrschrittverfahren hat die* **Ordnung** p, *wenn für jedes Anfangswertproblem* (8.1) *mit* f *hinreichend glatt eine der folgenden äquivalenten Bedingungen erfüllt ist:*

(a) *der lokale Fehler ist* $O(h^{p+1})$

(b) $L(y, t_0, h) = O(h^{p+1})$.

Eine handliche Charakterisierung der Ordnung basiert auf Dahlquist (1956), der die fundamentale Rolle der **generierenden Polynome**

$$\varrho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j, \qquad \sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j \tag{9.5}$$

erkannte.

Satz 9.4. *Ein Mehrschrittverfahren (9.3) hat Ordnung p genau dann, wenn eine der folgenden äquivalenten Bedingungen erfüllt ist:*

(a) *Für $q = 1, \dots, p$ gilt*

$$\sum_{j=0}^{k} \alpha_j = 0 \qquad \textit{und} \qquad \sum_{j=0}^{k} \alpha_j j^q = q \sum_{j=0}^{k} \beta_j j^{q-1}$$

(b) $\varrho(e^h) - h\sigma(e^h) = O(h^{p+1})$, $h \to 0$

(c) $\dfrac{\varrho(\zeta)}{\log \zeta} - \sigma(\zeta) = O((\zeta - 1)^p)$, $\zeta \to 1$.

Beweis. In (9.4) entwickeln wir $y(t + jh)$ und $y'(t + jh)$ in eine Taylor-Reihe:

$$\begin{aligned} L(y, t, h) &= \sum_{j=0}^{k} \left(\alpha_j \sum_{q \geq 0} \frac{j^q}{q!} h^q y^{(q)}(t) - h\beta_j \sum_{r \geq 0} \frac{j^r}{r!} h^r y^{(r+1)}(t) \right) \\ &= y(t) \sum_{j=0}^{k} \alpha_j + \sum_{q \geq 1} \frac{h^q}{q!} y^{(q)}(t) \left(\sum_{j=0}^{k} \alpha_j j^q - q \sum_{j=0}^{k} \beta_j j^{q-1} \right). \end{aligned}$$

Aus Definition 9.3 folgt daraus (a). Es bleibt noch zu zeigen, dass die drei Bedingungen äquivalent sind. Für $y(t) = e^t$ gilt nach Definition

$$L(\exp, 0, h) = \varrho(e^h) - h\sigma(e^h).$$

Setzen wir in obiger Taylor-Entwicklung von L speziell die Exponentialfunktion und $t = 0$ ein, so ergibt sich ebenfalls

$$L(\exp, 0, h) = \sum_{j=0}^{k} \alpha_j + \sum_{q \geq 1} \frac{h^q}{q!} \left(\sum_{j=0}^{k} \alpha_j j^q - q \sum_{j=0}^{k} \beta_j j^{q-1} \right).$$

(a) und (b) sind damit äquivalent. Nach der Transformation $\zeta = e^h$ bzw. $h = \log \zeta$ können wir (b) als

$$\varrho(\zeta) - \sigma(\zeta) \log \zeta = O((\log \zeta)^{p+1}), \qquad \zeta \to 1$$

schreiben und dies ist wegen

$$\log \zeta = (\zeta - 1) + O((\zeta - 1)^2), \qquad \zeta \to 1$$

äquivalent zu (c). □

Damit ist ein Mehrschrittverfahren **konsistent**, d. h. es hat mindestens Ordnung eins, wenn

$$\varrho(1) = 0, \qquad \varrho'(1) = \sigma(1).$$

Beispiel. (Ordnung von Adams- und BDF-Verfahren)
Nach Konstruktion integriert ein explizites Adams-Verfahren eine Differentialgleichung exakt, wenn bei der Interpolation kein Fehler entsteht. Dies ist der Fall, wenn $f(t, y)$ ein Polynom vom Grad $\leq k - 1$ in t ist. Insbesondere werden alle Differentialgleichungen der Form

$$y' = qt^{q-1}, \qquad q = 1, \ldots, k$$

exakt integriert, liefern also lokalen Fehler Null:

$$0 = L(t^q, 0, h) = h^q \left(\sum_{j=0}^{k} \alpha_j j^q - q \sum_{j=0}^{k} \beta_j j^{q-1} \right), \qquad q = 1, \ldots, k.$$

Das ist gerade Bedingung (a) aus Satz 9.4 für $p = k$, explizite Adams-Verfahren haben also Ordnung k.

Analog zeigt man, dass implizite Adams-Verfahren Ordnung $k + 1$ und BDF-Verfahren Ordnung k haben (Übung). $\diamond$

Beispiel. (Explizites 2-Schrittverfahren maximaler Ordnung)
Die allgemeine Form expliziter 2-Schrittverfahren ist

$$\alpha_2 y_{n+2} + \alpha_1 y_{n+1} + \alpha_0 y_n = h\big(\beta_1 f_{n+1} + \beta_0 f_n\big).$$

Da $\alpha_2 \neq 0$ vorausgesetzt ist, können wir ohne Einschränkung $\alpha_2 = 1$ setzen. Satz 9.4 liefert die folgenden Bedingungen:

$$\begin{aligned} q = 0: &\qquad \alpha_0 + \alpha_1 + 1 = 0 \\ q = 1: &\qquad \alpha_1 + 2 = \beta_0 + \beta_1 \\ q = 2: &\qquad \alpha_1 + 4 = 2\beta_1 \\ q = 3: &\qquad \alpha_1 + 8 = 3\beta_1 \end{aligned}$$

Aus den letzten beiden Gleichungen folgt $\beta_1 = \alpha_1 = 4$. Aus der zweiten Gleichung schließen wir damit $\beta_0 = 2$ und aus der ersten $\alpha_0 = -5$. Das Verfahren hat damit die Form:

$$y_{n+2} + 4y_{n+1} - 5y_n = h(4f_{n+1} + 2f_n).$$

Es hat die Ordnung 3, denn die Bedingungsgleichung für $q = 4$ ist nicht mehr erfüllt. Wenden wir dieses Verfahren auf das Testproblem

$$y' = y, \qquad y(0) = 1$$

an, so erhalten wir für die Schrittweiten $h = 1/10$, $h = 1/20$ und $h = 1/40$ den in Abbildung 9.1 dargestellten Lösungsverlauf. Im Gegensatz zu Runge-Kutta-Verfahren ist eine hohe Ordnung bei Mehrschrittverfahren offensichtlich nicht hinreichend für Konvergenz. Zur Erklärung betrachten wir die lineare Differenzengleichung

$$y_{n+2} + 4(1 - h)y_{n+1} - (5 + 2h)y_n = 0.$$

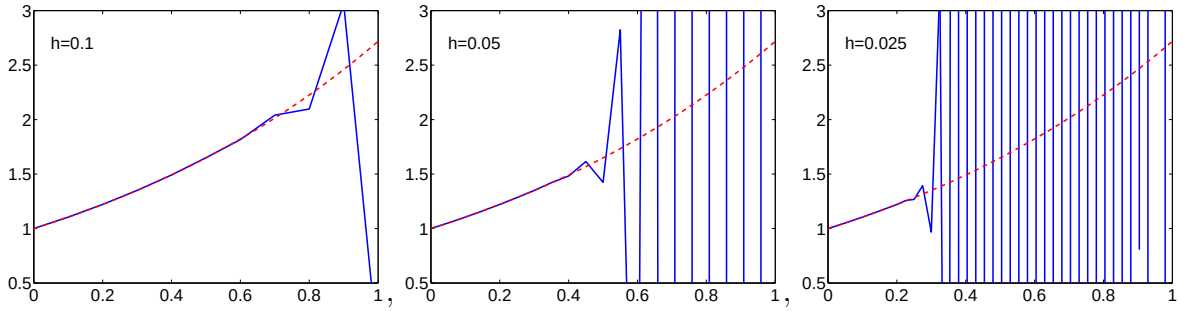

Abb. 9.1: Numerische Lösung des instabilen Verfahrens

Diese hat für $h \to 0$ das charakteristische Polynom

$$\zeta^2 + 4\zeta - 5 = (\zeta + 5)(\zeta - 1).$$

Die allgemeine Lösung der Differenzengleichung ist

$$y_n = c_1 1^n + c_2(-5)^n.$$

Der zweite oszillierende Term dominiert die numerische Lösung und erklärt das in Abbildung 9.1 dargestellte numerische Verhalten. ⋄

9.4 Stabilität von Mehrschrittverfahren

Definition 9.5. *Das Mehrschrittverfahren* (9.3) *heißt* **stabil** *(0-stabil), falls alle Lösungen der Differenzengleichung*

$$\alpha_0 y_n + \alpha_1 y_{n+1} + \cdots + \alpha_k y_{n+k} = 0 \tag{9.6}$$

für $n \to \infty$ beschränkt bleiben.

Satz 9.6. *Das Mehrschrittverfahren* (9.3) *ist stabil genau dann, wenn die Nullstellen ζ_j, $j = 1, \ldots, k$ von $\varrho(\zeta)$ die Wurzelbedingung erfüllen, d. h.*

(a) $|\zeta_j| \le 1$ *für* $j = 1, \ldots, k$ *und*

(b) *falls* $|\zeta_j| = 1$, *so ist* ζ_j *einfache Nullstelle.*

Beweis. Sind $\zeta_1, \ldots, \zeta_l$ die paarweise verschiedenen Nullstellen von ϱ mit Vielfachheiten $m_1, \ldots, m_l$, so ist jede Lösung von (9.6) von der Form

$$y_n = p_1(n)\zeta_1^n + \cdots + p_l(n)\zeta_l^n,$$

wobei p_j ein beliebiges Polynom vom Grad $m_j - 1$ ist (Übung). Daher ist Stabilität äquivalent zur Wurzelbedingung. □

Beispiel. (Stabilität von Adams-Verfahren)
Bei Adams-Verfahren ist

$$\varrho(\zeta) = \zeta^k - \zeta^{k-1} = \zeta^{k-1}(\zeta - 1).$$

ϱ hat also eine einfache Nullstelle 1 und eine $(k-1)$-fache Nullstelle bei 0. Adams-Verfahren sind daher stabil. ⋄

Man kann zeigen, dass BDF-Verfahren für $k \leq 6$ stabil und für $k \geq 7$ instabil sind Hairer, Norsett & Wanner (1991, Theorem III.3.4).

Für festes k kann man Schranken für die maximale Ordnung eines stabilen Mehrschrittverfahrens angeben (Dahlquist, 1956):

Satz 9.7. *(Erste Ordnungsschranke von Dahlquist)*
Für die Ordnung p eines stabilen, linearen k-Schrittverfahrens gilt

(a) $p \leq k+2$ *falls k gerade,*

(b) $p \leq k+1$ *falls k ungerade,*

(c) $p \leq k$ *falls $\beta_k/\alpha_k \leq 0$ (insbesondere für alle expliziten Verfahren).*

Beweis. Hairer et al. (1991, Theorem III.3.5) □

9.5 Konvergenz von Mehrschrittverfahren

Definition 9.8. *(Konvergenz)*

(a) *Das Mehrschrittverfahren (9.3) heißt* **konvergent der Ordnung** *p, wenn für alle Anfangswertprobleme (8.1) mit f genügend oft differenzierbar ein $h_0 > 0$ existiert, so dass für $h \leq h_0$*

$$\|y_n - y(t_n)\| \leq Mh^p, \qquad t_n = t_0 + nh \in [t_0, T]$$

mit M unabhängig von n, h gilt, wenn die Startwerte

$$\|y_j - y(t_j)\| \leq Kh^p, \qquad j = 0, 1, \ldots, k-1$$

für $h \leq h_0$ erfüllen.

(b) *Das Mehrschrittverfahren (9.3) heißt* **konvergent**, *wenn es (a) mit $p \geq 1$ erfüllt.*

Satz 9.9. *Ein stabiles Mehrschrittverfahren der Ordnung p ist konvergent der Ordnung p.*

Beweis. Die Näherungslösungen y_n erfüllen

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f(t_{n+j}, y_{n+j})$$

und für die exakte Lösung gilt wegen der Ordnung nach Definition 9.3(b) und dem Beweis von Satz 9.4

$$\sum_{j=0}^{k} \alpha_j y(t_{n+j}) = h \sum_{j=0}^{k} \beta_j f(t_{n+j}, y(t_{n+j})) + d_{n+k}, \qquad \|d_{n+k}\| \leq C_d h^{p+1}.$$

Subtrahieren wir beide Gleichungen voneinander und setzen

$$e_n = y_n - y(t_n), \qquad g_n = f(t_n, y_n) - f(t_n, y(t_n)),$$

so ergibt sich

$$\sum_{j=0}^{k} \alpha_j e_{n+j} = h \sum_{j=0}^{k} \beta_j g_{n+j} - d_{n+k}.$$

Jetzt verwenden wir die Methode der *erzeugenden Funktionen*, die auf Euler (1755) zurückgeht: Wir multiplizieren die Gleichung mit ζ^{n+k}:

$$\sum_{j=0}^{k} \alpha_j \zeta^{k-j} e_{n+j} \zeta^{n+j} = h \sum_{j=0}^{k} \beta_j \zeta^{k-j} g_{n+j} \zeta^{n+j} - d_{n+k} \zeta^{n+k}$$

und summieren von $n = 0$ bis ∞:

$$\sum_{j=0}^{k} \alpha_j \zeta^{k-j} \sum_{n=0}^{\infty} e_{n+j} \zeta^{n+j} = h \sum_{j=0}^{k} \beta_j \zeta^{k-j} \sum_{n=0}^{\infty} g_{n+j} \zeta^{n+j} - \sum_{n=0}^{\infty} d_{n+k} \zeta^{n+k}. \tag{9.7}$$

Definieren wir die formalen Potenzreihen (erzeugende Funktionen)

$$e(\zeta) = \sum_{n=0}^{\infty} e_n \zeta^n, \qquad g(\zeta) = \sum_{n=0}^{\infty} g_n \zeta^n, \qquad d(\zeta) = \sum_{n=0}^{\infty} d_n \zeta^n,$$

wobei $d_0 = \ldots = d_{k-1} = 0$, so ergibt sich für die linke Seite von (9.7)

$$\begin{aligned} \sum_{j=0}^{k} \alpha_j \zeta^{k-j} \sum_{n=0}^{\infty} e_{n+j} \zeta^{n+j} &= \sum_{j=0}^{k} \alpha_j \zeta^{k-j} \left(e(\zeta) - \sum_{n=0}^{j-1} e_n \zeta^n \right) \\ &=: \alpha(\zeta) e(\zeta) - p(\zeta) \end{aligned}$$

mit einem Polynom p vom Grad $\leq k-1$, welches nur von α_j, $j = 0, \ldots, k$ und e_j für $j = 0, \ldots, k-1$ abhängt. Analog definieren wir für die rechte Seite von (9.7)

$$\sum_{j=0}^{k} \beta_j \zeta^{k-j} \sum_{n=0}^{\infty} g_{n+j} \zeta^{n+j} =: \beta(\zeta) g(\zeta) - q(\zeta), \qquad q \in \mathcal{P}_{k-1}$$

und erhalten schließlich die zu (9.7) äquivalente Gleichung

$$e(\zeta) = h \frac{\beta(\zeta)}{\alpha(\zeta)} g(\zeta) + \frac{p(\zeta) - h q(\zeta)}{\alpha(\zeta)} - \frac{d(\zeta)}{\alpha(\zeta)}. \tag{9.8}$$

Jetzt setzen wir (unter Beachtung von $\alpha(0) = \alpha_k \neq 0$)

$$\frac{1}{\alpha(\zeta)} =: \sum_{n=0}^{\infty} r_n \zeta^n.$$

Koeffizientenvergleich in

$$\begin{aligned} 1 &= \left(\sum_{j=0}^{k} \alpha_j \zeta^{k-j} \right) \sum_{n=0}^{\infty} r_n \zeta^n \\ &= \sum_{n=0}^{\infty} \left(\sum_{j=0}^{k} \alpha_j r_{n+j} \right) \zeta^{n+k} + P_{k-1}(\zeta), \qquad P_{k-1} \in \mathcal{P}_{k-1} \end{aligned}$$

zeigt, dass

$$\sum_{j=0}^{k} \alpha_j r_{n+j} = 0 \qquad \text{für alle } n \geq 0$$

gelten muss. Nach Voraussetzung ist das Mehrschrittverfahren stabil. Die Lösungen $\{r_n\}_{n\geq 0}$ der obigen Differenzengleichung sind also beschränkt, d. h. $|r_n| \leq C_r$ für alle $n \geq 0$. Als Folgerung gilt dasselbe auch für die Koeffizienten von

$$\frac{\beta(\zeta)}{\alpha(\zeta)} =: \sum_{n=0}^{\infty} s_n \zeta^n = \sum_{n=0}^{\infty} \left(\sum_{j=0}^{k} \beta_j r_{n+j} \right) \zeta^{n+k} + P'_{k-1}(\zeta), \qquad P'_{k-1} \in \mathcal{P}_{k-1},$$

also $|s_n| \leq C_s$ für alle $n \geq 0$.

Koeffizientenvergleich in (9.8) liefert mit $r_j = 0$ für $j < 0$

$$e_n = h \sum_{j=0}^{n} s_{n-j} g_j + \sum_{j=0}^{k-1} r_{n-j}(p_j - hq_j) - \sum_{j=0}^{n} r_{n-j} d_j.$$

Nach Voraussetzung gilt

$$\|g_j\| \leq L\|e_j\|, \qquad \|p_j\| = O(h^p), \qquad \|q_j\| = O(h^p)$$

und aus $\|d_j\| \leq C_d h^{p+1}$ und $|r_j| \leq C_r$ folgt

$$\|\sum_{j=0}^{n} r_{n-j} d_j\| \leq C_d \widetilde{C}_r(nh) h^p.$$

Da auch $|s_j| \leq C_s$ gilt, gibt es für $t_0 + nh \leq T$ eine Konstante C mit

$$\|e_n\| \leq hC_s L \sum_{j=0}^{n} \|e_j\| + Ch^p.$$

Durch Induktion zeigt man für h hinreichend klein $\|e_n\| \leq \varepsilon_n$ mit

$$\varepsilon_n = hC_s L \sum_{j=0}^{n} \varepsilon_j + Ch^p, \qquad \varepsilon_0 = O(h^p)$$

woraus

$$\varepsilon_n - \varepsilon_{n-1} = hC_s L \varepsilon_n,$$

oder

$$\varepsilon_n = \frac{1}{1 - hC_s L} \varepsilon_{n-1} = \ldots = \frac{1}{(1 - hC_s L)^n} \varepsilon_0$$

folgt. Des Weiteren ist $1 + x \leq e^x$ für alle $x \in \mathbb{R}$ und damit

$$\frac{1}{1 - hC_s L} = 1 + \frac{hC_s L}{1 - hC_s L} \leq e^{hC_s L/(1 - hC_s L)},$$

so dass aus $nh \leq T - t_0$ und $h \leq h_0$

$$\|e_n\| \leq \varepsilon_n \leq e^{nhC_s L/(1 - h_0 C_s L)} \varepsilon_0 = Mh^p$$

folgt. □

9.6 Variable Schrittweite

Ein effizienter Integrator muss die Möglichkeit haben, die Schrittweite adaptiv zu wählen. Eine Änderung der Schrittweite bei Mehrschrittverfahren ist jedoch schwierig, da die Formeln, die wir etwa für Adams- und BDF-Verfahren hergeleitet haben, Näherungslösungen an äquidistanten Zeitpunkten benötigen. Es gibt prinzipiell zwei Möglichkeiten, diese Schwierigkeit zu beheben:

(a) Mit Polynominterpolation die benötigten Näherungen auf dem neuen äquidistanten Gitter berechnen oder

(b) Verfahren zu konstruieren, die direkt mit variablen Schrittweiten arbeiten.

Wir stellen hier den zweiten Ansatz vor. Hierzu bezeichnen wir die Schrittweiten mit

$$h_n = t_{n+1} - t_n, \qquad n \geq 0.$$

Die Näherungslösungen seien wieder $y_n \approx y(t_n)$ und es sei $f_n = f(t_n, y_n)$.

Explizite Adams-Verfahren

Wie im Fall konstanter Schrittweite konstruieren wir das Interpolationspolynom p durch die Punkte $(t_{n-k+1}, f_{n-k+1}), \ldots, (t_n, f_n)$. Mit Hilfe der Newton'schen Interpolationsformel gilt

$$p(t) = \sum_{j=0}^{k-1} \prod_{i=0}^{j-1} (t - t_{n-i}) \delta^j f[t_n, t_{n-1}, \ldots, t_{n-j}],$$

wobei die dividierten Differenzen $\delta^j f[t_n, t_{n-1}, \ldots, t_{n-j}]$ rekursiv durch

$$\begin{aligned} \delta^0 f[t_n] &= f_n, \\ \delta^j f[t_n, \ldots, t_{n-j}] &= \frac{\delta^{j-1} f[t_n, \ldots, t_{n-j+1}] - \delta^{j-1} f[t_{n-1}, \ldots, t_{n-j}]}{t_n - t_{n-j}} \end{aligned}$$

definiert sind. Für die praktische Implementierung schlägt Krogh (1969) vor, p gemäß

$$p(t) = \sum_{j=0}^{k-1} \prod_{i=0}^{j-1} \frac{t - t_{n-i}}{t_{n+1} - t_{n-i}} \Phi_j^*(n)$$

zu schreiben, wobei

$$\Phi_j^*(n) = \prod_{i=0}^{j-1} (t_{n+1} - t_{n-i}) \delta^j f[t_n, t_{n-1}, \ldots, t_{n-j}].$$

Die Approximation y_{n+1} des expliziten Adams-Verfahrens zur Zeit t_{n+1} ist dann durch

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} p(t) dt$$

oder

$$y_{n+1} = y_n + h_n \sum_{j=0}^{k-1} g_j(n) \Phi_j^*(n), \qquad h_n = t_{n+1} - t_n \tag{9.9}$$

mit

$$g_j(n) = \frac{1}{h_n} \int_{t_n}^{t_{n+1}} \prod_{i=0}^{j-1} \frac{t - t_{n-i}}{t_{n+1} - t_{n-i}} dt$$

definiert. Für konstante Schrittweiten $h_j = h$ reduziert sich diese Darstellung zu

$$g_j(n) = \gamma_j, \qquad \Phi_j^*(n) = \nabla^j f_n.$$

Der Beweis ist eine einfache Übung.

Implizite Adams-Verfahren

Es sei p^* das Interpolationspolynom durch die Punkte $(t_{n-k+1}, f_{n-k+1}), \ldots, (t_{n+1}, f_{n+1})$. Dann liefert die Newton'sche Interpolationsformel

$$p^*(t) = p(t) + \prod_{i=0}^{k-1} (t - t_{n-i}) \delta^k f[t_{n+1}, t_n, \ldots, t_{n-k+1}],$$

wobei p das Interpolationspolynom durch $(t_{n-k+1}, f_{n-k+1}), \ldots, (t_n, f_n)$ des expliziten Adams-Verfahrens ist. Die numerische Lösung ist durch

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} p^*(t) dt$$

oder

$$y_{n+1} = p_{n+1} + h_n g_k(n) \Phi_k(n+1) \tag{9.10}$$

gegeben, wobei p_{n+1} die numerische Lösung des expliziten Adams-Verfahrens definiert in (9.9) ist und

$$\Phi_k(n+1) = \prod_{i=0}^{k-1} (t_{n+1} - t_{n-i}) \delta^k f[t_{n+1}, t_n, \ldots, t_{n-k+1}].$$

Für eine effiziente Implementierung leiten wir jetzt Rekursionsformeln für die Koeffizienten $g_j(n)$, $\Phi_j(n)$ und $\Phi_j^*(n)$ her. Aus den rekursiv definierten dividierten Differenzen können sofort Rekursionen für die Werte $\Phi_j^*(n)$, $j = 0, \ldots, k-1$ und $\Phi_k(n+1)$ abgelesen werden:

$$\begin{aligned} \Phi_0(n) &= \Phi_0^*(n) = f_n \\ \Phi_{j+1}(n+1) &= \Phi_j(n+1) - \Phi_j^*(n) \\ \Phi_j^*(n) &= \beta_j(n) \Phi_j(n) \end{aligned} \tag{9.11}$$

mit

$$\beta_j(n) = \prod_{i=0}^{j-1} \frac{t_{n+1} - t_{n-i}}{t_n - t_{n-i-1}}.$$

Für $\beta_j(n)$ erhalten wir wiederum rekursiv

$$\beta_0(n) = 1, \qquad \beta_j(n) = \beta_{j-1}(n) \frac{t_{n+1} - t_{n-j+1}}{t_n - t_{n-j}}.$$

Trickreich ist die Berechnung der Koeffizienten $g_j(n)$ (Krogh, 1974). Wir führen das q-fache Integral

$$c_{jq}(t) = \frac{(q-1)!}{h_n^q} \int_{t_n}^{t} \int_{t_n}^{\xi_{q-1}} \cdots \int_{t_n}^{\xi_1} \prod_{i=0}^{j-1} \frac{\xi_0 - t_{n-i}}{t_{n+1} - t_{n-i}} d\xi_0 \ldots d\xi_{q-1}$$

ein. Nach Definition von $g_j(n)$ ist nämlich

$$g_j(n) = c_{j1}(t_{n+1}).$$

Lemma 9.10. *Es gilt*

$$c_{0q}(t_{n+1}) = \frac{1}{q}, \qquad c_{1q}(t_{n+1}) = \frac{1}{q(q+1)},$$
$$c_{jq}(t_{n+1}) = c_{j-1,q}(t_{n+1}) - c_{j-1,q+1}(t_{n+1}) \frac{h_n}{t_{n+1} - t_{n-j+1}}.$$

Beweis. Die ersten beiden Relationen folgen sofort aus der Definition: Für $j = 0$ ist

$$\begin{aligned} c_{0q}(t) &= \frac{(q-1)!}{h_n^q} \int_{t_n}^{t} \int_{t_n}^{\xi_{q-1}} \cdots \int_{t_n}^{\xi_1} 1 d\xi_0 \ldots d\xi_{q-1} \\ &= \frac{(q-1)!}{h_n^q} \int_{t_n}^{t} \int_{t_n}^{\xi_{q-1}} \cdots \int_{t_n}^{\xi_2} (\xi_1 - t_n) d\xi_1 \ldots d\xi_{q-1} \\ &= \frac{(q-1)!}{h_n^q} \int_{t_n}^{t} \int_{t_n}^{\xi_{q-1}} \cdots \int_{t_n}^{\xi_3} \frac{(\xi_2 - t_n)^2}{2!} d\xi_2 \ldots d\xi_{q-1} \\ &= \ldots \\ &= \frac{(q-1)!}{h_n^q} \frac{(t-t_n)^q}{q!}, \end{aligned}$$

also $c_{0q}(t_{n+1}) = 1/q$. Für $j = 1$ zeigt man die Behauptung ebenso oder man folgert sie später aus der Rekursionsformel. Um die Rekursionsformel zu beweisen, definieren wir

$$d(t) = c_{jq}(t) - c_{j-1,q}(t) \frac{t - t_{n-j+1}}{t_{n+1} - t_{n-j+1}} + c_{j-1,q+1}(t) \frac{h_n}{t_{n+1} - t_{n-j+1}}.$$

Offensichtlich gilt $d^{(i)}(t_n) = 0$ für $i = 0, \ldots, q-1$. Die q-te Ableitung von d verschwindet ebenfalls, denn zunächst ist nach Definition von c_{jq}

$$c_{jq}^{(q)}(t) = \frac{(q-1)!}{h_n^q} \prod_{i=0}^{j-1} \frac{t - t_{n-i}}{t_{n+1} - t_{n-i}} \tag{9.12}$$

und dann mit der Leibnizregel

$$\begin{aligned} &\frac{d^q}{dt^q} \cdot \left(c_{j-1,q}(t) \frac{t - t_{n-j+1}}{t_{n+1} - t_{n-j+1}} \right) \\ &\quad = c_{j-1,q}^{(q)}(t) \frac{t - t_{n-j+1}}{t_{n+1} - t_{n-j+1}} + q c_{j-1,q}^{(q-1)}(t) \frac{1}{t_{n+1} - t_{n-j+1}} \\ &\quad = c_{j,q}^{(q)}(t) + c_{j-1,q+1}^{(q)}(t) \frac{h_n}{t_{n+1} - t_{n-j+1}}, \end{aligned}$$

wobei die letzte Gleichung aus (9.12) folgt. Da $d^{(q)} \equiv 0$, gilt dasselbe auch für alle höheren Ableitungen von d. Insbesondere verschwinden alle Ableitungen von d an der Stelle t_n, das Polynom d muss das Nullpolynom sein. Die Behauptung ergibt sich dann durch Einsetzen von $t = t_{n+1}$. □

Mit den obigen Rekursionsformeln berechnet man nacheinander

$$\begin{aligned} &c_{2,q}(t_{n+1}) \text{ für } q = 1, \ldots, k-1, \\ &c_{3,q}(t_{n+1}) \text{ für } q = 1, \ldots, k-2, \\ &\vdots \\ &c_{k,q}(t_{n+1}) \text{ für } q = 1 \end{aligned}$$

und liest daraus die Koeffizienten $g_j(n) = c_{j,1}(t_{n+1})$ der Adams-Verfahren ab.

BDF-Verfahren

Analog zu den Adams-Verfahren kann man auch BDF-Verfahren in natürlicher Weise auf variable Schrittweiten verallgemeinern. Hierzu sei q das Interpolationspolynom vom Grad k, welches in den Punkten (t_j, y_j), $j = n-k+1, \ldots, n+1$ interpoliert. Mit dividierten Differenzen kann man q durch

$$q(t) = \sum_{j=0}^{k} \prod_{i=0}^{j-1} (t - t_{n+1-i}) \delta^j y[t_{n+1}, t_n, \ldots, t_{n-j+1}]$$

ausdrücken. Die Forderung

$$q'(t_{n+1}) = f(t_{n+1}, y_{n+1})$$

führt dann direkt auf die allgemeinen BDF-Formeln

$$\sum_{j=0}^{k} h_n \prod_{i=1}^{j-1} (t_{n+1} - t_{n+1-i}) \delta^j y[t_{n+1}, t_n, \ldots, t_{n-j+1}] = h_n f(t_{n+1}, y_{n+1}).$$

Erfreulicherweise kann man die Koeffizienten hier wesentlich einfacher als bei Adams-Verfahren berechnen.

9.7 Schrittweiten- und Ordnungssteuerung

Für die Schrittweitensteuerung beim impliziten Adams-Verfahren nehmen wir an, dass wir bereits erfolgreich bis zur Zeit t_n integriert haben und dass ein weiterer Schritt mit Schrittweite h_n und Ordnung $k+1$ die Näherung $y_{n+1} \approx y(t_{n+1})$ geliefert hat. Um zu entscheiden, ob y_{n+1} akzeptiert werden kann, benötigen wir eine Schätzung des lokalen Fehlers. Diese kann zum Beispiel durch

$$\texttt{err}_{k+1}(n+1) = y^*_{n+1} - y_{n+1}$$

erreicht werden, wobei y^*_{n+1} das Ergebnis einer impliziten Adams-Formel der Ordnung $k+2$ ist. Aus (9.9) können wir die Differenz der beiden expliziten Verfahren als

$$p^*_{n+1} - p_{n+1} = h_n g_k(n) \Phi^*_k(n)$$

schreiben. Subtrahieren wir dann die Gleichung (9.10) für k und $k+1$, so ergibt sich

$$\begin{aligned} y^*_{n+1} - y_{n+1} &= p^*_{n+1} - p_{n+1} + h_n \left[g_{k+1}(n)\Phi_{k+1}(n+1) - g_k(n)\Phi_k(n+1) \right] \\ &= h_n \left[g_k(n) \left(\Phi^*_k(n) - \Phi_k(n+1) \right) + g_{k+1}(n)\Phi_{k+1}(n+1) \right] \\ &= h_n \left[g_{k+1}(n) - g_k(n) \right] \Phi_{k+1}(n+1), \end{aligned}$$

wobei die letzte Gleichung aus (9.11) folgt. Man überlegt sich leicht, dass sich der führende Term in der Fehlerentwicklung nicht ändert, wenn man in diesem Ausdruck $\Phi_{k+1}(n+1)$ durch

$$\Phi^p_{k+1}(n+1) = \prod_{i=0}^{k} (t_{n+1} - t_{n-i}) \delta^{k+1} f^p[t_{n+1}, t_n, \ldots, t_{n-k}]$$

ersetzt, wobei der Index p bei f bedeutet, dass bei den dividierten Differenzen $f_{n+1} = f(t_{n+1}, y_{n+1})$ durch $f(t_{n+1}, p_{n+1})$ ersetzt wird. Gewöhnlich benutzt man ja p_{n+1} als Predictor für das implizite Verfahren und hat dann $\Phi^p_{k+1}(n+1)$ ohnehin schon berechnet. Die einzigen Kosten zur Berechnung der Schätzung

$$\widetilde{\mathtt{err}}_{k+1}(n+1) = h_n \left[g_{k+1}(n) - g_k(n) \right] \Phi^p_{k+1}(n+1)$$

entstehen also für die Berechnung von $g_{k+1}(n)$ und davon hatten wir ja bereits gezeigt, wie man das effizient bewerkstelligen kann. Wir akzeptieren den Schritt, wenn

$$\| \widetilde{\mathtt{err}}_{k+1}(n+1) \| \leq 1$$

in der gewichteten Norm ist. Im nächsten Schritt würden wir gerne die Schrittweite h_{n+1} so wählen, dass die Vorabschätzung des Fehlers klein genug ist:

$$h_{n+1} \left| g_{k+1}(n+1) - g_k(n+1) \right| \cdot \| \Phi^p_{k+1}(n+2) \| \leq 1.$$

Dies ist leider nicht möglich, da $g_j(n+1)$ und $\Phi^p_{k+1}(n+2)$ in einer komplizierten Weise von der unbekannten Schrittweite h_{n+1} abhängen und auch die Berechnung von $g_{k+1}(n+1)$ und $g_k(n+1)$ zu teuer wäre. Hätten wir in den letzten Schritten mit konstanter Schrittweite h gerechnet, so wüssten wir, dass der lokale Fehler des Verfahrens der Ordnung $k+1$ von der Form $C(t_{n+2})h^{k+2} + O(h^{k+3})$ ist, wobei C glatt von t abhängt. Der lokale Fehler kann dann genauso wie bei Einschrittverfahren geschätzt werden und wir erhalten

$$h^{(k+1)}_{opt} = h_n \cdot \left(\frac{1}{\| \widetilde{\mathtt{err}}_{k+1}(n+1) \|} \right)^{1/(k+2)}$$

als optimale Schrittweite. Unter der Annahme konstanter Schrittweite gilt auch

$$\widetilde{\mathtt{err}}_{k+1}(n+1) = h_n \gamma^*_{k+1} \Phi^p_{k+1}(n+1),$$

wobei γ^*_{k+1} die Koeffizienten des impliziten Adams-Verfahrens aus Abschnitt 9.1 sind.

Als nächstes diskutieren wir, wie man die optimale Ordnung bestimmen kann. Da die Zahl der benötigten Funktionsauswertungen für alle Ordnungen gleich ist, gibt es im Wesentlichen zwei Strategien. Man kann die Ordnung $k+1$ entweder so wählen, dass die lokale Fehlerschätzung minimiert oder die neue optimale Schrittweite maximiert wird. Wegen des Exponenten $1/(k+2)$ sind die beiden Strategien nicht immer äquivalent. Zudem haben wir gesehen, dass sich bei Adams-Verfahren die Berechnung der Koeffizienten $g_j(n)$ wesentlich

vereinfacht, wenn die Schrittweite konstant ist. Man wird also versuchen, möglichst lange mit konstanter Schrittweite zu rechnen.

Es ist zum Beispiel folgendes Vorgehen denkbar:

$$k_{neu} = \begin{cases} k-1, & \text{falls } \|\widetilde{\texttt{err}}_{k+1}(n+1)\| > \|\widetilde{\texttt{err}}_k(n+1)\| \\ k+1, & \text{falls die Schrittweite } k+2 \text{ Schritte konstant war und} \\ & \|\widetilde{\texttt{err}}_{k+2}(n+1)\| < \|\widetilde{\texttt{err}}_{k+1}(n+1)\| < \|\widetilde{\texttt{err}}_k(n+1)\| \\ k, & \text{sonst} \end{cases}$$

und

$$h_{n+1} = \begin{cases} 2h_n, & \text{falls } h_{opt} \geq 2h_n, \\ \max\{h_{opt}, h_n/2\}, & \text{falls } h_{opt} \leq 0.9h_n \\ h_n, & \text{sonst .} \end{cases}$$

In der Startphase wählen wir zum Beispiel

$$h_0 = \min\left\{ h_{\max}, \frac{1}{5}\sqrt{\frac{1}{\|f(t_0, y_0)\|}} \right\}$$

oder eine vom Benutzer vorgegebene Anfangsschrittweite und beginnen mit Ordnung $k_0 = 1$. Wir setzen $k_{n+1} = k_n + 1$ und $h_{n+1} = 2h_n$ bis entweder

(a) der Schritt verworfen wurde oder

(b) die maximale Ordnung erreicht wurde oder

(c) eine Erniedrigung der Ordnung (s.o.) vorgeschlagen wurde.

9.8 Numerische Vergleiche

Das numerische Verhalten eines Verfahrens hängt wesentlich von der Implementierung, vor allem von Details der Schrittweiten– und Ordnungssteuerung, ab. Man darf daher nicht nur die Verfahren vergleichen, sondern muss sich dabei immer auf eine bestimmte Implementierung beziehen. Vergleichskriterien sind zum Beispiel

- Anzahl der Funktionsauswertungen / erreichte Genauigkeit
- Rechenzeit / erreichte Genauigkeit
- Robustheit, Verlässlichkeit
- evtl. Speicherbedarf, Parallelisierbarkeit

In der Literatur wird häufig statt der erreichten die verlangte (etwa durch *Atol* und *Rtol* gegebene) Genauigkeit verglichen. Die erreichte Genauigkeit lässt sich nur exakt messen, wenn man die exakte Lösung kennt. Ist das nicht der Fall, so berechnet man mit einem robusten Code mit sehr kleiner Schrittweite (beziehungsweise sehr kleinen Toleranzen) eine Referenzlösung und verwendet diese statt der exakten Lösung.

Die Codes sollten an "geeigneten" Testbeispielen verglichen werden, die man häufig im Internet findet. Eine Vielzahl von numerischen Vergleichen ist in Hairer et al. (1991) wiedergegeben.