EIN ÜBERBLICK ÜBER EINSCHRITTVERFAHREN ZUR NUMERISCHEN INTEGRATION IN DER TECHNISCHEN SIMULATION

P. RENTROP, K. STREHMEL, R. WEINER

1. Modellbildung

Die mathematische Beschreibung in der technischen rechnergestützten Modellbildung – TCAD – führt häufig auf Systeme gewöhnlicher Differentialgleichungen oder differential-algebraische Systeme. Typische Anwendungen sind die mechanischen Mehrkörpersysteme der Fahrzeugtechnik und der Robotik [4], [89], [93], die elektrische Schaltungssimulation [10], die chemische Reaktionskinetik und die Verfahrenstechnik [41], sowie Gasströmungsnetzwerke zur Ladungswechselsimulation im Verbrennungsmotor [42].
Die klassische mathematische Modellierung arbeitet mit Zustandskoordinaten, einer sorgfältigen Wahl einer minimalen Anzahl von Koordinaten. Der **Zustandsform** entspricht ein System gewöhnlicher Differentialgleichungen in expliziter Form (mit einem Anfangszustand):

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \tag{1.1}$$

wobei $y(t)$ ein n-dimensionaler Vektor ist.
Die Festlegung der Zustandskoordinaten verlangt in der Mechanik analytische Vorarbeit, die in einer Rechnerumgebung – sie umfaßt die technische Modelleingabe, die mathematische Modellierung, die numerische Simulation und die grafische Ausgabe – nicht erwünscht oder nicht möglich ist. Ohne analytischen Aufwand läßt sich die mathematische Modellierung in redundanten Koordinaten durchführen, man erhält eine **Deskriptorform**. Der Deskriptorform entspricht ein gekoppeltes System von gewöhnlichen Differentialgleichungen und algebraischen Gleichungen (mit einem Anfangszustand) – das *differential-algebraische System*

$$\begin{aligned} u'(t) &= f(t, u(t), v(t)) \\ 0 &= g(t, u(t), v(t)), \end{aligned} \tag{1.2a}$$

bzw.

$$\begin{aligned} M(u)u' &= f(t, u(t), v(t)) \\ 0 &= g(t, u(t), v(t)) \end{aligned} \tag{1.2b}$$

wobei u ein N-dimensionaler Vektor, v ein $(n-N)$-dimensionaler Vektor und $M(u)$ eine reguläre (N, N)-Matrix ist.
Systeme der Struktur (1.2b) stehen z.B. für die mechanischen Mehrkörpersysteme. Bei einem modularen Aufbau eines Fahrzeugmodells beschreibt die Funktion $g(t, u, v)$ die Kopplung der Module. Jedes einzelne Modul kann in Zustandsform vorliegen. Die elektrische Schaltungssimulation führt auf quasilineare Differentialgleichungen [67]:

$$C(U) \cdot U'(t) = F(t, U(t)). \tag{1.3a}$$

$C(U)$ ist dabei die Kapazitätsmatrix (oft singulär) und $U(t)$ der n-dimensionale Knotenspannungsvektor. Abhängig von den verwendeten Transistormodellen kann auch eine implizite Differentialgleichung

$$F(t, U(t), Q'(t)) = 0, \tag{1.3b}$$

vorliegen, wobei $U(t)$ den n-dimensionalen Knotenspannungsvektor und $Q(t)$ den n-dimensionalen Ladungsvektor bezeichnet. Ist $C(U)$ bzw. $\partial F / \partial Q'$ singulär, dann lassen sich die impliziten Systeme (1.3a), (1.3b) in die Form (1.2) überführen.

Die bisherigen Ausführungen sollen nochmals an zwei Problemen aus der chemischen Kinetik und aus der Mechanik illustriert werden.

1.) **Chemische Reaktionen** lassen sich mit Hilfe des Massenwirkungsgesetzes mathematisch als Systeme gewöhnlicher Differentialgleichungen mit polynomialer rechter Seite beschreiben. Betrachtet man ein abgeschlossenes, homogenes System in dem N chemische Reaktionen zwischen n chemischen Substanzen G_i, $i = 1, ..., n$, gleichzeitig stattfinden und setzt hierbei konstantes Volumen und konstante Temperatur voraus, so läßt sich der j-te Teilschritt dieser Reaktion angeben in der Form

$$\sum_{i=1}^{n} p_{ij} G_i \xrightarrow{k_j} \sum_{i=1}^{n} q_{ij} G_i, \quad j = 1, ..., N.$$

Hierbei sind p_{ij}, q_{ij} die stöchiometrischen Koeffizienten und k_j Reaktionsgeschwindigkeitskonstanten.

Bezeichnen $y_1(t), ..., y_n(t)$ die Konzentrationen der miteinander reagierenden chemischen Substanzen G_i, so läßt sich unter den obigen Voraussetzungen die chemische Reaktion durch das folgende Anfangswertproblem gewöhnlicher Differentialgleichungen eindeutig beschreiben ([38]):

$$\begin{aligned} y_i'(t) &= \sum_{j=1}^{N} (q_{ij} - p_{ij}) k_j \prod_{l=1}^{n} (y_l(t))^{p_{lj}}, \quad i = 1, ..., n \\ y_i(t_0) &= y_{i0}. \end{aligned}$$

Bemerkung 1.1. Die Modellierung erfolgt hier in Zustandskoordinaten. Die Hinzunahme von Massenerhaltung führt auf differential-algebraische Systeme vom Typ (1.2). $\square$

Als Beispiel diene die chemische Pyrolyse (vgl. Aiken [1]).

$$\begin{aligned} G_1 &\xrightarrow{k_1} G_2 + G_3 \\ G_2 + G_3 &\xrightarrow{k_2} G_5 \\ G_1 + G_3 &\xrightarrow{k_3} G_4 \\ G_4 &\xrightarrow{k_4} G_3 + G_6. \end{aligned}$$

Das zugehörige Differentialgleichungssystem für die Konzentrationen $y_i(t)$ ergibt sich zu

$$
\begin{aligned}
y_1'(t) &= -k_1 y_1 - k_3 y_1 y_3\\
y_2'(t) &= k_1 y_1 - k_2 y_2 y_3\\
y_3'(t) &= k_1 y_1 - k_2 y_2 y_3 - k_3 y_1 y_3 + k_4 y_4\\
y_4'(t) &= k_3 y_1 y_3 - k_4 y_4\\
y_5'(t) &= k_2 y_2 y_3\\
y_6'(t) &= k_4 y_4
\end{aligned}
$$

Die stark unterschiedlichen Reaktionsgeschwindigkeitskonstanten

$$k_1 = 7.89 \cdot 10^{-10},\ k_2 = 1.13 \cdot 10^{9},\ k_3 = 1.1 \cdot 10^{7},\ k_4 = 1.13 \cdot 10^{3}$$

bewirken, daß das System *steif* ist (vgl. Abschnitt 2.2).
Die Generierung der Differentialgleichungen der Kinetik erfolgt durch entsprechende Softwaresysteme, z.B. [28], [37], [70], [96], [9], [34].
2.) Wir betrachten eine Teilaufgabe des ebenen **Stab-Balance Problems**, siehe Abbildung 1.

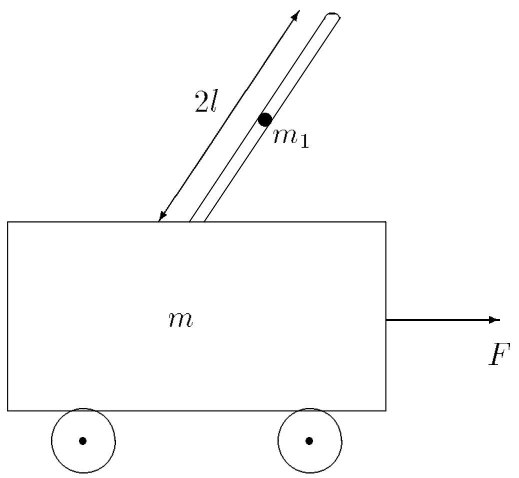

ABBILDUNG 1. Ebenes Stab-Balance-Problem

Die Aufgabe lautet: Welche Kraft F(t) muß an einem Wagen angreifen, damit der Stab senkrecht steht? Im folgenden soll die Modellierung des Wagen-Stab-Systems behandelt werden. Abbildung 2 zeigt die redundanten Koordinatensysteme.
Das mechanische System ist charakterisiert durch:

- Wagen: Masse m, Freiheitsgrad x, angreifende Kraft $F(t)$, Zwangskraft F_{z1} im Gelenk,
- Stab der Länge $2l$, in der Stabmitte: Masse m_1 und Trägheitsmoment I_1, Freiheitsgrade x_1, y_1, φ, Schwerkraft $-m_1 g$, Zwangskräfte F_{z2}, F_{z3} im Gelenk.

Damit lassen sich die Kräfte nach Newton-Euler formulieren. Das resultierende System besteht

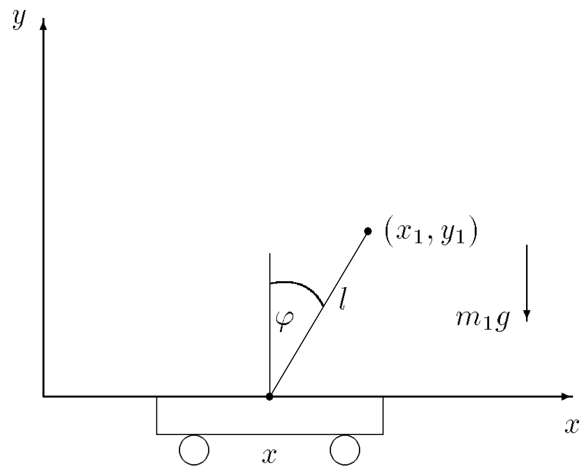

ABBILDUNG 2. Koordinatensysteme

aus Differentialgleichungen und nichtlinearen Gleichungen:

$$\begin{aligned} mx'' &= F + F_{z1} \\ m_1 x_1'' &= F_{z2} \\ m_1 y_1'' &= -m_1 g + F_{z3} \\ I_1 \varphi'' &= F_{z4} && \text{(Drallsatz)} \\ x_1 &= x + l \sin\varphi \\ y_1 &= l \cos\varphi \end{aligned}$$

Der Newton-Euler Formalismus ist ein denkbarer Zugang für eine rechnergestützte Modellierung. In den letzten Jahren sind in der Mechanik mehrere Formalismen entwickelt worden, die die Mehrkörpersystemmodellierung auf einem Rechner vornehmen. Einen Überblick findet man z.B. in [88]. Für eine ausführliche Diskussion der Methodik der Modellbildung und Gleichungsgenerierung sei auf Haug [62] und Roberson/Schwertassek [84] verwiesen. Für holonome Systeme (ohne redundante Zwangsbedingung) lautet die allgemeine Deskriptorform

$$\begin{aligned} M(p)p'' &= F(t, p, p', \lambda) \\ g(p) &= 0, \end{aligned} \tag{1.4}$$

dabei bezeichnet p den Vektor der Ortskoordinaten, M ist die der kinetischen Energie zugeordnete symmetrische, positiv definite Massematrix, und F den Vektor der Kräfte.
Die Zwangskräfte F_{zj} im Gelenk lassen sich nach D'Alembert durch Lagrange-Multiplikatoren $\lambda(t)$ ersetzen. Definiert man für die Freiheitsgrade den Ortsvektor p:

$$p = (x, x_1, y_1, \varphi)^T$$

und schreibt für das holonome, nichtlineare Gleichungssystem

$$g(p) = (x + l\sin\varphi - x_1, l\cos\varphi - y_1)^T = 0,$$

so gilt (Voraussetzungen der Glattheit sind erfüllt):

$$F_z = (F_{z1}, F_{z2}, F_{z3}, F_{z4})^T = -(\frac{\partial g}{\partial p})^T \lambda, \quad \text{wo} \quad \frac{\partial g}{\partial p} = \begin{pmatrix} 1 & -1 & 0 & l\cos\varphi \\ 0 & 0 & -1 & -l\sin\varphi \end{pmatrix},$$

und damit:

$$F_{z1} = -\lambda_1, \quad F_{z2} = \lambda_1, \quad F_{z3} = \lambda_2, \quad F_{z4} = -l\lambda_1 \cos\varphi + l\lambda_2 \sin\varphi.$$

In der resultierenden Deskriptorform (1.5) haben wir damit 6 Gleichungen für die 6 Unbekannten $x, x_1, y_1, \varphi, \lambda_1, \lambda_2$:

$$\begin{aligned} mx'' &= F - \lambda_1 \\ m_1 x_1'' &= \lambda_1 \\ m_1 y_1'' &= -m_1 g + \lambda_2 \\ I_1 \varphi'' &= -l\lambda_1 \cos\varphi + l\lambda_2 \sin\varphi \\ 0 &= x + l\sin\varphi - x_1 \\ 0 &= l\cos\varphi - y_1. \end{aligned} \tag{1.5}$$

Offensichtlich enthält die Deskriptorform redundante Information, denn für eine Beschreibung des Stab-Wagen-Systems sind die Koordinaten x und φ ausreichend. Formal entspricht der Übergang zu Zustandskoordinaten der Elimination der Lagrange-Multiplikatoren. Dieser Schritt erfordert im allgemeinen analytische Arbeit.
Zweimaliges Differenzieren der Nebenbedingung und Einsetzen der Differentialgleichung ergibt für die Lagrange-Multiplikatoren die Beziehungen:

$$\begin{aligned} \lambda_1 &= m_1 x'' + m_1 l \varphi'' \cos\varphi - m_1 l (\varphi')^2 \sin\varphi \\ \lambda_2 &= m_1 g - m_1 l \varphi'' \sin\varphi - m_1 l (\varphi')^2 \cos\varphi. \end{aligned}$$

Damit lautet die Zustandsform von (1.5):

$$\begin{pmatrix} m + m_1 & m_1 l \cos\varphi \\ m_1 l \cos\varphi & I_1 + m_1 l^2 \end{pmatrix} \cdot \begin{pmatrix} x'' \\ \varphi'' \end{pmatrix} = \begin{pmatrix} F + m_1 l (\varphi')^2 \sin\varphi \\ m_1 g l \sin\varphi \end{pmatrix}. \tag{1.6}$$

Erläuterungen zur Deskriptorform (1.5) und zur Zustandsform (1.6):

- Ist der Wagen fixiert, d.h. $x = 0$, so reduziert sich (1.6) auf die Pendelgleichung für den Stab:
  $$(I_1 + m_1 l^2)\varphi'' = m_1 g l \sin\varphi$$
- Die Zustandsform (1.6) besitzt eine kleinere Dimension als die Deskriptorform (1.5). Diese kleinere Dimension wird jedoch durch eine voll besetzte (symmetrische, positiv definite), ortsabhängige Massematrix erkauft. Die Massematrix der Deskriptorform besitzt Diagonalgestalt (Diagonalelemente sind positiv) und ist zeitlich konstant. Die Frage, ob man ‘a priori‘ die Zustandsform oder die Deskriptorform vorziehen soll, ist offen.
- Für die regelungstechnischen Aspekte der Aufgabenstellung verweisen wir auf [83].

Die vorgestellte Modellierung ist ohne Probleme auf mehrere Stäbe verallgemeinerbar.
Diese kurzen Ausführungen dienen als Hinweise für das Auftreten von Differential- und differential-algebraischen Gleichungen in der technischen Simulation. Eine Aufgabe der Numerischen Mathematik ist die Bereitstellung von zuverlässigen, genauen und effizienten Integrationsverfahren. Eine Übersicht zu der Entwicklung von Einschrittverfahren innerhalb der letzten Jahre soll diese Arbeit geben.

2. Einschrittverfahren für gewöhnliche Differentialgleichungen

Wir betrachten in diesem Abschnitt Einschrittverfahren zur Lösung von Anfangswertproblemen gewöhnlicher Differentialgleichungen

$$
\begin{aligned}
y'(t) &= f(t, y(t)) \\
y(t_0) &= y_0 \in \mathbb{R}^n, \quad t \in [t_0, t_e],
\end{aligned}
\tag{2.1}
$$

wie sie bei der Modellierung mit Zustandskoordinaten entstehen. Dabei setzen wir die eindeutige Lösbarkeit voraus, insbesondere soll die Funktion f einer Lipschitzbedingung bez. y mit der Lipschitz-Konstante L genügen.

Beginnend mit $u_0 = y_0$ berechnen *Einschrittverfahren* (ESV) ausgehend von einem Näherungswert u_m für $y(t_m)$ mit der Schrittweite h einen Näherungswert u_{m+1} für $y(t_m + h)$:

$$
u_{m+1} = u_m + h\Phi(t_m, u_m; h), \quad m = 0, 1, \ldots \tag{2.2}
$$

Die Darstellung ist nur formal explizit und umfaßt auch implizite Verfahren.

Die wichtigste Klasse von Einschrittverfahren sind die *Runge-Kutta-Verfahren* (RK-Verfahren). Hier wird die Verfahrensfunktion Φ in (2.2) als Linearkombination von Werten der Funktion $f(t, y)$ in diskreten Punkten gewählt. Dieser Ansatz geht auf die Arbeiten von Runge, Heun und Kutta um die Jahrhundertwende zurück. Ein s-stufiges RK-Verfahren ist gegeben durch

$$
\begin{aligned}
u_{m+1}^{(i)} &= u_m + h \sum_{j=1}^{s} a_{ij} f(t_m + c_j h, u_{m+1}^{(j)}), \quad i = 1, \ldots, s \\
u_{m+1} &= u_m + h \sum_{i=1}^{s} b_i f(t_m + c_i h, u_{m+1}^{(i)}).
\end{aligned}
\tag{2.3}
$$

Es wird durch das von Butcher eingeführte Parameterschema ("*Butcher-Schema*")

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

charakterisiert. Dabei ist $c = (c_1, \ldots, c_s)^T$ der *Knotenvektor*, $A = (a_{ij})$ die *Verfahrensmatrix* und $b = (b_1, \ldots, b_s)^T$ der *Wichtungsvektor*.

Die Parameter werden so bestimmt, daß das Verfahren eine möglichst hohe Konsistenzordnung und gute Stabilitätseigenschaften besitzt. Ein ESV besitzt die Konsistenzordnung p, wenn mit $u_0 = y_0$ für den lokalen Diskretisierungsfehler (Fehler nach einem Schritt) gilt

$$
u_1 - y(t_0 + h) = \mathcal{O}(h^{p+1}) \text{ für } h \to 0.
$$

Für Runge-Kutta-Verfahren folgt aus der Konsistenzordnung p auch die Konvergenzordnung p ([55], [59]), d.h., es gilt für den globalen Fehler

$$
u_m - y(t_m) = \mathcal{O}(h^p) \text{ für alle } m.
$$

2.1. Nichtsteife Probleme. Als nichtsteif bezeichnet man das Anfangswertproblem, wenn $(t_e - t_0)L \leq C$ gilt, und C von moderater Größe ist. Als grober Richtwert kann $C \leq 10^2$ gelten. Zur Lösung nichtsteifer Systeme sind *explizite* RK-Verfahren sehr gut geeignet. Hierbei gilt in (2.3)

$$c_1 = 0 \text{ und } a_{ij} = 0 \quad \text{für } j \geq i,$$

die Zwischenwerte $u_{m+1}^{(i)}$ können folglich sukzessiv berechnet werden.
Das einfachste explizite RK-Verfahren ist das *explizite Euler-Verfahren*

$$u_{m+1} = u_m + hf(t_m, u_m),$$

das nur die Konsistenzordnung 1 besitzt. Eines der gegenwärtig effektivsten expliziten RK-Verfahren für bei praktischen Aufgaben relevante Genauigkeitsforderungen ist das von Dormand/Prince [36] konstruierte 7-stufige Verfahren DOPRI5 der Konsistenzordnung 5:

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

Hierbei wird neben dem Näherungswert u_{m+1} der Konsistenzordnung 5 aus den berechneten Zwischenwerten noch ein Näherungswert $\widetilde{u}_{m+1}$ der Konsistenzordnung 4 mit Hilfe der Gewichte $\widetilde{b}$ berechnet. Dadurch läßt sich ohne wesentlichen Mehraufwand durch

$$err = \|u_{m+1} - \widetilde{u}_{m+1}\|$$

in einer geeigneten Norm der lokale Fehler des Verfahrens 4. Ordnung schätzen und die Schrittweite so bestimmen, daß der lokale Fehler in der Größenordnung einer vorgegebenen Toleranz TOL bleibt. Man spricht von einer Schrittweitensteuerung durch Einbettung. Ein analoges Verfahren der Ordnung 8 mit eingebetteten Verfahren der Ordnung 5 und 3 ist DOP853, das für schärfere Genauigkeitsforderungen effektiver als DOPRI5 wird. FORTRAN-Programme dieser beiden Verfahren findet man in Hairer/Nørsett/Wanner [59] bzw. [60].

Bemerkung 2.1. Eine andere Möglichkeit der Schrittweitensteuerung ist die *Richardson-Extrapolation.* Dabei werden zur Schätzung des lokalen Fehlers mit ein und demselben Verfahren ausgehend von u_m Näherungswerte $\widetilde{u}_{m+1}$ mit der Schrittweite h und u_{m+1} durch 2 Schritte mit der Schrittweite $h/2$ bestimmt. Neben einer Schätzung des lokalen Fehlers kann gleichzeitig durch

$$u_{m+1}^* = u_{m+1} + \frac{1}{2^p - 1}(u_{m+1} - \widetilde{u}_{m+1})$$

auch eine Näherungslösung der Ordnung $p + 1$ bestimmt werden. □

Bemerkung 2.2. Es existieren weiterhin zahlreiche effiziente Algorithmen für nichtsteife Systeme, die auf Extrapolationsverfahren oder auf linearen Mehrschrittverfahren vom Adams-Typ beruhen. Bekannte und erprobte Codes sind z.B. die Extrapolationsalgorithmen DIFEX1 von Deuflhard [32], ODEX von Hairer/Nørsett/Wanner [59], sowie die Mehrschrittverfahren DEABM von Shampine/Watts [91], EPISODE von Byrne/Hindmarsh [27], LSODE von Hindmarsh [63], [64] und VODE von Brown/Byrne/Hindmarsh [16]. □

2.2. Steife Systeme. 1952 stellten Curtiss/Hirschfelder [29] fest, daß explizite Runge-Kutta-Verfahren bei Gleichungen der chemischen Kinetik versagen und führten für solche Systeme den Begriff der *Steifheit* ein. Neben der chemischen Kinetik treten steife System in zahlreichen weiteren Anwendungen auf, eine Übersicht findet man in Aiken [1].
In der Literatur gibt es keine einheitliche Definition der Steifheit, allerdings lassen sich einige charakteristische Eigenschaften steifer Systeme herausstellen. Sie enthalten Lösungskomponenten, die sich nur langsam ändern, aber Störungen dieser langsam veränderlichen Lösung werden sehr schnell gedämpft ([31], [90], [72]). Quantitativ kann dieser Sachverhalt so ausgedrückt werden

$$(t_e - t_0)L \gg 1.$$

Jedoch existiert eine einseitige Lipschitz-Konstante μ, definiert durch

$$\langle f(t,u) - f(t,v), u - v\rangle \le \mu\|u - v\|^2 \quad \text{für alle } u, v \in \mathbb{R}^n,$$

($\langle .,. \rangle$ ein Skalarprodukt im $\mathbb{R}^n$) mit

$$(t_e - t_0)\mu \le C = \mathcal{O}(1).$$

Häufig ist sogar $\mu \le 0$.
Vom praktischen Standpunkt aus kann man auch sagen, ein System ist steif, wenn bei Anwendung eines expliziten RK-Verfahrens die Schrittweite durch Stabilitätsforderungen stärker eingeschränkt wird als durch die vorgegebene Genauigkeit. Äußeres Kennzeichen dafür sind bei einer Implementierung mit Schrittweitensteuerung eine relative Unabhängigkeit der Schrittzahl von der vorgegebenen Genauigkeit und häufig eine große Anzahl von wiederholten Schritten.

Beispiel 2.1. Wir betrachten folgendes einfaches Anfangswertproblem

$$y' = \lambda(y - e^{-t}) - e^{-t}, \quad y(0) = y_0, \quad t \in [0,1].$$

Die exakte Lösung ist $y = e^{-t} + e^{\lambda t}(y_0 - 1)$. Für $y_0 = 1$ erhält man die langsam veränderliche Lösung e^{-t}. Für $\lambda \ll -1$ nähert sich jede andere Lösung dieser glatten Lösung sehr schnell an, die Gleichung ist steif. In Tabelle 2.1 geben wir für $y_0 = 1$ für das explizite Euler-Verfahren und das implizite Euler-Verfahren

$$u_{m+1} = u_m + hf(t_{m+1}, u_{m+1}) \tag{2.4}$$

bei Rechnung mit konstanter Schrittweite $h = 2^{-k}$ den Fehler an der Stelle $t_e = 1$ an. Für den Parameter λ wählen wir dabei zwei verschiedene Werte. Für den nichtsteifen Fall $\lambda = -10$ liefern beide Verfahren für die verwendeten Schrittweiten fast identische Fehler. Für $\lambda = -1000$ ändert sich das grundlegend. Während das implizite Euler-Verfahren keine Probleme hat, liefert das explizite Euler-Verfahren für $k \le 8$ völlig unbrauchbare Werte, erst für $k = 10$ stimmt das Ergebnis wieder mit dem des impliziten Euler-Verfahrens überein. □

	$\lambda = -10$		$\lambda = -1000$	
k	expl. Euler	impl. Euler	expl. Euler	impl. Euler
4	1.2E−003	1.3E−003	1.3E+024	1.1E−005
6	3.2E−004	3.2E−004	2.9E+069	2.9E−006
8	8.0E−005	8.0E−005	8.0E+112	7.2E−007
10	2.0E−005	2.0E−005	1.8E−007	1.8E−007

TABELLE 2.1. Fehler für explizites und implizites Euler-Verfahren

1963 zeigte Dahlquist [30], daß die Schwierigkeiten expliziter Verfahren bei Anwendung auf steife Systeme als Stabilitätsprobleme gedeutet werden können. Anhand der skalaren Testgleichung

$$y' = \lambda y, \quad \operatorname{Re}\lambda \le 0, \tag{2.5}$$

führte er den Begriff der *A-Stabilität* ein, der auch heute noch von zentraler Bedeutung ist. Die gebräuchlichen Einschrittverfahren liefern für (2.5)

$$u_{m+1} = R_0(z)u_m, \quad z = h\lambda,$$

mit der sog. Stabilitätsfunktion $R_0(z)$.

Definition 2.1. Ein Einschrittverfahren heißt A-stabil, wenn gilt

$$|R_0(z)| \le 1 \text{ für alle } \operatorname{Re} z \le 0.$$

Es ist L-stabil, wenn es A-stabil ist und außerdem

$$\lim_{Re\, z\to-\infty} R_0(z) = 0$$

gilt. □

Die L-Stabilität ist eine von Ehle 1969 [40] eingeführte Verschärfung der A-Stabilität.
Die Stabilitätsfunktion eines s-stufigen expliziten RK-Verfahrens ist ein Polynom vom Grad $\le s$, die Beziehung $|R_0(z)| \le 1$ kann also nur in einem beschränkten Gebiet, dem sog. Stabilitätsgebiet, erfüllt sein. Für gegebenes λ bedeutet das eine Einschränkung an die Schrittweite h. Abbildung 3 zeigt die Stabilitätsgebiete des expliziten Euler-Verfahrens und von DOPRI5.
Ähnlich wie für die Testgleichung ergibt sich für allgemeine steife Systeme aus Stabilitätsgründen eine Einschränkung der Form

$$hL = \mathcal{O}(1)$$

an die Schrittweite expliziter RK-Verfahren, so daß diese Verfahren für diese Probleme nicht geeignet sind.
Die Notwendigkeit, andere Einschrittverfahren für steife Systeme zu finden, führte zur Entwicklung von impliziten RK-Verfahren (beginnend mit Butcher 1964 [22]) und linear-impliziten RK-Verfahren (erstmals Rosenbrock 1963 [86]).
Das einfachste implizite RK-Verfahren ist das implizite Euler-Verfahren aus Beispiel 2.1. Im Unterschied zu expliziten RK-Verfahren ist bei impliziten RK-Verfahren die Koeffizientenmatrix A vollbesetzt. Die Stabilitätsfunktion impliziter RK-Verfahren ist die rationale Funktion

$$R_0(z) = 1 + zb^T(I - zA)^{-1}\mathbb{1}, \quad \mathbb{1} = (1, ..., 1)^T.$$

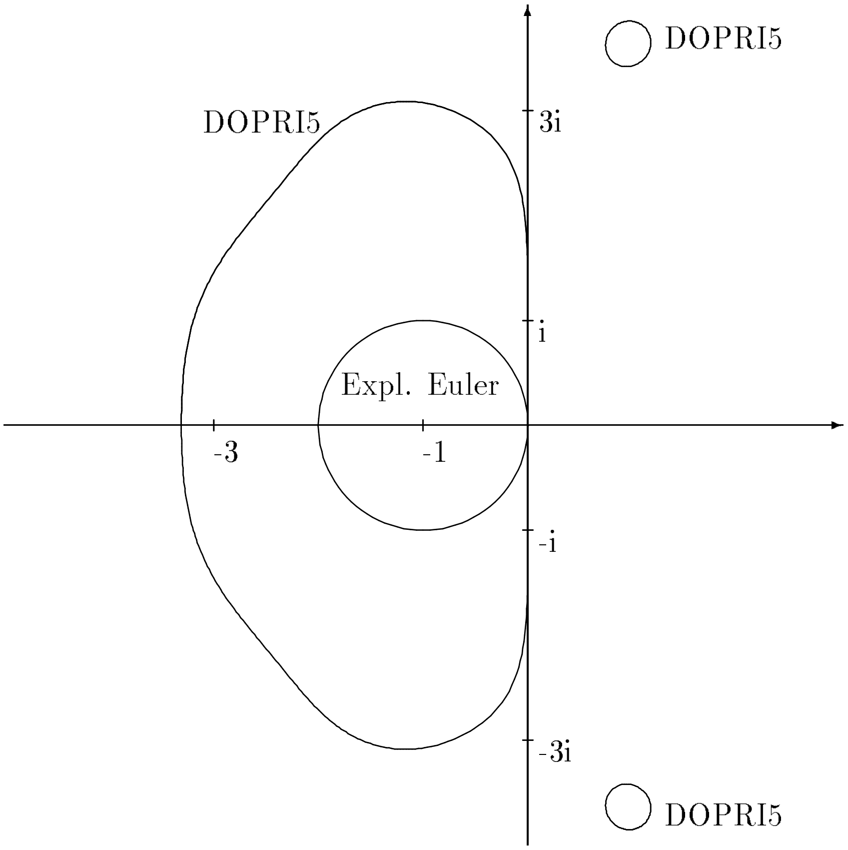

Abbildung 3. Stabilitätsgebiete des expliziten Euler-Verfahrens und von DOPRI5

Es existieren A- und L-stabile implizite RK-Verfahren beliebig hoher Ordnung. Ein bekannter Code ist RADAU5, ein FORTRAN-Programm findet man in Hairer/Wanner [61]. Er basiert auf dem folgenden dreistufigen L-stabilen Radau-IIA-Verfahren (Ehle [39]) der Konsistenzordnung $p = 5$

$$\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}$$

Der wesentliche Nachteil impliziter RK-Verfahren, der ihre breite Anwendung stark einschränkt, besteht darin, daß die Zwischenwerte $u_{m+1}^{(i)}$ nicht mehr sukzessiv berechnet werden können, sondern die Lösung eines nichtlinearen Gleichungssystems der Dimension ns erfordern. Bei steifen Systemen erfolgt die Lösung dieses Gleichungssystems mit Newton-ähnlichen Verfahren. Eine wesentliche Einsparung der Operationen kann bei Verwendung eines vereinfachten Newton-Verfahrens erzielt werden, d.h., für die Block-Jacobi-Matrix des ns-dimensionalen Gleichungssystems wird in den einzelnen Blöcken jeweils die Jacobi-Matrix $f_y(t_m, u_m)$ verwendet (Bickart [12]), Butcher [23]).
Besitzt die Koeffizientenmatrix A einen s-fachen Eigenwert γ, dann kann bei Verwendung des vereinfachten Newton-Verfahrens das lineare Gleichungssystem in s Gleichungssysteme der Di-

mension n mit gleicher Koeffizientenmatrix (d.h., nur noch eine LU-Zerlegung erforderlich) entkoppelt werden, man spricht von einfach-impliziten RK-Verfahren (singly-implicit RK methods, [75], [19]). Eine Implementierung dieser Verfahren ist der Code STRIDE von Burrage, Butcher und Chipman [21]. Besitzt die Koeffizientenmatrix A untere Dreiecksgestalt mit Diagonalelementen $a_{ii} = \gamma$, diese Verfahren werden als einfach diagonal-implizite RK-Verfahren (SDIRK-Verfahren) bezeichnet, dann kann die Entkopplung bereits auf der Ebene der nichtlinearen Gleichungssysteme erfolgen. Man hat noch s nichtlineare Gleichungssysteme der Dimension n zu lösen, bei Verwendung des vereinfachten Newton-Verfahrens ist wegen $a_{ii} = \gamma$ ebenfalls nur eine LU-Zerlegung erforderlich. SDIRK- und SIRK-Verfahren besitzen höchstens die Konsistenzordnung $s + 1$.

Bemerkung 2.3. Für eine Implementierung auf Parallelrechnern ist ein s-facher Eigenwert von A dagegen ungünstig. Besitzt A hier s verschiedene reelle Eigenwerte, so kann bei vereinfachter Newton-Iteration die Koeffizientenmatrix der linearen Gleichungssysteme transformiert werden in

$$\begin{pmatrix} I - h\gamma_1 f_y & & 0 \\ & \ddots & \\ 0 & & I - h\gamma_s f_y \end{pmatrix},$$

d.h., die s linearen Gleichungssysteme können jetzt parallel gelöst werden. Diese Methoden werden als *multi-implizite RK-Verfahren* bezeichnet ([71]). Eine anderes Vorgehen zur effektiven parallelen Implementierung von impliziten RK-Verfahren hoher Ordnung ist die von van der Houwen und Sommeijer [66] vorgeschlagene sog. parallele diagonal-implizite Iteration.
Eine Übersicht über parallele Methoden und ausführliche Literaturhinweise findet man in Burrage [20]. □

Eine andere Möglichkeit, den Aufwand zu reduzieren besteht darin, eine Approximation der Jacobi-Matrix direkt in die Verfahrensvorschrift einzubauen und dann nur noch lineare Gleichungssysteme zu lösen. Dieses Vorgehen führt zur Klasse der linear-impliziten RK-Verfahren, eine Übersicht über diese Methoden gibt [100]. Zur Verfahrensklasse der linear-impliziten RK-Verfahren gehören die W- ([97]) bzw. ROW-Methoden ([68], [69]) und die adaptiven RK-Verfahren ([98]).
Eine s-stufige W-Methode ist gegeben durch

$$\begin{aligned} (I - h\gamma T)k_i &= f(t_m + c_i h, u_{m+1}^{(i)}) + hT \sum_{j=1}^{i-1} \gamma_{ij} k_j, \quad i = 1, ..., s \\ u_{m+1}^{(i)} &= u_m + h \sum_{j=1}^{i-1} \alpha_{ij} k_j \\ u_{m+1} &= u_m + h \sum_{j=1}^{s} b_j k_j \end{aligned}$$

und erfordert pro Integrationsschritt nur eine LU-Zerlegung und s Rücksubstitutionen. Die Matrix T ist frei wählbar, für $T = 0$ erhält man ein explizites RK-Verfahren. Als wichtiger Spezialfall entstehen für autonome Systeme mit $T = f_y(u_m)$ die ROW-Methoden, bei denen sich durch die spezielle Wahl von T die Anzahl der Konsistenzbedingungen wesentlich reduziert. Linear-implizite RK-Verfahren besitzen ausgezeichnete lineare Stabilitätseigenschaften (es gibt A- und L-stabile Verfahren beliebig hoher Ordnung), aufgrund ihrer Struktur ist aber die Sta-

bilität für stark nichtlineare Systeme schlechter als bei impliziten RK-Verfahren, sie sind nicht B-stabil (zur B-Stabilität s. Dekker und Verwer [31]).
Eine bekannte effektive ROW-Methode der Ordnung 4 ist GRK4T von [68] mit den Koeffizienten

$$
\begin{array}{lll}
\alpha_{21} = 0.462, & \alpha_{31} = -0.0815668168327, & \alpha_{32} = 0.961775150166 \\
\alpha_{41} = \alpha_{31}, & \alpha_{42} = \alpha_{32}, & \alpha_{43} = 0 \\
\gamma_{21} = -0.270629667752, & \gamma_{31} = 0.311254483294, & \gamma_{32} = 0.00852445628482 \\
\gamma_{41} = 0.282816832044, & \gamma_{42} = -0.457959483281, & \gamma_{43} = -0.111208333333 \\
b_1 = 0.217487371653, & b_2 = 0.486229037990, & b_3 = 0 \\
b_4 = 0.296283590357, & \gamma = 0.231. &
\end{array}
$$

Mit den Gewichten

$$
\tilde{b}_1 = -0.717088504499, \quad \tilde{b}_2 = 1.77617912176, \quad \tilde{b}_3 = -0.0590906172617, \quad \tilde{b}_4 = 0
$$

läßt sich eine Näherungslösung 3. Ordnung bestimmen, wodurch eine Schrittweitensteuerung durch Einbettung möglich ist.

Bemerkung 2.4. Bei der klassischen Definition der Konsistenzordnung gehen in die $\mathcal{O}$-Terme partielle Ableitungen der Funktion f ein. Da speziell $\|f_y\|$ bei steifen Systemen sehr groß ist, sind für diesen Fall die Ordnungsangaben und Fehlerschranken für praktisch relevante Größen der Schrittweite nicht mehr aussagekräftig. Aus diesem Grund führten Frank, Schneid und Ueberhuber [44], [45] das Konzept der B-Konvergenz ein. Hier sind die Abschätzungen unabhängig von der Steifheit, anstelle der klassischen wird die einseitige Lipschitz-Konstante verwendet. Die B-Konsistenzordnung von impliziten RK-Verfahren ist i.allg. wesentlich geringer als die klassische Ordnung, es tritt eine sog. *Ordnungsreduktion* ein.
Für eine ausführliche Darstellung dieser Thematik und Literaturhinweise seien die Bücher von Dekker/Verwer [31] und Hairer/Wanner [61] empfohlen, die B-Konvergenz linear-impliziter RKV wird in Strehmel/Weiner [100] untersucht. □

Bemerkung 2.5. Die in der Praxis wohl am häufigsten genutzten Verfahren zur Lösung steifer Systeme sind die auf den rückwärtigen Differentiationsformeln beruhenden Mehrschrittverfahren (BDF, [47]). Die bekanntesten Codes sind LSODE von Hindmarsh [63], [64] und der Nachfolger VODE von Brown/Byrne/Hindmarsh [16]. Für hohe Genauigkeitsforderungen sind weiterhin Extrapolationsalgorithmen sehr effektiv, z.B. METAN1 von Bader/Deuflhard [8] (basiert auf der linear-impliziten Mittelpunktregel), EULSIM von Deuflhard [33] (basiert auf dem linear-impliziten Euler-Verfahren). Weitere Implementierungen von Extrapolationsverfahren findet man in [61]. □
Resultate numerischer Vergleiche verschiedener Codes für nichtsteife und steife Systeme und Einschätzungen werden in [60], [61] und [101] gegeben.

2.3. Partitionierte Verfahren. Wegen ihres beschränkten Stabilitätsgebietes und der damit verbundenen Schrittweiteneinschränkung sind explizite RK-Verfahren zur Lösung steifer Systeme nicht geeignet. Steife Integratoren erlauben für diese Probleme eine wesentlich größere Schrittweite, die den höheren Aufwand in einem Integrationsschritt (Berechnung der Jacobi-Matrix, Lösung nichtlinearer bzw. linearer Gleichungssysteme) mehr als kompensiert. Wendet man dagegen ein steifes Verfahren zur Lösung eines nichtsteifen Problems an, so entfällt der Vorteil der größeren Schrittweite, da diese jetzt durch die Genauigkeitsforderung bestimmt

wird. Ein steifes Verfahren ist daher für nichtsteife Systeme wesentlich aufwendiger als ein nichtsteifes, die Relation verschlechtert sich dabei mit wachsender Dimension des Systems.
Bei der Lösung praktischer Probleme ist häufig a priori nicht bekannt, ob ein steifes Problem vorliegt. Weiterhin ist oft nur ein Teilsystem steif. Es ist dann vorteilhaft, nur dieses mit einem impliziten oder linear-impliziten Verfahren zu lösen, da sich dadurch die Dimension der Gleichungssysteme auf die Anzahl der steifen Komponenten reduziert. Dieses Vorgehen bezeichnet man als Partitionierung des Differentialgleichungssystems, die Verfahren als *partitionierte Verfahren.*
Die Anzahl der Komponenten, die als steif angesehen werden müssen, ist abhängig vom verwendeten Verfahren, von der vorgegebenen Toleranz und vom Lösungsverlauf. Es ist folglich wünschenswert, daß ein Computer-Programm automatisch die steifen Komponenten erkennt und ein entsprechendes partitioniertes Verfahren verwendet, man spricht von *automatischer Partitionierung.*
In der Praxis erweist sich diese allgemeine Aufgabe aber als recht kompliziert. Zuverlässige und effektive Algorithmen existieren bereits für zwei spezielle Arten der Partitionierung:

(1) *Automatische Steifheitserkennung* für das Gesamtsystem. Es wird für jeden Integrationsschritt entschieden, ob das Gesamtsystem steif ist, und entsprechend wird ein steifes oder nichtsteifes Verfahren ausgewählt. Bekannte ESV-Codes sind RKF4RW [80] (Automatische Wahl zwischen ROW-Methode und explizitem RK-Verfahren), PAI4 [18] (adaptives und explizites RKV). Der bekannteste Mehrschritt-Code ist LSODA von Hindmarsh/Petzold ([64], [78]).

(2) *Feste, komponentenweise Partitionierung.* Hier werden die steifen Komponenten als bekannt vorausgesetzt, d.h., das System besteht aus miteinander gekoppelten steifen und nichtsteifen Teilsystemen. Ein typisches Beispiel dafür sind singulär gestörte Systeme

$$\begin{aligned} u' &= f(t,u,v), \quad 0 < \varepsilon \ll 1 \\ \varepsilon v' &= g(t,u,v). \end{aligned}$$

Sind die partiellen Ableitungen der Funktion f von moderater Größe, dann ist das erste Teilsystem nichtsteif, während das zweite für kleine ε sehr steif wird.
Durch geeignete Wahl der Matrix T als Jacobi-Matrix des steifen Teilsystems (die restlichen Elemente werden Null gesetzt) lassen sich speziell für W-Methoden und adaptive RK-Verfahren partitionierte Verfahren gewinnen, die die gleiche Ordnung wie das Grundverfahren besitzen, aber nur noch die Lösung linearer Gleichungssysteme geringerer Dimension (gleich der Anzahl der steifen Komponenten) erfordern ([100], [99]).
Weitere Beispiele für Verfahren mit fester, komponentenweiser Partitionierung findet man in [65], [53], [43], [80].

Das folgende Beispiel illustriert die Effektivität partitionierter Verfahren, sowie die Einschränkung der Schrittweite bei der Anwendung expliziter Verfahren auf steife Systeme:

Beispiel 2.2. Vereinfachtes Modell eines Kernreaktors [99].

$$\begin{aligned} y_1' =& 10^2(-5(y_2 - 7.5\cdot 10^2) - 7.2)y_1/3 + 10^5(3.02\cdot 10^{-4}y_3 + 8.28\cdot 10^{-4}y_4 \\ &+ 28.44\cdot 10^{-4}y_5 + 14.11\cdot 10^{-4}y_6 + 15.77\cdot 10^{-4}y_7 + 2.38\cdot 10^{-4}y_8)/3 \\ y_2' =& (1.36\cdot 10^5 y_1 - 3.3\cdot 10^2(y_2 - 30))/1.67 \\ y_3' =& 3(y_1 - y_3) \\ y_4' =& 1.13(y_1 - y_4) \end{aligned} \tag{2.6}$$

$$
\begin{aligned}
y_5' &= 0.301(y_1 - y_5)\\
y_6' &= 0.111(y_1 - y_6)\\
y_7' &= 3.05 \cdot 10^{-2}(y_1 - y_7)\\
y_8' &= 1.24 \cdot 10^{-2}(y_1 - y_8)
\end{aligned}
$$

mit $0 \leq t \leq 15$, $y_2(0) = 750$, $y_i(0) = 1$ für $i \neq 2$. Hierbei bezeichnet y_1 die durchschnittliche Neutronendichte, y_2 die Brennstofftemperatur und $y_3, ..., y_8$ die Kernkonzentrationen. Für dieses System ist bekannt, daß y_1 und y_2 die steifen Komponenten sind. □

Dieses Anfangswertproblem wurde mit DOPRI5 und mit folgenden Varianten von PAI4 gelöst:

- linear-implizit (adaptives RK-Verfahren 3. Ordnung),
- automatische Verfahrenswahl (zwischen dem Grundverfahren und dem zugehörigen expliziten RKV 3. Ordnung),
- feste, komponentenweise Partitionierung (2 steife Komponenten, nur lineare Gleichungssysteme der Dimension 2)

Tabelle 2.2 gibt den Aufwand der einzelnen Versionen für die Toleranzen $TOL = 10^{-2}$, 10^{-4} und 10^{-6} an. Die Rechnungen wurden in FORTRAN auf einem AT 486 mit 33 MHz durchgeführt.

Verf.	TOL	explizite Schritte	implizite Schritte	Fehler im Endpunkt	Rechenzeit in sec.
DOPRI5	10^{-2}	$***$			
	10^{-4}	34454		$6.2E-3$	27.5
	10^{-6}	34879		$5.9E-5$	27.8
PAI4 linear-implizit	10^{-2}		62	$4.8E-5$	0.3
	10^{-4}		346	$3.1E-6$	1.8
	10^{-6}		1338	$1.6E-7$	6.2
PAI4 autom. Verfahrenswahl	10^{-2}	0	62	$4.8E-5$	0.3
	10^{-4}	0	328	$6.0E-6$	1.7
	10^{-6}	184	1174	$1.6E-7$	5.6
PAI4, feste, komponentenweise Partitionierung	10^{-2}		72	$9.5E-4$	0.2
	10^{-4}		342	$1.1E-5$	0.5
	10^{-6}		1304	$4.7E-6$	1.7

TABELLE 2.2. Numerische Ergebnisse für Beipiel 2.6

Die numerischen Ergebnisse zeigen einige typische Erscheinungen auf:

- Bei expliziten Verfahren kann bei grober Toleranz die numerische Lösung instabil werden (im Beispiel Abbruch bei TOL=10^{-2}). Da die Schrittweite durch die Stabilität bestimmt wird, wächst der Aufwand nur unwesentlich mit schärferer Toleranz (TOL=10^{-4}, 10^{-6}).
- Mit schärferen Genauigkeitsforderungen nimmt die Anzahl der expliziten Schritte bei der automatischen Verfahrenswahl zu. Insbesondere werden häufig zu Beginn des Integrationsintervalls billige explizite Schritte ausgeführt.
- Bei exakter Kenntnis des steifen Teilsystems sind Verfahren mit fester, komponentenweiser Partitionierung sehr effektiv.

Für praktische Probleme führt eine modulare Modellbildung häufig auf Differentialgleichungssysteme mit nur schwach gekoppelten Teilsystemen, die sich unterschiedlich schnell ändern. Für derartige Systeme können sog. *Multirate-Methoden* [52] zu einer wesentlichen Senkung der

Rechenzeit führen. Neben einer eventuellen Behandlung der einzelnen Teilsysteme mit verschiedenen Verfahren, werden hierbei auch noch, den Änderungen der Teilsysteme angepaßte, unterschiedliche Schrittweiten verwendet. Da dabei für die einzelnen Teilsysteme Näherungswerte an verschiedenen Stellen berechnet werden, erfordern die Funktionsaufrufe eine Extrapolation bzw. Interpolation benötigter Zwischenwerte. Dieses Vorgehen hat Einfluß auf die Konsistenzordnung und die Stabilität der Verfahren. Untersuchungen dazu und Anwendungen auf elektrische Schaltkreise findet man in Günther/Rentrop [56], Stabilitätsuntersuchungen einfacher Multirate-Methoden in [3], [94], [95].

Während wir bisher eine bekannte Einteilung in steife und nichtsteife Teilsysteme vorausgesetzt haben, kommen wir jetzt kurz zum allgemeinen Fall der automatischen, komponentenweisen Partitionierung. Dafür gibt es im wesentlichen drei Zugänge:

(1) Nach einer gewissen Anzahl von Integrationsschritten wird jeweils ein Schritt für das Gesamtsystem mit einem expliziten und einem impliziten Verfahren ausgeführt. Aus dem Vergleich der jeweils geschätzten Fehler für die einzelnen Komponenten wird für die nächsten Schritte eine Einteilung in steife und nichtsteife Komponenten getroffen (Rentrop [80]).

(2) Die Einteilung in steife und nichtsteife Komponenten geschieht unter Verwendung der explizit berechneten Jacobi-Matrix. Erste Untersuchungen dazu für Mehrschrittverfahren findet man bei Enright/Kamel [43]. In Weiner u.a. [102] wird eine Übersicht über derartige Strategien gegeben. Numerische Tests zeigen, daß diese Partitionierung zuverlässig und effektiv ist, da man nur noch Gleichungssysteme geringerer Dimension zu lösen hat. Der wesentliche Nachteil besteht in der Notwendigkeit, von Zeit zu Zeit die volle Jacobi-Matrix zu berechnen.

(3) In den letzten Jahren wurden verstärkt Krylov-Techniken zur Lösung großer, steifer Systeme genutzt, so lösen Gear/Saad [51], Brown/Hindmarsh [17] und Byrne [26] in Mehrschrittverfahren die linearen Gleichungssysteme innerhalb der Newton-Iteration mit Hilfe eines Arnoldi-Prozesses. Saad/Gallopoulos [87] verwenden für lineare Differentialgleichungssysteme mit konstanten Koeffizienten die Integraldarstellung der Lösung und approximieren die Ausdrücke vom Typ $e^{At}u$ mit Hilfe von Krylov-Räumen. Für linear-implizite Runge-Kutta-Verfahren kann dieser Zugang als partitioniertes Verfahren mit $T_i = Q_i Q_i^T A$, $A = f_y$, in der i-ten Stufe interpretiert werden. Dabei ist $Q_i \in \mathbb{R}^{n,\varkappa_i}$ eine orthogonale Matrix, deren Spalten im Krylov-Raum $\mathrm{span}\{q_1, Aq_1, ..., A^{\varkappa_i - 1} q_1\}$ liegen. Durch die Approximation

$$Aq_j \approx (f(t_m, u_m + \delta q_j) - f(t_m, u_m))/\delta$$

ist dabei keine explizite Berechnung der Jacobi-Matrix erforderlich, man spricht von "Matrix-freien" Methoden. In den einzelnen Stufen des Verfahrens sind dann nur noch Gleichungssysteme der Dimension $\varkappa_i$ zu lösen. In den späteren Stufen wird dabei der Krylov-Raum der ersten Stufe erweitert. Stabilitätsaussagen, Abschätzungen für die benötigte Dimension des Krylovraumes und numerische Ergebnisse findet man in Büttner u.a. [24], [25].

3. Differential-algebraische Systeme und ihre numerische Behandlung

Die theoretische und numerische Behandlung von differential-algebraischen Systemen (auch Algebro-Differentialgleichungen, engl. differential algebraic equations – DAE's) ist ein junges und gegenwärtig aktuelles Forschungsgebiet. Das Lösungsverhalten dieser Systeme unterscheidet sich in starkem Maße von dem gewöhnlicher Differentialgleichungssysteme. Eine systematische Entwicklung von Diskretisierungsverfahren setzte Anfang der siebziger Jahre mit den Arbeiten von Gear [48] und Petzold [79] ein. Seitdem wurden zahlreiche Fortschritte sowohl in der theoretischen als auch in der numerischen Behandlung von DAE's erzielt. Eine Übersicht über den gegenwärtigen Stand der Forschung geben die Bücher Brenan/Cambell/Petzold [15], Griepentrog/März [54], Hairer/Lubich/Roche [58] und Hairer/Wanner [61].

3.1. Charakterisierung von differential-algebraischen Systemen. Zunächst wird das implizite Differentialgleichungssystem

$$F(t, y(t), y'(t)) = 0, \quad t \in [t_0, t_e], \quad F : [t_0, t_e] \times \mathbb{R}^n \times \mathbb{R}^n \tag{3.1}$$

mit entsprechenden Anfangsbedingungen betrachtet. Dabei ist F stetig und bez. y' stetig differenzierbar. Ist $\partial F/\partial y'$ in einer Umgebung der Lösung $y(t)$ regulär, so ist (3.1) nach dem Satz über implizite Funktionen lokal nach $y'(t)$ auflösbar,

$$y'(t) = f(t, y(t)),$$

so daß (3.1) ein System gewöhnlicher Differentialgleichungen in impliziter Form darstellt. Ist $\partial F/\partial y'$ dagegen singulär, so ist diese Auflösung nicht möglich. Da wir uns nicht für isolierte singuläre Punkte interesssieren, setzen wir im weiteren voraus, daß $\partial F/\partial y'$ im gesamten betrachteten Gebiet singulär ist. Das System (3.1) enthält dann (evtl. nach algebraischen Umformungen) algebraische Gleichungen, es liegt ein Algebro-Differentialgleichungssystem vor. Systeme der Form

$$\begin{aligned} u' &= f(t, u, v) \\ 0 &= g(t, u, v), \end{aligned} \tag{3.2}$$

oder allgemeiner

$$\begin{aligned} F_1(t, u, u', v) &= 0 \\ F_2(t, u, v) &= 0, \end{aligned} \tag{3.3}$$

wobei $\partial F_1/\partial u'$ regulär ist, werden *semi-explizite DAE's* genannt. Hier sind die unbekannten Variablen partitioniert in die differentiellen Variablen u und in die algebraischen Variablen v. Die allgemeine (implizite) DAE (3.1) läßt sich durch

$$\begin{aligned} y' &= z \\ F(t, y, z) &= 0 \end{aligned}$$

stets in semi-expliziter Form darstellen. Semi-explizite DAE's in *Hessenberg-Form der Größe* $r \geq 2$ sind durch

$$\begin{aligned} w_1' &= G_1(t, w_1, w_2, \ldots, w_r) \\ w_2' &= G_2(t, w_1, w_2, \ldots, w_{r-1}) \\ w_i' &= G_i(t, w_{i-1}, \ldots, w_{r-1}), \quad 3 \leq i \leq r-1 \\ 0 &= G_r(t, w_{r-1}) \end{aligned} \tag{3.4}$$

gegeben, wobei das Matrixprodukt

$$\left(\frac{\partial G_r}{\partial w_{r-1}}\right)\left(\frac{\partial G_{r-1}}{\partial w_{r-2}}\right)\cdots\left(\frac{\partial G_2}{\partial w_1}\right)\left(\frac{\partial G_1}{\partial w_r}\right)$$

in einer Umgebung der Lösung regulär ist.

Ein wesentliches Kennzeichen einer DAE (3.1) ist ihr Index. Man unterscheidet den von Gear 1988 [49] eingeführten *Differentiationsindex* und den von Hairer/Lubich/Roche 1989 [58] eingeführten *Störungsindex.* Der Differentiationsindex macht Aussagen über die Struktur der DAE und charakterisiert den algebraischen Teil der DAE. Der Störungsindex dagegen mißt die Sensitivität der Lösung $y(t)$ der DAE gegenüber Störungen in der Gleichung. Er ist für die numerische Behandlung einer DAE von grundlegender Bedeutung.
Gemäß Gear hat die nichtlineare DAE (3.1) den Differentiationsindex $di = k$, wenn k die kleinste Anzahl von Differentiationen

$$\begin{aligned} F(t,y,y') &= 0 \\ \frac{d}{dt}F(t,y,y') &= F_t + F_y y' + F_{y'}y'' = 0 \\ &\vdots \\ \frac{d^k}{dt^k}F(t,y,y') &= \frac{\partial^k}{\partial t^k}F + \cdots + F_{y'}y^{(k+1)} = 0 \end{aligned} \tag{3.5}$$

ist, die nötig sind, um die Gleichung (3.5) durch algebraische Umformungen in ein explizites gewöhnliches Differentialgleichungssystem $y'(t) = \varphi(t,y)$ zu überführen. Dieses System wird häufig das der DAE (3.1) *zugrundeliegende Differentialgleichungssystem* genannt. Im Sonderfall $\partial F/\partial y'$ regulär (gewöhnliches Differentialgleichungssystem) ist der Differentiationsindex $di = 0$. Die Bestimmung des Differentiationsindex kann für "unstrukturierte" Systeme schwierig sein. Bei semi-expliziten DAE's vereinfacht sich die Indexbestimmung, es genügt hier, den algebraischen Teil zu differenzieren.
Wir betrachten nun den Störungsindex. Gemäß Hairer/Lubich/Roche [58] hat eine DAE (3.1) längs einer Lösung $y(t)$ auf dem Intervall $[t_0, t_e]$ den Störungsindex $pi = k$, wenn k die kleinste natürliche Zahl ist, so daß für alle Funktionen $\tilde{y}(t)$, die einen Defekt

$$F(t,\tilde{y},\tilde{y}') = \delta(t)$$

haben, im Intervall $[t_0, t_e]$ die Ungleichung

$$\|y(t)-\tilde{y}(t)\| \le C(\|y(t_0)-\tilde{y}(t_0)\| + \max_{t_0\le\xi\le t}\|\delta(\xi)\| + \cdots + \max_{t_0\le\xi\le t}\|\delta^{(k-1)}(\xi)\| \tag{3.6}$$

gilt, solange der Ausdruck auf der rechten Seite hinreichend klein bleibt. Die Konstante C ist nur von F und der Länge des Integrationsintervalls abhängig. Die höchste Ordnung der in der Ungleichung (3.6) vorkommenden Ableitung des Defektes $\delta(t)$ bestimmt also den Index der DAE. Der Störungsindex ist 0, wenn

$$\|y(t)-\tilde{y}(t)\| \le C(\|y(t_0)-\tilde{y}(t_0)\| + \max_{t_0\le\xi\le t}\|\int_{t_0}^{\xi}\delta(\tau)d\tau\|)$$

gilt, was für gewöhnliche Differentialgleichungssysteme (2.1) erfüllt ist.
Für semi-explizite DAE's (3.2) mit regulärer Matrix g_v gilt $d_i = p_i = 1$ und für DAE's in Hessenbergform (3.4) $di = pi = r$.
Die numerische Behandlung von DAE's mit einem Störungsindex $pi > 1$ ist dadurch problematisch, daß nicht nur die Störung $\delta(t)$, sondern auch deren Ableitungen bis zur Ordnung $pi - 1$ in die Lösung eingehen. Diese können normmäßig groß sein, auch wenn die Störung $\delta(t)$ selbst klein ist. Kleine Störungen können durchaus beliebig große Änderungen der Lösung $y(t)$ hervorrufen. In diesem Sinne gehören DAE's vom Index > 1 zu der Klasse der schlecht konditionierten Probleme. Für die numerische Lösung spiegeln sich die Ableitungen der Störung $\delta(t)$ als Differenzenquotienten in h wider, in die Näherungslösung gehen Fehler in der Größenordnung

$\mathcal{O}(h^{1-p_i})$ ein. Für kleine Schrittweiten h können demzufolge Rundungs- und Diskretisierungsfehler die Näherungslösung verfälschen. In der Praxis läßt sich dieser Effekt für sehr kleine Werte von h nachweisen [5].

3.2. Einschrittverfahren für semi-explizite DAE's vom Index 1. Ohne Einschränkung der Allgemeinheit betrachten wir semi-explizite DAE's in autonomer Form

$$\begin{aligned} u'(t) &= f(u,v), \quad u(t_0) = u_0 \in \mathbb{R}^N \\ 0 &= g(u,v), \quad v(t_0) = v_0 \in \mathbb{R}^{n-N}. \end{aligned} \tag{3.7}$$

Aufgrund der Index-1-Voraussetzung ist $g_v(u,v)$ in einer Umgebung der Lösung $u(t), v(t)$ regulär. Die Anfangswerte u_0, v_0 werden als konsistent vorausgesetzt ($g(u_0,v_0) = 0$).
Ein Einschrittverfahren zur Bestimmung einer Näherungslösung U_{m+1}, V_{m+1} für $u(t_{m+1})$, $v(t_{m+1})$ hat die Gestalt (Deuflhard/Hairer/Zugck [35])

$$\begin{aligned} U_{m+1} &= U_m + h\Phi(U_m, V_m; h) \\ V_{m+1} &= \Psi(U_m, V_m; h), \quad m = 0, 1, \ldots \end{aligned} \tag{3.8}$$

Ebenso wie bei Einschrittverfahren für gewöhnliche Differentialgleichungen ist auch hier die Darstellung (3.8) nur formal explizit und umfaßt auch implizite Verfahren. Das Verfahren (3.8) besitzt die Konsistenzordnung p, wenn mit $U_0 = u_0, V_0 = v_0$ für den lokalen Diskretisierungsfehler gilt

$$U_1 - u(t_0+h) = \mathcal{O}(h^{p+1}), \quad V_1 - v(t_0+h) = \mathcal{O}(h^p) \quad \text{für} \;\; h \to 0.$$

Im Unterschied zur Theorie der Einschrittverfahren für gewöhnliche Differentialgleichungen, wo für ein Verfahren der Ordnung p der lokale Diskretisierungsfehler von der Ordnung $\mathcal{O}(h^{p+1})$ sein muß, wird hier für den lokalen Diskretisierungsfehler der algebraischen Komponente nur $\mathcal{O}(h^p)$ gefordert. Hat das ESV (3.8) die Konsistenzordnung p und ist es gegenüber kleinen Störungen in den rechten Seiten stabil, so ist es auch konvergent von der Ordnung p (Strehmel/Weiner [101]).
Zur Konstruktion von Diskretisierungsverfahren für Index-1-Probleme bieten sich neben der Index-Reduktionsmethode (Rückführung der DAE durch Differentiation der algebraischen Nebenbedingungen auf ein gewöhnliches Differentialgleichungssystem) die beiden folgenden Vorgehensweisen an (Deuflhard/Hairer/Zugck [35]):
1. *Der direkte Weg*: Das Anfangswertproblem (3.7) wird in das singulär gestörte Anfangswertproblem

$$\begin{aligned} u'(t) &= f(u,v), \quad u(t_0) = u_0 \in \mathbb{R}^N \\ \varepsilon v'(t) &= g(u,v), \quad v(t_0) = v_0 \in \mathbb{R}^{n-N} \end{aligned} \tag{3.9}$$

$(0 < \varepsilon \ll 1)$ eingebettet. Darauf wendet man ein Diskretisierungsverfahren für steife Systeme an und betrachtet anschließend den Grenzübergang $\varepsilon \to 0$, der ein Diskretisierungsverfahren für das Index-1-Problem (3.7) liefert.
Für ein s-stufiges implizites Runge-Kutta-Verfahren mit regulärer Koeffizientenmatrix A führt

der direkte Weg zu der Verfahrensvorschrift (Hairer/Lubich/Roche [58])

$$
\begin{aligned}
U_{m+1}^{(i)} &= U_m + h \sum_{j=1}^{s} a_{ij} f(U_{m+1}^{(j)}, V_{m+1}^{(j)}) \\
0 &= g(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \\
U_{m+1} &= U_m + h \sum_{i=1}^{s} b_i f(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \\
V_{m+1} &= (1 - \sum_{i,j=1}^{s} b_i \mu_{ij}) V_m + \sum_{i,j=1}^{s} b_i \mu_{ij} V_{m+1}^{(j)},
\end{aligned}
$$

wobei μ_{ij} die Elemente von A^{-1} sind. Für die Stabilitätsfunktion $R_0(z)$ eines RK-Verfahrens mit invertierbarer Matrix A gilt

$$
R_0(\infty) = 1 - b^T A^{-1} \mathbb{1} = 1 - \sum_{i,j=1}^{s} b_i \mu_{ij},
$$

was gerade der Faktor vor V_m ist. Die Näherungswerte $U_{m+1}^{(i)}, V_{m+1}^{(i)}$ liegen dabei auf der Zwangsmannigfaltigkeit $g(u,v) = 0$. Damit auch die Endwerte U_{m+1}, V_{m+1} auf $g(u,v) = 0$ liegen, wird die Beziehung für V_{m+1} durch

$$
g(U_{m+1}, V_{m+1}) = 0. \tag{3.10}
$$

ersetzt. Für *steif genaue* RK-Verfahren, das sind RK-Verfahren mit

$$
a_{si} = b_i, \quad i = 1, \ldots, s, \quad \text{d.h.} \quad A^T e_s = b \ \text{ mit } \ e_s = (0, \ldots, 0, 1)^T
$$

z.B. Radau-IIA-Verfahren, gilt für die Stabilitätsfunktion $R_0(\infty) = 0$. Wegen

$$
\sum_{i,j=1}^{s} b_i \mu_{ij} V_{m+1}^{(j)} = \underbrace{b^T A^{-1}}_{e_s^T} (V_{m+1}^{(1)}, \ldots, V_{m+1}^{(s)})^T = V_{m+1}^{(s)}
$$

erhält man

$$
U_{m+1} = U_{m+1}^{(s)}, \quad V_{m+1} = V_{m+1}^{(s)},
$$

d.h., (3.10) ist stets erfüllt.

Für Rosenbrock-Verfahren führt der direkte Weg zu der Verfahrensvorschrift

$$
\begin{pmatrix} I - h\gamma f_u & -h\gamma f_v \\ -h\gamma g_u & -h\gamma g_v \end{pmatrix} \begin{pmatrix} k_i \\ l_i \end{pmatrix} = \begin{pmatrix} f(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \\ g(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \end{pmatrix} + h \begin{pmatrix} f_u & f_v \\ g_u & g_v \end{pmatrix} \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \tag{3.11}
$$

mit

$$
\begin{pmatrix} U_{m+1}^{(i)} \\ V_{m+1}^{(i)} \end{pmatrix} = \begin{pmatrix} U_m \\ V_m \end{pmatrix} + h \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \quad \begin{pmatrix} U_{m+1} \\ V_{m+1} \end{pmatrix} = \begin{pmatrix} U_m \\ V_m \end{pmatrix} + h \sum_{i=1}^{s} b_i \begin{pmatrix} k_i \\ l_i \end{pmatrix}
$$

(Roche [85], Rentrop/Roche/Steinebach [81]). Die partiellen Ableitungen sind dabei im Punkt (U_m, V_m) zu nehmen. Die Koeffizientenmatrix des linearen Gleichungssystems (3.11) ist wegen der Index-1-Voraussetzung für hinreichend kleine Schrittweiten $h > 0$ und $\gamma \neq 0$ regulär.

Allgemeine linear-implizite Runge-Kutta-Verfahren werden von Strehmel/Weiner [100] betrachtet. Extrapolationsverfahren mit dem linear-impliziten Euler-Verfahren (einstufiges ROW-Verfahren)

$$\begin{pmatrix} I - hf_u & -hf_v \\ -g_u & -g_v \end{pmatrix} \begin{pmatrix} U_{m+1} - U_m \\ V_{m+1} - V_m \end{pmatrix} = \begin{pmatrix} hf(U_m, V_m) \\ g(U_m, V_m) \end{pmatrix}$$

als Grundverfahren und der harmonischen Unterteilungsfolge $h_l = H/n_l$, $\{n_l\} = \{1, 2, \dots\}$ werden von Deuflhard/Hairer/Zugck [35] untersucht.
2. *Der indirekte Weg*: Aufgrund der Regularität von $g_v(u, v)$ existiert in der Nachbarschaft der Lösung von (3.7) eine lokal eindeutige Lösung $v(t) = G(u(t))$ auf $[t_0, t_e]$. Wird diese in die erste Gleichung von (3.7) eingesetzt, so erhält man das Anfangswertproblem

$$u'(t) = f(u(t), G(u(t))), \quad u(t_0) = u_0,$$

das mit Standarddiskretisierungsmethoden gelöst werden kann.
Unter Zugrundelegung eines s-stufigen expliziten RK-Verfahrens für eine semi-explizite DAE mit nichtsteifem differentiellen Anteil (3.7) bekommt man das *halbexplizite RK-Verfahren* (Hairer/Lubich/ Roche [58])

$$\begin{aligned} U_{m+1}^{(i)} &= U_m + h \sum_{j=1}^{i-1} a_{ij} f(U_{m+1}^{(j)}, V_{m+1}^{(j)}), \quad i = 1, \dots, s \\ 0 &= g(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \\ U_{m+1} &= U_m + h \sum_{i=1}^{s} b_i f(U_{m+1}^{(i)}, V_{m+1}^{(i)}) \\ 0 &= g(U_{m+1}, V_{m+1}). \end{aligned}$$

Es erfordert in jeder Stufe die Lösung eines nichtlinearen Gleichungssystems der Dimension $n - N$. Für praktische Rechnungen ist es oft vorteilhaft, das nichtlineare Gleichungssystem nicht "exakt" zu lösen, sondern für $V_{m+1}^{(i)}, V_{m+1}$ Näherungswerte zu verwenden, deren Genauigkeit dem lokalen Diskretisierungsfehler angepaßt ist. Derartige Verfahren, die aus einer Kombination eines expliziten RK-Verfahrens und eines vereinfachten Newtonverfahrens (für die Lösung des algebraischen Teils der DAE) mit einer vorgegeben Anzahl von Iterationen bestehen, sog. *Runge-Kutta-Newton-Verfahren*, werden in Arnold/Strehmel/Weiner [6] betrachtet. So ist das Euler-Newton-Verfahren durch

$$\begin{aligned} U_{m+1} &= U_m + hf(U_m, V_m) \\ V_{m+1} &= U_m - (g_v(U_m, V_m))^{-1} g(U_m, V_m). \end{aligned}$$

gegeben . Es besitzt die Konvergenzordnung $p = 1$. Zur Bestimmung von V_{m+1} hat man ein lineares Gleichungssystem der Dimension $n - N$ zu lösen.
Der direkte Weg wurde erstmals von Gear [48] für die Klasse der linearen Mehrschrittverfahren vom BDF-Typ vorgeschlagen. Der indirekte Weg dagegen berücksichtigt die spezielle Struktur der DAE. In einigen Fällen führen beide Wege zum gleichen Diskretisierungsverfahren, z.B. für steif genaue RK-Verfahren.
Die Anwendung des indirekten Weges auf ROW-Verfahren sowie linear-implizite Runge-Kutta-Verfahren ist nicht zweckmäßig, da der Hauptvorteil gegenüber impliziten RK-Verfahren, nämlich das Lösen nur linearer Gleichungssysteme in jedem Integrationsschritt, verlorengeht.

Bemerkung 3.1. Eine breite Anwendung zur numerischen Lösung von Index-1-Systemen finden die Programmpakete LSODI (Hindmarsh [63]) und speziell in den letzten Jahren DASSL (Brenan/Campbell/Petzold [15]). Beide Codes beruhen auf den BDF-Formeln. Sie unterscheiden sich in der Darstellungsweise der zugrundeliegenden Interpolationspolynome. Während LSODI die Nordsieck-Darstellung verwendet, basiert DASSL auf der Darstellung mittels dividierter Differenzen. □

3.3. Einschrittverfahren für Mehrkörpersysteme. Die Bewegung eines mechanischen Mehrkörpersystems wird durch

$$\begin{aligned} M(p)p'' &= f(p,p',t) - G^T(p)\lambda \\ g(p) &= 0 \end{aligned}$$

mit

- den n_p Lagekoordinaten p
- den n_λ Lagrange-Multiplikatoren λ, $n_\lambda < n_p$
- der $(n_p \times n_p)$ Massematrix $M(p)$
- den n_p eingeprägten und äußeren Kräften $f(p,p',t)$
- den n_λ Zwangsbedingungen $g(p)$
- der $(n_\lambda \times n_p)$ Zwangsmatrix $G(p) = \partial g(p)/\partial p$

beschrieben (vgl. Abschnitt 1). Nach Transformation in ein System erster Ordnung erhalten wir daraus das semi-explizite differential-algebraische System

$$p' = v \tag{3.12a}$$

$$M(p)v' = f(p,v,t) - G^T(p)\lambda \tag{3.12b}$$

$$0 = g(p). \tag{3.12c}$$

Es besteht aus $2n_p$ Differentialgleichungen gekoppelt mit n_λ algebraischen Gleichungen. Differentiations- und Störungsindex sind äquivalent. Im weiteren wird vorausgesetzt:

$$\begin{aligned} &M(p) \text{ symmetrisch, positiv definit} \\ &g(p) \text{ hinreichend glatt} \\ &\text{rang } G(p) = n_\lambda. \end{aligned}$$

Bemerkung 3.2. Die Forderung, daß $G(p)$ Vollrang hat, wird in der Ingenieur-Literatur als *Grübler-Bedingung* bezeichnet. □

Differentiation der Lage-Nebenbedingung (3.12c) ergibt die Geschwindigkeits-Nebenbedingung

$$\frac{d}{dt} g(p) = G(p)p' = G(p)v = 0. \tag{3.13}$$

Wird (3.13) nach t differenziert, so erhält man die Beschleunigungs-Nebenbedingung

$$\frac{d^2}{dt^2} g(p) = G(p)v' + G_p(p)(v,v) = 0. \tag{3.14}$$

Die Beziehungen (3.12b) und (3.14) liefern für v' und λ das lineare System

$$\begin{pmatrix} M(p) & G^T(p) \\ G(p) & 0 \end{pmatrix} \begin{pmatrix} v' \\ \lambda \end{pmatrix} = \begin{pmatrix} f(p,v,t) \\ -G_p(p)(v,v) \end{pmatrix}. \tag{3.15}$$

Da $M(p)$ positiv definit ist und $G(p)$ Vollrang hat, ist die Koeffizientenmatrix regulär, so daß wir (3.15) nach v' und λ auflösen können. Wir erhalten

$$v' = M^{-1}(p)[f(p,v,t) - G^T(p)\lambda] \tag{3.16a}$$

$$\lambda = \Lambda(p,v,t) \tag{3.16b}$$

mit

$$\Lambda(p,v,t) = \left[G(p)M^{-1}(p)G^T(p)\right]^{-1}\left(G(p)M^{-1}(p)f(p,v,t) + G_p(p)(v,v)\right). \tag{3.17}$$

Damit ergibt sich die gewöhnliche Differentialgleichung (*zugrundeliegende Differentialgleichung*) für (p,v)

$$p' = v \tag{3.18a}$$

$$M(p)v' = f(p,v,t) - G^T(p)\Lambda(p,v,t). \tag{3.18b}$$

Eine weitere Differentiation von (3.16b) nach t und Einsetzen der Beziehungen (3.18a) für p' und (3.18b) für v' führen auf ein gewöhnliches Differentialgleichungssystem für $p(t), v(t), \lambda(t)$. Aufgrund der ausgeführten Differentiationen hat das mechanische Mehrkörpersystem (3.12) den Index 3. Für jeden konsistenten Anfangswert $(p_0,v_0)^T = (p(t_0), v(t_0))^T$ besitzt die zugrundeliegende Differentialgleichung (3.18) eine eindeutig bestimmte Lösung $(p(t), v(t))^T$ auf $[t_0, t_e]$. Dann ist $(p(t), v(t), \lambda(t))^T$ mit $\lambda(t) = \Lambda(p(t), v(t), t)$ die eindeutige Lösung des Index-1-Systems

$$p' = v \tag{3.19a}$$

$$M(p)v' = f(p,v,t) - G^T(p)\lambda \tag{3.19b}$$

$$0 = G_p(p)(v,v) + G(p)v'. \tag{3.19c}$$

Ebenso ist $(p(t), v(t), \lambda(t))^T$ die eindeutige Lösung des Index-2-Systems

$$\begin{aligned} p' &= v \\ M(p)v' &= f(p,v,t) - G^T(p)\lambda \\ 0 &= G(p)v \end{aligned} \tag{3.20}$$

sowie des Index-3-Systems (3.12).

Bemerkung 3.3. Anfangswerte (p_0, v_0) heißen konsistent, wenn sie den Konsistenzbedingungen

$$g(p_0) = 0$$
$$G(p_0)v_0 = 0$$

genügen. Der Anfangswert $\lambda_0 = \Lambda(p_0, v_0, t_0)$ ist dann eindeutig bestimmt. □

Für einen konsistenten Anfangswert $(p_0, v_0)^T$ liegt die Lösung $(p(t), v(t))^T$ der zugrundeliegenden Differentialgleichung (3.18) auf den durch die Lage- und Geschwindigkeits-Nebenbedingungen definierten Mannigfaltigkeiten. Diese Mannigfaltigkeiten sind demzufolge Invarianten von (3.18).

Die drei unterschiedlichen Formulierungen (3.12), (3.19), (3.20) eines mechanischen Mehrkörpersystem sind mathematisch äquivalent. Die analytische Lösung von (3.12) erfüllt neben der Lage-Nebenbedingung auch die Geschwindigkeits- und Beschleunigungs-Nebenbedingung. Die beiden letzteren Nebenbedingungen werden auch "versteckte" Nebenbedingungen genannt, da sie in der Formulierung des Index-3-Systems nicht explizit auftreten. Völlig anders liegen die

Verhältnisse bei der numerischen Behandlung dieser Formulierungen. Bezüglich des Störungsindex ist die Index-1-Formulierung (3.19) am vorteilhaftesten. In diesem Fall gehen, im Gegensatz zu der Index-3-Formulierung (3.12), keine Fehlerterme der Ordnungen $\mathcal{O}(h^{-2})$ und $\mathcal{O}(h^{-1})$ in die Näherungslösung ein. Andererseits besitzt die Index-1-Formulierung aber den Nachteil, daß statt der Lage-Nebenbedingung $g(p(t)) = 0$ die Beschleunigungs-Nebenbedingung

$$0 = \frac{d^2}{dt^2} g(p(t)) = g''(p(t)) \tag{3.21}$$

erfüllt wird. Aufgrund der im Integrationsprozeß unvermeidlich auftretenden Rundungs- und Abbruchfehler wird die Nebenbedingung (3.21) nicht exakt erfüllt. Geht man davon aus, daß in jedem Integrationsschritt $t_m \to t_m + h$ ein Fehler $e(t_m)$, $\max_{t_0 \le t \le t_e} \|e(t)\| \le \varepsilon$, gemacht wird, so wird statt (3.21) die Differentialgleichung

$$g''(p(t)) = e(t)$$

integriert. Daraus erhält man die Abschätzungen

$$\|G(p(t))v\| = \|g'(p(t))\| \le \varepsilon(t - t_0) + \varepsilon_1 \tag{3.22}$$

$$\|g(p(t))\| \le \frac{1}{2}\varepsilon(t - t_0)^2 + \varepsilon_1(t - t_0) + \varepsilon_0, \tag{3.23}$$

wobei $\varepsilon_1 = \|g'(p(t_0))\|$, $\varepsilon_0 = \|g(p(t_0))\|$ gesetzt wurde. Man erkennt, daß der Lage-Nebenbedingungsfehler von einer quadratischen und der Geschwindigkeits-Nebenbedingungsfehler von einer linearen Funktion der Zeit begrenzt werden. Die Lösung "driftet" demzufolge von der Lage- und Geschwindigkeits-Mannigfaltigkeit ab. Dieser Drift-Effekt wirkt sich auch auf die Approximationseigenschaften der Diskretisierung aus (Alishenas [2]) und kann die numerische Lösung völlig verfälschen. Er ist von der Schrittweite h und der Nichtlinearität der Zwangsbedingungen abhängig. Die Index-1-Formulierung erweist sich damit im Sinne von Ljapunov als instabil.

Für strukturierte Systeme mit bekanntem Index, zu denen die Mehrkörpersysteme gehören, läßt sich der Drift-Effekt durch Stabilisierungstechniken abschwächen bzw. vermeiden. Man unterscheidet für die Deskriptorform (3.12) zwei verschiedene Arten:

1. **Stabilisierung durch Invarianten**
   Das mechanische Mehrkörpersystem (3.12) vom Index 3 wird als indexreduziertes System unter Einbeziehung von Invarianten formuliert. Auf dieses erweiterte System wird eine Standarddiskretisierungsmethode angewendet.
2. **Stabilisierung durch Nebenbedingungen**
   Die Deskriptorform vom Index 3, 2, 1 wird diskretisiert. Anschließend wird zu jedem Zeitpunkt $t_m + h$, $m = 0, 1, \ldots$ eine Stabilisierung durch Projektion vorgenommen.

Zur ersten Art zählen:

a) *Baumgarte-Stabilisierung*
   In der Index-1-Formulierung (3.19) wird nach Baumgarte [11] die instabile Beschleunigungs-Nebenbedingung (3.21) durch eine Linearkombination

$$g''(p(t)) + 2\alpha g'(p(t)) + \alpha^2 g(p(t)) = 0 \ \text{ mit } \ \alpha > 0 \tag{3.24}$$

   ersetzt. Dies ist eine asymptotisch stabile Differentialgleichung, in der der Baumgarte-Parameter α frei wählbar ist.

Statt (3.19) ergibt sich mit (3.13) und (3.14) das stabilisierte Index-1-System

$$\begin{aligned} p' &= v && (3.25a) \\ M(p)v' &= f(p,v,t) - G^T(p)\lambda && (3.25b) \\ 0 &= G_p(p)(v,v) + (1+2\alpha)G(p)v + \alpha^2 g(p). && (3.25c) \end{aligned}$$

Gegenüber der algebraischen Gleichung (3.19c) treten in (3.25c) die beiden Terme

$$2\alpha G(p)v \quad \text{und} \quad \alpha^2 g(p)$$

(*stabilisierende Regelungsglieder*) auf. Für $\alpha = 0$ geht (3.25) in das nichtstabilisierte Index-1-System (3.19) über. Für $\alpha > 0$ werden bei einer numerischen Integration auftretende Störungen in der Zwangsbedingung (3.25c) über die Regelungsglieder exponentiell gedämpft. Schwierigkeiten bereitet allerdings die Festlegung des Parameters α. Wird α zu klein gewählt, so erhält man kaum bessere Ergebnisse als für die Index-1-Formulierung (3.19). Ein sehr großer Wert von α kann eine Steifheit hervorrufen, die nicht den physikalischen Eigenschaften des Systems entspricht. Sie führt damit zu einer Vergrößerung des Aufwandes. Die Wahl von α wird von zahlreichen Faktoren beeinflußt, wie z.B. vom gegebenen Mehrkörpersystem, vom Diskretisierungsverfahren, von der Anfangsschrittweite und von der Anfangsgeschwindigkeit des mechanischen Systems. Eine allgemeine Strategie zur Bestimmnung des Parameters α ist nicht möglich (Ascher/Petzold [7]). Dieser Umstand macht eine breite Anwendung der Baumgarte-Stabilisierung problematisch.

Bemerkung 3.4. Wir haben in (3.24) bereits den aperiodischen Grenzfall ($\alpha = \beta$) in die allgemeine Baumgarte-Formulierung

$$g''(p(t)) + (\alpha+\beta)g'(p(t)) + \alpha\beta g(p(t)) = 0$$

eingesetzt. □

b) *Ankopplung von Invarianten*
Hierzu zählt die **G**ear-**G**upta-**L**eimkuhler-Formulierung (GGL-Formulierung, [50]). Sie besteht darin, daß in das Index-2-System (3.20) ein zusätzlicher Lagrange-Multiplikator μ eingeführt wird, der die Lage-Nebenbedingung $g(p) = 0$ an das System ankoppelt. Die GGL-Formulierung lautet:

$$\begin{aligned} p' &= v - G^T(p)\mu \\ M(p)v' &= f(p,v,t) - G^T(p)\lambda \\ 0 &= g(p) \\ 0 &= G(p)v. \end{aligned} \tag{3.26}$$

Wegen

$$0 = G(p)p' = G(p)v - G(p)G^T(p)\mu.$$

folgt $\mu = 0$ für jede Lösung $p(t), v(t), \lambda(t)$. Das System (3.26) ist vom Index 2. Bezüglich des Störungsindex ist (3.26) für die numerische Behandlung besser geeignet als die Index-3-Formulierung (3.12).

Geht man von der Index-1-Formulierung (3.19) aus und will Lage- und Geschwindigkeits-Nebenbedingung an das System ankoppeln, so liefert eine Gear-Gupta-Leimkuhler-Formulierung das Index-2-System

$$\begin{aligned} p' &= v - G^T(p)\mu - \left[\frac{d^2}{dp^2} g(p)(v,\cdot)\right]^T \eta \\ M(p)v' &= f(p,v,t) - G^T(p)\lambda - G^T(p)\eta \\ 0 &= g(p) \\ 0 &= G(p)v \\ 0 &= G(p)v' + G_p(p)(v,v). \end{aligned} \tag{3.27}$$

Dieses Vorgehen steht in engem Zusammenhang mit dem von Führer [46], der (3.27) mit BDF-Verfahren diskretisiert und das resultierende überbestimmte nichtlineare Gleichungssystem mit einer speziellen Iterationstechnik löst.

c) *Formulierung als Projizierende Deskriptorform*
Die Invarianten können auch als Projektion in die Bewegungsgleichungen vorab integriert werden. Beim Ansatz nach Simeon [92] führt man dazu zusätzliche Lagevariablen q und Geschwindigkeitsvariablen u ein, die keiner Nebenbedingung genügen müssen. Mit den Multiplikatoren η und τ, Beschleunigungsvariablen w und der Krümmungsinformation $r(p,v,z,\eta) := M_p(p)(z,v) + (d^2/dp^2 g(p)(v,\cdot))^T\eta$, vgl. (3.27), lautet die Projizierende Deskriptorform der Bewegungsgleichungen

$$M(p)q' = M(p)v + r(p,p',p-q,\tau) \tag{3.28a}$$

$$M(p)u' = M(p)w + r(p,v,v-u,\eta) \tag{3.28b}$$

$$0 = \begin{pmatrix} M(p)w - f(p,v,t) + G(p)^T\lambda \\ G(p)w + G_p(p)(v,v) \end{pmatrix} \tag{3.28c}$$

$$0 = \begin{pmatrix} M(p)(v-u) + G(p)^T\eta \\ G(p)v \end{pmatrix} \tag{3.28d}$$

$$0 = \begin{pmatrix} M(p)(p-q) + G(p)^T\tau \\ g(p) \end{pmatrix} \tag{3.28e}$$

Die gesamte Information der Bewegungsgleichungen ist in diesem System vom Index 1 in expliziter Form enthalten. Die Lösung erfüllt in den Komponenten p, v, λ das ursprüngliche System (3.12) vom Index 3 mit sämtlichen Invarianten. Die Nebenbedingungen (3.28d) und (3.28e) stellen Projektionen dar, die inkonsistente Werte von q und u in einer durch die Massematrix definierten Metrik auf p und v abbilden. Insbesondere liefert die Projizierende Deskriptorform damit eine Vorschrift für die Berechnung konsistenter Anfangswerte und die Anwendung stabilisierender Projektionen, s.u. und Rheinboldt/Simeon [82].

Bemerkung 3.5. Der Krümmungsterm r ist a priori in den Bewegungsgleichungen vorhanden und stellt keine künstliche Information dar. Die Kräfte f umfassen u.a. Terme mit Ableitungen der Massematrix (verallgemeinerte Corioliskräfte), und die Ableitungen der Zwangsmatrix G finden sich in den Beschleunigungsnebenbedingungen wieder. □

Wir kommen nun zur zweiten Art von Diskretisierungsverfahren (Stabilisierung durch Nebenbedingungen). Hierzu gehören:

a) *Projektionsverfahren für die Index-1-Formulierung* (3.19)
Das System (3.19) wird direkt oder nach Elimination von λ gemäß (3.17) als gewöhnliches Differentialgleichungssystem diskretisiert. Das Abdriften (3.22) kann durch Projektion der Näherungslösung auf die Zwangsmannigfaltigkeiten vermieden werden.
Sei $\tilde{p}_1$ die Lagekoordinate der berechneten Näherungslösung des Index-1-Systems zum Zeitpunkt $t_0 + h$. Um die numerische Lösung bezüglich der Lage konsistent zu machen (Lage-Stabilisierung), hat man das unterbestimmte nichtlineare Gleichungssystem

$$g(p) = 0 \ \text{ mit } \ g : \mathbb{R}^{n_p} \to \mathbb{R}^{n_\lambda}, \ \ n_p > n_\lambda \tag{3.29}$$

zu lösen, das unendlich viele Lösungen besitzt. Alle diese Lösungen lassen sich als Projektionen der gestörten Lösung $\tilde{p}_1$ auf die Lage-Mannigfaltigkeit

$$L = \{p \in \mathbb{R}^{n_p} \mid g(p) = 0\}$$

des mechanischen Mehrkörpersystems auffassen. Eine dieser Projektionen ist dadurch gekennzeichnet, daß sie (in einer geeigneten Metrik) orthogonal in den linearen Unterraum projiziert, in dem der Vektor $p_1 \in L$ liegt. Eine näherungsweise Bestimmung dieser orthogonalen Projektion kann mit Hilfe des vereinfachten Newton-Verfahrens

$$\begin{aligned} G(\tilde{p}_1) d_p^{(k)} &= g(p_1^{(k)}) \\ p_1^{(k+1)} &= p_1^{(k)} - d_p^{(k)} \quad k = 0, 1 \ldots, \varkappa \text{ mit } p_1^{(0)} = \tilde{p}_1 \end{aligned} \tag{3.30}$$

erfolgen. Unter den unendlich vielen Lösungen des linearen Gleichungssystems (3.30) wird dabei diejenige minimaler Länge in der Norm

$$\|d\|_M = d^T M(\tilde{p}_1) d \tag{3.31}$$

gewählt. Die Jacobi-Matrix $G(\tilde{p}_1)$ braucht dabei nicht neu aufgestellt zu werden, sie ist bereits Bestandteil der Deskriptorform (3.12).
Eine ausführliche Diskussion der Stabilisierung findet man bei Alishenas [2].
Wir kommen nun zur Geschwindigkeits-Stabilisierung. Bezeichnen p_1, $\tilde{v}_1$ die Lage- und Geschwindigkeitskomponenten der berechneten Näherungslösung des Index-1-Systems (3.19) zum Zeitpunkt $t_0 + h$, dann ist diese gegeben durch

$$G(p_1) v = 0. \tag{3.32}$$

Die unendlich vielen Lösungen von (3.32) lassen sich als Projektionen der numerischen Lösung $\tilde{v}_1$ in den linearen Geschwindigkeitsraum $\{v \in \mathbb{R}^{n_p} \mid G(p_1)v = 0\}$ auffassen. Die Lösung minimaler Länge $v = 0$ scheidet aus physikalischen Gründen aus. Daher führt man die Variable $d_v = \tilde{v}_1 - v_1$ ein, so daß (3.32) in das äquivalente inhomogene lineare Gleichungssystem

$$G(p_1) d_v = G(p_1) \tilde{v}_1$$

übergeht. Die Lösung minimaler Länge in der Norm (3.31) mit $M(p_1)$ ist durch die Lösung des linearen Gleichungssystems

$$\begin{aligned} M(p_1) d_v &= G^T(p_1) \mu \\ G(p_1) v_1 &= 0 \end{aligned}$$

(vgl. Alishenas [2], Lubich [73]) gegeben. Dabei bestimmt man zuerst μ aus

$$G(p_1) M^{-1}(p_1) G^T \mu = G(p_1) \tilde{v}_1.$$

Bemerkung 3.6. Eine Eliminierung des Geschwindigkeitsfehlers wirkt sich auf den Integrationsprozeß effektiver aus als eine Eliminierung des Lagefehlers (Alishenas [2]). Außerdem erfordert eine Geschwindigkeits-Stabilisierung aufgrund der Linearität weniger Rechenaufwand.
Weitere Stabilisierungen (z.B. simultane Lage- und Geschwindigkeits-Stabilisierung oder Energie-Stabilisierung) findet man in Alishenas [2]. □

b) *Halbexplizite Verfahren für die Deskriptorform vom Index 2*
Ein expliziter Euler-Schritt mit konsistenten Anfangswerten p_0, v_0 für die zugrundeliegende Differentialgleichung (3.18) und anschließender Geschwindigkeits-Stabilisierung lautet

$$p_1 - p_0 = hv_0 \tag{3.33a}$$
$$M(p_0)(\tilde{v}_1 - v_0) = hf(p_0, v_0, t_0) - hG^T(p_0)\Lambda(p_0, v_0, t_0) \tag{3.33b}$$
$$M(p_1)(\tilde{v}_1 - v_1) = G^T(p_1)\mu \tag{3.33c}$$
$$G(p_1)v_1 = 0. \tag{3.33d}$$

Ersetzt man in (3.33c) das Argument p_1 durch p_0 und subtrahiert dieses Ergebnis von (3.33b), so ergibt sich die Verfahrensvorschrift

$$p_1 = p_0 + hv_0 \tag{3.34a}$$
$$M(p_0)(v_1 - v_0) = hf(p_0, v_0, t_0) - hG^T(p_0)(\underbrace{\Lambda(p_0, v_0, t_0) + \mu/h}_{\lambda}) \tag{3.34b}$$
$$G(p_1)v_1 = 0. \tag{3.34c}$$

Die erste Gleichung definiert p_1 explizit. Die Größen v_1 und λ bestimmen sich aus einem linearen, nichtsymmetrischen Gleichungssystem. Der Vorteil von (3.34) gegenüber (3.33) besteht darin, daß weder $\Lambda(p_0, v_0, t_0)$ noch $\tilde{v}_1$ berechnet werden muß, d.h., (3.34) erfordert keine Berechnung von $G_p(p)(v, v)$. Ein Nachteil ist, daß die Koeffizientenmatrix des linearen Systems (3.34b), (3.34c) nicht symmetrisch ist.
Die Anwendung dieser Vorgehensweise auf jede Stufe eines expliziten RK-Verfahrens führt zur Klasse der halbexpliziten RK-Verfahren von Brasey/Hairer ([13]).
Der Nachteil einer nichtsymmetrischen Koeffizientenmatrix in (3.34b), (3.34c) läßt sich vermeiden. Ersetzt man in (3.34b) das Argument in $M(p_0)$ und $G^T(p_0)$ durch p_1, so führt dies zu der halb-expliziten Euler-Methode

$$\begin{aligned} p_1 &= p_0 + hv_0 \\ M(p_1)(v_1 - v_0) &= hf(p_0, v_0, t_0) - hG^T(p_1)\lambda \\ G(p_1)v_1 &= 0. \end{aligned}$$

Die Koeffizientenmatrix des linearen Systems für v_1, λ ist jetzt symmetrisch und hat die gleiche Struktur wie (3.15). Die Ausdehnung dieser Vorschrift auf s-stufige explizite RK-Verfahren liefert die Klasse der symmetrischen halb-expliziten RK-Verfahren (Brasey/Hairer [14]).

Bemerkung 3.7. Der halbexplizite Euler-Schritt (3.33a),(3.33b) dient auch als Startschritt für ein von Lubich [73] entwickeltes Extrapolationsverfahren. Ein auf diesem Verfahren basierender Code MEXX (MEXanical systems eXtrapolation integrator) wurde von Lubich u.a. entwickelt (vgl. [74]). Numerische Ergebnisse zeigen, daß MEXX ein zuverlässiger und effektiver Code zur numerischen Lösung von mechanischen Mehrkörpersystemen ist. □

c) *Direkte Diskretisierung der Deskriptorform vom Index 3*
Wird das Mehrkörpersystem (3.12) in der Form

$$\begin{aligned} p' &= v \\ v' &= M^{-1}(p)[f(p,v,t) - G^T(p)\lambda] \\ 0 &= g(p) \end{aligned} \tag{3.35}$$

geschrieben, so liegt eine DAE in Hessenberg-Form vom Index 3 vor, in die der Lagrange-Parameter λ linear eingeht. Implizite Runge-Kutta-Verfahren für DAE's in Hessenberg-Form vom Index 3 wurden von Hairer/Lubich/Roche [58] und Hairer/Jay [57] untersucht. Für das Mehrkörpersystem (3.35) besitzt eine s-stufige Radau-IIA-Methode bez. der Variablen p die Konvergenzordnung $2s-1$, bez. der Variablen v die Ordnung s und bez. λ die Ordnung $s-1$. Führt man für v' eine neue Variable w ein, so geht das System (3.35) über in das System

$$\begin{aligned} p' &= v \\ v' &= w \\ 0 &= M(p)w - f(p,p',t) + G^T(p)\lambda \\ 0 &= g(p), \end{aligned} \tag{3.36}$$

so daß ein linear-implizites System der Form $Ay' = \phi(y,t)$ vom Index 3 mit konstanter singulärer Matrix A vorliegt. Für implizite RK-Verfahren sind die beiden Formulierungen (3.35) und (3.36) äquivalent. Sie unterscheiden sich lediglich in der Lösung des nichtlinearen Gleichungssystems mit einem vereinfachten Newton-Verfahren. Der von Hairer/Wanner [61] entwickelte Code RADAU5 für linear-implizite Systeme $Ay' = \phi(y,t)$ vom Index ≤ 3 kann auf (3.36) direkt angewendet werden. Für die Fehlerschätzung und Schrittweitensteuerung werden in diesem Code die Komponenten von v mit der Schrittweite h und die von w und λ mit h^2 multipliziert (skaliert).
Eine Klasse halb-expliziter RK-Verfahren für DAE's in Hessenberg-Form vom Index drei sowie und ein Extrapolationsverfahren wurden von Ostermann [76], [77] untersucht.

Literatur

1. R.C. Aiken, *Stiff Computation*, Oxford University Press, New York–Oxford, 1985.
2. T. Alishenas, *Zur numerischen Behandlung, Stabilisierung durch Projektion und Modellierung mechanischer Systeme mit Nebenbedingungen und Invarianten*, Dissertation, Royal Institute of Technology, Stockholm, 1992.
3. J.F. Andrus, *Stability of a multi-rate method for numerical integration of ode's*, Computers Math. Applic. **25** (1993), 3–14.
4. T. Andrzejewski, H.G. Bock, E. Eich, and R. von Schwerin, *Recent advances in the numerical integration of multibody systems*, In: Advanced multibody system dynamics (W. Schiehlen, ed.), Kluwer Academic Publ., 1993, pp. 127–152.
5. A. Arnold, K. Strehmel, and R. Weiner, *Small perturbations in differential-algebraic systems of index 2*, Preprint 93/1, University of Rostock, Dept. of Mathematics, 1993.
6. M. Arnold, K. Strehmel, and R. Weiner, *Half-explicit Runge-Kutta methods for semi-explicit differential-algebraic equations of index 1*, Numer. Math. **64** (1993), 409–431.
7. U. Ascher and L.R. Petzold, *Stability of computational methods for constrained dynamics systems*, SIAM J. Sci. Stat. Comput. **14** (1993), 95–120.

8. G. Bader and P. Deuflhard, *A semi-implicit mid-point rule for stiff systems of ordinary differential equations*, Numer. Math. **41** (1983), 373–398.
9. G. Bader, U. Novak, and P. Deuflhard, *An advanced simulation package for large chemical reaction systems*, Univ. Heidelberg, SFB 123, Tech. Rep. 149 (1982).
10. R.E. Bank, R. Bulirsch, and K. Merten (eds.), *Mathematical modelling and simulation of electrical circuits and semiconductor devices*, Birkhäuser, 1990.
11. J. Baumgarte, *Stabilization of Constraints and integrals of motion in dynamical systems*, Computer Methods in Applied Mechanics and Engineering **1** (1972), 1–16.
12. T.A. Bickart, *An efficient solution process for implicit Runge-Kutta methods*, SIAM J. Numer. Anal. **14** (1977), 1022–1027.
13. V. Brasey and E. Hairer, *Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2*, SIAM J. Numer. Anal. **30** (1993), 538–552.
14. ———, *Symmetrized half-explicit methods for constrained mechanical systems*, Appl. Numer. Math. **13** (1993), 23–31.
15. K.E. Brenan, S.L. Campbell, and L.R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*, North-Holland Publ. Co., Amsterdam, 1989.
16. P.N. Brown, G.D. Byrne, and A.C. Hindmarsh, *VODE: a variable coefficient ODE solver*, SIAM J. Sci. Stat. Comput. **10** (1989), 1038–1051.
17. P.N. Brown and A.C. Hindmarsh, *Matrix-free methods for stiff systems of ODEs*, SIAM J. Numer. Anal. **23** (1986), 610–638.
18. J. Bruder, K. Strehmel, and R. Weiner, *Partitioned adaptive Runge-Kutta methods for the solution of nonstiff and stiff systems*, Numerische Mathematik **52** (1988), 621–638.
19. K. Burrage, *A special family of RK methods for solving stiff differential equations*, BIT **18** (1978), 22–41.
20. ———, *Parallel methods for initial value problems*, Applied Numerical Mathematics **11** (1993), 5–25.
21. K. Burrage, J.C. Butcher, and F.H. Chipman, *An implementation of singly-implicit Runge-Kutta methods*, BIT **20** (1980), 326–340.
22. J.C. Butcher, *Implicit Runge-Kutta processes*, Math. Comp. **18** (1964), 50–64.
23. ———, *On the implementation of implicit Runge-Kutta methods*, BIT **16** (1976), 237–240.
24. M. Büttner, B.A. Schmitt, and R. Weiner, *Automatic partitioning in linearly-implicit Runge-Kutta methods*, Applied Numerical Mathematics **13** (1993), 41–55.
25. ———, *W-methods with automatic partitioning by Krylov techniques for large stiff systems*, SIAM J. Numer. Anal. **32** (1995), 260–284.
26. G.D. Byrne, *Pragmatic experiments with Krylov methods in the stiff ODE setting*, Computational ordinary differential equations, Clarendon Press, Oxford, 1992, pp. 323–356.
27. G.D. Byrne and A.C. Hindmarsh, *A polyalgorithm for the numerical solution of ordinary differential equations*, ACM Trans. on Math. Software **1** (1975), no. 1, 71–96.
28. A.R. Curtis, *The FACSIMILE numerical integrator for stiff initial value problems*, Techn. Rep. AERE - R 9352, Harwell (1979).
29. C.F. Curtiss and J.O. Hirschfelder, *Integration of stiff equations*, Proc. Nat. Acad. Sci. **38** (1952), 235–243.
30. G. Dahlquist, *A special stability problem for linear multistep methods*, Bit **3** (1963), 27–43.
31. K. Dekker and J.G. Verwer, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North Holland, 1984.
32. P. Deuflhard, *Order and step-size control in extrapolation methods*, Numer. Math. **41** (1983), 399–422.
33. ———, *Recent progress in extrapolation methods for ordinary differential equations*, SIAM J. Numer. Anal. **27** (1985), 505–535.
34. P. Deuflhard, G. Bader, and U. Novak, *LARKIN - A software package for the numerical simulation of LARge systems arising in chemical reaction KINetics*, Modelling of Chemical Reaction Systems (K.H. Ebert, P. Deuflhard, and W. Jäger, eds.), Springer-Verlag, 1981, pp. 38–55.
35. P. Deuflhard, E. Hairer, and J. Zugck, *One step and extrapolation methods for differential-algebraic systems*, Numer. Math. **51** (1987), 501–516.
36. J.R. Dormand and P.J. Prince, *A family of embedded Runge-Kutta formulae*, J. Comp. Appl. Math. **6** (1980), 19–26.
37. D. Edelson, *A simulation language and compiler to aid computer solution of chemical kinetic problems*, Comput. Chem. **1** (1976), 29–33.
38. L. Edsberg, *Integration package for chemical kinetics*, Stiff Differential Systems (R.A. Willoughby, ed.),

Plenum Press, New York, 1974, pp. 81–94.
39. B.L. Ehle, *High order A-stable methods for the numerical solution of systems of DEs*, BIT **8** (1968), 276–278.
40. ———, *On Padé approximations to the exponential function and A-stable methods for the numerical solution initial value problems*, Report CSRR 2010, Dept. AACS, Univ. of Waterloo (1969).
41. E. Eich, *Private Mitteilung – Linde AG*, 1993.
42. G. Engl and P. Rentrop, *Gas flow in a single cylinder internal combustion engine*, Int. J. Num. Meth. Heat Fluid Flow **2** (1992), 63–78.
43. W.H. Enright and M. Kamel, *Automatic partitioning of stiff systems and exploiting the resulting structure*, ACM-TOMS **5** (1979), 374–385.
44. R. Frank, J. Schneid, and C.W. Ueberhuber, *Order results for implicit Runge-Kutta methods applied to stiff systems*, SIAM J. Numer. Anal. **22** (1985), 515–534.
45. ———, *Stability properties of implicit Runge-Kutta methods*, SIAM J. Numer. Anal. **22** (1985), 497–514.
46. C. Führer, *Differential-Algebraische Gleichungssysteme in Mechanischen Mehrkörpersystemen*, Ph.D. thesis, TU München, 1988.
47. C.W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice Hall, New York, 1971.
48. ———, *Simultaneous numerical solution of differential-algebraic equations*, IEEE Trans. Circuit Theory **CT-18** (1971), 89–95.
49. ———, *Differential-algebraic index transformations*, SIAM J. Sci. Stat. Comp. **9** (1988), 39–47.
50. C.W. Gear, B. Leimkuhler, and G.K. Gupta, *Automatic integration of Euler-Lagrange equations with constraints*, J. Comp. Appl. Math. **12 & 13** (1985), 77–90.
51. C.W. Gear and Y. Saad, *Iterative solution of linear equations in ODE codes*, SIAM J. Sci. Stat. Comp. **4** (1983), 583–601.
52. C.W. Gear and R.R. Wells, *Multirate linear multistep methods*, BIT **24** (1984), 484–502.
53. E. Griepentrog, *Gemischte Runge-Kutta-Verfahren für steife Systeme*, Seminarbericht Nr. 11, Humboldt-Universität Berlin, Sektion Mathematik (1978).
54. E. Griepentrog and R. März, *Differential-algebraic equations and their numerical treatment*, Teubner-Texte zur Mathematik, Band 88, Leipzig, 1986.
55. R.D. Grigorieff, *Numerik gewöhnlicher Differentialgleichungen 1*, Teubner, Stuttgart, 1972.
56. M. Günther and P. Rentrop, *Partitioning and Multirate Strategies in Latent Electric Circuits*, Preprint TU München, TUM-M9301 (1993).
57. E. Hairer and L. Jay, *Implicit Runge-Kutta methods for higher index differential-algebraic systems*, WSSIAA, Contributions in numerical mathematics, vol. 2, 1993, pp. 213–224.
58. E. Hairer, Ch. Lubich, and M. Roche, *The numerical solution of differential-algebraic systems by Runge-Kutta methods*, Lecture Notes in Mathematics **1409** (1989), Springer-Verlag.
59. E. Hairer, S.P. Nørsett, and G. Wanner, *Solving ordinary differential equations I*, Springer-Verlag, 1987.
60. ———, *Solving Ordinary Differential Equations I*, 2. ed., Springer-Verlag, 1993.
61. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer-Verlag, 1991.
62. E. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*, Allyn and Bacon, 1989.
63. A.C. Hindmarsh, *LSODE and LSODI, two new initial value ordinary differential equation solvers*, ACM-SIGNUM Newsletter **15** (1980), 10–11.
64. ———, *ODEPACK, a systemized collection of ODE solvers*, Lawrence Livermore National Laboratory, Rept. UCRL–88007 (1982).
65. E. Hofer, *A partially implicit method for large stiff systems of ODEs with only few equations introducing small time constants*, SIAM J. Numer. Anal. **13** (1976), 645–663.
66. P.J. van der Houwen and B.P. Sommeijer, *Analysis of parallel diagonally implicit iteration of Runge-Kutta methods*, Applied Numerical Mathematics **11** (1993), 169–188.
67. W. Kampowsky, P. Rentrop, and W. Schmidt, *Classification and numerical simulation of electric circuits*, Surveys on Mathematics for Industry **2** (1992), 23–65.
68. P. Kaps and P. Rentrop, *Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations*, Numer. Math. **38** (1979), 55–68.
69. P. Kaps and G. Wanner, *A study of Rosenbrock-type methods of high order*, Numer. Math. **38** (1981), 279–298.

70. R.J. Kee, J.A. Miller, and T.H. Jefferson, *CHEMKIN: A general-purpose, problem-independent, transportable, fortran chemical kinetics code package*, Sandia National Laboratories, Livermore, Tech. Rep. SAND 80-8003 (1980).
71. S.L. Keeling, *On implicit Runge-Kutta methods with stability function having distinct real poles*, BIT **29** (1989), 91–109.
72. J.D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley & Sons, 1991.
73. Ch. Lubich, *Extrapolation integrators for constraint multibody systems*, IMPACT Comp. Sci. Eng. **3** (1991), 213–234.
74. Ch. Lubich, U. Novak, U. Pöhle, and Ch. Engstler, *MEXX — Numerical Software for the Integration of Constrained Mechanical Multibody Systems*, Preprint SC 92-12, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1992.
75. S.P. Nørsett, *Runge-Kutta methods with a multiple real eigenvalue only*, BIT **16** (1976), 388–393.
76. A. Ostermann, *Continuous extensions of Rosenbrock-type methods*, Computing **44** (1990), 59–68.
77. ———, *A class of half-explicit Runge-Kutta methods for differential-algebraic systems of index 3*, Appl. Numer. Math. **13** (1993), 165–179.
78. L. Petzold, *Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations*, SIAM J. Sci. Stat. Comput. **4** (1983), no. 1, 136–148.
79. L.R. Petzold, *Differential-algebraic equations are not ODE's*, SIAM J. Sci. Stat. Comput. **3** (1982), 367–384.
80. P. Rentrop, *Partitioned Runge-Kutta methods with stiffness detection and stepsize control*, Numer. Math. **47** (1985), 545–564.
81. P. Rentrop, M. Roche, and G. Steinebach, *The application of Rosenbrock-Wanner type methods with stepsize control in differential-algebraic equations*, Numer. Math. **55** (1989), 545–563.
82. W.C. Rheinboldt and B. Simeon, *Performance analysis of some methods for solving Euler-Lagrange equations*, Appl. Math. Lett. **8** (1995), 77–82.
83. H. Ritter, T. Martinetz, and K. Schulten, *Neuronale Netze*, Addison Wesley, 1991.
84. R.E. Roberson and R. Schwertassek, *Dynamics of Multibody Systems*, Springer Berlin, 1988.
85. M. Roche, *Rosenbrock methods for differential-algebraic systems*, Numer. Math. **52** (1988), 45–63.
86. H.H. Rosenbrock, *Some general implicit processes for the numerical solution of differential equations*, Comp. J. **5** (1963), 329–331.
87. Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. **29** (1992), no. 1, 209–228.
88. W. Schiehlen (ed.), *Multibody system handbook*, Springer-Verlag, 1990.
89. W. Schiehlen (ed.), *Advanced multibody system dynamics*, Kluwer Academic Publ., 1993.
90. L.F. Shampine and C.W. Gear, *A user's view of solving stiff ordinary differential equations*, SIAM Review **21** (1979), no. 1, 1–17.
91. L.F. Shampine and M.K. Gordon, *Computer Solution of Ordinary Differential Equations, The Initial Value Problem*, Freeman and Company, San Francisco, 1975.
92. B. Simeon, *Numerische Integration mechanischer Mehrkörpersysteme: Projizierende Deskriptorformen, Algorithmen und Rechenprogramme*, Dissertation, Fortschritt-Berichte VDI, Reihe 20, Nr. 130, Düsseldorf, VDI-Verlag (1994).
93. B. Simeon, C. Führer, and P. Rentrop, *Differential-algebraic equations in vehicle system dynamics*, Surv. Math. Ind. **1** (1991), 1–37.
94. S. Skelboe, *Stability properties of backward differentiation multirate formulas*, Appl. Numer. Math. **5** (1989), 151–160.
95. S. Skelboe and P.U. Anderson, *Stability properties of backward Euler multirate formulas*, SIAM J. Sci. Stat. Comp. **10** (1989), 1000–1009.
96. R.N. Stabler and J.P. Chesick, *A program system for computer integration of multistep reaction rate equations using the Gear integration method*, Int. J. Chem. Kin. **10** (1978), 461–469.
97. T. Steihaug and A. Wolfbrandt, *An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff ordinary differential equations*, Math. Comp. **33** (1979), 521–534.
98. K. Strehmel and R. Weiner, *Behandlung steifer Anfangswertprobleme gewöhnlicher Differentialgleichungen mit adaptiven Runge-Kutta-Methoden*, Computing **29** (1982), 153–165.
99. ———, *Partitioned adaptive Runge-Kutta methods and their stability*, Numer. Math. **45** (1984), 283–300.
100. ———, *Linear-implizite Runge-Kutta Methoden und ihre Anwendung*, Teubner-Verlag Stuttgart-Leipzig,

1992.
101. ———, *Numerik gewöhnlicher Differentialgleichungen*, Teubner Studienbücherei Mathematik, Teubner Stuttgart, 1994.
102. R. Weiner, M. Arnold, P. Rentrop, and K. Strehmel, *Partitioning strategies in Runge-Kutta type methods*, IMA Journal of Numer. Anal. **13** (1993), 303–319.

P. Rentrop
TH Darmstadt
FB Mathematik
Schloßgartenstr. 7
64289 Darmstadt

K. Strehmel und R. Weiner
FB Mathematik und Informatik
Martin-Luther-Universität
Halle-Wittenberg
Postfach
06099 Halle