Computing 29, 153—165 (1982)

Computing

Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge-Kutta-Methoden

K. Strehmel und **R. Weiner,** Halle a. d. Saale

Eingegangen am 17. März 1981

Zusammenfassung — Abstract

Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge-Kutta-Methoden. Es werden adaptive Runge-Kutta-Verfahren betrachtet und Stabilitätsuntersuchungen für diese linear impliziten Methoden durchgeführt. Für die Anwendung wird ein LS-stabiles Verfahren vierter Ordnung mit einer angepaßten Schrittweitenkontrolle vorgeschlagen. Testergebnisse von 25 stiff Anfangswertproblemen für verschiedene Toleranzen werden diskutiert.

Treatment of Stiff Initial Value Problems for Ordinary Differential Equations with Adaptive Runge-Kutta-Methods. Adaptive Runge-Kutta-methods are considered. Investigations of stability for these linear implicit methods are studied. For the application a LS-stable method of order four with an adaptive stepsize control is proposed. Test results for 25 stiff initial value problems for different tolerances are discussed.

AMS Subject Classifications: 65L05; CR: 5.17.

Key words: Numer. Analysis, verallgemeinerte Runge-Kutta-Methoden, steife Probleme.

1. Einleitung

Explizite Runge-Kutta-Verfahren (RK-Verfahren) sind aufgrund ihrer endlichen Gebiete der absoluten Stabilität zur Lösung steifer Anfangswertaufgaben gewöhnlicher Differentialgleichungen nicht geeignet. Implizite RK-Verfahren besitzen die besten Stabilitätseigenschaften (vgl. Hairer/Wanner [5]), sind jedoch schwierig zu implementieren und erfordern wegen der zu lösenden nichtlinearen Gleichungssysteme mittels Newton-Methoden einen beträchtlichen Aufwand. Wesentliche Fortschritte bezüglich der Implementierung wurden in den letzten Jahren durch Butcher, Burrage und Chipman [1], [2] erreicht. Die von ihnen betrachtete Familie von singly-impliziten RK-Verfahren gestattet eine einfache Schätzung des lokalen Diskretisierungsfehlers sowie eine variable Ordnungssteuerung. Ihr numerischer Aufwand entspricht dem von singly diagonal impliziten RK-Verfahren.

Verallgemeinerungen von expliziten RK-Verfahren (modifizierte Rosenbrock Methoden, genannt ROW-Methoden), wie sie von Kaps/Rentrop [7], Kaps/Wanner [8], Nørsett/Wolfbrandt [10] betrachtet wurden, benötigen in jedem Integrations-

0010-485X/82/0029/0153/$ 02.60

schritt einen Aufruf der Jacobi-Matrix des Systems und eine Matrixfaktorisierung. Um den numerischen Aufwand noch weiter zu senken, ohne einen wesentlichen Verlust an Genauigkeit und Stabilität in Kauf nehmen zu müssen, untersuchten Steighaug/Wolfbrandt [12] die ROW-Methoden, wenn keine exakten Jacobi-Matrizen zugrunde gelegt werden (W-Methoden). Verwer/Scholz [14] geben eine Klasse modifizierter Rosenbrock Methoden an, bei denen exakte aber zeitverzögerte Jacobi-Matrizen verwendet werden.

Die vorliegende Arbeit befaßt sich mit einer Klasse verallgemeinerter RK-Verfahren, die auf einer geeigneten Linearisierung des vorgelegten Differentialgleichungssystems beruht (vgl. Friedli [4]). Sie erfordert keine exakte Jacobi-Matrix, da die Konsistenzordnung im allgemeinen von der verwendeten Matrix unabhängig ist. Aus Stabilitätsgründen ist es jedoch zweckmäßig, eine Approximation an die Jacobi-Matrix zu verwenden. Es wird zunächst ein einfacheres Konstruktionsprinzip als in [4] vorgestellt, welches den Vorteil besitzt, daß sich die Anzahl der Bedingungsgleichungen für die Konsistenzordnung eines Verfahrens erheblich reduziert. Außer einigen Stabilitätsuntersuchungen dieser adaptiven RK-Verfahren wird ein LS-stabiler Algorithmus, der aus einem LS-stabilen Verfahren 4. Ordnung mit einem eingebetteten LS-stabilen Verfahren 3. Ordnung besteht, an 25 stiff Beispielen von Enright/Hull/Lindberg [3] getestet, so daß ein Vergleich bezüglich des numerischen Aufwandes mit den ROW-Methoden von Kaps/Rentrop [7] möglich ist.

2. Adaptive RK-Verfahren

Betrachtet wird ein Anfangswertproblem für ein stiff System gewöhnlicher Differentialgleichungen 1. Ordnung

$$y' = f(x, y), \quad x \in [x_0, x_e] \tag{2.1}$$

$$y(x_0) = y_0, \tag{2.2}$$

in einem n-dimensionalen reellen Vektorraum. Die eindeutige Lösbarkeit von (2.1), (2.2) wird vorausgesetzt.

Ein s-stufiges RK-Verfahren

$$\begin{aligned} u_{m+1}^{(1)} &:= u_m \ ^1 \\ u_{m+1}^{(i)} &:= u_m + h \sum_{j=1}^{i-1} a_{ij} f(x_m + c_j h, u_{m+1}^{(j)}), \quad i = 2, 3, \ldots, s+1 \\ u_{m+1} &:= u_{m+1}^{(s+1)} \qquad (c_{s+1} := 1), \end{aligned} \tag{2.3}$$

wobei der Knotenvektor $C := (0, c_2, \ldots, c_s)^T$ der Bedingung $0 < c_i \leqq 1$, $i = 2, 3, \ldots, s$ genügt, beruht im Prinzip auf der Lösung der zu (2.1), (2.2) äquivalenten Volterra-Integralgleichung

$$y(x) = y(x_m) + \int_{x_m}^{x} f(t, y(t))\, dt, \quad x \in I_m := [x_m, x_m + h],$$

[1] u_m sind Näherungswerte für $y(x_m)$ an den Stellen $x_m = x_0 + m \cdot h$.

indem in jeder Stufe die Funktion $f(x, y)$ durch ein Vektorpolynom approximiert wird.

Für die Konstruktion adaptiver RK-Verfahren wird (2.1) zunächst in jedem Diskretisierungsintervall I_m formal linearisiert in der Form

$$y' = T_m y + g(x, y) \quad \text{mit} \quad g(x, y) = f(x, y) - T_m y, \tag{2.4}$$

wobei T_m eine auf I_m beliebige konstante (n, n)-Matrix ist (vgl. Friedli [4]). Die Funktion $g(x, y(x))$ wird nun für $x \in I_m$ in jeder Stufe $i = 2, 3, \ldots, s+1$, durch ein Polynom

$$g_h^{(i)}(x) = \sum_{l=0}^{\rho_i} a_l^{(i)} (x - x_m)^l, \quad \rho_i \leqq \rho_{i+1} \tag{2.5}$$

mit

$$h^l a_l^{(i)} = \sum_{j=1}^{i-1} \lambda_{lj}^{(i)} g_j, \quad l = 0, 1, \ldots, \rho_i; \quad \lambda_{lj}^{(i)} \in R \tag{2.6}$$

approximiert, wobei zur Abkürzung $g_j := g(x_m + c_j h, u_{m+1}^{(j)})$ gesetzt wurde, und die Vektorkoeffizienten $a_l^{(i)}$ für $h \to 0$ normmäßig beschränkt sind. Mit (2.5) erhält man nun aus (2.4) durch Integration für (2.1), (2.2) die Näherungslösung

$$u_{m+1}^{(i)} = e_0(c_i h \cdot T_m) u_m + h \sum_{l=0}^{\rho_i} e_{l+1}(c_i h \cdot T_m) a_l^{(i)} h^l c_i^{l+1}, \quad i = 2, 3, \ldots, s+1. \tag{2.7}$$

Die Matrizenfunktionen $e_l(B)$ sind dabei definiert durch

$$e_0(B) := \exp(B); \quad e_1(B) := [e_0(B) - I] B^{-1}$$
$$e_{l+1}(B) := [l \cdot e_l(B) - I] B^{-1}, \quad l = 1, 2, \ldots, \rho_{s+1},$$

wobei I die (n, n)-Einheitsmatrix bezeichnet. Für eine numerische Rechnung ist die Darstellung (2.7) jedoch kaum geeignet, da sie die Berechnung der Exponentialmatrix erfordert (vgl. Moler/Van Loan [9]). Außerdem besitzen die Funktionen $e_l(B)$, $l = 1, 2, \ldots, \rho_{r+1} + 1$, für $\det(B) = 0$ eine numerische Singularität, was für $\det(B) \to 0$ zu weiteren numerischen Schwierigkeiten führt. Daher approximieren wir $e_0(B)$ durch eine rationale Matrizenfunktion $R_0(B)$ der Ordnung $k \geqq \rho_{s+1}$, d. h. es gilt

$$e_0(z) = R_0(z) + 0(z^{k+1}) \quad \text{für } z \to 0, \ z \in \mathbb{C}.$$

Für die Funktionen $e_l(B)$, $l = 1, \ldots, \rho_{s+1}$ erhält man dann die Approximationen

$$e_1(B) \approx [R_0(B) - I] B^{-1} =: R_1(B)$$
$$e_{l+1}(B) \approx [l \cdot R_l(B) - I] B^{-1} =: R_{l+1}(B), \quad l = 1, 2, \ldots, \rho_{s+1}.$$

Die Funktionen $R_l(z)$, $l = 0, 1, \ldots, k+1$, besitzen, wie man leicht sieht, den gleichen Nenner. Sie lassen sich daher für $\det(B) \to 0$ numerisch stabil berechnen. Aus (2.6) ergibt sich dann eine Näherungslösung (adaptives RK-Verfahren) für (2.1), (2.2) in der Form

$$\begin{aligned} u_{m+1}^{(1)} &= u_m \\ u_{m+1}^{(i)} &= R_0(c_i h \cdot T_m) u_m + h \sum_{j=1}^{i-1} A_{ij} g_j, \quad i = 2, 3, \ldots, s+1 \\ u_{m+1} &:= u_{m+1}^{(s+1)} \end{aligned} \tag{2.8}$$

mit

$$A_{ij}=\sum_{l=0}^{\rho_i} R_{l+1}(h c_i \cdot T_m)\,\lambda_{lj}^{(i)}\,c_i^{l+1}, \tag{2.9}$$

die für $T_m=0$ in das zugrunde gelegte explizite RK-Verfahren übergehen soll.

3. Stabilitätseigenschaften der adaptiven RK-Verfahren

Mit $T_m=f_y(x_m,u_m)$ ergibt sich sofort der

Satz 1: *Ein adaptives RK-Verfahren ist intern A bzw. L-stabil, wenn die Stabilitätsfunktion $R_0(z)$ A- bzw. L-verträglich ist.*

Wendet man (2.8) auf die Testdifferentialgleichung der S-Stabilität

$$y'=q\big(y-v(x)\big)+v'(x) \quad \text{mit} \quad \operatorname{Re} q\leqq q_0<0$$

an (vgl. Prothero/Robinson [11]), so erhält man (vgl. Strehmel [13]) den

Satz 2: *Ein adaptives RK-Verfahren ist genau dann S-stabil, wenn die Stabilitätsfunktion $R_0(z)$ der Ordnung $k\geqq\rho_{s+1}+1$ stark A-akzeptabel ist.*

Es seien $c_{j1}<c_{j2}<\ldots<c_{j\bar m}$ die voneinander verschiedenen Stützstellen c_i, $i=1,2,\ldots,s$, des expliziten RK-Verfahrens und $K_l\subset\{1,2,\ldots,s\}$ Indexmengen mit $K_l=\{i\,|\,c_i=c_{jl}\}$, $l=1,2,\ldots,\bar m$. Dann gilt der

Satz 3: *Ein adaptives RK-Verfahren mit einer L-verträglichen Stabilitätsfunktion $R_0(z)$ der Ordnung $k\geqq\rho_{s+1}+1$ ist genau dann LS-stabil, wenn die Beziehungen*

$$c_{j\bar m}=1,\ \sum_{j\in K_{\bar m}}\sum_{l=0}^{\rho_{s+1}}\lambda_{lj}^{(s+1)}=1,\ \sum_{j\in K_i}\sum_{l=0}^{\rho_{s+1}}\lambda_{lj}^{(s+1)}=0,\ i=1,2,\ldots,\bar m-1$$

bestehen.

Beweis: Aus (2.8) folgt mit $T_m=q$ für $\varepsilon_{m+1}:=u_{m+1}-v(x_{m+1})$ die Beziehung

$$\varepsilon_{m+1}=R_0(z)\,\varepsilon_m+h\,\beta_m(z,h,v) \tag{3.1}$$

mit

$$\beta_m(z,h,v)=\sum_{l=0}^{\rho_{s+1}}R_{l+1}(z)\sum_{j=1}^{s}\lambda_{lj}^{(s+1)}\left[v'(x_m+c_jh)-q\cdot v(x_m+c_jh)\right]+$$

$$\frac{1}{h}\left[R_0(z)\,v(x_m)-v(x_m+h)\right],\quad z=h\cdot q.$$

Aufgrund von Satz 2 ist $\beta_m(z,h,v)$ auf $M:=\{(h,z)\,|\,\operatorname{Re}(z)<0,\ h\in(0,h_0)\}$ gleichmäßig beschränkt. Nach (3.1) ist ein adaptives RK-Verfahren genau dann LS-stabil, wenn $R_0(z)$ L-verträglich ist und

$$\lim_{\operatorname{Re}(z)\to-\infty} h\,\beta_m(z,h,v)=0$$

gilt. Nach Voraussetzung folgt

$$\lim_{\operatorname{Re}(z)\to-\infty} R_l(z)=0 \text{ und } \lim_{\operatorname{Re}(z)\to-\infty} z\cdot R_{l+1}(z)=-1 \text{ für } l=0,1,\ldots,\rho_{s+1}.$$

Damit erhält man

$$\lim_{\mathrm{Re}(z)\to-\infty} h\beta_m(z,h,v)=\sum_{l=0}^{\rho_{s+1}}\sum_{j=1}^{s}\lambda_{lj}^{(s+1)}v(x_m+c_jh)-v(x_m+h),$$

woraus sich unmittelbar die Behauptung ergibt. □

Satz 4: *Das adaptive Euler-Verfahren*

$$u_{m+1}=R_0(h\cdot T_m)u_m+hR_1(h\cdot T_m)g(x_m,u_m) \tag{3.2}$$

ist AN- bzw. LN-stabil, wenn die Stabilitätsfunktion $R_0(z)$ der Konsistenzordnung $k\geqq 1$ A- bzw. L-verträglich ist.

Beweis: Bei Anwendung von (2.8) auf $y'=q(x)y$ folgt

$$u_{m+1}=R_0(h\cdot q(x_m))u_m,$$

was sofort die Behauptung liefert. □

Satz 5: *Ein konsistentes s-stufiges ($s\geqq 2$) adaptives RK-Verfahren mit $T_m=f_y(x_m,u_m)$ kann nicht AN-stabil sein.*

Beweis: Betrachtet wird das spezielle Anfangswertproblem

$$y'=-x^2y \quad \text{für} \quad x\in[0,x_e]$$
$$y(0)=1.$$

Aus (2.8) folgt mit $T_0=f_y(x_0,y_0)=0$ und $u_0=1$

$$u_1^{(i)}=1+h\sum_{j=1}^{i-1}a_{ij}f_j=1+h\sum_{j=1}^{i-1}a_{ij}c_j^2h^2u_1^{(j)}, \quad i=2,\ldots,s+1,$$

so daß

$$u_1=u_1^{(s+1)}=(1+P(h))$$

gilt, wobei $P(h)$ für $s\geqq 2$ ein Polynom vom Grad $\geqq 3$ mit $P(0)=0$ ist. Für genügend große h gilt $|u_1|>|u_0|$.

□

4. Konsistenzbedingungen

Die Konsistenzordnung p eines s-stufigen adaptiven RK-Verfahrens kann für beliebige Matrizen T_m nicht größer sein, als die des zugrunde gelegten s-stufigen expliziten RK-Verfahrens. Hingegen ist es möglich, daß adaptive RK-Verfahren bei Anwendung auf autonome Systeme für spezielle Matrizen T_m eine höhere Konsistenzordnung aufweisen. So besitzt z. B. das adaptive Euler-Verfahren (3.2) für $T_m=f_y(u_m+\frac{h}{3}\cdot f(u_m))$ die Ordnung $p=3$, falls $f(y)\in C^3(U)$, $U:=\{(x,y)\,|\,x_0\leqq x\leqq x_e, \|y-y(x)\|\leqq\rho\}$ ist und $k\geqq 3$ gilt (vgl. Verwer [15]). Hat ein adaptives RK-Verfahren für Differentialgleichungssysteme (2.1), (2.2) und beliebige Matrizen T_m die Konsistenzordnung p, aber für Anfangswertprobleme mit autonomer Funktion und $T_m=f_y(u_m)$ die Ordnung $p+1$, so kennzeichnen wir die Ordnung des adaptiven RK-Verfahrens mit p^+.

Ein s-stufiges adaptives RK-Verfahren muß außer den (erfüllten) Konsistenzbedingungen des zugrunde gelegten expliziten s-stufigen RK-Verfahrens noch zusätzliche Konsistenzbedingungen erfüllen. Da der numerische Aufwand eines adaptiven

Verfahrens (2.8) mit wachsendem Polynomgrad ρ_i zunimmt, wird man bestrebt sein, den Grad ρ_i in jeder Stufe $i=2,3,\ldots,s+1$ des Verfahrens so klein wie möglich zu halten, ohne aber einen Verlust an Genauigkeit und Stabilität in Kauf nehmen zu müssen. Im folgenden geben wir für

$$\begin{aligned}\rho_i &= 0 \quad \text{für} \quad i=2,3,\ldots,s-2\\ \rho_i &= \min(1,p-1) \quad \text{für} \quad i=s-1,s\\ \rho_{s+1} &= \min(2,p-1)\end{aligned}$$

die zusätzlichen Konsistenzbedingungen eines s-stufigen Verfahrens mit $p=1,2,3,4$ und $p^+=1^+,2^+,3^+$ an, wobei die Stabilitätsfunktion $R_0(z)$ eine Konsistenzordnung $k\geqq p$ besitzen soll.

Aus (2.8) und (2.9) folgt mit $T_m=0$

$$\lambda_{0j}^{(i)}=\frac{1}{c_i}\left[a_{ij}-\sum_{l=1}^{\rho_i}\frac{c_i^{l+1}}{l+1}\lambda_{lj}^{(i)}\right], \tag{4.1}$$

und aus (2.6) erhält man für $h\to 0$ die Bedingungsgleichungen

$$\begin{aligned}&\sum_{j=1}^{i-1}\lambda_{1j}^{(i)}=0 \qquad \text{für} \quad \rho_i\geqq 1\\ &\sum_{j=1}^{s}\lambda_{2j}^{(s+1)}c_j=0 \quad \text{für} \quad \rho_{s+1}=2\end{aligned} \tag{4.2}$$

Außer (4.2) bekommen wir dann, unter Berücksichtigung von (4.1), für die Koeffizienten $\lambda_{1j}^{(i)}$, $\lambda_{2j}^{(i)}$ eines s-stufigen adaptiven RK-Verfahrens die Konsistenzbedingungen:

p	Konsistenzbedingungen
1	keine
1^+	keine
2	keine
2^+	$\sum_{i=1}^{s} a_{s+1,i}c_i^2=\frac{1}{2}$ (zusätzliche Bedingung an das RK-Verfahren)
3	$\sum_{i=1}^{s}\lambda_{1i}^{(s+1)}c_i=1$
3^+	$\sum_{i=1}^{s} a_{s+1,i}c_i^3=\frac{1}{4}$ (zusätzliche Bedingung an das RK-Verfahren) $\sum_{i=1}^{s}(\lambda_{1i}^{(s+1)}+\lambda_{2i}^{(s+1)})c_i^2=1$
4	$\sum_{i=1}^{s}(\lambda_{1i}^{(s+1)}+\lambda_{2i}^{(s+1)})\sum_{j=1}^{i-1}a_{ij}c_j=\frac{1}{2}$ $a_{s+1,s-1}c_{s-1}^3\sum_{i=1}^{s}\lambda_{1i}^{(s-1)}c_i+a_{s+1,s}c_s^3\sum_{i=1}^{s}\lambda_{1i}^{(s)}c_i=\frac{1}{4}$

Soll das adaptive RK-Verfahren LS-stabil sein, so muß zusätzlich noch gelten:

$$c_{j\bar{m}}=1,\ \sum_{j\in K_i}(a_{s+1,j}+\tfrac{1}{2}\lambda_{1j}^{(s+1)}+\tfrac{2}{3}\lambda_{2j}^{(s+1)})=0,\ i=1,2,\ldots,\bar{m}-1$$

$$\sum_{j\in K_{\bar{m}}}(a_{s+1,s}+\tfrac{1}{2}\lambda_{1j}^{(s+1)}+\tfrac{2}{3}\lambda_{2j}^{(s+1)})=1.$$

Mit den aus den Bedingungsgleichungen sich ergebenden $\lambda_{ij}^{(i)}$ und den nach (2.8) folgenden Beziehungen

$$A_{ij}=a_{ij}R_1(c_i h\cdot T_m),\ i=2,3,\ldots,s-2,\ j=1,2,\ldots,i-1$$

$$A_{ij}=\left(a_{ij}-\frac{c_i^2}{2}\lambda_{1j}^{(i)}\right)R_1(c_i h\cdot T_m)+c_i^2\lambda_{1j}^{(i)}R_2(c_i h\cdot T_m),\ i=s-1,s;\quad j=1,2,\ldots,i-1,$$

$$A_{s+1,j}=\left(a_{s+1,j}-\frac{1}{2}\lambda_{1j}^{(s+1)}-\frac{1}{3}\lambda_{2j}^{(s+1)}\right)R_1(h\cdot T_m)+\lambda_{1j}^{(s+1)}R_2(h\cdot T_m)+\lambda_{2j}^{(s+1)}R_3(h\cdot T_m),\quad j=1,2,\ldots,s,$$

erhält man dann aus (2.8) zu einem vorgegebenen s-stufigen expliziten RK-Verfahren der Konsistenzordnung $p\leqq 4$ ein zugehöriges s-stufiges adaptives RK-Verfahren der Ordnung p oder p^+.

Durch analoges Vorgehen lassen sich adaptive RK-Verfahren höherer Konsistenzordnung erzeugen, wobei allerdings die Anzahl der benötigten LU-Zerlegungen pro Integrationsschritt (sie ist bei neuberechneter Matrix T_m gleich der Anzahl der voneinander verschiedenen c_i, $i=2,3,\ldots,s+1$, $c_{s+1}:=1$) zunimmt.

Zur Konstruktion von adaptiven RK-Verfahren der Ordnung $p=4$ sind die expliziten RK-Verfahren

0				
$\frac{1}{2}$	a_{21}			
$\frac{1}{2}$	a_{31}	a_{32}		
1	0	a_{42}	a_{43}	
	a_{51}	a_{52}	a_{53}	a_{54}

mit $p=4$ vorteilhaft, da bei Neuberechnung von T_m lediglich zwei LU-Zerlegungen pro Schritt erforderlich sind. Ein adaptives RK-Verfahren mit $p=4$ ist – unter der Voraussetzung, daß $R_0(z)$ eine Ordnung $k\geqq 4$ hat – durch das Parameterschema

0				
$\frac{1}{2}$	$\frac{1}{2}R_1$			
$\frac{1}{2}$	$\frac{1}{2}R_1+(2a_{31}-1)R_2$	$2a_{32}R_2$		
1	R_1-2R_2	$2a_{42}R_2$	$2a_{43}R_2$	
	$R_1+2(3a_{51}-2)R_2+3(1-2a_{51})R_3$	$6a_{52}(R_2-R_3)$	$6a_{53}(R_2-R_3)$	$2(3a_{54}-1)R_2+3(1-2a_{54})R_3$

(4.3)

gegeben, wobei in der i-ten Stufe, $i=2,3,\ldots,s+1$, zur Abkürzung $R_l:=R_l(c_i h\,T_m)$ gesetzt wurde. Werden in (4.3) die Matrizenfunktionen A_{5j}, $j=1,2,\ldots,4$, durch

$$A_{51}=2a_{51}(R_1-R_2);\quad A_{52}=2a_{52}(R_1-R_2)$$
$$A_{53}=2a_{53}(R_1-R_2);\quad A_{54}=2a_{54}(R_1-R_2)+2R_2-R_1 \tag{4.4}$$

ersetzt, so stellt (4.3) ein eingebettetes adaptives RK-Verfahren der Ordnung 3 dar. Beide Verfahren sind mit einer L-verträglichen Stabilitätsfunktion LS-stabil.

5. Numerische Implementierung

Für die numerische Rechnung wird der sich aus (4.3) und (4.4) – unter Zugrundelegung des klassischen expliziten RK-Verfahrens 4. Ordnung – ergebende Algorithmus verwendet. Er gestattet für stiff-Probleme mit nicht konstanter Jacobi-Matrix eine einfache Schrittweitensteuerung. Die Größe

$$\mathrm{EST}:=\max_{i=1}^{n}\frac{|u_i(x_m)-\bar{u}_i(x_m)|}{S_i},\quad m=1,2,\ldots, \tag{5.1}$$

wobei $u_i(x_m)$ bzw. $\bar{u}_i(x_m)$ die Komponenten des vom Verfahren 3. bzw. 4. Ordnung berechneten Näherungsvektors bezeichnen und

$$S_i:=\max_{i=1}^{n}\left(C_i,|\bar{u}_i(x_m)|\right)$$

mit $C_i>0$ ist, schätzt den Hauptteil des lokalen Fehlers vom Verfahren 3. Ordnung ab. Wird zu einer vorgegebenen Toleranz TOL der Schritt durch $\mathrm{EST}\leqq\mathrm{TOL}$ gesteuert, so erfolgt die Steuerung des lokalen Fehlers des Verfahrens 4. Ordnung „per unit step". Diesem Algorithmus ist eine Schrittweitensteuerung mittels Halbierung und Verdoppelung (SHV) angepaßt.

SHV: Falls $\mathrm{EST}\leqq\frac{1}{16}\mathrm{TOL}$: $x:=x_{\mathrm{trial}}$, $h:=2h$, $x_{\mathrm{trial}}:=x+h$,

sonst: $\mathrm{EST}>\mathrm{TOL}$: $x:=x_{\mathrm{trial}}-h$, $h:=h/2$, $x_{\mathrm{trial}}:=x+h$

sonst: $x:=x_{\mathrm{trial}}$, $h:=h$, $x_{\mathrm{trial}}:=x+h$.

Im Anfangspunkt (x_0,u_0) wird die Matrix T_0 durch $T_0:=f_y(x_0,u_0)$ festgelegt. Dann wird – um den Rechenaufwand zu reduzieren – die Matrix T_m nur von Zeit zu Zeit der momentanen Jacobi-Matrix angepaßt (in allen nachfolgenden Testbeispielen hat sich die Anpassung aller sechs Schritte bewährt; bei einer Schrittverdopplung wird, falls die Matrix T_m in mehr als drei zurückliegenden Schritten konstant blieb, ebenfalls eine Anpassung vorgenommen). Damit ist pro Integrationsschritt die in Tabelle 1 angegebene Anzahl von LU-Zerlegungen erforderlich.

Tabelle 1. *Anzahl der LU-Zerlegungen pro Integrationsschritt*

	LU-Zerlegungen	
	T_m neu	T_m alt
$h_{\mathrm{neu}}:=2h_{\mathrm{alt}}$	2	1
$h_{\mathrm{neu}}:=h_{\mathrm{alt}}$	2	0
$h_{\mathrm{neu}}:=\frac{1}{2}h_{\mathrm{alt}}$	2	1

Für Systeme

$$y' = A \cdot y;\ A \text{ konstante } (n,n)\text{-Matrix}$$

ist mit $T_m = A$, wegen

$$u_{m+1} = R_0(h \cdot A)\, u_m,$$

eine Schätzung des lokalen Diskretisierungsfehlers mittels eingebetteter Verfahren nicht möglich. Daher wird für Systeme

$$y' = A \cdot y + r(x)$$

eine ein-Schritt-zwei-halbe-Schritte Fehlerkontrolle verwendet.

Die Größe EST ist dabei festgelegt durch

$$\text{EST} := \max_{i=1}^{n} \frac{|u_i(x_m, h) - u_i(x_m, h/2)|}{S_i} \tag{5.2}$$

mit

$$S_i := \max_{i=1}^{n} (C_i, |u_i(x_m, h/2)|).$$

Diese Schrittsteuerung erfordert in jedem Integrationsschritt lediglich 4 Aufrufe von $r(x)$ (im Anfangsschritt 5). Bei einer Schrittwiederholung sind nur zwei Aufrufe erforderlich.

6. Numerische Resultate

Der vorgestellte LS-stabile Algorithmus wurde u. a. an den 25 stiff-Beispielen aus [3] erprobt. Diese Testbeispiele sind in die fünf Klassen

Klasse A: Autonome lineare Systeme mit reellen Eigenwerten
Klasse B: Autonome lineare Systeme mit komplexen Eigenwerten
Klasse C: Systeme mit nichtlinearer Kopplung
Klasse D: Nichtlineare Systeme mit reellen Eigenwerten
Klasse E: Nichtlineare Systeme mit komplexen Eigenwerten

eingeteilt. Alle Beispiele wurden mit den Toleranzen $\text{TOL} = 10^{-2}, 10^{-4}, 10^{-6}$ gerechnet. Die Schrittweitensteuerung erfolgt für die Klassen A und B mit (5.2) (eine Verdopplung von h wurde bei 64 $\text{EST} \leqq \text{TOL}$ vorgenommen) und für C, D und E mit (5.1). Dabei wurde für alle Klassen $C_i = 1$, $i = 1, 2, \ldots, n$, gesetzt. Als Stabilitätsfunktionen $R_0(z)$ wurden Padé-Approximationen verwendet, und zwar für die Klassen A und B eine (4, 3), für E 4 und E 5 eine (2, 2) und für die anderen Beispiele eine (3, 2)-Approximation. Die Konsistenzordnung für die Klassen A und B ist damit $p = 7$. Als Startschrittweiten wurden die in [3] angegebenen benutzt. Die Rechnungen wurden auf einer EDV-Anlage ES 1040 mit einer Mantissenlänge von 16 Dezimalstellen durchgeführt. Die Ergebnisse sind in den folgenden Tabellen wiedergegeben. Hierbei bezeichnen:

FA: Anzahl der Funktionsaufrufe
JA: Anzahl der Jacobi-Matrix-Aufrufe
GA: Anzahl der Gesamtaufrufe, $GA = FA + n \cdot JA$
ST: Anzahl der Integrationsschritte
W: Anzahl der Wiederholungen
LU: Anzahl der LU-Zerlegungen

$$\text{ERR: } \max_{i=1}^{n} \frac{|u_i - \tilde{u}_i|}{S_i} \text{ mit } u_i = u_i(x_e) \text{ bzw. } u_i(x_e, h/2).$$

Die Vergleichslösung $\tilde{u}_i = \tilde{u}_i(x_e)$ wurde mit einer verbesserten Gear-Fassung von Hindmarsh [6] (DRIVE) berechnet.[2]

Tabelle 2. $TOL = 1.0\,E-2$

Beispiel	FA	JA	GA	LU	ST	W	ERR
A1	45	1	49	13	11	0	7.3 E−8
A2	81	1	90	20	20	0	2.7 E−8
A3	85	1	89	23	21	0	<1.0 E−8
A4	69	1	79	19	17	0	<1.0 E−8
B1	145	1	149	12	36	0	<1.0 E−8
B2	45	1	51	13	11	0	<1.0 E−8
B3	45	1	51	13	11	0	<1.0 E−8
B4	57	1	63	13	14	0	<1.0 E−8
B5	119	1	125	19	31	3	4.0 E−7
C1	52	3	64	15	13	0	3.7 E−3
C2	44	3	56	14	11	0	1.8 E−7
C3	48	3	60	15	12	0	3.0 E−7
C4	88	4	104	17	22	0	2.0 E−6
C5	132	6	156	22	33	1	1.5 E−6
D1	124	5	134	25	31	2	2.2 E−2
D2	112	5	127	30	28	1	1.7 E−3
D3	92	5	112	26	23	0	1.2 E−8
D4	72	4	84	23	18	0	4.5 E−5
D5	80	4	88	25	20	0	8.8 E−3
D6	100	5	115	31	25	0	1.9 E−2
E1	32	2	40	11	8	0	<1.0 E−8
E2	176	7	190	39	44	9	8.4 E−2
E3	88	4	100	23	22	2	4.4 E−5
E4	148	7	176	32	37	2	7.9 E−5
E5	100	5	120	31	25	0	5.0 E−8

[2] Die Lösung $\tilde{u}_i$ ist die gleiche, wie bei Kaps/Rentrop [7], sie wurde den Autoren freundlicherweise von P. Rentrop zur Verfügung gestellt.

Tabelle 3. $TOL = 1.0\,E-4$

Beispiel	FA	JA	GA	LU	ST	W	ERR
A1	65	1	69	13	16	0	<1.0E−8
A2	125	1	134	19	31	0	<1.0E−8
A3	105	1	109	23	26	0	<1.0E−8
A4	105	1	115	19	26	0	<1.0E−8
B1	591	1	595	26	151	7	<1.0E−8
B2	61	1	67	13	15	0	<1.0E−8
B3	61	1	67	13	15	0	<1.0E−8
B4	81	1	87	13	20	0	<1.0E−8
B5	229	1	235	25	60	6	<1.0E−8
C1	104	5	124	23	26	3	2.4E−7
C2	72	4	88	15	18	0	4.8E−8
C3	124	6	148	20	31	0	<1.0E−8
C4	436	17	504	44	109	3	<1.0E−8
C5	604	22	692	58	151	3	<1.0E−8
D1	352	13	378	42	88	4	7.8E−4
D2	156	7	177	31	39	1	3.5E−4
D3	140	7	168	30	35	1	2.0E−8
D4	72	4	84	23	18	0	4.5E−5
D5	80	4	88	25	20	0	8.8E−3
D6	192	10	222	60	48	0	7.9E−5
E1	32	2	40	11	8	0	<1.0E−8
E2	376	13	402	53	94	10	1.5E−4
E3	132	6	150	23	33	1	2.2E−5
E4	344	13	396	51	86	4	3.7E−7
E5	100	5	120	31	25	0	5.0E−8

Der adaptive RK-Algorithmus bewältigte alle Beispiele – insbesondere für die Toleranzen 10^{-2} und 10^{-4} – gut. Ein Vergleich mit den ROW-Methoden GRK4A und GRK4T von Kaps und Rentrop [7] zeigt, daß der adaptive RK-Algorithmus weniger Gesamtaufrufe und LU-Zerlegungen (mit Ausnahme von D6) benötigt. Für die in [7] betrachtete Toleranz $TOL = 10^{-4}$ ist der FehlerERR für die Klassen D und E dabei mit dem in [7] vergleichbar; die Klassen A, B, C zeigen einen leichten Vorteil der adaptiven Verfahren.

Weitere Rechnungen für Beispiele aus der chemischen Industrie ergaben, daß der adaptive RK-Algorithmus – aufgrund seiner guten Stabilitätseigenschaften – auch für sehr steife Systeme geeignet ist.

Tabelle 4. $TOL = 1.0\,E-6$

Beispiel	FA	JA	GA	LU	ST	W	ERR
A1	93	1	97	13	23	0	<1.0E−8
A2	241	1	250	21	60	0	<1.0E−8
A3	149	1	153	22	37	0	<1.0E−8
A4	173	1	183	19	43	0	<1.0E−8
B1	591	1	595	40	155	15	<1.0E−8
B2	85	1	91	13	21	0	<1.0E−8
B3	93	1	99	13	23	0	<1.0E−8
B4	145	1	151	17	37	2	<1.0E−8
B5	435	1	441	39	115	13	<1.0E−8
C1	196	8	228	31	49	4	1.8E−7
C2	292	12	340	34	73	2	<1.0E−8
C3	696	26	800	62	174	3	<1.0E−8
C4	1832	67	2100	145	458	4	<1.0E−8
C5	2240	81	2564	177	560	5	<1.0E−8
D1	2448	87	2622	192	612	7	9.0E−8
D2	548	21	611	64	137	5	3.5E−6
D3	388	14	444	58	97	9	<1.0E−8
D4	80	4	92	22	20	0	1.2E−6
D5	104	5	114	28	26	1	7.0E−3
D6	376	16	424	77	94	7	1.3E−5
E1	32	2	40	11	8	0	<1.0E−8
E2	1104	39	1182	103	276	12	2.1E−6
E3	876	32	972	74	219	1	1.7E−7
E4	856	30	976	102	214	15	1.2E−7
E5	100	5	120	31	25	0	5.0E−8

Literatur

[1] Burrage, K., Butcher, J. C., Chipman, F. H.: An implementation of singly-implicit Runge-Kutta methods. BIT *20*, 326–340 (1980).

[2] Butcher, J. C., Burrage, K., Chipman, F. H.: STRIDE: Stable Runge-Kutta integrator for differential equations. Report No. 150, Dept. of Math., University of Auckland (1979).

[3] Enright, W. H., Hull, T. E., Lindberg, B.: Comparing numerical methods for stiff systems of ordinary differential equations. BIT *15*, 10–48 (1975).

[4] Friedli, A.: Verallgemeinerte Runge-Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme (Lecture Notes in Mathematics, Vol. 631), pp. 35–50. Berlin-Heidelberg-New York: Springer 1970.

[5] Hairer, E., Wanner, G.: Characterization of non-linearly stable implicit Runge-Kutta methods. Report 1980, University of Heidelberg.

[6] Hindmarsh, A. C.: GEAR-ordinary differential equation system solver. UCID-30001, Rev. 2, University of California: Lawrence Livermore Laboratory 1972.

[7] Kaps, P., Rentrop, P.: Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. Numer. Math. *33*, 55–68 (1979).

[8] Kaps, P., Wanner, G.: A study of Rosenbrock-type methods of high order. Report 1979, Universität Innsbruck.

[9] Moler, C. B., Van Loan, C. F.: Nineteen dubious ways to compute the exponential of a matrix. SIAM Review *20*, 801 – 836 (1978).
[10] Nørsett, S. P., Wolfbrandt, A.: Order condition for Rosenbrock-type methods. Numer. Math. *32*, 1 – 15 (1979).
[11] Prothero, A., Robinson, A.: On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. Math. Comp. *28*, 145 – 162 (1974).
[12] Steighaug, T., Wolfbrandt, A.: An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations. Math. Comp. *33*, 521 – 534 (1979).
[13] Strehmel, K.: Stabilitätseigenschaften adaptiver Runge-Kutta Verfahren. ZAMM *61*, 253 – 260 (1981).
[14] Verwer, J. G., Scholz, S.: Rosenbrock methods and time-lagged Jacobian matrices. Report NW 82/80, Mathematisch Centrum, Amsterdam (1980).
[15] Verwer, J. G.: On generalized Runge-Kutta methods using an exact Jacobian at a non-step point. ZAMM *60*, 263 – 265 (1980).

K. Strehmel
R. Weiner
Sektion Mathematik
Martin-Luther-Universität
Halle-Wittenberg
Weinbergweg 17
DDR-4020 Halle a. d. Saale
Deutsche Demokratische Republik