10. STEIFE DIFFERENTIALGLEICHUNGEN

10.1 Einführung

Steife Differentialgleichungen sind Probleme, für die explizite Verfahren nicht funktionieren. Betrachten wir dazu ein Beispiel einer chemischen Reaktion von Robertson (1966) Hairer & Wanner (1996, S. 3)). Drei Substanzen A, B, C reagieren mit den angegebenen Reaktionsgeschwindigkeiten gemäß

$$\begin{aligned} A &\xrightarrow{0.04} B && \text{(langsam)} \\ B+B &\xrightarrow{3\cdot 10^7} C+B && \text{(sehr schnell)} \\ B+C &\xrightarrow{10^4} A+C && \text{(schnell)} \end{aligned}$$

Nach dem Massenwirkungsgesetz führt dies auf ein System von drei Differentialgleichungen für die Konzentrationen der Substanzen:

$$\begin{array}{lrcrcrcl} A: & y_1' &=& -0.04y_1 &+& 10^4 y_2 y_3 & & \\ B: & y_2' &=& 0.04y_1 &-& 10^4 y_2 y_3 &-& 3\cdot 10^7 y_2^2 \\ C: & y_3' &=& & & & & 3\cdot 10^7 y_2^2 \end{array}$$

Die Reaktion ist durch Vorgabe von Anfangskonzentrationen

$$y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0.$$

eindeutig bestimmt. Die Lösung dieser Differentialgleichung für $t \in [0, 100]$ ist in Abbildung 10.1 dargestellt.

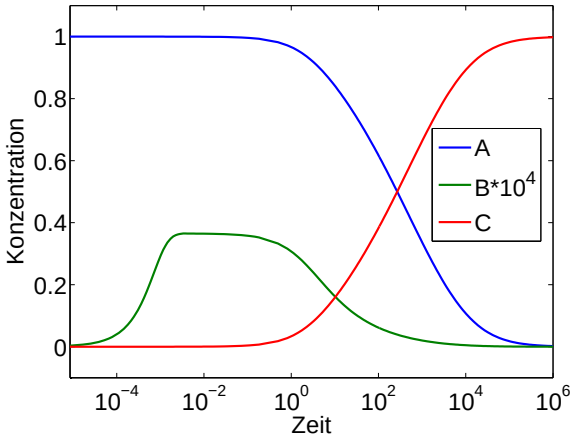

Abb. 10.1: Konzentrationsverlauf für die Substanzen A, B, C

In Abbildung 10.2 sind die Schrittweiten dargestellt, die von den MATLAB-Lösern `ode45` (explizites Runge-Kutta-Verfahren der Ordnung 5 mit Schrittweitensteuerung) und `ode15s`

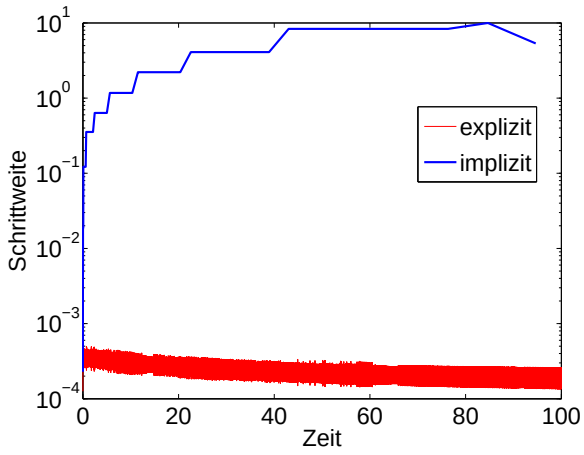

Abb. 10.2: Schrittweiten von `ode45` (rot) und `ode15s` (blau)

(implizites Mehrschrittverfahren mit Ordnungs- und Schrittweitensteuerung) mit Standardtoleranzen (absoluter Toleranz von 10^{-6} und relativer Toleranz von 10^{-3}) gewählt werden. Man erkennt, dass das explizite Verfahren trotz sehr glatter Lösungen extrem kleine Schrittweiten wählt, während das implizite Verfahren mit sehr viel größerer Schrittweite rechnet.

Derartiges beobachtet man auch bei vielen anderen Problemen, etwa aus der Wärmelehre, elektrischen Netzwerken, oder Ähnlichem. Obwohl die Lösung sehr "glatt" ist und obwohl die Differentialgleichung stabil ist, d. h. die Lösungen werden rasch gedämpft, wählt das Programm sehr kleine (annähernd konstante) Schrittweiten. Wird hingegen eine größere Schrittweite vorgeschrieben, so divergiert die numerische Lösung. Der Grund hierfür liegt im unterschiedlichen Stabilitätsverhalten von Differentialgleichung und numerischem Verfahren.

Betrachten wir hierzu die Differentialgleichung

$$y'(t) = f(t, y(t)), \qquad f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d. \tag{10.1}$$

$z(t)$ sei eine Lösung von (10.1) mit Anfangswert $z(t_0) = z_0$. Für eine beliebige Lösung der Differentialgleichung (10.1) gilt nach dem Satz von Taylor

$$y'(t) = f(t, y(t)) = f(t, z(t)) + \frac{\partial f}{\partial y}(t, \zeta)(y(t) - z(t))$$

mit gewissen Zwischenstellen $\zeta \in [z(t), y(t)]$, die von Zeile zu Zeile variieren können.

Wir treffen für das Folgende vereinfachte Annahmen, welche brauchbare Näherungen auf kurzen Zeitintervallen oder für kleine Störungen darstellen:

$$\frac{\partial f}{\partial y}(t, y) = A \qquad \forall t, y$$

mit einer konstanten Matrix $A \in \mathbb{R}^{d,d}$ und

$$z(t) \equiv z_0 \qquad \forall t \qquad \text{also} \quad z'(t) \equiv 0.$$

Mit diesen Vereinbarungen betrachten wir die numerische Lösung der Differentialgleichung

$$y'(t) = A(y(t) - z_0), \qquad y(t_0) = y_0$$

mit dem expliziten Euler-Verfahren

$$y_{n+1} = y_n + hA(y_n - z_0), \qquad y_0 \text{ gegeben.}$$

Der exakte Abstand $e(t) = y(t) - z(t)$ pflanzt sich gemäß

$$e'(t) = Ae(t), \qquad e(t_0) = e_0 := y_0 - z_0,$$

der des Euler-Verfahrens durch

$$e_{n+1} = e_n + hAe_n$$

fort.

Jetzt setzen wir noch A diagonalisierbar voraus. Dann gibt es eine nichtsinguläre Matrix T, so dass

$$T^{-1}AT = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d), \quad \lambda_j \in \mathbb{C}.$$

Definieren wir $v(t) = T^{-1}e(t)$ und entsprechend $v_n = T^{-1}e_n$, so gilt

$$v'(t) = \Lambda v(t), \qquad v(t_0) = v_0 := T^{-1}(y_0 - z_0),$$

bzw.

$$v_{n+1} = v_n + h\Lambda v_n.$$

Die d Differentialgleichungen werden durch diese Transformation entkoppelt. In den oben genannten Anwendungen ist

$$\operatorname{Re}\lambda_i \le 0 \quad \forall i \quad \text{und} \quad \exists i: \ \operatorname{Re}\lambda_i \ll 0.$$

Betrachten wir nun die i-te Komponente und bezeichnen v_i wieder mit y, so erhalten wir

$$y'(t) = \lambda y(t), \qquad y(t_0) = y_0. \tag{10.2}$$

Die exakte Lösung $y(t) = e^{\lambda(t-t_0)}y_0$ ist für alle $t \ge t_0$ beschränkt, falls $\operatorname{Re}\lambda \le 0$, und es gilt $y(t) \to 0$ für $t \to \infty$, falls $\operatorname{Re}\lambda < 0$.

Das explizite Euler-Verfahren angewendet auf (10.2) liefert

$$y_{n+1} = y_n + h\lambda y_n \qquad \text{oder} \qquad y_n = (1 + h\lambda)^n y_0.$$

Die Folge $\{y_n\}_n$ ist nur beschränkt für alle $t \ge t_0$, wenn $|1 + h\lambda| \le 1$ gilt, und es gilt $y_n \to 0$ für $n \to \infty$ nur, wenn $|1 + h\lambda| < 1$, vgl. Abb. 10.3. Für gegebene Eigenwerte λ liefert das eine Einschränkung an die Zeitschrittweite h. Bei Wahl größerer Schrittweiten divergiert die numerische Lösung.

Definition 10.1. (*Dahlquist* 1963)
Ein numerisches Verfahren heißt A-**stabil**, *falls die numerische Lösung* $(y_n)_{n\ge0}$ *beschränkt bleibt, wenn das Verfahren mit beliebiger konstanter Schrittweite* $h > 0$ *auf jedes beliebige Anfangswertproblem* (10.2) *mit* $\operatorname{Re}\lambda \le 0$ *angewandt wird. Ein Mehrschrittverfahren heißt A-stabil, wenn* $(y_n)_{n\ge0}$ *unter obigen Voraussetzungen für beliebig vorgegebene Startwerte beschränkt bleibt.*

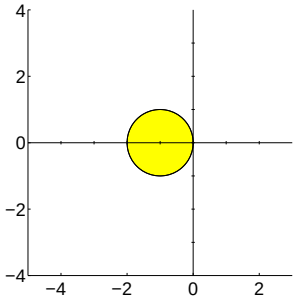

Abb. 10.3: Stabilitätsbereich des expliziten Euler-Verfahrens

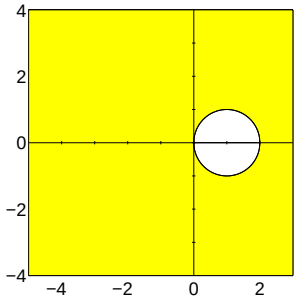

Abb. 10.4: Stabilitätsbereich des impliziten Euler-Verfahrens

Bemerkung. Das explizite Euler-Verfahren ist nicht A-stabil.
Beispiel. Das implizite Euler-Verfahren

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}), \qquad y_0 \text{ gegeben},$$

angewendet auf (10.2) lautet

$$y_{n+1} = y_n + h\lambda y_{n+1}.$$

Die numerische Lösung

$$y_n = \left(\frac{1}{1-h\lambda}\right)^n y_0$$

ist damit beschränkt für alle $n \geq 0$, wenn $1/|1-h\lambda| \leq 1$ gilt, insbesondere also für alle $\mathrm{Re}\,(h\lambda) \leq 0$, vgl. Abb. 10.4. Das implizite Euler-Verfahren ist also A-stabil. $\diamond$
Bemerkung. Ist A diagonalisierbar, $T^{-1}AT = \Lambda$, dann sind für praktisch alle Verfahren (Runge-Kutta-, Mehrschritt-, Extrapolations-Verfahren, ...) folgende Diagramme kommutativ:

$$\begin{array}{ccc} y' = Ay,\ y_0 & \xrightarrow{y=Tz} & z' = \Lambda z,\ z_0 = T^{-1}y_0 \\ \downarrow \text{num. Verfahren} & & \downarrow \text{num. Verfahren} \\ (y_n)_{n\geq 0} & \xrightarrow{y=Tz} & (z_n)_{n\geq 0} \end{array}$$

sowie für $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$

$$\begin{array}{ccc} y' = \Lambda y,\ y_0 & \xrightarrow{\text{Projektion}} & y_i' = \lambda_i y_i,\ y_{0,i} \\ \downarrow \text{num. Verfahren} & & \downarrow \text{num. Verfahren} \\ (y_n)_{n\geq 0} & \xrightarrow{\text{Projektion}} & (y_{n,i})_{n\geq 0} \end{array}$$

10.2 Stabilitätsbereiche, A-Stabilität

Um die Stabilität eines numerischen Verfahrens untersuchen zu können, definieren wir

Definition 10.2. *Als* **Stabilitätsbereich** *eines numerischen Verfahrens (Runge-Kutta-Verfahren, Mehrschrittverfahren, ...) bezeichnet man*

$$\mathcal{S} = \{z \in \mathbb{C} \mid z = h\lambda;\ \textit{das Verfahren liefert beschränkte Lösung } (y_n)_{n\geq 0},\ \textit{wenn es mit Schrittweite } h \textit{ auf } y' = \lambda y,\ y(0) = y_0 \textit{ angewandt wird.}\}$$

Die Definition ist natürlich nur sinnvoll, falls die numerische Lösung von (10.2) nur von $h\lambda$ abhängt. Dies ist für Runge-Kutta-Verfahren, Mehrschrittverfahren und Extrapolationsverfahren der Fall.

Bemerkung. Ein Verfahren ist A-stabil genau dann, wenn

$$\{z \in \mathbb{C} \mid \operatorname{Re} z \leq 0\} \subseteq \mathcal{S}$$

gilt, d. h. die abgeschlossene linke Halbebene im Stabilitätsbereich liegt.

Definition 10.3. *Ein numerisches Verfahren heißt* 0-**stabil** *genau dann, wenn* $0 \in \mathcal{S}$.

Ziel ist die Konstruktion numerischer Verfahren mit großem, unbeschränktem Stabilitätsbereich. Wir betrachten zunächst einige bekannte Verfahren:

Explizite Runge-Kutta-Verfahren

Ein allgemeines s-stufiges Runge-Kutta-Verfahren angewandt auf (10.1) ist von der Form

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} Y_j', \qquad Y_i' = f(t_0 + c_i h, Y_i),$$

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i Y_i'.$$

Bei expliziten Verfahren ist $a_{ij} = 0$ für $j \geq i$. Also erhalten wir für ein explizites Runge-Kutta-Verfahren angewandt auf die Testgleichung (10.2)

$$Y_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} Y_j', \qquad Y_i' = \lambda Y_i,$$

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i Y_i'$$

und damit

$$y_1 = P(h\lambda) y_0 \tag{10.3}$$

mit einem Polynom P vom Grad $\leq s$.

Bemerkung. Wir wissen, dass bei einem Verfahren der Ordnung p

$$P(z) = e^z + O(z^{p+1}), \qquad z \to 0$$

gilt. Im Spezialfall $p = s$ ist demnach

$$P(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^p}{p!},$$

P ist also unabhängig von den Koeffizienten a_{ij}, b_j, c_j des Runge-Kutta-Verfahrens.

Satz 10.4. *Der Stabilitätsbereich eines expliziten Runge-Kutta-Verfahrens ist beschränkt.*

Beweis. Wegen (10.3) ist

$$\mathcal{S} = \{z \in \mathbb{C} \mid |P(z)| \leq 1\}$$

der Stabilitätsbereich eines expliziten Runge-Kutta-Verfahrens. Dieser ist beschränkt, da $|P(z)| \to \infty$ für $z \to \infty$. □

Einige Beispiele sind in Abbildung 10.5 zu finden, weitere im Buch von Hairer & Wanner (1996, S. 17).

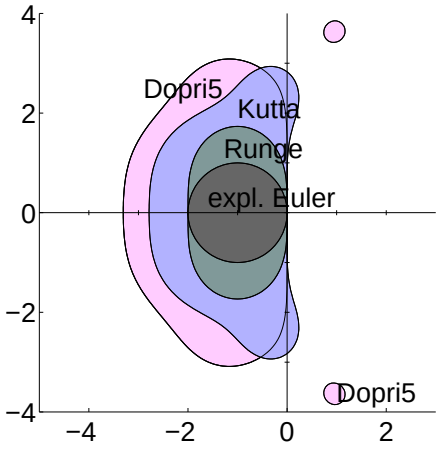

Abb. 10.5: Stabilitätsbereiche einiger expliziter Runge-Kutta-Verfahren

Gragg-Algorithmus (Extrapolations-Verfahren)

Man erhält auch hier für die Testgleichung (10.2)

$$T_{ij} = P_{ij}(h\lambda)y_0$$

mit Polynomen P_{ij}. Wie bei expliziten Runge-Kutta-Verfahren ist der Stabilitätsbereich damit beschränkt.

10.3 Mehrschrittverfahren

Ein allgemeines Mehrschrittverfahren ist von der Form

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \qquad f_{n+j} = f(t_{n+j}, y_{n+j}), \qquad \alpha_k \neq 0. \tag{10.4}$$

Angewandt auf die Testgleichung (10.2) ergibt sich $f_{n+j} = \lambda y_{n+j}$, also mit $z = h\lambda$

$$\sum_{j=0}^{k} (\alpha_j - z\beta_j) y_{n+j} = 0.$$

Die charakteristische Gleichung dieser homogenen Differenzengleichung ist

$$\sum_{j=0}^{k} (\alpha_j - z\beta_j)\zeta^j = \rho(\zeta) - z\sigma(\zeta) = 0 \tag{10.5}$$

mit

$$\rho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j, \qquad \sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j.$$

Wir nehmen an, dass $\mathrm{ggt}(\rho, \sigma) = 1$ gilt, anderenfalls können wir ein äquivalentes Verfahren mit kleinerem k angeben. Mehrschrittverfahren mit dieser Eigenschaft heißen irreduzibel.

Satz 10.5. *Der Stabilitätsbereich eines durch ρ und σ gegebenen Mehrschrittverfahrens ist*

$$\mathcal{S} = \{z \in \mathbb{C} \mid \begin{array}{l} \text{alle Nullstellen } \zeta(z) \text{ von (10.5) haben Betrag} \leq 1, \\ \text{mehrfache Nullstellen haben Betrag} < 1\}. \end{array}$$

Beweis. Satz 9.6. □

Lemma 10.6. *Es gilt*

$$\mathcal{S} \subseteq \mathbb{C} \setminus \left\{ \frac{\rho(\zeta)}{\sigma(\zeta)} \mid |\zeta| > 1 \right\}.$$

Beweis. Die erste Bedingung in Satz 10.5 ist äquivalent zu

$$\rho(\zeta) - z\sigma(\zeta) \neq 0 \quad \text{für} \quad |\zeta| > 1$$

oder

$$\begin{cases} z \neq \frac{\rho(\zeta)}{\sigma(\zeta)} \text{ für alle } \zeta \text{ mit } |\zeta| > 1 \text{ und } \sigma(\zeta) \neq 0, \\ \rho(\zeta) \neq 0 \text{ für alle Nullstellen } \zeta \text{ von } \sigma \text{ mit } |\zeta| > 1. \end{cases}$$

Da nach Voraussetzung $\operatorname{ggt}(\rho, \sigma) = 1$ gilt, ist dies äquivalent zu

$$z \neq \frac{\rho(\zeta)}{\sigma(\zeta)} \quad \text{für} \quad |\zeta| > 1,$$

was die behauptete Inklusion beweist. □

Es gilt sogar Gleichheit für alle wichtigen Verfahren (insbesondere für alle in dieser Vorlesung behandelten). In jedem Fall gilt aber

$$\mathcal{S} \supseteq \mathbb{C} \setminus \left\{ \frac{\rho(\zeta)}{\sigma(\zeta)} \mid |\zeta| \geq 1 \right\}.$$

Liegt $h\lambda$ in der Menge auf der rechten Seite, so konvergiert $y_n \to 0$ für $n \to \infty$.

Aus diesem Lemma ergibt sich sofort

Satz 10.7. *Ist $\sigma(\zeta) = 0$ für ein $|\zeta| > 1$, so ist $\mathcal{S}$ beschränkt.* □

Satz 10.8. *Der Stabilitätsbereich eines expliziten Mehrschrittverfahrens ist beschränkt.*

Beweis. Wir betrachten

$$\omega(\zeta) = \frac{\beta_k + \beta_{k-1}\zeta + \cdots + \beta_0\zeta^k}{\alpha_k + \alpha_{k-1}\zeta + \cdots + \alpha_0\zeta^k} = \frac{\zeta^k\sigma(\zeta^{-1})}{\zeta^k\rho(\zeta^{-1})}.$$

Für den Stabilitätsbereich gilt nach Lemma 10.6

$$\mathcal{S} \subseteq \mathbb{C} \setminus \left\{ \frac{\rho(\zeta)}{\sigma(\zeta)} \mid |\zeta| > 1 \right\} = \mathbb{C} \setminus \left\{ \frac{1}{\omega(\zeta)} \mid |\zeta| < 1 \right\}.$$

Bei expliziten Mehrschrittverfahren ist $\beta_k = 0$ und damit $\omega(0) = 0$. Da analytische Funktionen offene Abbildungen sind, ist $\{\omega(\zeta) \mid |\zeta| < 1\}$ eine Umgebung von 0, also $\{\frac{1}{\omega(\zeta)} \mid |\zeta| < 1\}$ eine Umgebung von unendlich. Damit ist $\mathcal{S}$ beschränkt. □

Die A-Stabilität eines Mehrschrittverfahrens lässt sich nun durch das Bild der Funktion $\frac{\rho(\zeta)}{\sigma(\zeta)}$ für ζ außerhalb des Einheitskreises charakterisieren:

Satz 10.9. *Ein irreduzibles Mehrschrittverfahren ist A-stabil genau dann, wenn*

$$\mathrm{Re}\left(\frac{\rho(\zeta)}{\sigma(\zeta)}\right) > 0 \quad \textit{für} \quad |\zeta| > 1. \tag{10.6}$$

Beweis. Ist das Mehrschrittverfahren A-stabil, so erfüllen alle Nullstellen ζ der charakteristischen Gleichung (10.5) $|\zeta| \leq 1$ für alle $\mathrm{Re}\, z \leq 0$. Dies ist äquivalent zu $\mathrm{Re}\, z > 0$ für alle $|\zeta| > 1$ und damit (10.6), da aus (10.5) $z = \rho(\zeta)/\sigma(\zeta)$ folgt.

Nehmen wir nun umgekehrt an, dass (10.6) gilt. Für ein fest gewähltes z_0 mit $\mathrm{Re}\, z_0 \leq 0$ sei ζ_0 eine Lösung der charakteristischen Gleichung (10.5). Wegen der Annahme $\mathrm{ggt}(\rho, \sigma) = 1$ muss $\sigma(\zeta_0) \neq 0$ gelten. Also ist $z_0 = \rho(\zeta_0)/\sigma(\zeta_0)$ und damit nach Voraussetzung $|\zeta_0| \leq 1$. Wir müssen noch zeigen, dass ζ_0 eine einfache Nullstelle ist, wenn $|\zeta_0| = 1$ gilt. Aus Stetigkeitsgründen folgt aus (10.6), dass $|\zeta_0| = 1$ und $\mathrm{Re}\, z_0 < 0$ unmöglich ist. Also bleibt noch zu zeigen, dass für $\mathrm{Re}\, z_0 = 0$ eine Nullstelle ζ_0 mit $|\zeta_0| = 1$ einfach ist. In der Umgebung einer solchen Nullstelle gilt

$$\frac{\rho(\zeta)}{\sigma(\zeta)} - z_0 = C_1(\zeta - \zeta_0) + C_2(\zeta - \zeta_0)^2 + \dots$$

Aus Voraussetzung (10.6) folgt $C_1 \neq 0$, ζ_0 ist also eine einfache Nullstelle. □

Da explizite Mehrschrittverfahren für steife Differentialgleichungen nicht geeignet sind, betrachten wir im Weiteren implizite Mehrschrittverfahren.

Implizite Adams-Verfahren

Zur Konstruktion impliziter Adams-Verfahren bestimmt man ein Interpolationspolynom q durch die Punkte $(t_n, f_n), \dots, (t_{n+k}, f_{n+k})$. In der Darstellung

$$y(t_{n+k}) = y(t_{n+k-1}) + \int_{t_{n+k-1}}^{t_{n+k}} f(s, y(s))ds$$

für die exakte Lösung ersetzt man nun den Integranden mit der unbekannten Funktion y durch das Interpolationspolynom q. Dies führt mit Hilfe der Newton'schen Interpolationsformel auf

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

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. Implizite Adams-Verfahren haben nach Konstruktion die Ordnung $k+1$.

Eingesetzt in die Testgleichung (10.2) ergibt sich wieder $f_{n+j} = \lambda y_{n+j}$. Setzen wir $y_m = \zeta^m$, so erhalten wir wegen

$$\nabla^j \zeta^k = \zeta^k \left(1 - \frac{1}{\zeta}\right)^j$$

die Polynome

$$\rho(\zeta) = \zeta^k \left(1 - \frac{1}{\zeta}\right), \qquad \sigma(\zeta) = \zeta^k \sum_{j=0}^{k} \gamma_j^* \left(1 - \frac{1}{\zeta}\right)^j$$

(Übung).

Die Koeffizienten der impliziten Adams-Verfahren sind durch

$$\gamma_j^* = (-1)^j \int_0^1 \binom{-s+1}{j} ds,$$

gegeben, vgl. Abschnitt 9.1. Für $0 \le k \le 2$ haben die impliziten Adams-Verfahren folgende Form:

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

Für $k = 0$ entspricht das implizite Adams-Verfahren dem impliziten Euler-Verfahren, von dem wir bereits gezeigt haben, dass es A-stabil ist. Für $k = 1$ erhält man die Trapezregel mit

$$\frac{\rho(\zeta)}{\sigma(\zeta)} = 2\,\frac{\zeta - 1}{\zeta + 1}.$$

Nach Satz 10.9 ist die Trapezregel also A-stabil. Für $k = 2$ ist

$$\sigma(\zeta) = \frac{5}{12}\zeta^2 + \frac{8}{12}\zeta - \frac{1}{12}.$$

Die Nullstellen von σ sind

$$\zeta_{1,2} = \frac{-8 \pm \sqrt{64+20}}{10} = -\frac{4}{5} \pm \frac{\sqrt{21}}{5}.$$

Es ist also $|\zeta_2| > 1$ und wegen Satz 10.7 der Stabilitätsbereich beschränkt.

Ebenso kann man zeigen, dass die impliziten Adams-Verfahren für alle $k \ge 2$ einen beschränkten Stabilitätsbereich haben und damit für steife Differentialgleichungen ungeeignet sind. Die Stabilitätsbereiche einiger impliziter Adams-Verfahren findet man in Abbildung 10.6.

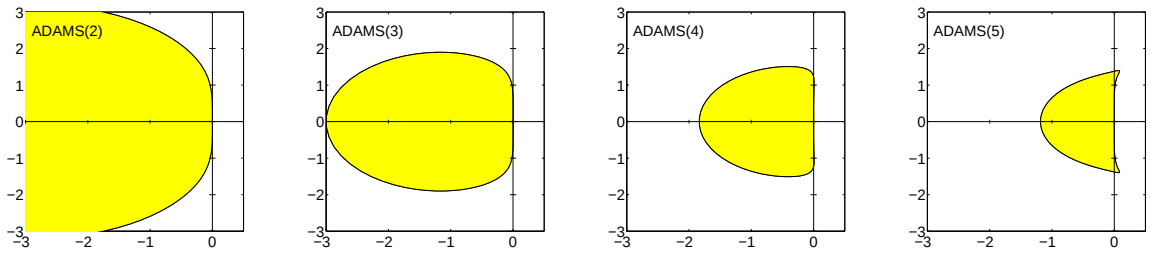

Abb. 10.6: Stabilitätsbereiche von impliziten Adams-Verfahren für $k = 2, 3, 4, 5$.

BDF-Verfahren

BDF-Verfahren ("backward differentiation formula") sind Mehrschrittverfahren, bei denen ein Interpolationspolynom q durch $(t_n, y_n), (t_{n+1}, y_{n+1}), \ldots, (t_{n+k}, y_{n+k})$ konstruiert wird, welches zusätzlich zur Zeit t_{n+k} die Differentialgleichung löst: $q'(t_{n+k}) = f(t_{n+k}, y_{n+k})$. Es ergibt sich ein Verfahren der Form

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

Die Koeffizienten α_j können durch Rückwärtsdifferenzen ausgedrückt werden, so dass man das BDF-Verfahren auch als

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

schreiben kann. Daraus erhält man

$$\rho(\zeta) = \zeta^k \sum_{j=1}^{k} \frac{1}{j} \left(1 - \frac{1}{\zeta}\right)^j, \qquad \sigma(\zeta) = \zeta^k \tag{10.7}$$

(Übung). Für $k > 6$ ist BDF(k) nicht 0-stabil. BDF(1) ist das implizite Euler-Verfahren, BDF(2) hat die Form:

$$\frac{3}{2} y_{n+1} - 2 y_n + \frac{1}{2} y_{n-1} = h f_{n+1}$$

Auch BDF(2) ist A-stabil, denn setzen wir $\zeta = r e^{i\theta}$ mit $r > 1$, so gilt

$$\begin{aligned}
\operatorname{Re} \frac{\rho(\zeta)}{\sigma(\zeta)} &= \operatorname{Re} \left(\frac{3}{2} - \frac{2}{\zeta} + \frac{1}{2\zeta^2} \right) \\
&= \operatorname{Re} \left(\frac{3}{2} - \frac{2}{r} e^{-i\theta} + \frac{1}{2r^2} e^{-2i\theta} \right) \\
&= \frac{3}{2} - \frac{2}{r} \cos\theta + \frac{1}{2r^2} \underbrace{\cos 2\theta}_{2\cos^2\theta - 1} \\
&= \left(1 - 2\frac{\cos\theta}{r} + \frac{\cos^2\theta}{r^2} \right) + \frac{1}{2} - \frac{1}{2r^2} \\
&= \left(1 - \frac{\cos\theta}{r} \right)^2 + \frac{1}{2} \left(1 - \frac{1}{r^2} \right) \\
&> 0.
\end{aligned}$$

Die A-Stabilität folgt nun aus Satz 10.9.

Wegen (10.7) ist $\frac{\rho(\zeta)}{\sigma(\zeta)}$ ein Polynom vom Grad k in ζ^{-1}. Dieses ist beschränkt für $|\zeta| > 1$, der Stabilitätsbereich also unbeschränkt. BDF(k) ist jedoch für $k = 3, \ldots, 6$ nicht A-stabil, sondern nur $A(\alpha)$-stabil:

Definition 10.10. *Ein Verfahren mit Stabilitätsbereich $\mathcal{S}$ heißt* $A(\alpha)$-**stabil**, $0 < \alpha < \pi/2$, *wenn*

$$\{z \in \mathbb{C} \mid |\arg(-z)| \le \alpha\} \subseteq \mathcal{S}.$$

BDF(k)-Verfahren, $k \leq 6$ sind $A(\alpha)$-stabil mit folgenden Werten für α:

k	1	2	3	4	5	6
α	90°	90°	86°	73°	51°	18°

Für $k \geq 3$ erwartet man also schlechte Ergebnisse, wenn die Eigenwerte von $\frac{\partial f}{\partial y}$ nahe an der imaginären Achse liegen. Die Stabilitätsbereiche sind in Abbildung 10.7 abgebildet.

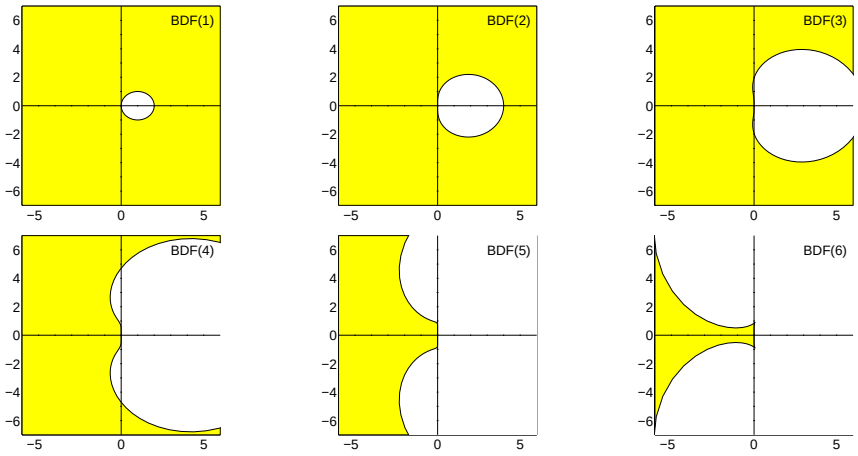

Abb. 10.7: Stabilitätsbereiche von BDF(1), ..., BDF(6)

BDF-Verfahren sind die derzeit am meisten verwendeten Methoden für steife Differentialgleichungen. Es existieren ausgereifte Programmpakete, z. B. DIFSUB von Gear (1968), LSODE von Hindmarsh 1975-1990, VODE und DASSL, die über Netlib erhältlich sind. In der Regel verfügen sie über eine Schrittweiten- und Ordnungssteuerung ($k \leq 5$), ähnlich wie wir sie bei Adams-Verfahren in Abschnitt 9.7 kennengelernt haben.

Im Allgemeinen müssen wir in jedem Zeitschritt ein nichtlineares Gleichungssystem

$$y_{n+k} - \frac{h}{\alpha_k} f(t_{n+k}, y_{n+k}) = -\sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} y_{n+j}$$

lösen. Fixpunktiteration

$$y_{n+k}^{(m+1)} = \frac{h}{\alpha_k} f(t_{n+k}, y_{n+k}^{(m)}) - \sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} y_{n+j}$$

konvergiert bekanntlich, falls die Abbildung kontraktiv ist, also falls $\frac{hL}{\alpha_k} < 1$ mit der Lipschitzkonstanten L von f gilt. Dies führt zu Schrittweitenbeschränkungen proportional zu $1/L$, was Fixpunktiteration für steife Differentialgleichungen (L groß) ungeeignet macht. Besser ist es, die nichtlinearen Gleichungssysteme mit einem vereinfachten Newton-Verfahren zu lösen. Das Newton-Verfahren zur Lösung von $F(y) = 0$ hat die Form

$$y^{(m+1)} = y^{(m)} - F'(y^{(m)})^{-1} F(y^{(m)}), \qquad y^{(0)} \text{ gegeben.}$$

Hierbei muss in jedem Newton-Schritt die Jacobi-Matrix F' an der aktuellen Näherung $y^{(m)}$ neu ausgewertet werden. Hingegen verwendet man beim vereinfachten Newton-Verfahren nur eine Jacobi-Matrix:

$$y^{(m+1)} = y^{(m)} - F'(y^{(0)})^{-1} F(y^{(m)}), \qquad y^{(0)} \text{ gegeben.}$$

In jedem Schritt muss ein lineares Gleichungssystem der Form

$$F'(y^{(0)})\Delta y^{(m)} = -F(y^{(m)}), \qquad y^{(m+1)} = y^{(m)} + \Delta y^{(m)}$$

gelöst werden. In unserem Fall ist

$$F(y) = y - \frac{h}{\alpha_k} f(t_{n+k}, y) + \sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} y_{n+j}, \qquad F'(y) = I - \frac{h}{\alpha_k} \frac{\partial f}{\partial y}(t_{n+k}, y).$$

Man berechnet nun iterativ

$$y_{n+k}^{(m+1)} = y_{n+k}^{(m)} + \Delta y_{n+k}^{(m)}, \qquad y_{n+k}^{(0)} = y_{n+k-1},$$

wobei $\Delta y_{n+k}^{(m)}$ durch Lösen des linearen Gleichungssystems

$$(I - \frac{h}{\alpha_k} J)\Delta y_{n+k}^{(m)} = -y_{n+k}^{(m)} + \frac{h}{\alpha_k} f(t_{n+k}, y_{n+k}^{(m)}) - \sum_{j=0}^{k-1} \frac{\alpha_j}{\alpha_k} y_{n+j}, \qquad J \approx \frac{\partial f}{\partial y}(t_{n+k}, y_{n+k-1})$$

berechnet wird. Oft sind nur wenige (ein bis drei) Iterationen ausreichend, um

$$\|y_{n+k}^{(m+1)} - y_{n+k}^{(m)}\| \leq \text{ lokaler Fehler des BDF-Verfahrens}$$

zu erreichen. Eine höhere Genauigkeit bei der Lösung der nichtlinearen Gleichungen ist nicht sinnvoll.

Eine weitere Vereinfachung erreicht man, indem man die Jacobi-Matrix nicht nur bei allen Newton-Schritten eines Zeitschritts, sonderen gleich über mehrere Zeitschritte festhält. Solange auch die Schrittweite konstant bleibt, kann man dann die LU-Zerlegung von $(I - \frac{h}{\alpha_k} J)$ weiterverwenden. Ist die LU-Zerlegung bekannt, so müssen in jedem Newton-Iterationsschritt zwei Dreieckssysteme gelöst sowie die Funktion f einmal ausgewertet werden.

10.4 Ordnungsschranke für A-stabile Mehrschrittverfahren

Alle bisher genannten A-stabilen Mehrschrittverfahren haben Ordnung $p \leq 2$. Wir werden nun zeigen, dass dies für alle Mehrschrittverfahren gilt.

Erinnern wir uns zunächst, dass nach Satz 9.4 ein Mehrschrittverfahren die Ordnung p mit der Fehlerkonstante $C = C_{p+1}/\sigma(1)$ hat, wenn

$$\rho(e^h) - h\sigma(e^h) = C_{p+1} h^{p+1} + \ldots, \qquad h \to 0 \tag{10.8}$$

gilt. Ein Verfahren heißt konsistent, wenn es mindestens die Ordnung eins hat. In diesem Fall folgt aus (10.8)

$$\rho(1) = 0, \quad \rho'(1) = \sigma(1)$$

und damit

$$\rho(e^h) = \rho(1 + h + \ldots) = h\sigma(1) + \ldots, \qquad h \to 0. \tag{10.9}$$

Tatsächlich gilt nun

Satz 10.11. (*Dahlquist* 1963)
Für die Ordnung p eines A-stabilen Mehrschrittverfahrens gilt $p \le 2$. Ist die Ordnung 2, so gilt für die Fehlerkonstante $C \le -1/12$. Die Trapezregel ist das einzige A-stabile Mehrschrittverfahren der Ordnung 2 mit $C = -1/12$.

Beweis. Teilen wir die Ordnungsbedingung (10.8) durch $h\rho(e^h)$, so erhalten wir wegen (10.9)

$$\frac{1}{h} - \frac{\sigma(e^h)}{\rho(e^h)} = \frac{C_{p+1}}{\sigma(1)} h^{p-1} + \dots, \qquad h \to 0.$$

Mit $\zeta = e^h$ ist dies äquivalent zu

$$\frac{1}{\log \zeta} - \frac{\sigma(\zeta)}{\rho(\zeta)} = C(\zeta - 1)^{p-1} + \dots, \qquad \zeta \to 1,$$

wobei C die Fehlerkonstante des Verfahrens ist. In dieser Formel setzen wir $p = 2$. Ist das Mehrschrittverfahren von höherer Ordnung, so gilt $C = 0$. Ist die Ordnung eins, so ist nichts zu zeigen. Im Spezialfall der Trapezregel gilt

$$\rho_T(\zeta) = \zeta - 1, \qquad \sigma_T(\zeta) = \frac{1}{2}(\zeta + 1)$$

woraus sich durch Taylor-Entwicklung

$$\frac{1}{\log \zeta} - \frac{\sigma_T(\zeta)}{\rho_T(\zeta)} = -\frac{1}{12}(\zeta - 1) + \dots, \qquad \zeta \to 1$$

ergibt. Wir subtrahieren nun die beiden Entwicklungen voneinander und erhalten

$$\delta(\zeta) := \frac{\sigma(\zeta)}{\rho(\zeta)} - \frac{\sigma_T(\zeta)}{\rho_T(\zeta)} = \left(-C - \frac{1}{12}\right)(\zeta - 1) + \dots, \qquad \zeta \to 1. \tag{10.10}$$

Nun wissen wir aus Satz 10.9, dass für ein A-stabiles Mehrschrittverfahren gilt

$$\operatorname{Re}\left(\frac{\sigma(\zeta)}{\rho(\zeta)}\right) > 0 \quad \text{für} \quad |\zeta| > 1.$$

Für die Trapezregel ist $\operatorname{Re}(\dots) = 0$ für $|\zeta| = 1$, da der Stabilitätsbereich der Trapezregel genau $\mathbb{C}^-$ ist. Wir können also

$$\lim_{\substack{\zeta \to \zeta_0 \\ |\zeta| > 1}} \operatorname{Re} \delta(\zeta) \ge 0 \qquad \text{für} \qquad |\zeta_0| = 1. \tag{10.11}$$

schließen. Die Pole von $\delta(\zeta)$ sind gerade die Nullstellen von $\rho(\zeta)$, die wegen der A-Stabilität nicht außerhalb des Einheitskreises liegen. Nach dem Maximumprinzip gilt (10.11) damit überall außerhalb des Einheitskreises. Wählen wir $\zeta = 1 + \varepsilon$ mit $\operatorname{Re}\varepsilon > 0$ und $|\varepsilon|$ klein, dann ergibt sich aus (10.10), dass entweder $-C - 1/12 > 0$ oder $\delta(\zeta) \equiv 0$. Dies zeigt die letzte Behauptung. □

Nach diesem negativen Resultat wollen wir nun die Frage nach A-stabilen Runge-Kutta-Verfahren höherer Ordnung positiv beantworten.

10.5 Kollokationsverfahren

Wir betrachten wieder das Anfangswertproblem

$$y' = f(t,y), \qquad y(t_0) = y_0.$$

Es seien $c_1 < \cdots < c_s \in [0,1]$ sowie eine Schrittweite h gegeben. Gesucht ist ein Polynom u vom Grad s, welches die Kollokationsbedingungen

$$\begin{aligned} u(t_0) &= y_0 \\ u'(t_0 + c_i h) &= f(t_0 + c_i h, u(t_0 + c_i h)), \qquad i = 1, \ldots, s \end{aligned}$$

erfüllt. (Das Kollokationspolynom u hat die richtigen Steigungen an den Kollokationspunkten $t_0 + c_i h$, $i = 1, \ldots, s$ und den richtigen Anfangswert $u(t_0)$.) Mit diesem Polynom setzen wir dann $y_1 = u(t_0 + h)$.

Satz 10.12. *Das Kollokationsverfahren ist äquivalent zum impliziten Runge-Kutta-Verfahren*

$$\begin{array}{c|c} c_i & a_{ij} \\ \hline & b_j \end{array}, \qquad i,j = 1, \ldots, s$$

mit den Koeffizienten

$$a_{ij} = \int_0^{c_i} \ell_j(x)dx, \qquad b_j = \int_0^1 \ell_j(x)dx,$$

wobei

$$\ell_j(x) = \frac{\prod_{k \neq j}(x - c_k)}{\prod_{k \neq j}(c_j - c_k)}$$

das j-te Lagrange-Polynom zu den Knoten $c_1, \ldots, c_s$ ist.

Bemerkung. $\ell_j(x)$ ist ein Polynom vom Grad $\leq s - 1$ mit

$$\ell_j(c_k) = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases}$$

Beweis. Setzen wir $Y_i' = u'(t_0 + c_i h)$ und $Y_i = u(t_0 + c_i h)$, so folgt aus der Kollokationsbedingung $Y_i' = f(t_0 + c_i h, Y_i)$. Da u' ein Polynom vom Grad $s - 1$ ist, stimmt es mit dem Interpolationspolynom in den s Knoten $t_0 + c_i h$ überein, d. h.

$$u'(t_0 + xh) = \sum_{j=1}^{s} \ell_j(x) Y_j'$$

(Lagrange-Interpolation). Außerdem erhalten wir

$$\begin{aligned} Y_i = u(t_0 + c_i h) &= y_0 + h \int_0^{c_i} u'(t_0 + xh)dx \\ &= y_0 + h \sum_{j=1}^{s} \underbrace{\int_0^{c_i} \ell_j(x)dx}_{a_{ij}} Y_j' \end{aligned}$$

und daraus mit

$$\begin{aligned} y_1 = u(t_0+h) &= y_0 + h\int_0^1 u'(t_0+xh)dx \\ &= y_0 + h\sum_{j=1}^s \underbrace{\int_0^1 \ell_j(x)dx}_{b_j}\, Y_j' \end{aligned}$$

die Behauptung. □

Bemerkung. Für genügend kleines h existiert das Kollokationspolynom u eindeutig.

Wir zeigen zunächst, dass die Koeffizienten a_{ij}, b_j von Satz 10.12

$$\sum_{j=1}^s b_j c_j^{k-1} = \frac{1}{k}, \qquad \sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \qquad \text{für } i,k = 1,\dots,s.$$

erfüllen. Es gilt

$$\sum_{j=1}^s \ell_j(x) c_j^{k-1} = x^{k-1} \qquad \text{falls} \qquad k \le s,$$

denn die linke Seite ist das Interpolationspolynom zur Funktion x^{k-1} in den Knoten c_j, $j = 1,\dots,s$ und stimmt daher mit x^{k-1} überein, falls $k \le s$. Daraus folgt

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \sum_{j=1}^s \int_0^{c_i} \ell_j(x)dx c_j^{k-1} = \int_0^{c_i} x^{k-1}dx = \frac{1}{k} c_i^k, \qquad k = 1,\dots,s \tag{10.12}$$

und analog

$$\sum_{j=1}^s b_j c_j^{k-1} = \sum_{j=1}^s \int_0^1 \ell_j(x)dx c_j^{k-1} = \int_0^1 x^{k-1}dx = \frac{1}{k}, \qquad k = 1,\dots,s.$$

Für gegebene Knoten $c_1,\dots,c_s$ können die Koeffizienten a_{ij}, b_j aus obigem linearen Gleichungssystem berechnet werden.

Satz 10.13. *Das Kollokationsverfahren hat dieselbe Ordnung wie die zugehörige Quadraturformel, d. h. das implizite Runge-Kutta-Verfahren aus Satz 10.12 hat die Ordnung p genau dann, wenn die Quadraturformel*

$$\int_0^1 g(x)dx \approx \sum_{j=1}^s b_j g(c_j)$$

die Ordnung p hat, also exakt ist für alle Polynome bis zum Grad $p-1$.

Bevor wir diesen Satz beweisen, wiederholen wir zunächst den Begriff der Resolvente. Wir beginnen mit einer homogenen linearen Differentialgleichung

$$y'(t) = A(t)y(t), \qquad y(t_0) = y_0$$

deren Lösung *linear* vom Anfangswert abhängt:

$$y(t) = R(t,t_0)y_0, \qquad R(t,t_0) \in \mathbb{R}^{d,d} \text{ Resolvente.}$$

Insbesondere gilt $R(t_0, t_0) = I$. Durch Differentiation nach t erhält man aus der Differentialgleichung sofort

$$y'(t) = \frac{\partial R}{\partial t}(t, t_0)y_0 = A(t)y(t) = A(t)R(t, t_0)y_0. \tag{10.13}$$

Ist A konstant, so gilt $R(t, t_0) = e^{A(t-t_0)}$. Die Lösung der inhomogenen Differentialgleichung

$$y'(t) = A(t)y(t) + g(t), \qquad y(t_0) = y_0$$

ist durch die *Variation der Konstanten-Formel*

$$y(t) = R(t, t_0)y_0 + \int_{t_0}^{t} R(t, s)g(s)ds \tag{10.14}$$

gegeben, denn wegen $R(s, s) = I$ für alle s gilt $y(t_0) = y_0$ und aus (10.13) folgt

$$\begin{aligned} y'(t) &= \frac{\partial R}{\partial t}(t, t_0)y_0 + R(t, t)g(t) + \int_{t_0}^{t} \frac{\partial R}{\partial t}(t, s)g(s)ds \\ &= A(t)R(t, t_0)y_0 + g(t) + \int_{t_0}^{t} A(t)R(t, s)g(s)ds \\ &= A(t)y(t) + g(t). \end{aligned}$$

Beweis von Satz 10.13. Ohne Einschränkung sei f autonom, d. h. $y' = f(y)$. Es gilt

$$u' = f(u) + \delta(t) \qquad \text{mit} \qquad \delta(t_0 + c_i h) = 0, \quad i = 1, \ldots, s.$$

Wir betrachten die Homotopie

$$z' = f(z) + \tau\delta(t), \qquad 0 \le \tau \le 1 \text{ Parameter} \tag{10.15}$$

mit Anfangswert $z(t_0) = y_0$. Die Lösung bezeichnen wir mit $z(t, \tau)$. Es gilt dann $z(t, 0) = y(t)$ und $z(t, 1) = u(t)$, also

$$u(t) - y(t) = z(t, 1) - z(t, 0) = \int_0^1 \frac{\partial z}{\partial \tau}(t, \tau)d\tau. \tag{10.16}$$

Zur Berechnung von $\frac{\partial z}{\partial \tau}$ differenzieren wir (10.15) nach τ und erhalten

$$\left(\frac{\partial z}{\partial \tau}\right)' = f'(z(t, \tau))\frac{\partial z}{\partial \tau} + \delta(t) =: A_\tau(t)\frac{\partial z}{\partial \tau} + \delta(t), \qquad \frac{\partial z}{\partial \tau}(t_0, \tau) = 0.$$

Die Variation der Konstanten-Formel (10.14) liefert

$$\frac{\partial z}{\partial \tau}(t, \tau) = \int_{t_0}^{t} R_\tau(t, s)\delta(s)ds.$$

Durch Einsetzen in (10.16) und Vertauschen der Integrale ergibt sich

$$\begin{aligned} u(t_0 + h) - y(t_0 + h) &= \int_{t_0}^{t_0+h} \underbrace{\int_0^1 R_\tau(t_0 + h, s)d\tau\delta(s)}_{=: g(s)} ds \\ &= h\sum_{j=1}^{s} b_j g(t_0 + c_j h) + O(h^{p+1}) \end{aligned}$$

wobei wir für die zweite Gleichung verwendet haben, dass die Quadraturformel die Ordnung p hat. Ferner gilt $g(t_0 + c_j h) = 0$, da aus der Kollokationsbedingung $\delta(t_0 + c_j h) = 0$ folgt. Es bleibt noch ein technisches Detail zu zeigen, nämlich dass $\|\delta^{(p+1)}\|$ gleichmäßig beschränkt ist. Da δ lediglich von u und $f(y)$ abhängt, genügt es dafür zu zeigen, dass alle Ableitungen von u beschränkt bleiben. Dies zeigen wir im folgenden Lemma. □

Lemma 10.14. *Für das Kollokationspolynom gilt*

$$\|y^{(k)}(t) - u^{(k)}(t)\| \le C h^{s+1-k}, \qquad k = 0, \dots, s.$$

Beweis. Die exakte Lösung y erfüllt überall die Kollokationsbedingung, insbesondere in den Kollokationspunkten $t_0 + c_i h$. Mit Hilfe der Lagrange-Interpolationsformel folgt wie im Beweis von Satz 10.12

$$y'(t_0 + th) = \sum_{j=1}^{s} f(t_0 + c_j h, y(t_0 + c_j h))\ell_j(t) + h^s r(t, h)$$

mit einer Funktion r die glatt in t und h ist. Durch Integration der Differentialgleichung ergibt sich

$$y(t_0 + th) = y(t_0) + h \int_0^t y'(t_0 + \tau h) d\tau,$$

also

$$y(t_0 + th) - u(t_0 + th) = h \sum_{j=1}^{s} \Delta f_j \int_0^t \ell_j(\tau) d\tau + h^{s+1} \int_0^t r(\tau, h) d\tau$$

mit

$$\Delta f_j = f(t_0 + c_j h, y(t_0 + c_j h)) - f(t_0 + c_j h, u(t_0 + c_j h)).$$

Für die k-te Ableitung nach t gilt somit

$$h^k \left(y^{(k)}(t_0 + th) - u^{(k)}(t_0 + th) \right) = h \sum_{j=1}^{s} \Delta f_j \ell_j^{(k-1)}(t) + h^{s+1} \frac{\partial^{k-1}}{\partial t^{k-1}} r(t, h).$$

Der letzte Summand auf der rechten Seite ist beschränkt. Wir zeigen jetzt noch $\|\Delta f_j\| = O(h^{s+1})$. Wegen (10.12) ist

$$y(t_0 + c_i h) = y_0 + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, y(t_0 + c_j h)) + O(h^{s+1})$$

und nach Definition des Kollokationspolynoms gilt

$$u(t_0 + c_i h) = y_0 + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, u(t_0 + c_j h)).$$

Subtraktion der beiden Gleichungen liefert

$$y(t_0 + c_i h) - u(t_0 + c_i h) = h \sum_{j=1}^{s} a_{ij} \Delta f_j + O(h^{s+1}). \tag{10.17}$$

Aus der Lipschitz-Bedingung an f folgt

$$\|\Delta f_i\| \le L\|y(t_0 + c_i h) - u(t_0 + c_i h)\|$$

und aus (10.17) damit

$$\|y(t_0 + c_i h) - u(t_0 + c_i h)\| = O(h^{s+1}).$$

Also ist $\|\Delta f_i\| = O(h^{s+1})$. □

Bemerkung. Das fehlende technische Detail aus dem Beweis von Satz 10.13 ist durch dieses Lemma gezeigt, denn aus der hinreichenden Glattheit der Lösung y folgt mit der Dreiecksungleichung die Beschränktheit aller Ableitungen von u. (Höhere Ableitungen von u verschwinden, da u ein Polynom vom Grad s ist). Ein völlig anderer Beweis ist direkt durch Verifikation der Ordnungsbedingungen mit Hilfe von (10.12) möglich (Übung). Allerdings kann man daraus im Gegensatz zu dem hier vorgeführten Beweis den Hintergrund der Aussage nicht erkennen.

10.6 *A*-stabile Runge-Kutta-Verfahren

Wir wollen nun den Stabilitätsbereich von Runge-Kutta-Verfahren bestimmen. Dazu wenden wir ein (implizites) Runge-Kutta-Verfahren auf die Testgleichung $y' = \lambda y$, $y(t_0) = y_0$ an:

$$Y_i = y_0 + h\lambda \sum_{j=1}^{s} a_{ij} Y_j, \qquad i = 1, \dots, s,$$

$$y_1 = y_0 + h\lambda \sum_{i=1}^{s} b_i Y_i.$$

In Matrixschreibweise lauten die Relationen für die inneren Stufen

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix} = y_0 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} + z\mathcal{A} \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix}, \qquad \mathcal{A} = (a_{ij})_{i,j=1}^{s}, \quad z = h\lambda.$$

Definieren wir

$$\mathbb{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_s \end{bmatrix}, \qquad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix},$$

so ist dies äquivalent zu

$$(I - z\mathcal{A})Y = y_0 \mathbb{1}.$$

Unter der Voraussetzung, dass $(I - z\mathcal{A})$ invertierbar ist, d. h. z^{-1} kein Eigenwert von $\mathcal{A}$ ist, gilt

$$y_1 = y_0 + zb^T Y = [1 + zb^T (I - z\mathcal{A})^{-1}\mathbb{1}]y_0 =: R(z)y_0$$

mit der **Stabilitätsfunktion**

$$R(z) = \frac{P(z)}{Q(z)}, \qquad \deg P, \deg Q \le s.$$

Genauer gilt

$$R(z) = \frac{\det(I - z\mathcal{A} + z\mathbb{1}b^T)}{\det(I - z\mathcal{A})}.$$

(Übung). Damit ist

$$\mathcal{S} = \{z \in \mathbb{C} \mid (I - z\mathcal{A}) \text{ invertierbar und } |R(z)| \leq 1\}$$

der Stabilitätsbereich eines impliziten Runge-Kutta-Verfahrens. Ein implizites Runge-Kutta-Verfahren ist also A-stabil genau dann, wenn

$$|R(z)| \leq 1 \qquad \text{für} \qquad \operatorname{Re} z \leq 0.$$

Gauß-Kollokationsverfahren

Wegen Satz 10.13 erreicht man die maximale Ordnung $p = 2s$ eines Kollokationsverfahrens mit s Knoten, indem man eine Gauß-Quadraturformel verwendet. Wir wollen einige Beispiele angeben:

$s = 1$: $c_1 = \frac{1}{2}$, $a_{11} = \frac{1}{2}$, $b_1 = 1$

$$Y_1 = y_0 + \frac{h}{2} Y_1', \qquad Y_1' = f(t_0 + \frac{h}{2}, Y_1),$$
$$y_1 = y_0 + hY_1'.$$

Dieses Runge-Kutta-Verfahren ist äquivalent zur impliziten Mittelpunktsregel

$$y_1 = y_0 + hf(t_0 + \frac{h}{2}, \frac{y_0 + y_1}{2}).$$

Wegen $s = 1$ hat die implizite Mittelpunktsregel die Ordnung 2. Angewendet auf die Testgleichung $y' = \lambda y$ erhält man mit $z = h\lambda$

$$y_1 = y_0 + \frac{z}{2}(y_0 + y_1) \iff y_1 = \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}} y_0.$$

Das Verfahren ist also A-stabil, denn die Stabilitätsfunktion ist die gleiche wie bei der Trapezregel, die ja gerade das implizite Adams-Verfahren mit $k = 2$ ist.

$s = 2$: $c_{1,2} = \frac{1}{2} \mp \frac{\sqrt{3}}{6}$, $b_1 = b_2 = \frac{1}{2}$. Aus den Ordnungsbedingungen (10.12), $\sum a_{ij} = c_i$ und $\sum a_{ij} c_j = \frac{c_i^2}{2}$, ergibt sich das Runge-Kutta-Verfahren

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

der Ordnung 4. Eine einfache Rechnung liefert die Stabilitätsfunktion

$$R(z) = \frac{1 + \frac{z}{2} + \frac{z^2}{12}}{1 - \frac{z}{2} + \frac{z^2}{12}}.$$

Um die Frage nach der A-Stabilität (also $|R(z)| \leq 1$ für $\operatorname{Re} z \leq 0$) zu beantworten, betrachten wir

$$|R(iy)|^2 = \frac{\left(1 - \frac{y^2}{12}\right)^2 + \frac{y^2}{4}}{\left(1 - \frac{y^2}{12}\right)^2 + \frac{y^2}{4}} = 1.$$

Die Pole von $R(z)$ sind die Nullstellen von $z^2 - 6z + 12$, also $z_{1,2} = 3 \pm \sqrt{9-12}$. Beide Pole haben offensichtlich positiven Realteil, so dass aus dem Maximumprinzip folgt, dass $|R(z)| \leq 1$ sogar für alle $\operatorname{Re} z \leq 0$ gilt. Auch dieses Gauß-Kollokationsverfahren ist damit A-stabil.

Man kann sogar zeigen, dass alle Gauß-Kollokationsverfahren A-stabil sind, denn ihre Stabilitätsfunktionen sind (s,s) Padé-Approximationen an die Exponentialfunktion. Mit Hilfe von Ordnungssternen kann man elegant zeigen, dass (k,j) Padé-Approximationen genau dann A-stabil sind, wenn $k \leq j \leq k+2$. Gauß-Kollokationsverfahren haben jedoch den Nachteil, dass wegen

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

sehr steife Komponenten nicht gedämpft werden.

Radau-Kollokationsverfahren

Radau-Quadraturformeln sind die eindeutig bestimmten Verfahren der Ordnung $2s-1$ mit dem Knoten $c_s = 1$. Wir betrachten auch hier einige Beispiele.

$s = 1$: $c_1 = 1$, $a_{11} = 1$, $b_1 = 1$ ist gerade das wohlbekannte implizite Euler-Verfahren. Es hat Ordnung eins und ist A-stabil.

$s = 2$: $c_1 = \frac{1}{3}$, $c_2 = 1$ ergibt das Runge-Kutta-Verfahren Radau IIA

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

der Ordnung 3. Man beachte, dass die letzten beiden Zeilen des Schemas identisch sind. Die Stabilitätsfunktion ist

$$R(z) = \frac{1 + \frac{z}{3}}{1 - \frac{2z}{3} + \frac{z^2}{6}}.$$

Um die A-Stabilität zu zeigen, betrachten wir $R(z)$ wieder auf der imaginären Achse:

$$|R(iy)|^2 = \frac{1 + \frac{y^2}{9}}{\left(1 - \frac{y^2}{6}\right)^2 + \frac{4}{9}y^2} = \frac{1 + \frac{y^2}{9}}{1 + \frac{y^2}{9} + \frac{y^4}{36}} \leq 1$$

Da die Pole von $R(z)$, $z_{1,2} = 2 \pm \sqrt{4-6}$, positiven Realteil haben, folgt aus dem Maximumprinzip die A-Stabilität. Das Stabilitätsgebiet ist in Abbildung 10.8 zu sehen.

Allgemein folgt aus Satz 10.12 wegen $c_s = 1$

$$a_{sj} = b_j, \qquad j = 1, \ldots, s \qquad \text{oder} \qquad b^T = e_s^T \mathcal{A},$$

wobei e_s der s-te Einheitsvektor ist. Bei invertierbarem $\mathcal{A} = (a_{ij})$ folgt daraus

$$\lim_{z\to\infty} R(z) = \lim_{z\to\infty} [1 + b^T (\tfrac{1}{z} I - \mathcal{A})^{-1} \mathbb{1}] = 1 - b^T \mathcal{A}^{-1} \mathbb{1} = 1 - e_s^T \mathcal{A}\mathcal{A}^{-1}\mathbb{1} = 0,$$

sehr steife Komponenten werden bei Radau-Verfahren gedämpft.

Man kann allgemein zeigen, dass Radau-Verfahren A-stabil für jedes s sind, denn ihre Stabilitätsfunktionen sind $(s-1,s)$ Padé-Approximationen an die Exponentialfunktion. Eine hervorragende Implementierung ist der Code RADAU5 ($s = 3$ und $p = 5$) von Hairer und Wanner 1988.

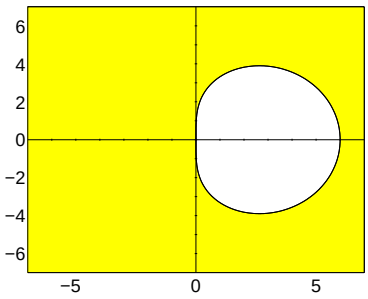

Abb. 10.8: Stabilitätsbereich des Radau-Verfahrens für $s = 2$

10.7 Implementierung impliziter Runge-Kutta-Verfahren

Bei impliziten Runge-Kutta-Verfahren müssen im allgemeinen nichtlineare Gleichungssysteme

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, Y_j), \qquad i = 1, \ldots, s$$

der Dimension ds gelöst werden. Mit den Lösungen Y_i berechnet man dann explizit

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(t_0 + c_i h, Y_i).$$

Hierzu werden s zusätzliche Funktionsauswertungen benötigt.

Um den Einfluss von Rundungsfehlern zu reduzieren löst man die Runge-Kutta-Gleichungen nicht nach Y_i, sondern nach $Z_i = Y_i - y_0$, $i = 1, \ldots, s$ auf. Definieren wir

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix}, \qquad f(Y) = \begin{bmatrix} f(t_0 + c_1 h, Y_1) \\ \vdots \\ f(t_0 + c_s h, Y_s) \end{bmatrix},$$

und entsprechend Z, dann sind die Runge-Kutta-Gleichungen äquivalent zu

$$Y = \mathbb{1} \otimes y_0 + h(\mathcal{A} \otimes I) f(Y).$$

Hierbei bezeichnet $A \otimes B$ das Kronecker-Produkt zweier Matrizen: Für $A \in \mathbb{C}^{m,n}$ und $B \in \mathbb{C}^{p,q}$ ist

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ldots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{C}^{mp,nq}.$$

Mit obiger Notation ist $Z = Y - \mathbb{1} \otimes y_0$, also erhalten wir

$$\begin{aligned} Z &= h(\mathcal{A} \otimes I) f(\mathbb{1} \otimes y_0 + Z), \\ y_1 &= y_0 + h(b^T \otimes I) f(\mathbb{1} \otimes y_0 + Z). \end{aligned}$$

Bei der Berechnung von y_1 können Fehler in Z durch die große Lipschitz-Konstante von f verstärkt werden. Falls $\mathcal{A}$ invertierbar ist, kann man dieses Problem durch eine weitere Umformung umgehen. Aus

$$(\mathcal{A}^{-1} \otimes I)Z = hf(\mathbb{1} \otimes y_0 + Z)$$

folgt nämlich

$$\begin{aligned} y_1 &= y_0 + (b^T \otimes I)(\mathcal{A}^{-1} \otimes I)Z \\ &=: y_0 + (d^T \otimes I)Z, \qquad\qquad \text{mit } d^T = b^T\mathcal{A}^{-1}. \end{aligned}$$

Zusätzliche Funktionsauswertungen sind hier nicht mehr nötig. Im Spezialfall, dass $a_{si} = b_i$ für alle i gilt, etwa bei Radau-Verfahren, ist $d^T = e_s^T$ und damit $y_1 = y_0 + Z_s = Y_s$. Verfahren mit dieser Eigenschaft heißen "stiffly accurate".

Wie bei impliziten Mehrschrittverfahren bietet sich auch hier die Lösung der nichtlinearen Gleichungssysteme für Z_i, $i = 1, \ldots, s$ mit einem vereinfachten Newton-Verfahren an. Da $\|Z_i\| = O(h)$, kann man als Startwert $Z_i^{(0)} = 0$ wählen. Definieren wir

$$F_i(Z) = Z_i - h\sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, y_0 + Z_j), \qquad i = 1, \ldots, s$$

und $F(Z) = (F_i(Z))_{i=1}^s$, so ist das nichtlineare Gleichungssystem $F(Z) = 0$ zu lösen. Es gilt

$$F'(Z) = I_{sd} - h \begin{bmatrix} a_{11}\frac{\partial f}{\partial y}(t_0 + c_1 h, y_0 + Z_1) & \cdots & a_{1s}\frac{\partial f}{\partial y}(t_0 + c_s h, y_0 + Z_s) \\ \vdots & & \vdots \\ a_{s1}\frac{\partial f}{\partial y}(t_0 + c_1 h, y_0 + Z_1) & \cdots & a_{ss}\frac{\partial f}{\partial y}(t_0 + c_s h, y_0 + Z_s). \end{bmatrix}.$$

Im vereinfachten Newton-Verfahren ersetzen wir alle Jacobi-Matrizen $\frac{\partial f}{\partial y}(t_0 + c_j h, y_0 + Z_j)$ durch eine Approximation $J \approx \frac{\partial f}{\partial y}(t_0, y_0)$:

$$F'(Z) \approx I_{sd} - h\mathcal{A} \otimes J.$$

Die Iterationsvorschrift des vereinfachten Newton-Verfahrens lautet somit

$$\begin{aligned} (I_{sd} - h\mathcal{A} \otimes J)\Delta Z^{(k)} &= -F(Z^{(k)}) \\ &= -Z^{(k)} + h(\mathcal{A} \otimes I)f(\mathbb{1} \otimes y_0 + Z^{(k)}), \\ Z^{(k+1)} &= Z^{(k)} + \Delta Z^{(k)} \end{aligned}$$

woraus man durch Multiplikation mit $(h\mathcal{A} \otimes I)^{-1} = h^{-1}(\mathcal{A}^{-1} \otimes I)$

$$(h^{-1}(\mathcal{A}^{-1} \otimes I) - I \otimes J)\Delta Z^{(k)} = -h^{-1}(\mathcal{A}^{-1} \otimes I)Z^{(k)} + f(\mathbb{1} \otimes y_0 + Z^{(k)})$$

erhält. Die direkte Lösung dieses linearen Gleichungssystems etwa mit Gauß-Elimination, bzw. die LU-Zerlegung der Koeffizientenmatrix, kostet $\frac{1}{3}(sd)^3$ Operationen. Effizienter ist es, $\mathcal{A}^{-1}$ zu diagonalisieren:

$$T^{-1}\mathcal{A}^{-1}T = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_s).$$

Man beachte, dass diese Diagonalisierung nur vom Verfahren abhängt, jedoch nicht von der Differentialgleichung. Es reicht also, sie ein einziges Mal zu berechnen. Mit einfachen Rechenregeln für Kronecker-Produkte erhält man für die transformierten Variablen $W = (T^{-1} \otimes I)Z$

$$(h^{-1}\Lambda \otimes I - I \otimes J)\Delta W^{(k)} = -h^{-1}(\Lambda \otimes I)W^{(k)} + (T^{-1} \otimes I)f(\mathbb{1} \otimes y_0 + (T \otimes I)W^{(k)}).$$

Die Koeffizientenmatrix dieses linearen Gleichungssystems ist blockdiagonal. Die s Systeme der Dimension d sind also entkoppelt. Damit man nicht für alle s Eigenwerte λ_i eine neue LU-Zerlegung berechnen muss, ist es günstig, J zunächst auf obere Hessenberg-Form zu transformieren, $H = Q^H J Q$ mit einer unitären Matrix Q. Der Aufwand hierfür beträgt $\frac{2}{3}d^3$ Operationen. Die LU-Zerlegung von $h^{-1}\lambda_i I - H$ kann dann in nur $O(d^2)$ Operationen berechnet werden.

Algorithmus 10.1 fasst die einzelnen Schritte zusammen und zeigt, welcher Aufwand notwendig ist. Die Jacobi-Matrix J wird im Algorithmus als bekannt vorausgesetzt.

Algorithmus 10.1 Vereinfachtes Newton-Verfahren	Aufwand
$W^{(0)} = (T^{-1} \otimes I)Z^{(0)}$	$s^2 d$
transformiere J auf obere Hessenberg-Form: $H = Q^H J Q$	$2d^3/3$
berechne die LU-Zerlegungen von $h^{-1}\lambda_i I - H$ für $i = 1, \ldots, s$	sd^2
setze $k = 0$	
while $(k < k_{\max})$ **do**	
berechne $f(t_0 + c_i h, y_0 + Z_i^{(k)})$, $i = 1, \ldots, s$	s f-Ausw.
berechne $r := (T^{-1} \otimes I) f(\mathbb{1} \otimes y_0 + Z^{(k)})$	$s^2 d$
berechne $r := -h^{-1}(\Lambda \otimes I)W^{(k)} + r$	sd
löse $(h^{-1}\lambda_i I - H)\big(Q\Delta W_i^{(k)}\big) = Qr_i$, $i = 1, \ldots, s$	$2sd^2$
berechne $\|\Delta W^{(k)}\|$, überprüfe Konvergenz	sd
setze $W^{(k+1)} = W^{(k)} + (Q^H \otimes I)\big((Q \otimes I)\Delta W^{(k)}\big)$	$sd + sd^2$
berechne $Z^{(k+1)} = (T \otimes I)W^{(k+1)}$	$s^2 d$
setze $k := k + 1$	
end while	

Interessanterweise liefert die Konvergenz oder Divergenz des vereinfachten Newton-Verfahrens Hinweise über die Wahl der Schrittweite oder über die Möglichkeit, die Jacobi-Matrix für weitere Zeitschritte festzuhalten. So wird man die Schrittweite reduzieren, wenn in der maximalen Anzahl $k_{\max}$ von Newton-Schritten keine Konvergenz eintritt. Noch effizienter ist es, die Konvergenzgeschwindigkeit nach wenigen Schritten zu schätzen und mit kleinerer Schrittweite neu zu beginnen, bevor man $k_{\max}$ Schritte gerechnet hat.

Zur Schrittweitensteuerung werden wie in Abschnitt 8.9 eingebettete Verfahren verwendet. Wichtig ist außerdem, dass das eingebettete Verfahren ebenfalls für steife Differentialgleichungen geeignet ist. Die Fehlerschätzung angewendet auf die Testgleichung sollte sich wie der exakte Fehler verhalten.

Derartige Details zur Implementierung werden in den Übungen ausführlich diskutiert. Man findet sie in Hairer & Wanner (1996, Abschnitt IV.8).

10.8 Implizites Euler-Verfahren bei Dgl. mit einseitiger Lipschitz-Bedingung

Für die rechte Seite der Differentialgleichung (10.1) nehmen wir nun an, dass $f : [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ stetig differenzierbar ist und die *einseitige Lipschitz-Bedingung* (8.4)

$$\langle f(t, y) - f(t, z), y - z \rangle \le \ell \|y - z\|^2 \qquad \text{für alle } y, z \in \mathbb{R}^d$$

für ein Skalarprodukt $\langle \cdot, \cdot \rangle$ auf $\mathbb{R}^d$ und die dadurch induzierte Norm $\|\cdot\|$ erfüllt. In Satz 8.2 haben wir gezeigt, dass zwei Lösungen $y, z : [t_0, T] \to \mathbb{R}^d$ der Differentialgleichung (8.1) zu

den Anfangswerten $y(t_0) = y_0$, $z(t_0) = z_0$ die Ungleichung

$$\|y(t) - z(t)\| \le e^{\ell(t-t_0)}\|y_0 - z_0\| \qquad \text{für alle } t \in [t_0, T]$$

erfüllen. Wir leiten jetzt eine entsprechende Aussage für die Näherungen des impliziten Euler-Verfahrens $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ her.

Satz 10.15. *Unter der Voraussetzung* (8.4) *und* $h\ell < 1$ *gilt für den Fehler des impliziten Euler-Verfahrens*

$$\|y_n - y(t_n)\| \le mh \qquad \textit{für} \qquad t_n \in [t_0, T]$$

mit

$$m = \frac{e^{\ell(T-t_0)/(1-h\ell)} - 1}{\ell} \, \frac{1}{2} \max_{t\in[t_0,T]} \|y''(t)\|.$$

Beweis. (Vergleichen Sie hierzu den Beweis von Satz 8.3 für das explizite Euler-Verfahren).

(a) Lokaler Fehler: Wir untersuchen zunächst den Fehler nach einem Schritt des Verfahrens, wenn man auf der exakten Lösung $(t_n, y(t_n))$ startet. Der Satz von Taylor liefert für die exakte Lösung

$$\begin{aligned} y(t_{n+1}) &= y(t_n) + hy'(t_{n+1}) - \frac{1}{2}h^2 y''(\tau), \quad t_n \le \tau \le t_{n+1} \\ &= y(t_n) + hf(t_{n+1}, y(t_{n+1})) + d_{n+1}, \end{aligned}$$

wobei

$$\|d_{n+1}\| \le \frac{1}{2}h^2 \max_{t\in[t_0,T]} \|y''(t)\| =: Ch^2.$$

Subtrahieren wir hiervon die Approximation z_{n+1} des impliziten Euler-Verfahrens (ausgehend von der exakten Lösung) im nächsten Schritt,

$$z_{n+1} = y(t_n) + hf(t_{n+1}, z_{n+1}),$$

so ergibt sich

$$\begin{aligned} \|z_{n+1} - y(t_{n+1})\|^2 &= \langle h\big(f(t_{n+1}, z_{n+1}) - f(t_{n+1}, y(t_{n+1})\big) - d_{n+1}, z_{n+1} - y(t_{n+1})\rangle \\ &\le h\ell\|z_{n+1} - y(t_{n+1})\|^2 + \|d_{n+1}\| \, \|z_{n+1} - y(t_{n+1})\|, \end{aligned}$$

also

$$\|z_{n+1} - y(t_{n+1})\| \le \frac{1}{1-h\ell} \|d_{n+1}\| \le \frac{Ch^2}{1-h\ell}.$$

(b) Fehlerfortpflanzung: Ausgehend von den Anfangswerten v_n bzw. w_n ergeben sich durch einen impliziten Euler-Schritt die Näherungen

$$\begin{aligned} v_{n+1} &= v_n + hf(t_{n+1}, v_{n+1}) \\ w_{n+1} &= w_n + hf(t_{n+1}, w_{n+1}) \\ &= v_n + hf(t_{n+1}, w_{n+1}) + (w_n - v_n). \end{aligned}$$

Wie in Teil (a) folgt hieraus

$$\|v_{n+1} - w_{n+1}\| \le \frac{1}{1-h\ell} \|v_n - w_n\|.$$

(c) Fehlerakkumulation: Wie im Beweis von Satz 8.3 folgt mit Hilfe von Lady Windermere's Fächer die Abschätzung

$$\|y_n - y(t_n)\| \le \frac{Ch^2}{(1-h\ell)} \sum_{j=0}^{n-1} \frac{1}{(1-h\ell)^j}.$$

Aus $1 + x \le e^x$ für alle $x \in \mathbb{R}$ folgt

$$\frac{1}{1-hl} = 1 + \frac{hl}{1-hl} \le e^{h\ell/(1-h\ell)},$$

so dass mit Hilfe der Formel für die geometrische Reihe

$$\begin{aligned}\|y_n - y(t_n)\| &\le \frac{Ch^2}{1-h\ell} \frac{\frac{1}{(1-h\ell)^n} - 1}{\frac{1}{1-h\ell} - 1} \\ &\le \frac{Ch}{\ell}(e^{nh\ell/(1-h\ell)} - 1) \\ &\le mh.\end{aligned}$$

Die letzte Ungleichung folgt aus $nh \le T - t_0$. □

Bemerkungen.

(a) Für $h\ell \le \alpha < 1$ kann m unabhängig von h beschränkt werden. Auf jeden Fall ist m im Gegensatz zum expliziten Euler-Verfahren unabhängig von der Lipschitz-Konstanten L.

(b) (ohne Beweis) Die Gleichung $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$ hat eine eindeutige Lösung falls $h\ell < 1$.

10.9 Kontraktive Runge-Kutta-Verfahren

Mit den Bezeichnungen aus dem vorigen Abschnitt gilt für zwei Lösungen $y(t)$, $z(t)$ der Differentialgleichung (10.1)

$$\begin{aligned}\frac{d}{dt}\|y(t) - z(t)\|^2 &= \frac{d}{dt}\langle y(t) - z(t), y(t) - z(t)\rangle \\ &= \langle y'(t) - z'(t), y(t) - z(t)\rangle + \langle y(t) - z(t), y'(t) - z'(t)\rangle \\ &= 2\,\mathrm{Re}\,\langle y'(t) - z'(t), y(t) - z(t)\rangle \\ &= 2\,\mathrm{Re}\,\langle f(t, y(t)) - f(t, z(t)), y(t) - z(t)\rangle.\end{aligned}$$

Im Folgenden setzen wir

$$\mathrm{Re}\,\langle f(t,y) - f(t,z), y - z\rangle \le 0 \qquad \text{für alle } t \in \mathbb{R},\ y, z \in \mathbb{C}^d \tag{10.18}$$

voraus. Mit dem oben Gezeigten wird folgende Definition klar:

Definition 10.16. (*Kontraktivität*)

(a) *Die Differentialgleichung* (10.1) *heißt* **kontraktiv**, *falls*

$$\|y(t) - z(t)\| \le \|y(t_0) - z(t_0)\| \qquad \textit{für alle } t \ge t_0$$

gilt.

(b) *Ein Runge-Kutta-Verfahren heißt* **kontraktiv** *oder* **B-stabil**, *falls für jede Differentialgleichung die* (10.18) *erfüllt und für beliebige Schrittweiten* $h > 0$

$$\|y_1 - z_1\| \leq \|y_0 - z_0\|$$

gilt, wobei y_1, z_1 *die numerischen Lösungen zu den Anfangswerten* y_0, z_0 *sind.*

Differentialgleichungen mit einseitiger Lipschitz-Konstante $\ell \leq 0$ sind kontraktiv. Teil (b) des Beweises von Satz 10.15 zeigt, dass das implizite Euler-Verfahren kontraktiv ist, denn für $\ell \leq 0$ gilt $1/(1 - h\ell) \leq 1$.

Lemma 10.17. *Ein kontraktives Runge-Kutta-Verfahren ist A-stabil.*

Beweis. Die Testgleichung $f(t, y) = \lambda y$ mit $\operatorname{Re}\lambda \leq 0$ erfüllt offensichtlich (10.18). Die Runge-Kutta-Näherungen zu den Anfangswerten y_0, z_0 sind durch

$$y_1 = R(h\lambda)y_0, \qquad z_1 = R(h\lambda)z_0$$

gegeben, wobei R die Stabilitätsfunktion des Runge-Kutta-Verfahrens ist. Aus der Kontraktivität folgt daher $|R(\mu)| \leq 1$ für $\operatorname{Re}\mu \leq 0$, was gerade bedeutet, dass das Verfahren A-stabil ist. □

Wir werden jetzt zeigen, dass Gauß- und Radau-Kollokationsverfahren nicht nur A-stabil, sondern sogar kontraktiv sind Zur Erinnerung: Gauß-Verfahren verwenden die Nullstellen $c_j \in (0, 1)$ der Legendre-Polynome und haben die Ordnung $p = 2s$. Bei Radau-Verfahren hingegen wählt man $c_s = 1$. Die übrigen Knoten $0 < c_1 < \cdots < c_s = 1$ sind eindeutig durch die Forderung $p = 2s - 1$ bestimmt.

Lemma 10.18. *Die Quadraturgewichte einer* s*-stufigen Quadraturformel der Ordnung* $p \geq 2s - 1$ *sind positiv:* $b_j > 0$ *für* $j = 1, \ldots, s$.

Beweis. Lagrange-Polynome $\ell_j(x)$ zu den Knoten $c_1, \ldots, c_s$ erfüllen $l_i(c_j) = \delta_{ij}$, $\deg \ell_i = s-1$. Damit ist $\deg \ell_i^2 = 2s - 2 \leq p - 1$ und es gilt

$$b_i = \sum_{j=1}^{s} b_j \ell_i^2(c_j) = \int_0^1 \ell_i^2(x)dx > 0. \quad \square$$

Insbesondere haben Gauß- und Radau-Quadraturformeln positive Gewichte.

Satz 10.19. *Gauß-Kollokationsverfahren sind kontraktiv.*

Beweis. Es seien u und $\widetilde{u}$ die Kollokationspolynome des Gauß-Verfahrens angewandt auf (10.1) zu den Startwerten y_0 und z_0. Wir definieren den Abstand

$$d(t) = \|u(t) - \tilde{u}(t)\|^2. \tag{10.19}$$

Offensichtlich ist d ein Polynom vom Grad $2s$. Die Ableitung ist also ein Polynom vom Grad $2s-1$, welches von einer Gauß-Quadraturformel exakt integriert wird. Da u und $\widetilde{u}$ die Kollokationsbedingungen erfüllen, folgt

$$\begin{aligned} d(t_1) &= d(t_0) + \int_{t_0}^{t_0+h} d'(t)\,dt \\ &= d(t_0) + h\sum_{i=1}^{s} b_i d'(\tau_i) \\ &= d(t_0) + h\sum_{i=1}^{s} b_i 2\mathrm{Re}\,\langle f(\tau_i,u(\tau_i)) - f(\tau_i,\tilde{u}(\tau_i)), u(\tau_i)-\tilde{u}(\tau_i)\rangle \\ &\le d(t_0), \end{aligned}$$

wobei $\tau_i := t_0 + c_i h$. Die letzte Ungleichung folgt aus der Voraussetzung (10.18) und Lemma 10.18. □

Satz 10.20. *Radau-Kollokationsverfahren sind kontraktiv.*

Beweis. Der Beweis von Satz 10.19 ist nicht unmittelbar auf Radau-Verfahren anwendbar, da Radau-Quadraturformelen nur Polynome bis zum Grad $2s-2$ exakt integrieren.

Wir betrachten das in Gleichung (10.19) definierte Polynom $d \in \mathcal{P}_{2s}$ und schreiben es in der Form

$$d(t) = c(t-t_0)^{2s} + r(t), \qquad r \in \mathcal{P}_{2s-1}.$$

Wegen $d(t) \ge 0$ für alle t ist $c \ge 0$. Somit folgt

$$d'(t) = 2cs(t-t_0)^{2s-1} + r'(t).$$

Der Höchstkoeffizient von d' ist $2cs \ge 0$, so dass wir für $\tau_i = t_0 + c_i h$ auch

$$d'(t) = m(t) + q(t), \qquad m(t) := 2cs(t-\tau_1)^2 \cdot \ldots \cdot (t-\tau_{s-1})^2(t-\tau_s)$$

mit einem Polynom $q \in \mathcal{P}_{2s-2}$ schreiben können. Wie im Beweis des letzten Satzes gilt dann

$$\begin{aligned} d(t_1) &= d(t_0) + \int_{t_0}^{t_0+h} d'(t)\,dt \\ &= d(t_0) + \int_{t_0}^{t_0+h} m(t)dt + \int_{t_0}^{t_0+h} q(t)dt \\ &\le d(t_0) + \int_{t_0}^{t_0+h} q(t)\,dt, \end{aligned}$$

denn wir haben nach Definition $m(t) \le 0$ für $t_0 \le t \le t_0+h$. Da die Radau-Quadraturformel die Ordnung $2s-2$ hat, wird das Integral durch die Quadraturformel exakt integriert. Ferner ist $m(\tau_i) = 0$, so dass $d'(\tau_i) = q(\tau_i)$ für $i = 1,\ldots,s$ gilt. Daraus erhalten wir

$$\int_{t_0}^{t_0+h} q(t)\,dt = h\sum_{i=1}^{s} b_i q(\tau_i) = h\sum_{i=1}^{s} b_i d'(\tau_i).$$

Analog zum Beweis von Satz 10.19 folgt $d'(\tau_i) \le 0$ und damit $d(t_1) \le d(t_0)$. □

10.10 Beispiel von Prothero und Robinson

Prothero und Robinson (1974) waren die ersten, die beobachtet haben, dass bei der Anwendung impliziter Runge-Kutta-Verfahren auf steife Differentialgleichungen Ordnungsreduktionen auftreten können. Sie haben auch bemerkt, dass A-stabile Verfahren instabile Lösungen produzieren können, wenn sie auf nichtlineare Differentialgleichungen angewendet werden. Frank, Schneid und Ueberhuber (1981) haben das Konzept der B-Konvergenz eingeführt.

Wir betrachten zunächst das folgende nichtlineare Testproblem von Prothero und Robinson

$$y' = \lambda\Big(y - \varphi(t)\Big) + \varphi'(t), \qquad y(t_0) = y_0,\ \operatorname{Re}\lambda \le 0. \tag{10.20}$$

Offensichtlich ist $y(t) = \varphi(t)$ die exakte Lösung und dies erlaubt, lokale und globale Fehler explizit anzugeben.

Wenden wir ein Runge-Kutta-Verfahren auf (10.20) an, so ergibt sich

$$Y_i = y_0 + h\sum_{j=1}^{s} a_{ij}\Big(\lambda\big(Y_j - \varphi(t_0 + c_j h)\big) + \varphi'(t_0 + c_j h)\Big), \tag{10.21a}$$

$$y_1 = y_0 + h\sum_{i=1}^{s} b_i\Big(\lambda\big(Y_i - \varphi(t_0 + c_i h)\big) + \varphi'(t_0 + c_i h)\Big). \tag{10.21b}$$

Ersetzen wir in dieser Formel Y_i, y_0 und y_1 durch die exakte Lösung, so ergeben sich Defekte

$$\varphi(t_0 + c_i h) = \varphi(t_0) + h\sum_{j=1}^{s} a_{ij}\varphi'(t_0 + c_j h) + \Delta_{i,h}(t_0), \tag{10.22a}$$

$$\varphi(t_0 + h) = \varphi(t_0) + h\sum_{i=1}^{s} b_i\varphi'(t_0 + c_i h) + \Delta_{0,h}(t_0). \tag{10.22b}$$

Definition 10.21. *Die* **Stufenordnung** q *eines Runge-Kutta-Verfahrens ist die größte natürliche Zahl für die*

$$\sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{c_i^k}{k}, \qquad k = 1, \dots, q$$

für $i = 1, \dots, s$ *gilt.*

Bemerkung. Wegen (10.12) haben Kollokationsverfahren mindestens die Stufenordnung s. Dass sie tatsächlich die Stufenordnung s haben, kann man sich mit Hilfe der Differentialgleichung $y' = t^s$ überlegen.

Taylor-Entwicklung von φ in (10.22) liefert dann

$$\Delta_{0,h}(t_0) = O(h^{p+1}), \qquad \Delta_{i,h}(t_0) = O(h^{q+1}), \tag{10.23}$$

wobei p die Ordnung der Quadraturformel (b_i, c_i) (und damit auch die Ordnung des Runge-Kutta-Verfahrens) und q die Stufenordnung ist.

Subtraktion von (10.22) und (10.21) liefert

$$Y_i - \varphi(t_0 + c_i h) = y_0 - \varphi(t_0) + h\sum_{j=1}^{s} a_{ij}\lambda\big(Y_j - \varphi(t_0 + c_j h)\big) - \Delta_{i,h}(t_0).$$

Schreiben wir

$$Y = (Y_i)_{i=1}^s, \quad \Phi = (\varphi(t_0 + c_i h))_{i=1}^s, \quad \Delta_h = (\Delta_{i,h}(t_0))_{i=1}^s$$

als Vektoren, so gilt

$$Y - \Phi = (I - z\mathcal{A})^{-1}\Big(\mathbb{1}(y_0 - \varphi(t_0)) - \Delta_h\Big).$$

und

$$\begin{aligned} y_1 - \varphi(t_0 + h) &= y_0 - \varphi(t_0) + hb^T\lambda(Y - \Phi) - \Delta_{0,h} \\ &= R(z)(y_0 - \varphi(t_0)) - zb^T(I - z\mathcal{A})^{-1}\Delta_h - \Delta_{0,h} \end{aligned}$$

mit der Stabilitätsfunktion $R(z) = 1 + zb^T(I - z\mathcal{A})^{-1}\mathbb{1}$ und $z = h\lambda$.

Der lokale Fehler $\delta_h(t)$ ergibt sich, wenn man $y_0 = \varphi(t_0)$ setzt

$$\delta_h(t) = -zb^T(I - z\mathcal{A})^{-1}\Delta_h - \Delta_{0,h}.$$

Damit genügt der globale Fehler der Rekursion

$$y_{n+1} - \varphi(t_{n+1}) = R(z)(y_n - \varphi(t_n)) + \delta_h(t_n). \tag{10.24}$$

Auflösen der Rekursion liefert (für konstante Schrittweite)

$$y_{n+1} - \varphi(t_{n+1}) = R(z)^{n+1}(y_0 - \varphi(t_0)) + \sum_{j=0}^{n} R(z)^{n-j}\delta_h(t_j). \tag{10.25}$$

Die klassische (nicht-steife) Theorie behandelt den Fall $z = O(h)$ und dann verhält sich der globale Fehler wie $O(h^p)$. Bei steifen Differentialgleichungen ist man aber an Schrittweiten $h \gg |\lambda|^{-1}$ interessiert. Für unser Testproblem (10.20) betrachten wir daher die Grenzübergänge $h \to 0$ und $z = \lambda h \to \infty$ simultan.

Beispiel. Gauß-Kollokations-Verfahren

Nach Hairer & Wanner (1996, Theorem IV.14.5) ist $\mathcal{A}$ invertierbar, so dass

$$-zb^T(I - z\mathcal{A})^{-1} = b^T\mathcal{A}^{-1}(I - z^{-1}\mathcal{A}^{-1})^{-1} = b^T\mathcal{A}^{-1} + O(z^{-1}).$$

Wegen (10.23) und $q = s$ gilt für den lokalen Fehler $\delta_h(t) = O(h^{s+1})$. Aus (10.25) folgt, dass der globale Fehler sich wie $O(h^s)$ verhält, denn für $\operatorname{Re}\lambda \le 0$ ist $|R(z)| \le 1$. Für ungerades s kann man $R(\infty) = -1$ ausnutzen, indem man in (10.25) partielle Summation anwendet:

$$\sum_{j=0}^{n} \rho^{n-j}\delta(t_j) = \frac{1 - \rho^{n+1}}{1 - \rho}\delta(t_0) + \sum_{j=1}^{n} \frac{1 - \rho^{n+1-j}}{1 - \rho}\big(\delta(t_j) - \delta(t_{j-1})\big) \tag{10.26}$$

(Übung). Wegen $\delta(t_j) - \delta(t_{j-1}) = O(h^{q+2})$ gewinnt man hier eine zusätzliche h-Potenz, der globale Fehler ist also $O(h^{s+1})$. $\diamond$

Beispiel. Radau-Verfahren (Radau IIA)

Auch für Radau-Verfahren ist $\mathcal{A}$ invertierbar. Außerdem erfüllen diese Verfahren $a_{si} = b_i$ für $i = 1, \ldots, s$. Die letzte Stufe ist also identisch mit der numerischen Lösung, so dass für den Fehler

$$\delta_h(t) = -e_s^T(I - z\mathcal{A})^{-1}\Delta_h(t) = e_s^T z^{-1}\mathcal{A}^{-1}(I - z^{-1}\mathcal{A}^{-1})^{-1}\Delta_h(t) = O(z^{-1}h^{s+1}).$$

Wegen $R(\infty) = 0$ sind lokaler und globaler Fehler im Wesentlichen identisch. $\diamond$

Die Beispiele zeigen, dass die Konvergenzordnung signifikant schlechter sein kann als die (klassische) Ordnung des Verfahrens. Zudem liefern steif genaue Verfahren wie Radau-Verfahren asymptotisch das korrekte Verhalten für $z \to \infty$.

10.11 Konvergenz bei kontraktiven Differentialgleichungen

Für die Differentialgleichung (10.1) mit $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ setzen wir in diesem Abschnitt die einseitige Lipschitz-Bedingung (8.4) mit $\ell = 0$ voraus, d. h.

$$\langle f(t,y) - f(t,z), y - z\rangle \le 0 \qquad \text{für alle } y, z \in \mathbb{R}^d. \tag{10.27}$$

Alle Resultate dieses Abschnitts bleiben jedoch mit den gleichen Argumenten auch für $\ell \le 0$ richtig.

In Satz 8.5 haben wir die globale Fehlerabschätzung

$$\|y_n - y(t_n)\| \le C(L,T)h^p, \qquad 0 \le t_n \le T,$$

für Runge-Kutta-Verfahren der Ordnung p kennengelernt. Da in die Konstante $C(L,T)$ die Lipschitz-Konstante L wie e^{LT} eingeht, ist eine solche Abschätzung für steife Differentialgleichungen, bei denen L groß ist, unbrauchbar. Wir wollen im Folgenden eine Abschätzung der Form

$$\|y_n - y(t_n)\| \le ch^q, \qquad 0 \le t_n \le T, \quad h \le h_0,$$

bei der c und h_0 unabhängig von L sind, herleiten. Eine solche Abschätzung mit $q = 1$ haben wir in Satz 10.15 bereits für das implizite Euler-Verfahren gezeigt. Im allgemeinen tritt für algebraisch stabile Runge-Kutta-Verfahren leider eine Ordnungsreduktion, d. h. $q < p$ ein. Das klassische Beweisschema für die Abschätzung des globalen Fehlers besteht aus drei Schritten (vgl. Satz 10.15): Abschätzen des lokalen Fehlers, Fehlerfortpflanzung und Fehlerakkumulation. Da algebraisch stabile Runge-Kutta-Verfahren kontraktiv sind, werden Fehler gedämpft:

$$\|y_{n+1} - z_{n+1}\| \le \|y_n - z_n\|,$$

es tritt also keine Fehlerfortpflanzung auf. Falls wir also für den lokalen Fehler eine Abschätzung der Form

$$\|y_1(t,h) - y(t+h)\| \le \bar{c}h^{q+1}$$

zeigen können, so gilt für den globalen Fehler

$$\|y_n - y(t_n)\| \le n\bar{c}h^{q+1} \le \bar{c}(T - t_0)h^q, \qquad nh \le T - t_0.$$

Zusätzlich zur algebraischen Stabilität benötigen wir noch folgende Voraussetzung

$$\begin{cases} \mathcal{A} = (a_{ij}) \text{ invertierbar} \\ \exists D = \operatorname{diag}(d_1, \dots, d_s), d_j > 0 \text{ und } \alpha > 0 : \langle D\mathcal{A}^{-1}x, x\rangle \ge \alpha\|x\|^2 \ \ \forall x \in \mathbb{R}^s. \end{cases} \tag{10.28}$$

Bemerkung. Die zweite Bedingung ist genau dann erfüllt, wenn der symmetrische Anteil von $D\mathcal{A}^{-1}$, d. h. $(D\mathcal{A}^{-1} + (D\mathcal{A}^{-1})^T)/2$ positiv definit ist: Sei α der kleinste Eigenwert dieser Matrix, so ist für jedes $x \in \mathbb{R}^d$

$$\langle D\mathcal{A}^{-1}x, x\rangle = \tfrac{1}{2}(\langle D\mathcal{A}^{-1}x, x\rangle + \langle x, D\mathcal{A}^{-1}x\rangle) = \tfrac{1}{2}\langle (D\mathcal{A}^{-1} + (D\mathcal{A}^{-1})^T)x, x\rangle \ge \alpha\|x\|^2.$$

$\alpha > 0$ ist also eine untere Schranke für den kleinsten Eigenwert von $(D\mathcal{A}^{-1} + (D\mathcal{A}^{-1})^T)/2$.

Satz 10.22. *Die Differentialgleichung (10.1) sei kontraktiv, das Runge-Kutta-Verfahren der Ordnung p und der Stufenordnung $q \le p$ sei kontraktiv und erfülle zusätzlich (10.28). Dann gilt für den lokalen Fehler*

$$\|y_1 - y(t_0 + h)\| \le ch^{q+1},$$

wobei c nur von der Lösung y abhängt und nicht von f.

Beweis. Setzen wir

$$\widetilde{Y}_i = y(t_0 + c_i h), \qquad \widetilde{Y}_i' = y'(t_0 + c_i h),$$

so erhalten wir (mit Hilfe von Taylor-Entwicklung)

$$\widetilde{Y}_i = y_0 + \int_{t_0}^{t_0+c_i h} y'(t)dt = y_0 + h\sum_{j=1}^{s} a_{ij}\widetilde{Y}_j' + \delta_i,$$

wobei δ_i der Quadraturfehler ist, also $\|\delta_i\| = O(h^{q+1})$. Andererseits ist

$$Y_i = y_0 + h\sum_{j=1}^{s} a_{ij}Y_j'.$$

Wir definieren nun die Differenzvektoren

$$\Delta Y_i = Y_i - \widetilde{Y}_i, \qquad \Delta Y_i' = Y_i' - \widetilde{Y}_i' = f(t_0 + c_i h, Y_i) - f(t_0 + c_i h, \widetilde{Y}_i)$$

und fassen die Vektoren zusammen

$$\Delta Y = \begin{bmatrix} \Delta Y_1 \\ \vdots \\ \Delta Y_s \end{bmatrix}, \qquad \Delta Y' = \begin{bmatrix} \Delta Y_1' \\ \vdots \\ \Delta Y_s' \end{bmatrix}, \qquad \delta = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_s \end{bmatrix}.$$

Dann gilt

$$\Delta Y = h(\mathcal{A} \otimes I)\Delta Y' - \delta \Longleftrightarrow h\Delta Y' = (\mathcal{A}^{-1} \otimes I)(\Delta Y + \delta).$$

Multiplizieren wir diese Gleichung mit $\Delta Y^T(D \otimes I)$ von links, so ergibt sich aus der Kontraktivität und Voraussetzung (10.28)

$$h\sum_{i=1}^{s} \underbrace{d_i}_{>0}\underbrace{\langle \Delta Y_i', \Delta Y_i\rangle}_{\le 0} = \underbrace{\langle (D\mathcal{A}^{-1} \otimes I)\Delta Y, \Delta Y\rangle}_{\ge \alpha\|\Delta Y\|^2} + \langle (D\mathcal{A}^{-1} \otimes I)\delta, \Delta Y\rangle$$

und daraus

$$0 \ge \alpha\|\Delta Y\|^2 + \langle (D\mathcal{A}^{-1} \otimes I)\delta, \Delta Y\rangle.$$

Mit Hilfe der Cauchy-Schwarz'schen Ungleichung folgt

$$\alpha\|\Delta Y\|^2 \le \|D\mathcal{A}^{-1} \otimes I\|\|\delta\|\|\Delta Y\|$$

und daraus die Abschätzung $\|\Delta Y\| \le c\|\delta\|$. Für $\Delta Y'$ erhalten wir aus der Dreiecksungleichung

$$h\|\Delta Y'\| \le \|\mathcal{A}^{-1} \otimes I\|\big(\|\Delta Y\| + \|\delta\|\big) \le c'\|\delta\|.$$

Analog schätzen wir jetzt den lokalen Fehler ab. Für die Runge-Kutta-Näherung gilt

$$y_1 = y_0 + h\sum_{i=1}^{s} b_i Y_i',$$

für die exakte Lösung

$$y(t_0 + h) = y_0 + \int_{t_0}^{t_0+h} y'(t)dt = y_0 + h\sum_{i=1}^{s} b_i\widetilde{Y}_i' + O(h^{p+1}).$$

Somit ergibt sich für den lokalen Fehler

$$\begin{aligned}\|y_1 - y(t_0+h)\| &\le C \underbrace{h\|\Delta Y'\|}_{\le c'\|\delta\|} + O(h^{p+1}) \\ &= O(h^{q+1}),\end{aligned}$$

da $\|\delta\| = O(h^{q+1})$. Die Konstante hängt nur von den Ableitungen der exakten Lösung ab, nicht jedoch von f. □

Wie oben erwähnt erhalten wir aus der Kontraktivität des Runge-Kutta-Verfahrens und durch Aufsummieren

Satz 10.23. *Unter den Voraussetzungen von Satz 10.22 ist das Runge-Kutta-Verfahren konvergent von der Ordnung q, d. h.*

$$\|y_n - y(t_n)\| \le ch^q, \qquad t_0 \le t_n \le T,$$

wobei c unabhängig von der Lipschitz-Konstante von f ist. □

Bemerkung. Die Ordnungsreduktion von p auf die Stufenordnung q, z. B. von $2s$ auf s bei Gauß- und von $2s-1$ auf s bei Radau-Verfahren, ist in Anwendungen oft zu pessimistisch.

Wir weisen nun die Voraussetzung (10.28) für Gauß- und Radau-Kollokationsverfahren nach. Nach Hairer & Wanner (1996, Theorem IV.14.5) ist $\mathcal{A}$ invertierbar. Zum Beweis der zweiten Voraussetzung sei

$$B = \operatorname{diag}(b_1, \dots, b_s), \qquad C = \operatorname{diag}(c_1, \dots, c_s).$$

Lemma 10.24. *Für Gauß-Kollokationsverfahren gilt (10.28) mit $D = B(C^{-1} - I)$.*

Beweis. Wir zeigen $D\mathcal{A}^{-1} + (D\mathcal{A}^{-1})^T = BC^{-2}$. Da $c_j > 0$ und nach Lemma 10.18 auch $b_j > 0$ gilt, ist BC^{-2} eine Diagonalmatrix mit positiven Einträgen.

Um $\mathcal{A}^{-1}$ zu eliminieren, multiplizieren wir von links mit $\mathcal{A}^T$ und von rechts mit $\mathcal{A}$. Anschließend führen wir eine Kongruenztransformation mit der Vandermonde-Matrix $V = (c_i^{j-1})$ durch. Dann ist

$$V^T(\mathcal{A}^T D + D\mathcal{A} - \mathcal{A}^T BC^{-2}\mathcal{A})V = 0,$$

denn das (l,m)-te Element ist

$$\begin{aligned}&\sum_{i,j} c_i^{l-1} a_{ji} b_j (c_j^{-1} - 1) c_j^{m-1} + \sum_{i,j} c_i^{l-1} b_i (c_i^{-1} - 1) a_{ij} c_j^{m-1} - \sum_{i,j,k} c_i^{l-1} a_{ji} b_j c_j^{-2} a_{jk} c_k^{m-1} \\ &= \sum_j b_j (c_j^{-1} - 1) c_j^{m-1} \underbrace{\sum_i a_{ji} c_i^{l-1}}_{c_j^l/l} + \sum_i b_i (c_i^{-1} - 1) c_i^{l-1} \underbrace{\sum_j a_{ij} c_j^{m-1}}_{c_i^m/m} \\ &\qquad - \sum_j b_j c_j^{-2} \underbrace{\sum_i a_{ji} c_i^{l-1}}_{c_j^l/l} \underbrace{\sum_k a_{jk} c_k^{m-1}}_{c_j^m/m} \\ &= \left(\frac{1}{l} + \frac{1}{m}\right) \underbrace{\sum_j b_j \left(c_j^{m+l-2} - c_j^{m+l-1}\right)}_{\frac{1}{m+l-1} - \frac{1}{m+l}} - \frac{1}{lm} \underbrace{\sum_j b_j c_j^{m+l-2}}_{\frac{1}{m+l-1}} \\ &= \frac{[m+l-(m+l-1)](m+l) - (m+l)}{(m+l-1)(m+l)lm} \\ &= 0.\end{aligned}$$

Damit ist auch $\mathcal{O}^T D + D\mathcal{O} - \mathcal{O}^T B C^{-2} \mathcal{O} = 0$, was äquivalent zur Behauptung ist. □

Lemma 10.25. *Für Radau-Kollokationsverfahren gilt* (10.28) *mit* $D = BC^{-1}$.

Beweis. Ähnlich wie oben zeigt man $D\mathcal{O}^{-1} + (D\mathcal{O}^{-1})^T = BC^{-2} + e_s e_s^T$. □

10.12 Existenz von Runge-Kutta-Lösungen

In den vorigen Abschnitten haben wir stillschweigend die Existenz von Runge-Kutta-Lösungen vorausgesetzt. Wir wollen nun den Existenzbeweis nachholen. Aus dem Banach'schen Fixpunktsatz wissen wir, dass für hL hinreichend klein eine eindeutige Lösung existiert. Dieses Resultat ist hier leider unbrauchbar.

Wir treffen die Annahmen aus dem letzten Abschnitt, insbesondere (10.27), wobei auch hier die Argumente auf einseitige Lipschitz-Konstanten $\ell \le 0$ erweiterbar sind.

Satz 10.26. *Erfüllt ein Runge-Kutta-Verfahren die Voraussetzung* (10.28)*, dann haben die Runge-Kutta-Gleichungen eine eindeutige Lösung.*

Beweis. Eindeutigkeit: Seien Y_i, $\widetilde{Y}_i$ zwei Lösungen der Gleichungen für die inneren Stufen. Wie im Beweis von Satz 10.22 mit $\delta_i = 0$ können wir $\Delta Y_i = Y_i - \widetilde{Y}_i = 0$ schließen. Daraus ergibt sich sofort auch $\Delta Y_i' = Y_i' - \widetilde{Y}_i' = 0$ und damit $y_1 = \widetilde{y}_1$.

Existenz: Für $0 \le \tau \le 1$ betrachten wir die Homotopie

$$Y_i(\tau) = y_0 + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, Y_j(\tau)) + (\tau - 1) h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, y_0). \tag{10.29}$$

Hierbei gehen wir von der bekannten Lösung für $\tau = 0$ aus, $Y_i(0) = y_0$, $i = 1, \dots, s$. Ferner ist $Y_i(1)$ die gesuchte Runge-Kutta-Lösung, falls diese existiert. Wir differenzieren die Gleichung formal nach τ und bezeichnen mit $\dot{}$ die Ableitung nach τ:

$$\dot{Y}_i(\tau) = h \sum_{j=1}^{s} a_{ij} \frac{\partial f}{\partial y}(t_0 + c_j h, Y_j(\tau)) \dot{Y}_j(\tau) + h \sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, y_0). \tag{10.30}$$

Zur Vereinfachung führen wir einige Bezeichnungen ein:

$$F_y(Y) = \operatorname{diag}\left(\frac{\partial f}{\partial y}(t_0 + c_j h, Y_j(\tau)) \right), \quad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix}, \quad f_0 = (f(t_0 + c_j h, y_0))_{j=1}^{s}.$$

Damit ist (10.30) äquivalent zu

$$(I - h(\mathcal{O} \otimes I) F_y(Y)) \dot{Y} = h(\mathcal{O} \otimes I) f_0.$$

Das Ziel ist, $\|\dot{Y}\| \le$ const zu zeigen, also die gleichmäßige Beschränktheit von $\dot{Y}$. Wir multiplizieren dazu mit $\dot{Y}^T (D\mathcal{O}^{-1} \otimes I)$ von links:

$$\dot{Y}^T (D\mathcal{O}^{-1} \otimes I) \dot{Y} - h \dot{Y}^T (D \otimes I) F_y \dot{Y} = h \dot{Y}^T (D \otimes I) f_0.$$

Hierbei ist wegen Voraussetzung (10.28) $\dot{Y}^T (D\mathcal{O}^{-1} \otimes I) \dot{Y} \ge \alpha \|\dot{Y}\|^2$ für ein $\alpha > 0$. Schwieriger ist es, den zweiten Term abzuschätzen:

$$\dot{Y}^T (D \otimes I) F_y \dot{Y} = \sum_{i=1}^{s} d_i \langle \dot{Y}_i, \frac{\partial f}{\partial y}(t_0 + c_i h, Y_i) \dot{Y}_i \rangle \tag{10.31}$$

Die Idee besteht darin, die Ableitungen durch Grenzwerte von Differenzenquotienten zu ersetzen:

$$\frac{\partial f}{\partial y}(t_0 + c_i h, Y_i)\dot{Y}_i = \lim_{\epsilon \to 0+} \frac{1}{\epsilon}\Big[f(t_0 + c_i h, Y_i + \epsilon \dot{Y}_i) - f(t_0 + c_i h, Y_i)\Big]$$

und

$$\dot{Y}_i = \lim_{\epsilon \to 0+} \frac{1}{\epsilon}(Y_i + \epsilon \dot{Y}_i - Y_i).$$

In (10.31) eingesetzt ergibt sich aus der Kontraktivität (10.27)

$$\begin{aligned}\langle \dot{Y}_i, \frac{\partial f}{\partial y}(t_0 + c_i h, Y_i)\dot{Y}_i\rangle &= \lim_{\epsilon \to 0+} \frac{1}{\epsilon^2}\langle (Y_i + \epsilon \dot{Y}_i) - Y_i, f(t_0 + c_i h, Y_i + \epsilon \dot{Y}_i) - f(t_0 + c_i h, Y_i)\rangle \\ &\le 0.\end{aligned}$$

Insgesamt haben wir $\alpha \|\dot{Y}\|^2 \le h\|\dot{Y}\|\|D\|\|f_0\|$, oder $\|\dot{Y}\| \le ch$. Das lineare Gleichungssystem für $\dot{Y}$ hat somit eine beschränkte Lösung (gleichmäßig in Y):

$$\dot{Y} = (I - h(\mathcal{A} \otimes I)F_y(Y))^{-1} h(\mathcal{A} \otimes I) f_0, \qquad Y(0) = \begin{bmatrix} y_0 \\ \vdots \\ y_0 \end{bmatrix}.$$

Diese Differentialgleichung hat für alle $\tau \in \mathbb{R}$ eine Lösung, insbesondere für $\tau = 1$. Da $Y(\tau)$ eine Lösung von (10.29) ist, ist $Y(1)$ Lösung der Runge-Kutta-Gleichungen. □

10.13 Exponentielle Integratoren

Wir betrachten das nichtlineare Anfangswertproblem

$$y'(t) = f(t, y(t)), \qquad y(t_0) = y_0. \tag{10.32}$$

Die Grundidee exponentieller Integratoren ist motiviert dadurch, dass für gewisse *einfache* Probleme, die exakte Lösung explizit angegeben werden kann. Ein solches Problem ist etwa

$$y' = Ay + g(t), \qquad y(t_0) = y_0$$

mit einer konstanten Matrizen A und einer Funktion $g : \mathbb{R} \to \mathbb{C}^d$. Mit Hilfe der Variation-der-Konstanten Formel gilt für die Lösung

$$y(t_n + h) = \exp(hA)y(t_n) + \int_0^h \exp\big((h - \tau)A\big)g(t_n + \tau)d\tau.$$

Approximieren wir hier $g(t_n + \tau) \approx g(t_n)$ und integrieren exakt, so ergibt sich

$$y(t_n + h) \approx \exp(hA)y(t_n) + \varphi_1(hA)g(t_n),$$

mit der analytischen Funktion

$$\varphi_1(z) = \frac{e^z - 1}{z}. \tag{10.33}$$

Gleichheit gilt, wenn g konstant ist.

Bessere Approximationen können erwartet werden, wenn g durch ein geeignetes Polynom ersetzt wird, zum Beispiel durch das Interpolationspolynom in vorgegebenen Kollokationsknoten.

Wir betrachten etwas allgemeiner das semilineare Problem

$$y'(t) = Ay(t) + g(t, y(t)), \qquad y(t_0) = y_0. \tag{10.34}$$

Ist $y_n \approx y(t_n)$, so kann man durch folgende Gleichungen eine allgemeine Klasse von Einschrittverfahren definieren:

$$y_{n+1} = \chi(hA)y_n + h\sum_{i=1}^{s} b_i(hA)G_{ni}, \tag{10.35a}$$

$$Y_{ni} = \chi_i(hA)y_n + h\sum_{j=1}^{s} a_{ij}(hA)G_{nj}, \tag{10.35b}$$

$$G_{nj} = g(t_n + c_j h, Y_{nj}) \tag{10.35c}$$

Die Koeffizienten χ, χ_i, a_{ij} und b_i werden aus der Exponentialfunktion oder aus rationalen Approximationen solcher Funktionen konstruiert, welche jeweils an der Matrix oder dem Operator hA ausgewertet werden.

Bemerkung. Um ein konstistentes Verfahren zu bekommen, nehmen wir immer $\chi(0) = \chi_i(0) = 1$ an. Wenn die Koeffizienten skalare Konstanten sind, also $b_i = b_i(0)$ and $a_{ij} = a_{ij}(0)$, dann reduziert sich (10.35) auf ein Runge-Kutta-Verfahren, wenn wir den Grenzwert $A \to 0$ betrachten. Dieses Verfahren heißt im Folgenden das *zu Grunde liegende Runge-Kutta-Verfahren* und wir nennen (10.35) deshalb im Folgenden ein *exponentielles Runge-Kutta-Verfahren.* Außerdem nehmen wir an, dass das zu Grunde liegende Runge-Kutta-Verfahren die folgenden vereinfachten Bedingungen erfüllt:

$$\sum_{j=1}^{s} b_j(0) = 1, \qquad \sum_{j=1}^{s} a_{ij}(0) = c_i, \quad i = 1, \dots, s.$$

Dies ist wichtig, weil das Schema dann wie Runge-Kutta-Verfahren invariant unter einer Transformation von (10.34) in autonome Form ist (Übung).

Wir können uns daher auf autonome Probleme

$$y'(t) = Ay(t) + g(y(t)), \qquad y(t_0) = y_0, \tag{10.36}$$

beschränken.

Eine wünschenswerte Eigenschaft eines numerischen Verfahrens ist die Erhaltung von Gleichgewichtspunkten $y^\star$ des autonomen Problems (10.36). Aus der Forderung $Y_{ni} = y_n = y^\star$ für alle i und $n \geq 0$, erhalten wir notwendige und hinreichende Bedingungen. Die Koffizienten müssen die vereinfachten Bedingungen

$$\sum_{j=1}^{s} b_j(z) = \frac{\chi(z) - 1}{z}, \qquad \sum_{j=1}^{s} a_{ij}(z) = \frac{\chi_i(z) - 1}{z}, \quad i = 1, \dots, s \tag{10.37}$$

erfüllen (Übung). Diese werde wir im Folgenden immer voraussetzen.

Ein weiterer Vorteil von (10.37) ist, dass die Funktionen χ und χ_i aus (10.35) eliminiert werden können. Das Verfahren hat dann die Form

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i(hA)\big(G_{ni} + Ay_n\big), \tag{10.38a}$$

$$Y_{ni} = y_n + h\sum_{j=1}^{s} a_{ij}(hA)\big(G_{nj} + Ay_n\big). \tag{10.38b}$$

Noch allgemeiner betrachten wir das nichtlineare autonome Problem

$$y'(t) = f(y(t)), \qquad y(t_0) = y_0. \tag{10.39}$$

Für eine gegebene Näherung $y_n \approx y(t_n)$, linearisieren wir diese Gleichung an der Stelle y_n

$$y'(t) = J_n y(t) + g_n(y(t)), \tag{10.40a}$$

$$J_n = \mathrm{D}f(y_n) = \frac{\partial f}{\partial y}(y_n), \qquad g_n(y(t)) = f(y(t)) - J_n y(t) \tag{10.40b}$$

wobei wir mit J_n die Jacobi-Matrix von f und mit g_n den nichtlineare Rest ausgewertet an y_n bezeichnen. In unserem numerischen Verfahren verwenden wir diese Funktionen *explizit*.

Wenden wir ein explizites exponentielles Runge-Kutta-Verfahren auf (10.40a) an, so ergibt sich die folgende Klasse expliziter Einschrittverfahren, sogenannte *exponentielle Rosenbrock Verfahren*.

$$Y_{ni} = e^{c_i h_n J_n} y_n + h_n \sum_{j=1}^{i-1} a_{ij}(h_n J_n) g_n(Y_{nj}), \qquad 1 \le i \le s \tag{10.41a}$$

$$y_{n+1} = e^{h_n J_n} y_n + h_n \sum_{i=1}^{s} b_i(h_n J_n) g_n(Y_{ni}). \tag{10.41b}$$

Hierbei ist $h_n > 0$ eine positive Schrittweite und y_{n+1} die numerische Approximation an die exakte Lösung zur Zeit $t_{n+1} = t_n + h_n$.

Setzen wir

$$\chi(z) = e^z \qquad \text{und} \qquad \chi_i(z) = e^{c_i z}, \quad 1 \le i \le s, \tag{10.42}$$

so lauten die vereinfachten Bedingungen (10.37) für diese Verfahren

$$\sum_{i=1}^{s} b_i(z) = \varphi_1(z), \qquad \sum_{j=1}^{i-1} a_{ij}(z) = c_i \varphi_1(c_i z), \quad 1 \le i \le s. \tag{10.43}$$

Die Bedingung (10.43) impliziert $c_1 = 0$ und damit $Y_{n1} = y_n$.

Für die Implementierung ist es wichtig, die Produkte von Funktionen von Matrizen mit Vektoren effizient zu berechnen. Hier hilft die folgende Umformulierung, bei der die Vektoren $g_n(Y_{nj})$ in der Form

$$g_n(Y_{nj}) = g_n(y_n) + D_{nj}, \qquad 2 \le j \le s$$

geschrieben werden. Mit den vereinfachten Bedingungen (10.43) kann das Verfahren (10.41) äquivalent als

$$Y_{ni} = y_n + c_i h_n \varphi_1(c_i h_n J_n) f(y_n) + h_n \sum_{j=2}^{i-1} a_{ij}(h_n J_n) D_{nj}, \tag{10.44a}$$

$$y_{n+1} = y_n + h_n \varphi_1(h_n J_n) f(y_n) + h_n \sum_{i=2}^{s} b_i(h_n J_n) D_{ni}. \tag{10.44b}$$

geschrieben werden. Die Hauptmotivation dieser Umformulierung ist, dass die Vektoren D_{ni} kleine Norm haben. Wenn die Anwendung der Matrixfunktionen auf diese Vektoren mit Krylov-Verfahren approximiert wird, sollte dies in einem niedrigdimensionalen Raum möglich

sei. Daher ist in jedem Zeitschritt nur eine teure Krylov-Approximation nötig, nämlich die mit dem Vektor $f(y_n)$.

Um die Ordnung der Verfahren zu analysieren, betrachten wir jetzt die Defekte. Wir schreiben abkürzend $G_n(t) = g_n(y(t))$. Die Herleitung basiert auf der Darstellung der exakten Lösung mit Hilfe der Variation-der-Konstante Formel

$$y(t_n + \theta h) = e^{\theta h J_n} y(t_n) + \int_0^{\theta h} e^{(\theta h - \tau) J_n} G_n(t_n + \tau)\, \mathrm{d}\tau. \tag{10.45}$$

Entwicklen wir G in eine Taylor-Reihe mit Restglied in Integralform, so erhalten wir

$$G_n(t_n + \tau) = \sum_{j=1}^{q} \frac{\tau^{j-1}}{(j-1)!} G_n^{(j-1)}(t_n) + \int_0^{\tau} \frac{(\tau - \sigma)^{q-1}}{(q-1)!} G_n^{(q)}(t_n + \sigma)\, \mathrm{d}\sigma. \tag{10.46}$$

Wir definieren nun ganze Funktionen

$$\varphi_j(z) = \int_0^1 e^{(1-\tau)z} \frac{\tau^{j-1}}{(j-1)!}\, \mathrm{d}\tau, \qquad j \geq 1. \tag{10.47}$$

Es gilt $\varphi_0(z) = e^z$ und die Rekursion

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - 1/k!}{z}, \qquad \varphi_k(0) = \frac{1}{k!}, \qquad k \geq 0 \tag{10.48}$$

(Übung). Einsetzen in die rechte Seite von (10.45) liefert

$$\begin{aligned} y(t_n + c_i h) = e^{c_i h J_n} y(t_n) &+ \sum_{j=1}^{q_i} (c_i h)^j \varphi_j(c_i h J_n) G_n^{(j-1)}(t_n) \\ &+ \int_0^{c_i h} e^{(c_i h - \tau) J_n} \int_0^{\tau} \frac{(\tau - \sigma)^{q_i - 1}}{(q_i - 1)!} G_n^{(q_i)}(t_n + \sigma)\, \mathrm{d}\sigma\, \mathrm{d}\tau, \end{aligned} \tag{10.49}$$

während Einsetzen der exakten Lösung in das numerische Verfahren die Darstellung

$$y(t_n + c_i h) = e^{c_i h J_n} y(t_n) + h \sum_{j=1}^{i-1} a_{ij}(h J_n) G_n(t_n + c_j h) + \Delta_{ni}, \tag{10.50a}$$

$$y(t_{n+1}) = e^{h J_n} y(t_n) + h \sum_{i=1}^{s} b_i(h J_n) G_n(t_n + c_i h) + \delta_{n+1} \tag{10.50b}$$

mit Defekten Δ_{ni} und δ_{n+1} ergibt. Durch Einsetzen von (10.46) in (10.50a) erhalten wir

$$\begin{aligned} y(t_n + c_i h) = e^{c_i h J_n} y(t_n) &+ h \sum_{k=1}^{i-1} a_{ik}(h J_n) \sum_{j=1}^{q_i} \frac{(c_k h)^{j-1}}{(j-1)!} G_n^{(j-1)}(t_n) \\ &+ h \sum_{k=1}^{i-1} a_{ik}(h J_n) \int_0^{c_k h} \frac{(c_k h - \sigma)^{q_i - 1}}{(q_i - 1)!} G_n^{(q_i)}(t_n + \sigma)\, \mathrm{d}\sigma + \Delta_{ni}. \end{aligned} \tag{10.51}$$

Subtrahieren wir (10.49) von (10.51), so ergibt sich die folgende explizite Darstellung der Defekte

$$\Delta_{ni} = h_n \psi_{1,i}(h_n J_n) G_n(t_n) + h_n^2 \psi_{2,i}(h_n J_n) G_n'(t_n) + \Delta_{ni}^{[2]}, \tag{10.52}$$

mit Restgliedern $\Delta_{ni}^{[2]}$, wobei

$$\psi_{j,i}(z) = \varphi_j(c_i z)c_i^j - \sum_{k=1}^{i-1} a_{ik}(z)\frac{c_k^{j-1}}{(j-1)!} \tag{10.53}$$

und

$$\left\|\Delta_{ni}^{[2]}\right\| \le Ch_n^3. \tag{10.54}$$

Wegen (10.43) haben wir $\psi_{1,i}(z) \equiv 0$, aber für explizite Verfahren ist es *nicht* möglich, $\psi_{2,i}(z) \equiv 0$ für alle i zu erfüllen.

Durch die Linearisierung erhalten wir wegen $\frac{\partial g_n}{\partial y}(y_n) = 0$ die Darstellung

$$G_n'(t_n) = \frac{\partial g_n}{\partial y}\big(y(t_n)\big)y'(t_n) = \left(\frac{\partial g_n}{\partial y}\big(y(t_n)\big) - \frac{\partial g_n}{\partial y}(y_n)\right)y'(t_n).$$

Hieraus folgt

$$\|G_n'(t_n)\| \le C\|e_n\| \tag{10.55}$$

mit $e_n = y_n - y(t_n)$, so dass die Defekte der internen Stufen durch

$$\|\Delta_{ni}\| \le Ch^2\|e_n\| + Ch^3 \tag{10.56}$$

beschränkt sind.

Analog gilt für die Defekte zur Zeit t_{n+1}

$$\delta_{n+1} = \sum_{j=1}^{q} h^j \psi_j(hJ_n)G_n^{(j-1)}(t_n) + \delta_{n+1}^{[q]}, \tag{10.57}$$

wobei

$$\psi_j(z) = \varphi_j(z) - \sum_{k=1}^{s} b_k(z)\frac{c_k^{j-1}}{(j-1)!}. \tag{10.58}$$

Die Restglieder $\delta_{n+1}^{[q]}$ sind durch

$$\left\|\delta_{n+1}^{[q]}\right\| \le Ch^{q+1} \tag{10.59}$$

beschränkt. Wegen (10.43) gilt $\psi_1(z) \equiv 0$. Der h^2-Term in (10.57) ist klein wegen (10.55). Weitere Terme verschwinden, wenn wir $\psi_j = 0$, $j \ge 3$ fordern.

Die Gewichte $b_i(z)$ sind also Linearkombinationen der Funktionen $\varphi_k(z)$ und die Koeffizienten $a_{ij}(z)$ sind Linearkombinationen von $\varphi_k(c_i z)$. Die Ordnungsbedingungen sind in Tabelle 10.1 zusammengefasst.

Wir nehmen im Folgenden an, dass

$$f(y) = Ay + g(y) \tag{10.60}$$

mit Lipschitz-stetiger Funktion g gilt. Dann ist die Jacobi-Matrix $J(y) = Df(y)$ ebenfalls Lipschitz-stetig,

$$\|J(u) - J(v)\| \le C\|u - v\|. \tag{10.61}$$

Unter gewissen Voraussetzungen kann man zeigen, dass das Verfahren mit der Ordnung p konvergiert unabhängig von $\|J_n\|$.

No.	Bedingung im Defekt	Ordnungbedingung	order
1	$\psi_1(z) \equiv 0$	$\sum_{i=1}^{s} b_i(z) = \varphi_1(z)$	1
2	$\psi_{1,i}(z) \equiv 0$	$\sum_{j=1}^{i-1} a_{ij}(z) = c_i\varphi_1(c_i z), \quad 2 \le i \le s$	2
3	$\psi_3(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z)c_i^2 = 2\varphi_3(z)$	3
4	$\psi_4(z) \equiv 0$	$\sum_{i=2}^{s} b_i(z)c_i^3 = 6\varphi_4(z)$	4

Tabelle 10.1: Steife Ordnungsbedingungen für exponentielle Rosenbrock-Verfahren für autonome Probleme.

Lemma 10.27. *Wenn die Ordnungsbedingungen aus 10.1 bis zur Ordnung $p \le 4$ erfüllt sind, so gilt*

$$\|\delta_{n+1}\| \le Ch^2\|e_n\| + Ch^{p+1}. \tag{10.62}$$

Beweis. Dies folgt aus (10.57) und (10.59). □

Wir betrachten nun die Fehler

$$e_n = y_n - y(t_n) \quad \text{und} \quad E_{ni} = Y_{ni} - y(t_n + c_i h).$$

Subtrahieren wir (10.50) von der numerischen Lösung (10.41), so ergibt sich die Fehlerrekursion

$$E_{ni} = e^{c_i h J_n} e_n + h \sum_{j=1}^{i-1} a_{ij}(hJ_n)\left(g_n(Y_{nj}) - G_n(t_n + c_j h)\right) - \Delta_{ni}, \tag{10.63a}$$

$$e_{n+1} = e^{hJ_n} e_n + h \sum_{i=1}^{s} b_i(hJ_n)\left(g_n(Y_{ni}) - G_n(t_n + c_i h)\right) - \delta_{n+1}. \tag{10.63b}$$

Hierfür leiten wir jetzt Fehlerabschätzungen unter der Voraussetzung her, dass die rechte Seite f der Differentialgleichung die einseitige Lipschitz-Bedingung (10.27) erfüllt.

Lemma 10.28. *Unter der Voraussetzung (10.27) gilt*

$$\|e^{tJ(y)}\| \le 1 \qquad \textit{für alle} \qquad y \in \mathbb{R}^d. \tag{10.64}$$

Beweis. Wir wählen $y \in \mathbb{R}^d$ beliebig und definieren beliebige Vektoren x, z so, dass y auf der Strecke von x nach z liegt, d. h. $x = y + \epsilon v$ und $z = y - \epsilon v$ für ein $\epsilon \in \mathbb{R}$ und $v \in \mathbb{R}^d$. Nach dem Satz von Taylor gilt

$$f(x) - f(z) = \int_0^1 J\big(\theta x + (1-\theta)z\big)(x - z)d\theta.$$

Die einseitige Lipschitz-Bedingung liefert

$$\begin{aligned} 0 &\ge \langle f(x) - f(z), x - z\rangle \\ &= \int_0^1 \langle J\big(\theta x + (1-\theta)z\big)(x - z), x - z\rangle d\theta \\ &= 4\epsilon^2 \int_0^1 \langle J\big(y + (2\theta - 1)\epsilon v\big)v, v\rangle d\theta. \end{aligned}$$

Teilen wir dies durch $4\epsilon^2 > 0$ und betrachten den Grenzübergang $\epsilon \to 0$, so folgt nach Wahl von v

$$\langle J(y)v, v\rangle \le 0 \qquad \text{für alle } y, v \in \mathbb{R}^d.$$

Die Differentialgleichung $z' = J(y)z$ ist also kontraktiv und daraus folgt die Behauptung. □

Lemma 10.29. *Unter den Voraussetzung (10.27) und (10.61) gelten folgende Abschätzungen*

$$\|g_n(Y_{ni}) - G_n(t_n + c_i h)\| \le C\,(h + \|e_n\| + \|E_{ni}\|)\,\|E_{ni}\|, \tag{10.65a}$$

$$\|g_n(y_n) - G_n(t_n)\| \le C\|e_n\|^2, \tag{10.65b}$$

$$\left\| \frac{\partial g_n}{\partial u}\,(y(t_n)) \right\| \le C\|e_n\|. \tag{10.65c}$$

Beweis. Die Abschätzung (10.65c) folgt direkt aus der Linearisierung und der Lipschitz-Bedingung (10.61). Mit Taylor-Entwicklung erhalten wir

$$g_n(Y_{ni}) - G_n(t_n + c_i h) = \frac{\partial g_n}{\partial u}\,(y(t_n + c_i h))\,E_{ni} + \int_0^1 (1-\tau)\frac{\partial^2 g_n}{\partial u^2}\,(y(t_n + c_i h) + \tau E_{ni})\,(E_{ni}, E_{ni})\,\mathrm{d}\tau.$$

Für $i = 1$ ergibt sich (10.65b). Um (10.65a) zu zeigen entwickeln wir den ersten Terme auf der rechten Seite um t_n und verwenden die Relation

$$\frac{\partial g_n}{\partial u}\,(y(t_n)) = -\int_0^1 \frac{\partial^2 g_n}{\partial u^2}\,(y(t_n) + \tau e_n)\,e_n\,\mathrm{d}\tau.$$

Dies beweist (10.65a). □

Wir können nun eine Fehlerschranke für die inneren Stufen angeben.

Lemma 10.30. *Unter den Voraussetzung von Lemma 10.29 gilt*

$$\|E_{ni}\| \le C\|e_n\| + Ch^3,$$

wenn die globalen Fehler e_n hinreichend klein sind.

Beweis. Die Behauptung folgt aus (10.63a), Lemma 10.29 und (10.56). □

Damit ergibt sich das folgende Konvergenzresultat:

Satz 10.31. *Das Anfangswertproblem (10.1) erfülle die Voraussetzungn von Lemma 10.29 und das explizite exponentielle Rosenbrock-Verfahren (10.41) erfülle die Ordnungsbedingungen aus Tabelle 10.1 bis zur Ordnung p mit $2 \le p \le 4$. Dann gilt für den Fehler*

$$\|y_n - y(t_n)\| \le C \sum_{j=0}^{n-1} h^{p+1} \tag{10.66}$$

gleichmäßig im Intervall $t_0 \le t_n \le T$.

Beweis. Aus (10.63b) erhalten wir die Fehlerrekursion

$$e_{n+1} = e^{hJ_n} e_n + h\varrho_n - \delta_{n+1}, \qquad e_0 = 0, \tag{10.67}$$

mit

$$\varrho_n = \sum_{i=1}^{s} b_i(hJ_n)\left(g_n(Y_{ni}) - G_n(t_n + c_i h)\right).$$

Auflösen der Fehlerrekursion liefert wegen $e_0 = 0$

$$e_n = \sum_{j=0}^{n-1} h\, e^{hJ_{n-1}} \cdots e^{hJ_{j+1}} \left(\varrho_j - h^{-1}\delta_{j+1}\right). \tag{10.68}$$

Aus Lemma 10.27, 10.29 und 10.30 ergibt sich die Schranke

$$\|\varrho_j\| + h^{-1}\|\delta_{j+1}\| \le C\left(h\|e_j\| + \|e_j\|^2 + h^p\right). \tag{10.69}$$

Mit dieser Abschätzung, Gleichung (10.68) und dem Stabilitätslemma 10.28 folgt

$$\|e_n\| \le C \sum_{j=0}^{n-1} h\left(\|e_j\|^2 + h\|e_j\| + h^p\right). \tag{10.70}$$

Hierauf kann man nun das Gronwall-Lemma (eine Abschätzung wie sie am Ende vom Beweis von Satz 9.9 verwendet wurde) anwenden, und erhält die gewünschte Schranke (10.66). □

Beispiel.

(a) Das exponentielle Euler-Verfahren

$$\begin{aligned} y_{n+1} &= e^{h_n J_n} y_n + h_n \varphi_1(h_n J_n) g_n(y_n) \\ &= y_n + h_n \varphi_1(h_n J_n) f(y_n) \end{aligned} \tag{10.71}$$

konvergiert mit der Ordnung 2.

(b) Das Verfahren `exprb32` besteht aus einem exponentiellen Rosenbrock-Verfahren 3. Ordnung mit einem eingebetteten Verfahren 2. Ordnung zur Fehlerschätzung (dem exponentiellen Rosenbrock–Euler Verfahren). Die Koeffizienten sind gegeben durch

$$\begin{array}{c|cc} c_1 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \\ & \widehat{b}_1 & \end{array} \quad = \quad \begin{array}{c|cc} 0 & & \\ 1 & \varphi_1 & \\ \hline & \varphi_1 - 2\varphi_3 & 2\varphi_3 \\ & \varphi_1 & \end{array}$$

(c) Das Verfahren `exprb43` hat die Ordnung 4 mit einem Fehlerschätzer der Ordnung 3:

$$\begin{array}{c|ccc} c_1 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \\ & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 \end{array} \quad = \quad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2}\varphi_1\left(\frac{1}{2}\cdot\right) & & \\ 1 & 0 & \varphi_1 & \\ \hline & \varphi_1 - 14\varphi_3 + 36\varphi_4 & 16\varphi_3 - 48\varphi_4 & -2\varphi_3 + 12\varphi_4 \\ & \varphi_1 - 14\varphi_3 & 16\varphi_3 & -2\varphi_3 \end{array}$$

Verfahren höherer Ordnung als 4 zu analysieren ist äußerst schwierig. ⋄

Zur Implementierung von exponentiellen Integratoren müssen Produkte von Funktionen von Matrizen mit Vektoren berechnet werden. Wir geben zunächst verschiedene Definitionen von Matrixfunktionen an. Im Folgenden sei ϕ eine Funktion, die analytisch in einer Umgebung des Spektrums der Matrix $A \in \mathbb{C}^{d,d}$ ist. A habe die Jordan-Normalform

$$Z^{-1}AZ = J = \operatorname{diag}(J_1, \ldots, J_p),$$
$$J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k,m_k},$$

wobei Z nicht singulär ist und $m_1 + \ldots + m_p = d$ gilt. n_k sei die Dimension des größten Jordan-Kästchens zum Eigenwert λ_k, $k = 1, \ldots, s$, wobei s die Anzahl der paarweise verschiedenen Eigenwerte von A ist.

Definition 10.32. *Die Matrixfunktion $\phi(A)$ ist definiert als*

$$\phi(A) := Z\phi(J)Z^{-1} = Z\operatorname{diag}\big(\phi(J_1), \ldots, \phi(J_p)\big)Z^{-1}, \tag{10.72}$$

wobei

$$\phi(J_k) = \phi\big(J_k(\lambda_k)\big) = \begin{bmatrix} \phi(\lambda_k) & \phi'(\lambda_k) & \cdots & \frac{\phi^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & \phi(\lambda_k) & \ddots & \vdots \\ & & \ddots & \phi'(\lambda_k) \\ & & & \phi(\lambda_k) \end{bmatrix} \in \mathbb{C}^{m_k,m_k}.$$

Definition 10.33. *Es sei ψ das Minimalpolynom von A. Die Matrixfunktion $\phi(A)$ ist definiert als*

$$\phi(A) := p(A),$$

wobei p das Hermite-Interpolationspolynom vom Grad kleiner als $\deg \psi = n_1 + \ldots + n_s$ ist, d. h.

$$p^{(j)}(\lambda_k) = \phi^{(j)}(\lambda_k), \qquad j = 0, \ldots, n_k - 1, \qquad k = 1, \ldots, s. \tag{10.73}$$

Die aus der Funktionentheorie bekannte Cauchy-Integralformel besagt, dass für eine in einem Gebiet $\Omega \subset \mathbb{C}$ analytische Funktion

$$f(a) = \frac{1}{2\pi i} \int_\Gamma f(\lambda)(\lambda - a)^{-1} d\lambda \tag{10.74}$$

gilt, wobei Γ eine einfach geschlossene, positiv orientierte, Kurve in Ω und $a \in \mathbb{C}$ ein Punkt ist, der nicht auf Γ liegt.

Die Cauchy-Integralformel kann auch auf quadratische Matrizen und sogar Operatoren verallgemeinert werden.

Definition 10.34. *Die Matrixfunktion $\phi(A)$ ist definiert als*

$$\phi(A) := \frac{1}{2\pi i} \int_\Gamma \phi(z)(zI - A)^{-1} dz, \tag{10.75}$$

wobei Γ eine einfach geschlossene, positiv orientierte, Kurve ist, die das Spektrum von A so umschliesst, dass ϕ analytisch auf und innerhalb von Γ ist.

Satz 10.35. *Die Defintionen 10.32, 10.33 und 10.34 sind äquivalent.*

Beweis. Die Äquivalenz von Definition 10.32 und 10.33 überlassen wir als Übung.

Die Äquivalenz von 10.32 und 10.34 zeigen wir nur für den Spezialfall, dass A diagonalisierbar ist, also $X^{-1}AX = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ gilt. Viel allgemeiner kann sie sogar für unbeschränkte Operatoren mit funktionalanalytischen Mitteln gezeigt werden. Wir wollen hier aus Zeitgründen aber auf einen Beweis verzichten und es bei der Motivation belassen.

Nach Definition 10.32 ist

$$\phi(A) = X\phi(\Lambda)X^{-1}.$$

Indem wir für jedes $\phi(\lambda_j)$ die Cauchy-Integralformel (10.74) anwenden ergibt sich

$$\phi(A) = X\operatorname{diag}\left(\frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda-\lambda_1)^{-1}, \ldots, \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda-\lambda_n)^{-1}, d\lambda\right) X^{-1}.$$

Die Kurve Γ muss so gewählt werden, dass sie alle Eigenwerte von A umschließt. Aus

$$X\operatorname{diag}\Big((\lambda-\lambda_1)^{-1}, \ldots, (\lambda-\lambda_n)^{-1}\Big)X^{-1} = (\lambda I - A)^{-1}$$

folgt dann

$$\phi(A) = \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda I - A)^{-1} d\lambda. \tag{10.76}$$

Die Umkehrung gilt ebenfalls. □

Ist A groß und dünn besetzt, dann ist im Allgemeinen A^j für große j nicht mehr dünn besetzt und damit gilt dasselbe auch für $\phi(A)$. In Anwendungen tritt nicht nur der in Kapitel 7 betrachtete Fall $\phi(\lambda) = \lambda^{-1}$ (Lösung des linearen Gleichungssystems $Ax = b$) auf, sondern auch andere Funktionen wie $\phi(\lambda) = \exp(\lambda)$, $\phi(\lambda) = \varphi_k(\lambda)$ oder auch trigonometrische Funktionen. Allerdings muss man – wie bei exponentiellen Integratoren – glücklicherweise meist nicht $\phi(A)$ selbst berechnen, sondern nur $\phi(A)b$.

Um Approximationen an solche Matrixfunktionen zu bekommen, nutzt man aus, dass in der Cauchy-Integralformel

$$\phi(A)b = \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda I - A)^{-1} b d\lambda = \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)x(\lambda)d\lambda \tag{10.77}$$

für jedes $\lambda \in \Gamma$ die Lösung eines linearen Gleichungssystems auftritt, nämlich

$$(\lambda I - A)x(\lambda) = b. \tag{10.78}$$

Dieses können wir mit einem Krylov-Verfahren näherungsweise lösen. Hier betrachten wir exemplarisch das Arnoldi-Verfahren, aber mit dem Lanczos-Verfahren geht es analog. Es gilt dann

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \qquad V_m^H V_m = I_m,$$

wobei V_m eine Basis von $\mathcal{K}_m(A, b)$ ist. Aus der Definition eines Krylov-Raums folgt, dass $\mathcal{K}_m(A, b) = \mathcal{K}_m(\lambda I - A, b)$ für jedes $\lambda \in \mathbb{C}$ gilt. Daher ist V_m auch Basis all dieser verschobenen Krylov-Räume und es gilt

$$(\lambda I - A)V_m = V_m(\lambda I - H_m) - h_{m+1,m} v_{m+1} e_m^T.$$

Die Galerkin-Iterierte zu (10.78) ist

$$x_m(\lambda) = V_m(\lambda I - H_m)^{-1}(\beta e_1). \tag{10.79}$$

Wählen wir die Kurve Γ so, dass Sie den Wertebereich $\mathcal{F}(A)$ im Inneren enthält, dann enthält sie wegen $\mathcal{F}(H_m) \subset \mathcal{F}(A)$ erst recht den Wertebereich von H_m im Inneren. Die Galerkin-Iterierte existiert daher für jedes m (denn die Eigenwerte von H_m liegen im Wertebereich und daher nicht auf der Kurve Γ). Eine Approximation an $\phi(A)b$ erhält man nun, indem man in der Integralformel die Näherung $x_m(\lambda)$ einsetzt:

$$\begin{aligned}
\phi(A)b &\approx \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)x_m(\lambda)d\lambda \\
&= \frac{1}{2\pi i}\int_\Gamma \phi(\lambda)V_m(\lambda I - H_m)^{-1}(\beta e_1)d\lambda \\
&= V_m\frac{1}{2\pi i}\int_\Gamma \phi(\lambda)(\lambda I - H_m)^{-1}d\lambda\ (\beta e_1) \\
&= V_m\phi(H_m)\beta e_1,
\end{aligned} \tag{10.80}$$

denn es tritt hier wieder eine Cauchy-Integralformel auf, nur dieses Mal mit der Matrix H_m. H_m ist nun eine Matrix kleiner Dimension, von der man $\phi(H_m)$ zum Beispiel durch Diagonalisierung oder durch Padé-Approximation direkt berechnen kann. Wir haben diese Approximation zwar über die Lösung linearer Gleichungssysteme hergeleitet, aber diese wurden hier nur für die Herleitung der Approximation benötigt. In der Praxis berechnet man lediglich $\phi(H_m)$.

Alternativ kann man auf Kosten der Lösung einiger linearer Gleichungssysteme $x(\theta_k)$ für gewisse Knoten θ_k berechnen und dann eine Quadraturformel zur Approximation des Integrals verwenden.