TA eu.

FORSCHUNG - AUSBILDUNG - WEITERBILDUNG

BERICHT NR. 29

SEMI-IMPLIZITE EINSCHRITTVERFAHREN ZUR NUMERISCHEN LÖSUNG DIFFERENTIAL-ALGEBRAISCHER GLEICHUNGEN TECHNISCHER MODELLE

GERD STEINEBACH

UNIVERSITÄT KAISERSLAUTERN
FACHBEREICH MATHEMATIK
ERWIN-SCHRÖDINGER-STRASSE
D - 6750 KAISERSLAUTERN

JANUAR 1988

Vorwort

Die vorliegende Arbeit beschäftigt sich mit der numerischen Behandlung Differential-Algebraischer Gleichungen (DAE's).
DAE's treten beispielsweise bei der Modellierung der Dynamik mechanischer Systeme, der Schaltkreissimulation sowie der chemischen Reaktionskinetik auf.
Es werden Rosenbrock-Wanner ähnliche Verfahren zu deren Lösung hergeleitet und an technischen Modellen (Fahrzeugachse und Verstärker) getestet.

Diese Arbeit entstand auf Anregung und unter der Leitung von Herrn Prof. Dr. P. Rentrop, dem an dieser Stelle mein besonderer Dank gilt.
Weiterhin möchte ich Frau Dr. L. Petzold, Herrn Dipl. Math. C. Führer sowie den Herren Prof. Dr. H.J. Oberle und Prof. Dr. K. Glashoff danken, die zum Zustandekommen der Arbeit beigetragen haben.
In ihrer ursprünglichen Form liegt die Arbeit als Diplomarbeit an der Universität Kaiserslautern vor.

INHALTSVERZEICHNIS

1. Einleitung

In der vorliegenden Arbeit beschäftigen wir uns mit der numerischen Behandlung Differential-Algebraischer Gleichungen, genannt DAE's.
Unter DAE's versteht man implizite Differentialgleichungssysteme der Form $F(t,y,y') = 0$ mit $F : R \times R^n \times R^n \to R^n$, $y : R \to R^n$ und passenden Anfangswerten.
Sie umfassen als Spezialfall die gewöhnlichen Differentialgleichungssysteme $y' = F(y)$.

DAE's treten bei der Modellierung der Dynamik mechanischer Systeme, der Schaltkreissimulation sowie der chemischen Reaktionskinetik auf.

Die Arbeit gliedert sich in drei Teile.
Der erste Teil (§1,§2) gibt eine Übersicht über die Lösungstheorie und die Indexklassifikation von DAE's.
Der zweite Teil (§3,..,§6) diskutiert Ansätze zur numerischen Lösung.
Der dritte Teil (§7,§8,§9) umfaßt Fragen der Implementierung und eine Diskussion der Testbeispiele.
Hervorheben möchten wir die Untersuchung der Konvergenz von allgemeinen Einschrittverfahren in §4 und die Übertragung der bekannten Butcherreihentheorie auf Index-Eins DAE's in §5. Auf dieser Grundlage können wir in §6 Ordnungs- und Konvergenzbedingungen für zwei verschiedene Lösungsansätze herleiten. Es handelt sich um einen in /17/ vorgeschlagenen

Rosenbrock-Wanner Ansatz sowie um die Kombination eines Rosenbrock-Wanner Ansatzes mit einem expliziten Runge-Kutta Ansatz.

Zu beiden Ansätzen entwickeln und implementieren wir verschiedene Verfahren und testen schließlich ihr Verhalten an einigen Beispielen.

In §8 stellen wir drei Testbeispiele vor, um die vorgeschlagenen Verfahren bzgl. Zuverlässigkeit, Genauigkeit und Schnelligkeit zu untersuchen.

Alle numerischen Resultate vergleichen wir mit denen des Programms DASSL /12/. DASSL ist ein auf einem BDF-Code basierendes Mehrschrittverfahren zur Lösung von DAE's.

Als wichtiges Anwendungsbeispiel berechnen wir das dynamische Verhalten einer Fahrzeughinterachse bei schneller Kurvenfahrt. Die Modellierung dieses Problems führt auf ein Index-Drei DAE-System und wurde in Zusammenarbeit mit Herrn Dipl. Math. C. Führer von der DFVLR Oberpfaffenhofen behandelt.

Als ein interessantes Beispiel aus der Schaltkreissimulation untersuchen wir in §9 das Eingangs-Ausgangsverhalten eines Verstärkers. Der Verstärker wird durch ein DAE-System in Nicht-Normalform $A\,y' = f(y)$ beschrieben, wo A eine singuläre Matrix ist. Die Elemente von A sind die Kapazitäten des Schaltkreises.

2. Existenz und Eindeutigkeit der Lösungen von Differential-Algebraischen Gleichungen

In diesem Abschnitt fassen wir die für uns wichtigen Eigenschaften von DAE's zusammen.

Um Probleme aufzeigen zu können, die bei der numerischen Behandlung von DAE's auftreten, beschränken wir uns zunächst auf lineare Systeme der Form

(2.1) $A\,y' = B\,y + g(t)\,,\ y(t_o) = y_o.$

Ist die Matrix A regulär, so ist (2.1) ein gewöhnliches Anfangswertproblem (AWP). Jedoch ist es für die numerische Behandlung nicht sinnvoll, A zu invertieren, weil dadurch die Struktur der Gleichung (z.B. seien A und B dünn besetzte Matrizen) zerstört werden kann.

Es seien also A und B, möglicherweise singuläre, (n,n)-Matrizen.

In /5/ wird die Matrizenschar $\{A-\lambda B\}$, $\lambda \in R$ untersucht.

Die wichtigsten Aussagen aus /5/ und /20/ sind:

Es gibt reguläre Matrizen P und Q, die $A-\lambda B$ auf eine kanonische Form reduzieren. Werden P und Q auf (2.1) angewendet, so erhält man die *kanonische Kroneckerzerlegung:*

(2.2) $P\,A\,Q\,Q^{-1}y' = P\,B\,Q\,Q^{-1}y + P\,g(t)$

$$Q^{-1}y(t_o) = Q^{-1}y_o$$

Ist $\det(A-\lambda B) = 0$ für alle $\lambda \in R$, so hat (2.2) entweder keine Lösung oder unendlich viele Lösungen. In diesem Fall ist (2.1) numerisch nicht ohne weitere Informationen lösbar.

Ist $\det(A-\lambda B)$ nicht identisch Null, so zerfällt (2.2) in

zwei entkoppelte Systeme der Form

$$\text{(2.3a)} \quad \tilde{y}_1' = E_1\tilde{y}_1 + \tilde{g}_1(t) \quad , \quad \tilde{y}_1(t_o) = \tilde{y}_{1o}$$

$$\text{(2.3b)} \quad E_2\tilde{y}_2' = \tilde{y}_2 + \tilde{g}_2(t) \quad , \quad \tilde{y}_2(t_o) = \tilde{y}_{2o}$$

mit

$$P\,A\,Q = \begin{pmatrix} I_1 & 0 \\ 0 & E_2 \end{pmatrix} , \quad P\,B\,Q = \begin{pmatrix} E_1 & 0 \\ 0 & I_2 \end{pmatrix}$$

$$\begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{pmatrix} = Q^{-1}y \ , \quad \begin{pmatrix} \tilde{g}_1 \\ \tilde{g}_2 \end{pmatrix} = P\,g \ , \quad \begin{pmatrix} \tilde{y}_{1o} \\ \tilde{y}_{2o} \end{pmatrix} = Q^{-1}y_o \ .$$

I_1, I_2 sind Einheitsmatrizen. $E_2 = \mathrm{diag}(E_{21},..,E_{21})$ ist eine Blockdiagonalmatrix mit

$$E_{2i} = \begin{pmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & 0 & . & 0 \\ . & . & . & . & . & . \\ 0 & . & . & . & 0 & 1 \\ 0 & . & . & . & . & 0 \end{pmatrix}_{m_i \times m_i} .$$

Definition 2.4:

$m := \max_{i=1}^{l} (\dim E_{2i})$ heißt *Index* des Problems (2.1).

Bemerkung 2.5:

Es gilt: $E_2^m = 0$ und $E_2^{m-1} \neq 0$, d.h. E_2 ist eine nilpotente Matrix mit der Nilpotenz m.

Obwohl E_2 nicht eindeutig definiert ist, ist der Index m wohldefiniert.

Wie wir leicht nachrechnen, ist die Lösung von (2.3b) durch

$$\text{(2.6)} \quad \tilde{y}_2(t) = -\sum_{i=0}^{m-1} E_2^i\, \tilde{g}_2^{(i)}(t) \quad \text{gegeben.}$$

Zu beachten ist, daß $\tilde{y}_2(t)$ unabhängig von den Anfangswerten ist und Unstetigkeiten hat, falls $\tilde{g}_2 \notin C^{m-1}$.

Die Anfangswerte in (2.1) bzw. (2.3) sollten *konsistent* sein, d.h. $\tilde{y}_{2o} = \tilde{y}_2(t_o)$.

Fassen wir zusammen:

Ist der Index m = 0, so reduziert sich (2.2) auf ein gewöhnliches AWP.

Ist m= 1, so gilt $E_2 = 0$ und $\tilde{y}_2(t) = -\tilde{g}_2(t)$.

Für $m \geq 2$ treten Unstetigkeiten in der Lösung auf, falls $g \notin C^{m-1}$.

Versuchen wir (2.1) mit numerischen Methoden zur Integration gewöhnlicher Differentialgleichungen zu lösen, so können in vielfacher Weise Probleme auftreten, falls das System (2.1) Index m > 1 hat. Einige dieser Schwierigkeiten sind z.B. in /13/ beschrieben. Die Folge ist meist ein Versagen der automatischen Schrittweitensteuerung.

Das Konzept der Index-Klassifizierung kann auch auf nichtlineare Systeme übertragen werden (siehe /6/).

Betrachten wir etwa Systeme der Form

(2.7a) $y' = f(t,y,z)$, $y(t_o) = y_o$, $y : R \to R^n$, $f : R \times R^n \times R^m \to R^n$

(2.7b) $0 = g(t,y,z)$, $z(t_o) = z_o$, $z : R \to R^m$, $g : R \times R^n \times R^m \to R^m$

Der Index des Systems kann nun auf folgende Weise interpretiert werden:

Sei k die Anzahl der Differentiationen der algebraischen Nebenbedingung (2.7b), die wir durchführen müssen, um ein gewöhnliches Differentialgleichungssystem zu erhalten.

Dann ist der Index des Systems (2.7) gleich k (siehe /11/).

Sei vorausgesetzt, daß in (2.7) $(\partial g/\partial z)^{-1}$ existiert und in einer Umgebung der Lösung beschränkt ist. Dann können wir (2.7b) mit Hilfe des Satzes über implizite Funktionen lokal nach z auflösen. Differentiation nach t liefert zusammen mit (2.7a) ein System gewöhnlicher Differentialgleichungen.

Unter den oben genannten Voraussetzungen ist (2.7) somit ein Index-Eins System.

Ein Anwendungsbeispiel eines Index-Drei Systems ist die Simulation mechanischer Systeme, bestehend aus starren Körpern, die über Gelenke, Federn und Dämpfer miteinander verbunden sind /11/.

Der Vektor q der Koordinaten der Körper erfüllt die Gleichungen

(2.8a) $M(q)\, q'' = f(q,q',t) + G(q)\cdot\lambda, \quad q(0) = q_o$

(2.8b) $0 = \Phi(q)$

M ist die reguläre Massenmatrix und λ der Vektor der Lagrange'schen Multiplikatoren. Für Φ gilt: $(\partial\Phi/\partial q) = G^t$.

(2.8b) beschreibt die geometrischen Zwangsbedingungen des Systems. Die Anfangsbedingung sei konsistent, d.h. es gelte $\Phi(q_o) = 0$.

Dann ist (2.8b) äquivalent mit

(2.9) $0 = G^t(q)\, q'$.

Für $q'_o := q'(0)$ gilt die Beziehung $G^t(q_o)\, q'_o = 0$, und daher erhalten wir wiederum durch Differenzieren nach t

(2.10) $0 = G^t\, q'' + (G^t)'q'$.

Einsetzen von q'' aus (2.8a) in (2.10) und auflösen nach λ liefert

(2.11) $\lambda = -(G^tM^{-1}G)^{-1}(G^tM^{-1}f + (G^t)'q')$.

Setzen wir voraus, daß λ_o, q_o, q'_o Bedingung (2.11) erfüllen, so können wir ein weiteres mal differenzieren.

Setzen wir dann wiederum q'' aus (2.8a) ein, so erhalten wir eine neue zu (2.8b) äquivalente Differentialgleichung der

Form $\lambda' = \tilde{f}(t,q,q',\lambda)$.
Diese Gleichung, zusammen mit (2.8a), ist ein gewöhnliches Differentialgleichungssystem. Der Index des Originalsystems (2.8) ist somit drei, da wir dreimal die algebraische Nebenbedingung (2.8b) differenziert hatten, um das gewünschte Differentialgleichungssystem zu erhalten.

Rheinboldt /16/ zeigte, daß Systeme, deren Index größer als eins ist, algebraisch unvollständig sind, d.h. Existenz und Eindeutigkeit von Lösungen sind nicht garantiert. Daher werden wir in den nächsten Kapiteln nur Index-Eins Probleme der Form (2.7) behandeln.

3. Übersicht über numerische Lösungsmethoden bei Differential-Algebraischen Gleichungen

3.1. Einschrittverfahren

Zur numerischen Lösung von Index-Eins Systemen der Form (2.7) können folgende Ansätze verfolgt werden:

(i) Der Satz über implizite Funktionen erlaubt die explizite Darstellung von (2.7b) in der Form

(3.1) $z = \tilde{g}(t,y)$,

die eingesetzt in (2.7a) das gewöhnliche Differentialgleichungssystem

(3.2) $y' = f(t,y,\tilde{g}(t,y))$ liefert.

Dieses kann mit den bekannten Methoden bearbeitet werden. Der Nachteil besteht darin, daß jede Auswertung von f die Lösung der nichtlinearen Gleichungen (3.1) erfordert.

(ii) Ohne Beschränkung der Allgemeinheit sei das System (2.7) autonom. Semi-implizite Euler-Diskretisierung führt zu

$$\frac{y_{n+1}-y_n}{h} = f(y_n,z_n) + (f_y(y_n,z_n), f_z(y_n,z_n)) \begin{pmatrix} y_{n+1}-y_n \\ z_{n+1}-z_n \end{pmatrix}$$

$$0 = g(y_n,z_n) + (g_y(y_n,z_n), g_z(y_n,z_n)) \begin{pmatrix} y_{n+1}-y_n \\ z_{n+1}-z_n \end{pmatrix}$$

$\leftrightarrow$

$$(3.3)\quad \left(\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} - h \begin{pmatrix} f_y & f_z \\ g_y & g_z \end{pmatrix} \right) \begin{pmatrix} y_{n+1}-y_n \\ z_{n+1}-z_n \end{pmatrix} = h \begin{pmatrix} f(y_n,z_n) \\ g(y_n,z_n) \end{pmatrix}$$

Hierauf aufbauend wurde in /3/ eine Extrapolationsmethode entwickelt.

(iii) Gleichung (3.3) ist ein Spezialfall der Rosenbrock-Wanner Methoden (ROW). Bei Anwendung der bisher bekannten ROW-Verfahren erfahren diese jedoch eine Ordnungsreduktion, wenn die Koeffizienten der Verfahren nicht zusätzlichen Bedingungen genügen. Diese werden in §5 und §6 hergeleitet.
Die ROW-Verfahren vierter Ordnung von Kaps und Rentrop /9/ sind bei Anwendung auf DAE's beispielsweise nur von Ordnung zwei.
Derselbe Effekt tritt auch bei Anwendung impliziter Runge-Kutta Methoden auf DAE's auf (siehe /14/).

3.2. Das Mehrschrittverfahren DASSL

Weil wir alle numerischen Resultate mit denen des bekannten Verfahrens DASSL vergleichen, beschreiben wir dieses hier kurz.

DASSL bearbeitet Probleme der allgemeinen Form

$$(3.4)\quad F(t,y,y') = 0\ ,\quad y(t_0) = y_0\ ,\ y'(t_0) = y'_0\ .$$

Die Idee ist, y' durch eine Differenzenapproximation zu ersetzen. Als Beispiel nehmen wir die Rückwärtsdifferenz erster Ordnung und erhalten

$$(3.5)\quad F(t_n, y_n, \frac{y_n - y_{n-1}}{t_n - t_{n-1}}) = 0\ .$$

Gleichung (3.5) wird dann durch ein Newton-Verfahren gelöst:

$$(3.6)\quad y_n^{m+1} = y_n^m - ((\partial F/\partial y) + \Delta t_n^{-1}(\partial F/\partial y'))^{-1}\, F(t_n, y_n^m, (y_n^m - y_{n-1})/\Delta t_n)$$

DASSL approximiert y' durch Rückwärtsdifferenzen k-ter Ordnung, wobei k von eins bis fünf reicht.
Nach jedem Schritt wird eine Ordnung k und eine neue Schrittweite Δt_n in Abhängigkeit des Lösungsverhaltens gewählt.
Das Newtonverfahren (3.6) konvergiert schnell, falls der Startwert y_n^o gut gewählt wurde. DASSL wählt y_n^o durch Auswerten des Polynoms, das die Lösung in den letzten (k+1) Zeitpunkten $t_{n-1},\ldots,t_{n-(k+1)}$ interpoliert, im Zeitpunkt t_n.
Es ist wichtig, das nichtlineare System (3.5) effizient zu lösen. Aus diesem Grund arbeitet DASSL mit einem modifizierten Newton-Verfahren. Die Iterationsmatrix dieses Verfahrens wird für möglichst viele Integrationsschritte beibehalten. Daher braucht die Jacobi-Matrix von F nicht in jedem Schritt neu berechnet zu werden. Weil die Ableitung i.allg. durch Differenzenquotienten approximiert wird, spart man somit Funktionsauswertungen von F.
Einzelheiten hierzu und zu den Strategien, nach denen die Schrittweite und die Ordnung k gewählt wird, sind in /12/ zu finden.

Vorteile von Mehrschrittverfahren gegenüber Einschrittverfahren liegen darin, daß sie i.allg. weniger Funktionsauswertungen benötigen. Daher sind sie geeigneter für die Integration großer Systeme und für große Integrationsintervalle bei hohen Genauigkeitsanforderungen.

Ein Nachteil von Mehrschrittverfahren besteht darin, daß sie einen verhältnismäßig großen Verwaltungsaufwand benötigen. Daher sind i.allg. Einschrittverfahren für kleinere Systeme effizienter.
Die hier gemachten Aussagen werden durch die in §8 und §9 gewonnenen Resultate bestätigt.

4. Ordnung und Konvergenz von Einschrittverfahren

In diesem Paragraphen untersuchen wir die Konvergenz von Einschrittverfahren für $h \to 0$ analog zur Arbeit /3/.
Dazu betrachten wir zur Lösung des Index-Eins Problems

$$(4.1)\quad y' = f(y,z)\ ,\ y(x_o) = y_o\ ,\quad f : R^N \times R^M \to R^N$$
$$0 = g(y,z)\ ,\ z(x_o) = z_o\ ,\quad g : R^N \times R^M \to R^M$$

die folgende (nur formal explizite) Klasse von Einschrittmethoden der Form

$$(4.2a)\quad y_1 = y_o + h\,\Phi(y_o,z_o,h)$$
$$(4.2b)\quad z_1 = \Psi(y_o,z_o,h)\ .$$

y_o, z_o seien konsistente Anfangswerte und h die Verfahrensschrittweite. Φ und Ψ bezeichnen die Verfahrensfunktionen (vergl. /22/ und /3/).

Definition 4.3:

Seien y,z die exakten Lösungen von (4.1).

a) Die Methode (4.2) ist von *Ordnung* p (man spricht auch von *lokaler Konsistenzordnung* p), falls gilt

$$y(x_o+h) - y_1 = O(h^{p+1})$$
$$z(x_o+h) - z_1 = O(h^p)$$

b) Die Methode (4.2) hat global die *Konvergenzordnung* p, falls für festes $x = n\cdot h$ gilt:

$$y_n - y(x) = O(h^p)$$
$$z_n - z(x) = O(h^p)\ .$$

y_n, z_n bezeichnet die durch n-maliges Anwenden von (4.2) gewonnene numerische Lösung von (4.1).

Satz 4.4:

Die Methode (4.2) sei von Ordnung p und erfülle zusätzlich

$$|| \partial \Psi / \partial z(y,z,0) || \leq \alpha < 1$$

in einer Umgebung der Lösung.

Dann hat (4.2) auch Konvergenzordnung p.

Der Beweis des Satzes ist in /3/ und /21/ zu finden.

5. Anwendung der Butcherreihentheorie auf Index-Eins

Probleme

5.1. Taylorreihe als Baummodelle

Beschäftigen wir uns im folgenden wieder mit Index-Eins Problemen der Form (4.1), wobei $(\partial g/\partial z)^{-1}$ existiert und in der Umgebung der exakten Lösung beschränkt ist. Die Anfangswerte seien konsistent ,d.h. $g(y_0,z_0) = 0$.

Um die Taylorreihen der exakten Lösung y und z darzustellen, benötigen wir y', z', y'', z'',...

Jedes $y^{(p)}$ bzw. $z^{(p)}$ besteht aus einzelnen Summanden, die wir *elementare Differentiale* nennen.

Die Idee ist nun, jedes elementare Differential mit einem Baum zu identifizieren. Wir folgen einem Vorschlag von Roche /17/ und der dort zitierten Literatur. Dazu definieren wir rekursiv:

Definition 5.1:

Es seien τ_y und τ_z die folgenden Bäume:

$\tau_y :=$ • , $\tau_z :=$  .

DAT , DAT_y und DAT_z bezeichnen Mengen von Bäumen, definiert durch:

a) $\tau_y \in DAT_y$; $\tau_z \in DAT_z$

b) Seien $t_1,..,t_n \in DAT_y \cup DAT_z$, dann ist $[t_1,..,t_n]_y \in DAT_y$

c) Seien $t_1,..,t_m \in DAT_y \cup DAT_z$ mit $m > 1$ oder $m = 1$ und $t_1 \in DAT_y$, dann ist $[t_1,..,t_m]_z \in DAT_z$.

d) $DAT := DAT_y \cup DAT_z$

Hierbei erhält man $t = [t_1,..,t_n]_y$, indem man die Wurzeln

der Bäume $t_1,..,t_n$ durch n Kanten zu einer neuen dünnen Wurzel verbindet, z.B.:

t_1 =  , t_2 =  $\rightarrow$ $t = [t_1,t_2]_y$ =  .

Entsprechend entsteht $t = [t_1,..,t_m]_z$, jedoch mit neuer dicker Wurzel. (hier : $t_1 = [t_1,t_2]_z$ = )

Bemerkung 5.2:

a) $[t_1,t_2]_y = [t_2,t_1]_y$ und $[t_1,t_2]_z = [t_2,t_1]_z$ für alle $t_1,t_2 \in DAT$.

b) Ist die Wurzel eines Baumes $t \in DAT$ dünn, so ist $t \in DAT_y$, andernfalls gilt $t \in DAT_z$.

c) Hat ein dicker Knoten keine Verzweigung, dann ist der oberhalb liegende Knoten ein dünner Knoten, d.h. t =  ist z.B. nicht in DAT.

Definition 5.3:

Die Anzahl dünner Knoten eines Baumes $t \in DAT$ bezeichnet die *Ordnung* $\rho(t)$ von t.

$t \in DAT$ heißt *monoton indiziert*, falls jedem dünnen Knoten von t bijektiv eine Zahl $i \in \{1,2,..,\rho(t)\}$ zugeordnet ist und falls diese Zahlen entlang jedes Astes von t streng monoton wachsen.

Die Anzahl aller möglichen Indizierungen eines Baumes $t \in DAT$ bezeichnen wir mit $\alpha(t)$.

$LDAT_y$ (bzw. $LDAT_z$) bezeichnet die Menge aller Bäume $t \in DAT_y$ (bzw. DAT_z) mit monotoner Indizierung.

$LDAT := LDAT_y \cup LDAT_z$.

Beispiel 5.4:

$t =$  $\in DAT_z$ mit $\rho(t) = 3$ und $\alpha(t) = 3$.
Sei $t_1 =$  , $t_2 =$  , $t_3 =$  , $t_4 =$  , dann gilt: $t_1, t_2, t_3 \in LDAT_z$, aber $t_4 \notin LDAT_z$.

Wir definieren nun rekursiv Funktionen $F(t) : R^N \times R^M \to R^N$ und $G(u) : R^N \times R^M \to R^M$, die jedem Baum $t \in DAT_y$ bzw. $u \in DAT_z$ wie folgt genau ein elementares Differential zuordnen:

Definition 5.5:

a) $F(\tau_y)\ (y,z) := f$; $G(\tau_z)\ (y,z) := -(D_z g)^{-1}(D_y g)\ f$

b) $F(t)\ (y,z) := D_y^k\ D_z^l\ f\ (F(t_1),..,F(t_k),G(u_1),..,G(u_l))$,
falls $t = [t_1,..,t_k,u_1,..,u_l]_y$

c) $G(u)\ (y,z) := -(D_z g)^{-1} D_y^k\ D_z^l\ g\ (F(t_1),...,F(t_k),G(u_1),...,G(u_l))$,
falls $u = [t_1,..,t_k,u_1,..,u_l]_z$,

wobei $t_1,..,t_k \in DAT_y$ und $u_1,..,u_l \in DAT_z$.

Da wir ausreichende Glattheit von f und g fordern, sind deren partielle Ableitungen unabhängig von Vertauschungen. Definition 5.5 ist ebenfalls von Permutationen der $t_1,..,t_k,u_1,..,u_l$ unabhängig; daher sind F und G wohldefiniert.

Beispiel 5.6:

Durch Definition 5.5 werden den elementaren Differentialen auf folgende Weise Bäume zugeordnet:

Zunächst gilt $0 = g(y,z)$

oder $0 = (D_y g)\, y' + (D_z g)\, z'$

bzw. $z' = (-D_z g)^{-1}(D_y g)\, y'$

Wegen $0 = (Id)' = ((-D_z g)(-D_z g)^{-1})'$

$= (-D_z g)'(-D_z g)^{-1} + (-D_z g)\,((-D_z g)^{-1})'$

folgt $((-D_z g)^{-1})' = (-D_z g)^{-1}(D_y D_z g\, y' + D_z^2 g\, z')(-D_z g)^{-1}$.

Wir erhalten:

$y' = f$

•

$z' = (-D_z g)^{-1}(D_y g)\, f$

$y'' = D_y f \cdot f + D_z f\,(-D_z g)^{-1}(D_y g)\, f$

$z'' = (-D_z g)^{-1}\Big[D_y D_z g\,(f,(-D_z g)^{-1}(D_y g) f)$

$+ D_z^2 g\,[(-D_z g)^{-1} D_y g \cdot f, (-D_z g)^{-1}(D_y g)\, f)$

$+ D_y^2 g\,(f,f) + D_z D_y g\,(f,(-D_z g)^{-1}(D_y g) f)$

$+ D_y g\, D_y f \cdot f + D_y g\, D_z f\,(-D_z g)^{-1} D_y g\, f\Big]$.

$y''' = \ldots$

$z''' = \ldots$

Satz 5.7:

Für die exakte Lösung von (4.1) gilt:

$$y^{(p)}(x_0) = \sum_{\substack{t\in LDAT_y \\ \rho(t)=p}} F(t)\,(y_0,z_0) = \sum_{\substack{t\in DAT_y \\ \rho(t)=p}} \alpha(t)\,F(t)\,(y_0,z_0)$$

$$z^{(p)}(x_0) = \sum_{\substack{u\in LDAT_z \\ \rho(u)=p}} G(u)\,(y_0,z_0) = \sum_{\substack{u\in DAT_z \\ \rho(u)=p}} \alpha(u)\,G(u)\,(y_0,z_0)$$

und damit

$$y(x_0+h) = y(x_0) + \sum_{t\in LDAT_y} F(t)\,(y_0,z_0)\,\frac{h^{\rho(t)}}{\rho(t)!}$$

$$z(x_0+h) = z(x_0) + \sum_{u\in LDAT_z} G(u)\,(y_0,z_0)\,\frac{h^{\rho(u)}}{\rho(u)!} \quad .$$

Der Beweis des Satzes ist in /17/ und /21/ zu finden.

Wir führen nun analog zu /17/ DA-Reihen ein. Sie sind einen Verallgemeinerung der in /8/ eingeführten Butcher-Reihen und erlauben eine einfache Herleitung von Ordnungsbedingungen für numerische Methoden.

Definition 5.8:

Seien $A : LDAT_y \to R$ und $B : LDAT_z \to R$ beliebige Abbildungen.

Die Reihe

$$DA_y(A,y_0,z_0) := y_0 + \sum_{t\in LDAT_y} A(t)\,F(t)\,(y_0,z_0)\,\frac{h^{\rho(t)}}{\rho(t)!}$$

bzw.

$$DA_z(B,y_0,z_0) := z_0 + \sum_{u\in LDAT_z} B(u)\,G(u)\,(y_0,z_0)\,\frac{h^{\rho(u)}}{\rho(u)!}$$

heißt *DA_y-Reihe* bzw. *DA_z-Reihe*.

Nach Satz 5.7 sind insbesondere die Lösungen von (4.1) DA-Reihen:

(5.10) $y(x_o+h) = DA_y(P_y,y_o,z_o)$

$z(x_o+h) = DA_z(P_z,y_o,z_o)$ mit

$P_y(t) = 1$ für alle $t \in DAT_y$ und $P_z(u) = 1$ für alle $u \in DAT_z$.

6. Rosenbrock-Wanner und ähnliche Methoden zur Lösung von Index-Eins Problemen

In diesem Paragraphen stellen wir zwei Lösungsansätze vor. Für diese leiten wir Ordnungs- und Konvergenzbedingungen bis zur Ordnung vier her. Zusätzlich geben wir an, welche maximale Ordnung Verfahren gegebener Stufenzahl erreichen können.

6.1. Explizite/Implizite Lösungsansätze

A) Impliziter Ansatz

Um die Idee der Rosenbrock-Wanner (ROW) oder verallgemeinerten Runge-Kutta Methoden /9/ auf DAE's der Form (4.1) übertragen zu können, gehen wir wie folgt vor /17/ :

Wir ersetzen (4.1) durch das Ersatzproblem

(6.1a) $y' = f(y,z)$, $\quad y(x_o) = y_0$

(6.1b) $z' = (1/\varepsilon)g(y,z)$, $\quad z(x_o) = z_o$

wobei ε sehr klein ist.

Wenden wir die ROW-Methoden auf (6.1) an, multiplizieren die zweite Gleichung mit ε und setzen $\varepsilon = 0$, so erhalten wir die Verfahrensklasse

$$(6.2a) \quad a_i = y_o + \sum_{j=1}^{i-1} \alpha_{ij} l_j$$

$$(6.2b) \quad b_i = z_o + \sum_{j=1}^{i-1} \alpha_{ij} k_j$$

$$(6.2c)\quad l_i = h\, f(a_i,b_i) + h \sum_{j=1}^{i} \gamma_{ij}\left((D_y f)_o l_j + (D_z f)_o k_j\right)$$

$$(6.2d)\quad 0 = g(a_i,b_i) + \sum_{j=1}^{i} \gamma_{ij}\left((D_y g)_o l_j + (D_z g)_o k_j\right)$$

$$i = 1,..,s$$

$$(6.2e)\quad y_1 = y_o + \sum_{i=1}^{s} \mu_i l_i$$

$$(6.2d)\quad z_1 = z_o + \sum_{i=1}^{s} \mu_i k_i$$

α_{ij}, γ_{ij} und μ_i sind reelle Parameter mit $\gamma_{ii} = \gamma \neq 0$ für $i = 1,..,s$; s ist die Stufenzahl und $(D_y f)_o$, $(D_z f)_o$, $(D_y g)_o$, $(D_z g)_o$ sind die partiellen Ableitungen im Punkt (y_o,z_o). a_i, b_i, k_i, l_i und y_1, z_1 sind Funktionen in h; y_1, z_1 sind Näherungen von $y(x_o+h)$, $z(x_o+h)$.

B) Expliziter Ansatz

Enthält die y-Komponente der Lösung keine steifen Lösungsanteile, so können wir auf Gleichung (6.1a) auch eine explizite Runge-Kutta Methode anwenden.
Auf diese Weise erhalten wir den zu (6.2) analogen Ansatz (6.3), indem wir (6.2c) durch
(6.3c) $l_i = h\, f(a_i,b_i)$ ersetzen.

Vor- und Nachteile der beiden Ansätze werden in den Paragraphen 7 und 8 erörtert.

6.2. Herleitung der Ordnungsbedingungen

Zunächst befassen wir uns analog zu /17/ mit dem impliziten Ansatz A:

Satz 6.4:

Die durch (6.2) definierten Funktionen a_i, b_i, l_i, k_i, $i = 1,..,s$ und y_1, z_1 sind DA-Reihen, deren Koeffizienten A_i, B_i, L_i, K_i, $i = 1,..,s$ und Y_1, Z_1 rekursiv definiert sind durch:

$$(6.4a)\quad A_i(t) = \sum_{j=1}^{i-1} \alpha_{ij} L_j(t) \quad \text{für alle } t \varepsilon LDAT_y$$

$$(6.4b)\quad B_i(u) = \sum_{j=1}^{i-1} \alpha_{ij} K_j(u) \quad \text{für alle } u \varepsilon LDAT_z$$

$$(6.4c)\quad L_i(\tau_y) = 1$$

$$L_i(t) = \rho(t)\, A_i(t_1)\cdot\cdot A_i(t_k) B_i(u_1)\cdot\cdot B_i(u_l)$$

$$+ \begin{cases} 0 & \text{falls } k+l > 1 \\ \rho(t) \sum_{j=1}^{i} \gamma_{ij} L_j(t_1) & \text{falls } k=1,\ l=0 \\ \rho(t) \sum_{j=1}^{i} \gamma_{ij} K_j(u_1) & \text{falls } k=0,\ l=1 \end{cases}$$

$$\text{wobei } t = \left[t_1,\ldots,t_k,u_1,\ldots,u_l\right]_y \text{ mit } t_1,\ldots,t_k \varepsilon LDAT_y,\ u_1,\ldots,u_l \varepsilon LDAT_z.$$

$$(6.4d)\quad K_i(\tau_z) = 1$$

$$0 = \begin{cases} A_i(t_1)\cdot\cdot A_i(t_k) B_i(u_1)\cdot\cdot B_i(u_l) - \\ \quad \sum_{j=1}^{i} (\alpha_{ij} + \gamma_{ij})\, K_j(u), & \text{falls } k+l > 1 \\ \sum_{j=1}^{i} (\alpha_{ij} + \gamma_{ij})(L_j(t_1) - K_j(u)), & \text{falls } k=1, l=0 \end{cases}$$

wobei $u = \lceil t_1,..,t_k,u_1,..,u_l \rceil_z$ mit $t_1,..,t_k \in LDAT_y$

$u_1,..,u_l \in LDAT_z$

und $\alpha_{ii} := 0$ für $i=1,..,s$.

(6.4e) $Y_1(t) = \sum_{i=1}^{s} \mu_i L_i(t)$ für alle $t \in LDAT_y$

(6.4f) $Z_1(u) = \sum_{i=1}^{s} \mu_i K_i(u)$ für alle $u \in LDAT_z$.

Der vollständige Beweis des Satzes ist in /21/ widergegeben.

Setzen wir nun $\beta_{ij} := \alpha_{ij} + \gamma_{ij}$, $\beta_{ii} := \gamma$, so vereinfacht sich (6.4c) und (6.4d) zu

(6.4c)' $L_i(\tau_y) = 1$

$$L_i(t) = \begin{cases} \rho(t) \sum_{\substack{(n_1,..,n_k,m_1,..,m_l) \\ \in\{1,..,i-1\}^{k+l}}} \alpha_{in_1}..\alpha_{in_k}\alpha_{im_1}..\alpha_{im_l} L_{n_1}(t_1).. \\ \qquad L_{n_k}(t_k)K_{m_1}(u_1)..K_{m_l}(u_l) \text{ , falls } k+l>1 \\ \rho(t) \sum_{j=1}^{i} \beta_{ij} L_j(t_1) \qquad \text{falls } k=1,\ l=0 \\ \rho(t) \sum_{j=1}^{i} \beta_{ij} K_j(u_1) \qquad \text{falls } k=0,\ l=1 \end{cases}$$

(6.4d)′ $K_i(\tau_z) = 1$

$$0 = \begin{cases} \sum\limits_{\substack{(n_1,..,n_k,m_1,...,m_k) \\ \varepsilon\{1,..,i-1\}^{l+k}}} \alpha_{in_1}\cdot\cdot\alpha_{in_k}\alpha_{im_1}\cdot\cdot\alpha_{im_l} L_{n_1}(t_1)\cdot\cdot \\ \qquad L_{n_k}(t_k)K_{m_1}(u_1)\cdot\cdot K_{m_l}(u_l) - \sum\limits_{j=1}^{i} \beta_{ij}K_j(u) \\ \qquad\qquad \text{falls } k+l > 1 \\ \sum\limits_{j=1}^{i} \beta_{ij}(L_j(t_1) - K_j(u)) \qquad \text{falls } k=1,\ l=0 \end{cases}$$

Proposition 6.5:

Sei $u = [t_1]_z$, $t_1 \varepsilon \text{LDAT}_y$.

Dann gilt: $L_i(t_1) = K_i(u)$ für $i=1,..,s$.

Beweis

Für i=1 folgt die Behauptung wegen $\gamma \neq 0$ aus (6.4d)′.

Durch weitere Anwendung von (6.4d)′ folgt induktiv die Behauptung.

Beispiel 6.6:

Für $u =$  und $t =$  gilt: $L_i(t) = K_i(u)$.

Wir definieren nun Matrizen β und W:

$\beta := (\beta_{ij})_{i=1,..,s}^{j=1,..,i}$ und Null für $j>1$; $W := (w_{ij}) := \beta^{-1}$.

W existiert wegen $\beta_{ii} = \gamma \neq 0$.

Damit vereinfacht sich (6.4d)′ weiter zu

(6.4d)′′ $K_i(\tau_z) = 1$

$$K_i(u) = \begin{cases} \sum_{j=1}^{i} w_{ij} \sum_{(n_1,..,n_k,m_1,..,m_l) \in \{1,..,j-1\}^{l+k}} \alpha_{jn_1} \cdot\cdot \alpha_{jn_k} \alpha_{jm_1} \cdot\cdot \alpha_{jm_l} \cdot \\ \quad \cdot L_{n_1}(t_1) \cdot\cdot L_{n_k}(t_k) K_{m_1}(u_1) \cdot\cdot K_{m_l}(u_l) \quad \text{falls } k+l>1 \\ L_i(t_1) \qquad \text{falls } k=1,\ l=0 \end{cases}$$

Proposition 6.7:

Sei $u = [t_1,..,t_k,u_1,..,u_l]_z$ mit $t_1,..,t_k \in LDAT_y$

$u_1,..,u_l \in LDAT_z$.

Dann gilt $L_i([u]_y) = L_i([t_1,..,t_k,u_1,..,u_l]_y)$.

Beweis

Aus (6.4c)′ und (6.4d)′′ folgt

$L_i([u]_y) = \rho([u]_y) \sum_{j=1}^{i} \beta_{ij} K_j(u) =$

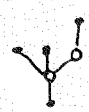
$$= \rho(\lceil t_1,..,u_l \rceil_y) \sum_{j=1}^{i} \beta_{ij} \sum_{p=1}^{j} w_{jp} \sum_{\substack{(n_1,..,m_l) \\ \varepsilon\{1,..,j-1\}^{k+l}}} \alpha_{pn_1} \cdot\cdot \alpha_{pm_l} L_{n_1}(t_1) \cdot$$

$$\cdot\cdot K_{m_l}(u_l)$$

$$= \rho(\lceil t_1,..,u_l \rceil_y) \sum_{\substack{(n_1,..,m_l) \\ \varepsilon\{1,..,i-1\}^{l+k}}} \alpha_{in_1} \cdot\cdot \alpha_{im_l} L_{n_1}(t_1) \cdot\cdot K_{m_l}(u_l)$$

$$= L_i(\lceil t_1,..,u_l \rceil_y) \qquad \text{q.e.d.}$$

Beispiel 6.8:

Für t_1 =  und t_2 =  gilt: $L_i(t_1) = L_i(t_2)$.

Durch Koeffizientenvergleich der DA-Reihen für die exakte Lösung des Index-Eins Problems (siehe Satz 5.8) und der DA-Reihen (6.4e), (6.4f) können wir nun mit Hilfe von (6.4c)′ und (6.4d)′′ Ordnungsbedingungen für die Koeffizienten μ_i, α_{ij}, γ_{ij} der Methode (6.2) herleiten.

Beispiel 6.9:

Gegeben sei der Baum u =  .

Für die Koeffizienten $P_y(t)$ und $P_z(u)$ der exakten Lösung gilt $P_y(t) = P_z(u) = 1$ für alle $u,t \in$ LDAT (siehe Satz 5.8).

Für die Koeffizienten der DA-Reihe z_1 in (6.4f) gilt

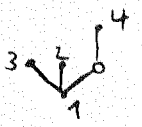
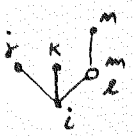
$$Z_1(u) = \sum_{i=1}^{s} \mu_i K_i(u) = \sum_{i=1}^{s} \mu_i \sum_{j=1}^{i} w_{ij} \sum_{k=1}^{j-1} \sum_{l=1}^{j-1} \alpha_{jk}\alpha_{jl} \cdot$$

Als Bedingung ergibt sich somit in abgekürzter Schreibweise:

$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} = 1$.

Diese Bedingung ist dem Baum u =  zugeordnet.

Betrachten wir beispielsweise den Baum

 bzw.  , so ergibt sich:

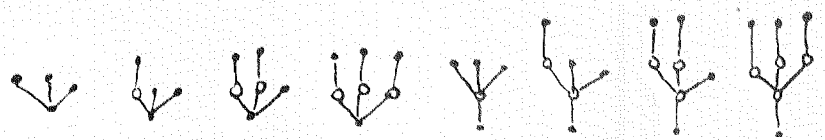
$4 \sum \mu_i \alpha_{ij} \alpha_{ik} \alpha_{il} w_{lm} r_{mn} = 1.$

Bäume, die sich nur in der Indizierung unterscheiden, liefern natürlich identische Bedingungen.

Eine wichtige Folgerung aus Proposition 6.5 und 6.7 ist, daß viele verschiedene Bäume gleiche Ordnungsbedingungen liefern.

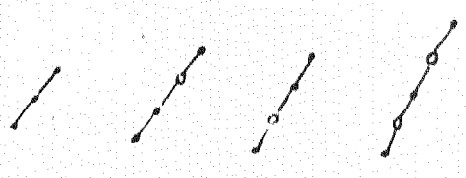
Beispiele dafür sind:

(6.10a) 

(6.10b) 

(6.10c) 

(6.10d) alle Bäume aus $LDAT_y$ der Ordnung n, die keine Verzweigungen haben, z.B. für n = 3:



Proposition 6.11:

Die Ordnungsbedingungen der Bäume $t \in LDAT_y$ mit ausschließlich dünnen Knoten sind mit den Ordnungsbedingungen der klassischen Rosenbrock-Wanner Methoden identisch.

Beweis

Wir erhalten die Ordnungsbedingungen für Bäume t mit ausschließlich dünnen Knoten, indem wir in (6.4c)' $1 = 0$ setzen. Vergleichen wir dann (6.4c)' mit Theorem 2 aus /10/, so folgt die Behauptung.

Für ein Verfahren der Ordnung vier nach Ansatz A ergeben sich folgende Ordnungsbedingungen (identische Bedingungen für verschiedene Bäume werden weggelassen).

y-Komponente

$\rho(t)$	t	Ordnungsbedingung	
1		$\sum \mu_i = 1$	(6.12a)
2		$\sum \mu_i \beta_{ij} = 1/2$	(6.12b)
3		$\sum \mu_i \alpha_{ij} \alpha_{ik} = 1/3$	(6.12c)
		$\sum \mu_i \beta_{ij} \beta_{jk} = 1/6$	(6.12d)
4		$\sum \mu_i \alpha_{ij} \alpha_{ik} \alpha_{il} = 1/4$	(6.12e)
		$\sum \mu_i \alpha_{ij} \alpha_{ik} \beta_{kl} = 1/8$	(6.12f)

$$\sum \mu_i \beta_{ij} \alpha_{jk} \alpha_{jl} = 1/12 \qquad (6.12g)$$

$$\sum \mu_i \alpha_{ij} \alpha_{ik} w_{kl} \alpha_{lm} \alpha_{ln} = 1/4 \qquad (6.12h)$$

$$\sum \mu_i \beta_{ij} \beta_{jk} \beta_{kl} = 1/24 \qquad (6.12i)$$

z-Komponente

$\rho(u)$	u	Ordnungsbedingung	
2		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} = 1$	(6.12j)
3		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} \alpha_{jm} = 1$	(6.12k)
		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} \beta_{lm} = 1/2$	(6.12l)
		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} w_{lm} \alpha_{mn} \alpha_{mp} = 1$	(6.12m)

Behandeln wir nun den expliziten Ansatz B.

Analog zu Satz 6.4 erhalten wir:

Satz 6.13:

Die durch (6.3) definierten Funktionen a_i, b_i, l_i, k_i, $i=1,..,s$ und y_1, z_1 sind DA-Reihen, deren Koeffizienten A_i, B_i, L_i, K_i, $i=1,..,s$ und Y_1, Z_1 rekursiv definiert sind durch:

(6.13a) $A_i(t) = \sum_{j=1}^{i-1} \alpha_{ij} L_j(t)$ für alle $t \in LDAT_y$

(6.13b) $B_i(u) = \sum_{j=1}^{i-1} \alpha_{ij} K_j(u)$ für alle $u \in LDAT_z$

(6.13c) $L_i(\tau_y) = 1$

$L_i(t) = \rho(t) A_i(t_1) \cdot\cdot A_i(t_k) B_i(u_1) \cdot\cdot B_i(u_l)$,

falls $t = [t_1,..,t_k,u_1,..,u_l]_y$ mit $t_1,..,t_k \in LDAT_y$

$u_1,..,u_l \in LDAT_z$.

(6.13d) $K_i(\tau_z) = 1$

$$0 = \begin{cases} A_i(t_1) \cdot\cdot A_i(t_k) B_i(u_1) \cdot\cdot B_i(u_l) - \sum_{j=1}^{i} (\alpha_{ij} + \gamma_{ij}) K_j(u) & \\ & \text{falls } k+l > 1 \\ \sum_{j=1}^{i} (\alpha_{ij} + \gamma_{ij})(L_j(t_1) - K_j(u)) & \text{falls } k=1, l=0 \end{cases}$$

wobei $u = [t_1,..,t_k,u_1,..,u_l]_z$; $t_1,..,t_k \in LDAT_y$;

$u_1,..,u_l \in LDAT_z$ und $\alpha_{ii} = 0$ für $i=1,..,s$.

(6.13e) $Y_1(t) = \sum_{i=1}^{s} \mu_i L_i(t)$ für alle $t \in LDAT_y$

(6.13f) $Z_1(u) = \sum_{i=1}^{s} \mu_i K_i(u)$ für alle $u \in LDAT_z$.

Der Beweis von Satz 6.4 kann fast wörtlich übertragen werden.

Wie im impliziten Ansatz A können wir (6.13c) und (6.13d) vereinfachen:

(6.13c)' $L_i(\tau_y) = 1$

$$L_i(t) = \rho(t) \sum_{(n_1,..,n_k,m_1,..,m_l) \in \{1,..,i-1\}^{k+l}} \alpha_{in_1} \cdot\cdot \alpha_{in_k} \alpha_{im_1} \cdot\cdot \alpha_{im_l} L_{n_1}(t_1) \cdot\cdot L_{n_k}(t_k) K_{m_1}(u_1) \cdot\cdot K_{m_l}(u_l)$$

(6.13d)′ $K_i(\tau_z) = 1$

$$0 = \begin{cases} \sum_{\substack{(n_1,\ldots,n_k,m_1,\ldots,m_l)\\ \varepsilon\ \{1,..,i-1\}^{k+l}}} \alpha_{n_1}\cdot\cdot\alpha_{m_l} L_{n_1}(t_1)\cdot\cdot K_{m_l}(u_l) - \\ \qquad \sum_{j=1}^{i} \beta_{ij} K_j(u) \qquad \text{falls } k+l > 1 \\ \sum_{j=1}^{i} \beta_{ij}(L_j(t_1)-K_j(u)) \qquad \text{falls } k=1,\ l=0 \end{cases}$$

Damit erhalten wir:

Proposition 6.14:

Für $u = \left[t_1\right]_z$, $t_1 \varepsilon\ LDAT_y$ gilt $L_i(t_1) = K_i(u)$, $i=1,..,s$.

Beweis siehe Proposition 6.5.

Durch Definition der Matrizen β und W folgt wie im Ansatz A:

(6.13d)′′ $K_i(\tau_z) = 1$

$$K_i(u) = \begin{cases} \sum_{j=1}^{i} w_{ij} \sum_{\substack{(n_1,\ldots,m_l)\\ \varepsilon\ \{1,..,j-1\}^{k+l}}} \alpha_{jn_1}\cdot\cdot\alpha_{jm_l} L_{n_1}(t_1)\cdot\cdot K_{m_l}(u_l) \\ \qquad \text{falls } k+l > 1 \\ L_i(t_1) \qquad \text{falls } k=1, l=0 \end{cases}$$

Bemerkung 6.15:

Proposition 6.7 ist für den expliziten Ansatz B nicht gültig.

Proposition 6.16:

Die Ordnungsbedingungen der Bäume $t \in LDAT_y$ mit ausschließlich dünnen Knoten sind mit den Ordnungsbedingungen der klassischen Runge-Kutta Methoden identisch.

Beweis

Setzen wir in (6.13c)' $1 = 0$ und vergleichen die resultierenden Ordnungsbedingungen mit denen in /2/, dann folgt die Behauptung.

Für ein Verfahren der Ordnung vier nach Ansatz B ergeben sich mit Hilfe von (6.13c)', (6.13d)'' und Proposition 6.14 folgende Ordnungsbedingungen:

y-Komponente

$\rho(t)$	t	Ordnungsbedingung	
1		$\sum \mu_i = 1$	(6.17a)
2		$\sum \mu_i \alpha_{ij} = 1/2$	(6.17b)
3		$\sum \mu_i \alpha_{ij} \alpha_{ik} = 1/3$	(6.17c)
		$\sum \mu_i \alpha_{ij} \alpha_{jk} = 1/6$	(6.17d)
		$\sum \mu_i \alpha_{ij} w_{jk} \alpha_{kl} \alpha_{km} = 1/3$	(6.17e)
4		$\sum \mu_i \alpha_{ij} \alpha_{ik} \alpha_{il} = 1/4$	(6.17f)

$$\sum \mu_i \alpha_{ij} \alpha_{ik} \alpha_{kl} = 1/8 \qquad (6.17g)$$

$$\sum \mu_i \alpha_{ij} \alpha_{ik} w_{kl} \alpha_{lm} \alpha_{ln} = 1/4 \qquad (6.17h)$$

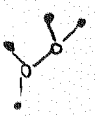
$$\sum \mu_i \alpha_{ij} \alpha_{jk} \alpha_{kl} = 1/24 \qquad (6.17i)$$

$$\sum \mu_i \alpha_{ij} \alpha_{jk} \alpha_{jl} = 1/12 \qquad (6.17j)$$

$$\sum \mu_i \alpha_{ij} \alpha_{jk} w_{kl} \alpha_{lm} \alpha_{ln} = 1/12 \qquad (6.17k)$$

$$\sum \mu_i \alpha_{ij} w_{jk} \alpha_{kl} \alpha_{km} \alpha_{kn} = 1/4 \qquad (6.17l)$$

$$\sum \mu_i \alpha_{ij} w_{jk} \alpha_{kl} \alpha_{km} \alpha_{mn} = 1/8 \qquad (6.17m)$$



$$\sum \mu_i \alpha_{ij} w_{jk} \alpha_{kl} \alpha_{km} w_{mn} \alpha_{np} \alpha_{nq} = 1/4 \qquad (6.17n)$$

<u>z-Komponente</u>

$\rho(u)$	u	Ordnungsbedingung	
2		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} = 1$	(6.17o)
3		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} \alpha_{jm} = 1$	(6.17p)
		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} \alpha_{lm} = 1/2$	(6.17q)
		$\sum \mu_i w_{ij} \alpha_{jk} \alpha_{jl} w_{lm} \alpha_{mn} \alpha_{mp} = 1$	(6.17r)

Bemerkung 6.18:

In den Ordnungsbedingungen für den expliziten Ansatz B treten die Koeffizienten γ_{i1} für i=2,..,s nicht auf.

Beweis

Die Koeffizienten γ_{ij} treten im Ansatz B nur in Gleichung (6.3d) auf. Weil wir bei der Herleitung von konsistenten Anfangsbedingungen ausgingen, folgt aus (6.3d) wegen $\gamma \neq 0$:

$(D_y g)_o l_1 + (D_z g)_o k_1 = 0$.

Daher sind aber γ_{i1} für i=2,..,s frei wählbar und können somit nicht in den Ordnungsbedingungen auftreten.

6.3. Herleitung der Konvergenzbedingungen

Mit Hilfe des vorangegangenen Abschnitts können wir Verfahren der Ordnung p herleiten. Damit auch Konvergenz der Ordnung p sichergestellt ist, muß zusätzlich gelten:

$$\alpha = ||(\partial\Psi/\partial z)(h=0)|| < 1 \quad \text{(siehe Satz 4.4)}$$

Dazu führen wir die Stabilitätsfunktion R(z) der ROW-Methoden für die skalare Testgleichung $y' = \lambda y$, $y(0) = 1$, $\lambda \in \mathbb{C}$, $z = \lambda h$ ein:

$$R(z) = 1 + \sum_{j=1}^{s} \left(\frac{z}{1-\gamma z} \right)^j \vec{\mu}^t \beta^{j-1} \vec{1}$$

mit $\beta = (\beta_{ij})_{i=1,..,s}^{j=1,..,i-1}$ und Null für $j \geq i$

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}$$

$$\vec{1} = (1,..,1)^t \quad , \quad \vec{\mu}^t = (\mu_1,..,\mu_s) \quad .$$

Bemerkung 6.19:

Wenden wir die ROW-Methoden für gewöhnliche Differentialgleichungen auf die oben angegebene skalare Testgleichung an, dann erfüllt R(z): $y_1 = R(z)y_o$ mit $z = \lambda h$ (siehe /10/ und /15/).

Eine Methode heißt *A-stabil*, falls $|R(z)| \leq 1$ für alle z mit $Re(z) < 0$.

Eine Methode heißt *stabil im Unendlichen*, falls

$\lim_{z\to\infty} |R(z)| = 0$.

Die exakte Lösung der skalaren Testgleichung ist gegeben durch $y(x) = \exp(\lambda x)$.

A-Stabilität und Stabilität im Unendlichen sind also Maße dafür, wie gut die fallende e-Funktion approximiert wird.

Explizite Runge-Kutta Methoden können nicht A-stabil oder stabil im Unendlichen sein, da deren Stabilitätsfunktionen Polynome sind.

Für die Konvergenzbedingung erhalten wir:

Satz 6.20:

Seien (y_o, z_o) konsistent gewählt.

Dann gilt sowohl für den impliziten Ansatz A wie für den expliziten Ansatz B:

$$\tilde{\alpha} := \left|\frac{\partial z_1}{\partial z_o}\right| = \lim_{z\to\infty} |R(z)| = |1 + \sum_{j=1}^{s} (-1)^j \gamma^{-j} \vec{\mu}^t \beta^{j-1} \vec{1}|$$

Der Beweis ist in /21/ zu finden.

Folgerung 6.21:

Weil wir im obigen Satz von konsistenten Werten (y_o, z_o) ausgingen, ist die Bedingung $\tilde{\alpha} < 1$ notwendig für die Voraussetzung von Satz 4.4, d.h. $\left|\frac{\partial \Psi}{\partial z}(y,z,0)\right| \leq \alpha < 1$.

Weil wir weiterhin voraussetzen, daß $\tilde{\alpha} = \left|\frac{\partial z_1}{\partial z_o}\right|$ stetig in z_o ist, folgt aus $\tilde{\alpha} < 1$ die Bedingung $\left|\frac{\partial \Psi}{\partial z}(y,z,0)\right| \leq \alpha < 1$ in einer geeigneten kompakten Umgebung der Lösung (y,z). Damit ist $\tilde{\alpha} < 1$ auch hinreichende Bedingung für die Konvergenz der Ordnung p eines Verfahrens der Konsistenzordnung p.

Im folgenden verzichten wir daher auf eine Unterscheidung zwichen $\tilde{\alpha}$ und α und bezeichnen $|R(\infty)|$ mit α.

6.4. Beziehungen zwischen Stufenzahl und Konvergenzordnung

Nachfolgend geben wir die maximal zu erreichende Konvergenzordnung für Verfahren gegebener Stufenzahl an. Wir beschränken uns dabei auf Konvergenzordnungen $p \leq 4$ und Stufenzahlen $s \leq 5$ für den Ansatz A bzw. $s \leq 4$ für den Ansatz B.

Für Methoden gemäß dem impliziten Ansatz A erhalten wir:

(6.22)

Stufenzahl s	1	2	3	4	5
Konvergenzordnung p	1	2	3	3	4

Bemerkung 6.23:

a) Für $s = 1$ erhalten wir mit $\mu_1 = 1$ und $\gamma = 1$ ein Verfahren der Konvergenzordnung $p = 1$, da $\alpha = 0$.
Dieses Verfahren ist mit der semi-impliziten Euler-Diskretisierung (3.3) identisch.

b) Es existiert kein Verfahren mit $s = 1$ und $p = 2$, denn:
Aus den Ordnungsbedingungen folgt $\mu_1 = 1$, $\beta_{11} = \gamma = 1/2$.
Jedoch ist die Konvergenzbedingung $\alpha < 1$ wegen
$\alpha = |1 - 1/\gamma| = 1$ verletzt.

c) Ein Verfahren mit $s = 2$ und $p = 2$ erhalten wir, indem wir z. B. $\gamma = 1$, $\mu_1 = 0$, $\mu_2 = 1$, $\alpha_{21} = 1$ und $\gamma_{21} = -3/2$ setzen. Mit diesen Werten folgt $\alpha = 1/2$.

d) Es exsistiert ein Verfahren mit $s = 3$ und $p = 3$ (siehe Herleitung der Methode DAE3S im nächten Kapitel).

e) Ebenso leiten wir im Abschitt 7.2 die Methode DAE4SF mit $s = 5$ und $p = 4$ her.

Proposition 6.24:

Es exsistiert keine Methode mit s = 2 und p = 3.

Beweis

Gleichungen (6.12c) und (6.12j) liefern $\mu_2\alpha_{21}^2 = 1/3$ und $\mu_2\alpha_{21}^2/\gamma = 1$. Daraus folgt $\gamma = 1/3$.

Einsetzen von (6.12a) in (6.12b) liefert $\mu_2\beta_{21} = 1/2 - \gamma = 1/6$. Setzen wir die gewonnenen Werte und (6.12a) in (6.12d) ein, so ergibt sich ein Widerspruch.

Proposition 6.25:

Es existiert keine Methode mit s = 4 und p = 4.

Der Beweis kann in /17/ und /21/ nachgelesen werden.

Damit ist (6.22) gezeigt, und wir können uns dem expliziten Ansatz B zuwenden. Wir erhalten:

(6.26)

Stufenzahl s	1	2	3	4
Konvergenzordnung p	1	2	3	3

Bemerkung 6.27:

a) Ein Verfahren der Konvergenzordnung p = 1 erhalten wir mit $\mu_1 = 1$, $\gamma = 1$.

b) Wegen der Bedingung $\sum\mu_i\alpha_{ij} = 1/2$ exsistiert kein Verfahren mit s = 1 und p = 2.

c) Mit $\mu_1 = \mu_2 = 1/2$, $\alpha_{21} = 1$, $\gamma = 1$ und $\gamma_{21} = -1$ erhalten wir eine Methode der Stufenzahl $s = 2$ und Konvergenzordnung $p = 2$.

d) Wegen der Bedingung $\sum \mu_i \alpha_{ij} \alpha_{jk} = 1/6$ exsistiert keine Methode mit $s = 2$ und $p = 3$.

e) Die Methode DAE3NS aus Abschnitt 7.2 hat Stufenzahl $s = 3$ und Konvergenzordnung $p = 3$.

f) Es exsistieren Methoden mit $s = 6$ und $p = 4$ (siehe Methode DAE34NS aus Abschnitt 7.2).

Proposition 6.28:

Es exsistiert kein Verfahren mit $s = 4$ und $p = 4$.

Beweis

Durch Vereinfachung der Gleichungen (6.17) erhalten wir:

(6.28e) $\mu_4 \alpha_{43} w_{32} \alpha_2^2 = 1/3 - 1/(12\gamma)$

(6.28i) $\mu_4 \alpha_{43} \alpha_{32} \alpha_2 = 1/24$

(6.28j) $\mu_3 \alpha_{32} \alpha_2^2 + \mu_4 \alpha_{42} \alpha_2^2 + \mu_4 \alpha_{43} \alpha_3^2 = 1/12$

(6.28k) $\mu_4 \alpha_{43} \alpha_{32} \alpha_2^2 = \gamma/12$

(6.28l) $\mu_3 \alpha_{32} \alpha_2^3/\gamma + \mu_4 \alpha_{42} \alpha_2^3/\gamma + \mu_4 \alpha_{43} \alpha_3^3/\gamma + \mu_4 \alpha_{43} w_{32} \alpha_2^3 = 1/4$

(6.28m) $\mu_4 \alpha_{43} \alpha_3 \alpha_{32} \alpha_2 = \gamma/8$

(6.28o) $\mu_3 w_{32} \alpha_2^2 + \mu_4 w_{42} \alpha_2^2 + \mu_4 w_{43} \alpha_3^2 = 1 - 1/(3\gamma)$

(6.28p) $\mu_3 w_{32}\alpha_2^3 + \mu_4 w_{42}\alpha_2^3 + \mu_4 w_{43}\alpha_3^3 = 1 - 1/(4\gamma)$

(6.28q) $\mu_4 w_{43}\alpha_3\alpha_{32}\alpha_2 = 1/2 - 1/(8\gamma)$

Aus (6.28i) und (6.28k) folgt $\alpha_2 = 2\gamma$.
(6.28m) und (6.28q) liefern $w_{43}/\alpha_{43} = (4\gamma - 1)/\gamma^2$ und
(6.28e) und (6.28i) liefern $w_{32}/\alpha_{32} = (4\gamma - 1)/\gamma^2$.
Setzen wir diese Werte in (6.28o) ein, so erhalten wir zusammen mit (6.28j) die Gleichung

$$(4\gamma - 1)\mu_4\alpha_{42}\alpha_2^2 = (4\gamma - 1)/12 + \mu_4 w_{42}\alpha_2^2\gamma^2 - \gamma(3\gamma - 1)/3$$

Entsprechend erhalten wir aus (6.28e), (6.28p) und (6.28l) die Gleichung

$$(4\gamma - 1)\mu_4\alpha_{42}\alpha_2^2 = -(4\gamma - 1)^2/12 + \mu_4 w_{42}\alpha_2^2\gamma^2 .$$

Damit folgt $\gamma = 0$, und die Behauptung ist bewiesen.

Auf diese Weise ist (6.26) vollständig gezeigt.

7. Implementierte Verfahren

7.1. Implementierungsfragen und Aufrufliste

Zur Vermeidung von Matrix-Vektor Produkten formen wir den impliziten Ansatz A wie folgt um:

$$l_i = h\, f(a_i,b_i) + h \sum_{j=1}^{i} \gamma_{ij}\left((D_y f)_o l_j + (D_z f)_o k_j\right)$$

$$0 = g(a_i,b_i) + \sum_{j=1}^{i} \gamma_{ij}\left((D_y g)_o l_j + (D_z g)_o k_j\right)$$

$\leftrightarrow$

$$(7.1)\quad \left[\begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} - \gamma \begin{pmatrix} h(D_y f)_o & h(D_z f)_o \\ (D_y g)_o & (D_z g)_o \end{pmatrix}\right] \begin{pmatrix} l_i \\ k_i \end{pmatrix} + \sum_{j=1}^{i-1} \tilde{\gamma}_{ij} \begin{pmatrix} l_j \\ k_j \end{pmatrix} = \begin{pmatrix} h\, f(a_i,b_i) \\ g(a_i,b_i) \end{pmatrix} + \sum_{j=1}^{i-1} \tilde{\gamma}_{ij} \begin{pmatrix} l_j \\ 0 \end{pmatrix},$$

$\tilde{\gamma}_{ij} = \gamma_{ij}/\gamma$ für $i = 1,..,s$ und E ist Einheitsmatrix der Dimension (N,N).

In jedem Integrationsschritt muß also ein lineares Gleichungssystem der Dimension N + M mit s verschiedenen rechten Seiten gelöst werden. Dazu führen wir eine LU-Zerlegung der (N+M,N+M)-Matrix durch und speichern diese.

Weil bei Index-Eins Problemen $(D_z g)^{-1}$ existiert, ist sichergestellt, daß obige Matrix bei genügend kleiner Schrittweite h regulär ist.

Zusätzlich sind dann in jedem Integrationsschritt s Rücksubstitutionen, s Auswertungen von f und g und eine Berechnung

der Jacobi-Matrix von (f,g) nötig.
Da die partiellen Ableitungen von f und g i. allg. durch Differenzenquotienten berechnet werden, benötigen wir noch-einmal N+M Auswertungen von f und g in jedem Integrations-schritt.

Die Implementierung des expliziten Ansatzes B geschieht auf folgende Weise:

(7.2) $l_i = h\, f(a_i, b_i)$

$$(D_z g)_o k_i = -g(a_i, b_i)/\gamma - \sum_{j=1}^{i-1} \tilde{\gamma}_{ij} \left((D_y g)_o l_j + (D_z g)_o k_j\right)$$

$$-(D_y g)_o l_i \; ; \qquad \tilde{\gamma}_{ij} = \gamma_{ij}/\gamma, \; i=1,..,s.$$

Hierbei sind im wesentlichen pro Integrationsschritt nötig:
- s Auswertungen von f und g
- die Berechnung aller partieller Ableitungen von f und g
- eine LU-Zerlegung der (M,M)-Matrix $(D_z g)_o$
- s Rücksubstitutionen und s Matrix-Vektor Produkte der Form $(D_y g)_o l_i$.

Der explizite Ansatz B bietet also insbesondere bei einer großen Anzahl N von Differentialgleichungen im Verhältnis zur Anzahl M der algebraischen Nebenbedingungen Vorteile, vorausgesetzt, daß die Differentialgleichungen keine steifen Lösungsanteile besitzen.

Alle implementierten Verfahren werden mit einer lokalen Fehlerschätzung und einer Schrittweitensteuerung, wie in /9/ und /22/ beschrieben, ausgestattet.

Aus diesem Grund werden nur *eingebettete Verfahren* implementiert. Eine Methode der Ordnung p

$$(7.3)\quad \begin{pmatrix} y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{i=1}^{s} \mu_i \begin{pmatrix} l_i \\ k_i \end{pmatrix}$$

und eine Methode der Ordnung p-1

$$(7.4)\quad \begin{pmatrix} \hat{y}_1 \\ \hat{z}_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{i=1}^{\hat{s}} \hat{\mu}_i \begin{pmatrix} l_i \\ k_i \end{pmatrix}, \quad \hat{s} \leq s$$

werden miteinander kombiniert, wobei die Koeffizienten γ, γ_{ij}, α_{ij} $(i = 1,..,s;\ j = 1,..,i-1)$ und daher auch l_i und k_i $(i = 1,..,s)$ für beide Methoden gleich sind.

Auf diese Weise entstehen für die Implementierung eingebetteter Verfahren nur sehr geringe zusätzliche Kosten.

I. allg. liefert (7.3) die numerische Lösung, d.h. (7.4) dient nur zur Schrittweitenkontrolle.

Zur Erläuterung der Schrittweitensteuerung beschränken wir uns zunächst nur auf die y-Komponenten der Lösung.

Seien y_{i+1} bzw. $\hat{y}_{i+1}$ die durch (7.3) bzw. (7.4) erzeugten numerischen Lösungen im Zeitpunkt x_{i+1} zu dem gemeinsamen Anfangswert y_i.

Lokal gilt die Gleichung

$$y(x_{i+1}) - y_{i+1} = C_1(x_i)h^{p+1} + O(h^{p+2})$$

bzw. $$y(x_{i+1}) - \hat{y}_{i+1} = C_2(x_i)h^{p} + O(h^{p+1}),$$

wobei $y(\cdot)$ die exakte Lösung bezeichnet.

Durch Vernachlässigen der Terme höherer Ordnung erhalten

wir $y_{i+1} - \hat{y}_{i+1} = C_2(x_i)h^p$.

Die Integration von x_i nach x_{i+1} sei erfolgreich gewesen, d.h. zu gegebener *Fehlerschranke* $TOL > 0$ sei $||y_{i+1} - \hat{y}_{i+1}|| \leq TOL$ erreicht worden, wobei $||\cdot||$ eine beliebige Norm ist. Damit ist auch $||C_2(x_i)||\cdot h^p \leq TOL$.

Soll die neue Schrittweite $h_{neu} = x_{i+2} - x_{i+1}$ erfolgreich sein, so muß gelten $||C_2(x_{i+1})||\cdot h_{neu}^p \leq TOL$.

Taylorentwicklung von C_2 liefert $C_2(x_{i+1}) = C_2(x_i) + O(h)$.

Damit erhalten wir wieder durch Vernachlässigung der Terme höherer Ordnung $||C_2(x_i)||\cdot h_{neu}^p \leq TOL$

$$\text{bzw. } \frac{||y_{i+1} - \hat{y}_{i+1}||}{h^p} h_{neu}^p \leq TOL.$$

Dies kann zum Vorschlag der neuen Schrittweite genutzt werden:

$$(7.5) \quad h_{neu} = h \left(\frac{TOL}{||y_{i+1} - \hat{y}_{i+1}||} \right)^{1/p} .$$

Formel (7.5) wird noch durch einen Sicherheitsfaktor und eine Nebenbedingung ergänzt, die verhindert, daß die Schrittweitensteuerung einen zu großen Zick-Zack Charakter hat.

Wir erhalten:

$$(7.6) \quad h_{neu} = 0.9\, h_{alt}(TOL/EST)^{1/p} \text{ mit der Nebenbedingung}$$

$$0.5\, h_{alt} \leq h_{neu} \leq 1.5\, h_{alt} \text{ , wobei}$$

$$EST = \max_{1 \leq j \leq N} (1/S_j)\, |y_{i+1}^{(j)} - \hat{y}_{i+1}^{(j)}| \text{ ;}$$

$y_{i+1}^{(j)}$ bezeichnet die j-te Komponente von y_{i+1}.

Der Skalierungsvektor $S = (S_1, \ldots, S_N)^t$ ist definiert durch $S_j := \max_{0 \leq l \leq i+1} (1, |y_l^{(j)}|)$.

EST liefert also im wesentlichen eine Schätzung des lokalen Fehlers der Methode (7.4). Der Fehler der Methode (7.3) wird durch EST daher meist überschätzt.
h_{neu} wird akzeptiert, falls nach dem mit der Schrittweite h_{neu} ausgeführten Schritt gilt: EST $\leq$ TOL.
Andernfalls wird Formel (7.6) mit dem eben berechnetem EST und h_{neu} erneut angewendet. Dies führt wegen EST > TOL zu einem kleineren h_{neu}. Dieser Vorgang wird solange wiederholt, bis entweder h_{neu} akzeptiert wird oder $h_{neu} \leq h_{min}$ gilt.
Die minimale Schrittweite h_{min} hängt von der relativen Maschinengenauigkeit und der Intervallänge $x_{END} - x_o$ ab.

Die z-Komponente der Lösung wird nun wie folgt berücksichtigt:
Für Verfahren der Ordnung p gilt

$$y(x_o+h) - y_1 = O(h^{p+1})$$
$$z(x_o+h) - z_1 = O(h^{p}) .$$

Daher wird die Schrittweitenkontrolle (7.6) gesplittet, d.h.

(7.7) $$h_{neu} = 0.9\, h_{alt} \min\left((TOL/EST_y)^{1/p} , (TOL/EST_z)^{1/(p-1)} \right),$$

wobei sich EST_y nur auf die y-Komponenten der Lösung und EST_z sich nur auf die z-Komponenten bezieht.

Die weiteren Formeln in (7.6) gelten für die z-Komponenten analog.

Bemerkung 7.8:

Will man das Splitten der Schrittweitenkontrolle vermeiden und die lokale Genauigkeit in den z-Komponenten erhöhen, so kann man Verfahren der "Ordnung pp" konstruieren, d.h.

$$y(x_o+h) - y_1 = O(h^{p+1})$$
$$z(x_o+h) - z_1 = O(h^{p+1}) \ .$$

(siehe Methode DAE33NS im nächsten Abschnitt).

Schnell fallende Lösungskomponenten, wie sie beispielsweise in steifen Differentialgleichungen auftreten, sind für den Benutzer meist von geringerem Interesse.
Die Wahl des Skalierungsvektors S in (7.6) impliziert, daß solche Lösungskomponenten durch den Fehlertest EST $\leq$ TOL leichter akzeptiert werden. Weiterhin impliziert die Wahl von S, daß für Lösungskomponenten, die dem Betrage nach kleiner eins sind, der absolute Abbrechfehler und andernfalls der relative Abbrechfehler kontrolliert wird.

Der Aufruf eines Verfahrens nach Ansatz A (hier z.B. DAE3S) innerhalb eines FORTRAN-Programms geschieht auf folgende Weise:

```
call DAE3S (N,M,FCN,T,Y,TEND,TOL,HMAX,HI,IER)
```

Die Parameter in der *Aufrufliste* haben bei der Eingabe folgende Bedeutung:

N: Gesamtanzahl aller Gleichungen (d.h. Differentialgleichungen und algebraische Nebenbedingungen).

M: Anzahl der vorkommenden Differentialgleichungen.

FCN: Name des FORTRAN-Unterprogramms, das die rechte Seite f und g auswertet (Beispiel dazu siehe Anhang).

T: Anfangszeitpunkt x_o.

Y: Vektor der Dimension N, der die Anfangswerte (y_o, z_o) enthält.

TEND: Zeitpunkt, bis zu dem das Verfahren integrieren soll.

TOL: Lokale Fehlertoleranz.

HMAX: Maximal erlaubte Schrittweite.

HI: Eingangsschrittweite, die zum Start des Programms vorgeschlagen wird.

IER: Fehlerparameter, bei Eingabe IER = 0.

Nach Programmstop ändert sich die Bedeutung der folgenden Parameter:

T : Zeitpunkt, zu dem das Programm stoppte; bei erfolgreicher Integration T = TEND.

Y : Numerische Lösung zum Zeitpunkt T.

HI : Letzte vorgeschlagene Schrittweite.

IER: IER = 0: Programmlauf erfolgreich abgeschlossen.

IER = 1: Nach wiederholt gescheitertem Fehlertest gilt schließlich $h \leq h_{min}$.

IER = 2: Die zu invertierende Matrix aus (7.1) ist numerisch singulär.

In der Aufrufliste der Verfahren nach Ansatz B ist lediglich eine kleine Änderung enthalten:

FCN bezeichnet jetzt den Namen des FORTRAN-Unterprogramms zur Auswertung der Funktion f.

Die Aufrufliste wird ergänzt durch den Namen FCNA des Unterprogramms, das g auswertet.

Falls bei der Ausgabe IER = 2 gilt, ist die Matrix $(D_z g)$ singulär.

7.2. Charakterisierung der entwickelten Verfahren

In diesem Abschnitt leiten wir Koeffizientensätze für verschiedene Verfahren nach Ansatz A und nach Ansatz B her.

Bemerkung 7.9:
Mit den in dieser Arbeit vorgeschlagenen Verfahrensklassen (Ansatz A und B) können nur *autonome* Index-Eins oder Index-Null Probleme integriert werden.
Bekanntlich können nichtautonome Probleme durch Hinzufügen der Gleichung $y_o' = 1$ in autonome Probleme transformiert werden, wobei y_o die Zeit darstellt.
Bearbeiten wir nun solche Probleme mit unseren Methoden, so fordern wir, daß die rechte Seite f bzw. g des Problems nur zu Zeitpunkten ausgewertet wird, die innerhalb des aktuellen Integrationsintervalls $[x_i, x_{i+1}]$ liegen.
Aus dieser Forderung ergibt sich die Bedingung $0 \leq \alpha_i \leq 1$ für $i=1,..,s$.

a) Das Verfahren DAE3S
Es handelt sich um eine Methode der Konvergenzordnung $p = 3$ mit Stufenzahl $s = 3$. Eine Methode der Konvergenzordnung zwei mit Stufenzahl $\hat{s} = 3$ ist eingebettet.
Weil es sich um eine semi-implizite Methode handelt, symbolisiert der Buchstabe S aus DAE3S, daß die Methode auch für steife Differentialgleichungen geeignet ist.
Obwohl die Stufenzahl $s = 3$ ist, werden nur zwei Funktionsauswertungen von f und g pro Integrationsschritt benötigt.
Weiterhin ist DAE3S für "reine Differentialgleichungen"

A-stabil:

Die Wahl $\alpha_{32} = 0$, $\alpha_{31} = \alpha_{21}$ impliziert, daß pro Schritt nur zwei Funktionsauswertungen benötigt werden.

Die Koeffizienten der Methode dritter Ordnung müssen fünf Bedingungen genügen. Durch Vereinfachung erhalten wir:

(1) $\mu_1 = 1 - \mu_2 - \mu_3$

(2) $\mu_2\beta_2 + \mu_3\beta_3 = 1/2 - \gamma$

(3) $\alpha_{21}^2(\mu_2 + \mu_3) = 1/3$

(4) $\mu_3\beta_{32}\beta_{21} = 1/6 - \gamma + \gamma^2$

(5) $\beta_{21} = (1/6 - \gamma + \gamma^2)\cdot\alpha_{21}^2/(-\gamma^2 + \gamma/3)$

Die Konvergenzbedingung aus Satz 6.20 lautet damit

$$\alpha = |1 - 3/\gamma + 3/(2\gamma^2) - 1/(6\,\gamma^3)| < 1.$$

Durch die Wahl $\gamma = 0.4$ ist $\alpha = 0.2708$ bestimmt.

Diese Wahl impliziert außerdem, daß die Methode A-stabil ist, da die A-Stabilität i. allg. nur durch γ beeinflußt wird (siehe /9/ und /23/).

Weiterhin sind z.B. α_{21} und μ_2 frei wählbar.

Durch die Wahl $\alpha_{21} = 0.75$ und $\mu_2 = 192/1593$ sind somit die restlichen Koeffizienten bestimmt.

Wählen wir für das eingebettete Verfahren zweiter Ordnung z.B. $\hat{\mu}_3 = 0.5$, so folgen die Werte für $\hat{\mu}_1$ und $\hat{\mu}_2$ aus den Bedingungen $\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3 = 1$ und $\hat{\mu}_2\beta_2 + \hat{\mu}_3\beta_3 = 1/2 - \gamma$.

Damit ist auch $\hat{\alpha} = 0.338.. < 1$ bestimmt, obwohl für das eingebettete Verfahren nur Konsitenz gefordert wird, da es nur lokal zur Schrittweitenkontrolle dient.
Durch die Wahl $\hat{\mu}_3 = 0.5$ hat das eingebettete Verfahren wegen $\mu_3 = 0.472..$ einen kleinen Abbrechfehler. Dies führt dazu, daß der lokale Fehler der Methode nicht zu oft überschätzt wird. Außerdem werden so größere Schrittweiten vorgeschlagen, was zu einer Reduzierung der Rechenzeit führt.

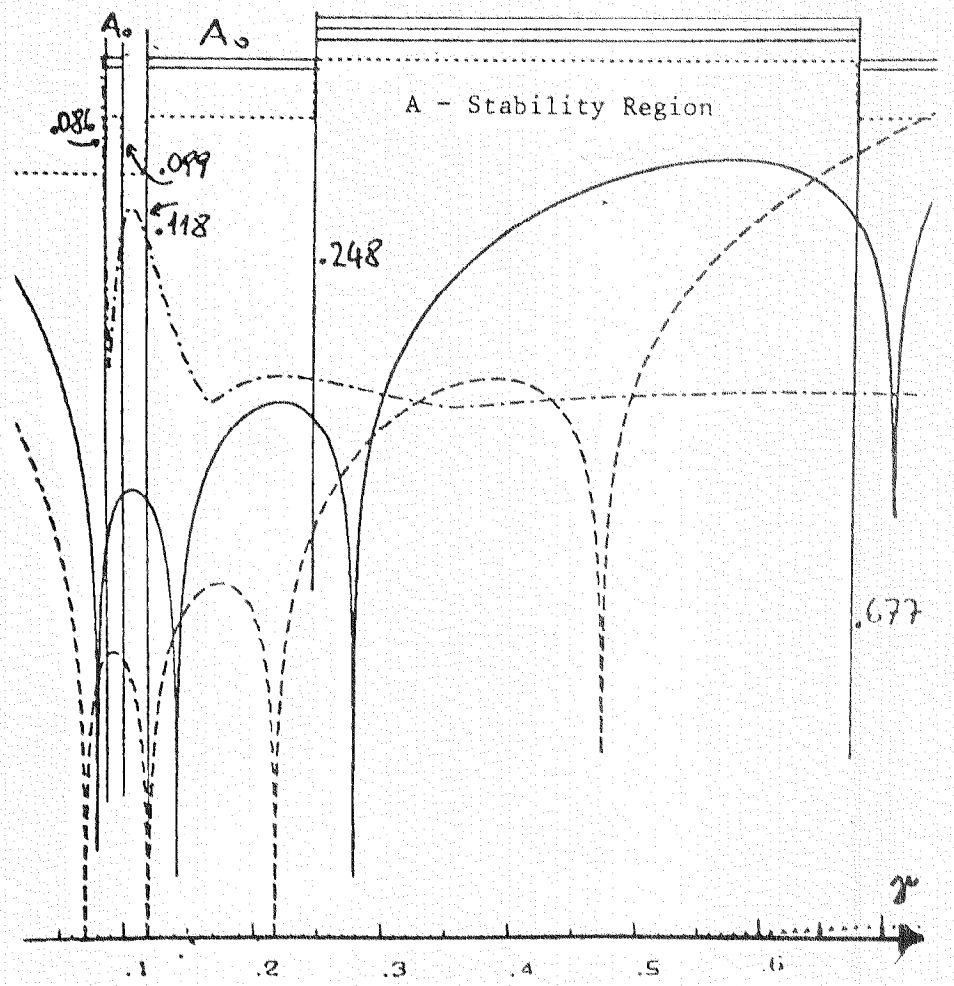
Die vollständigen Koeffizientensätze aller in diesem Abschnitt hergeleiteten Verfahren sind im Anhang 12.1 zu finden.

b) Das Verfahren DAE4S

DAE4S ist ein Verfahren der Konvergenzordnung vier mit Stufenzahl $s = 5$ und $\alpha = 0$. Das eingebettete Verfahren der Ordnung drei hat ebenfalls Stufenzahl $\hat{s} = 5$.
Der von Roche /17/ vorgeschlagene Koeffizientensatz wurde durch ein Newtonverfahren berechnet.
Wegen $\gamma = 0.707..$ ist DAE4S nicht A-stabil (siehe Skizze 1 unten).

Skizze 1 (aus /22/):

Stabilitätsbereiche eines ROW-Verfahrens mit Stufenzahl s = 5 und Ordnung p = 4 in Abhänigkeit von γ.



c) Das Verfahren DAE4SF

DAE4SF ist ebenfalls eine Methode der Konvergenzordnung vier und Stufenzahl fünf. Wie in /17/ vorgeschlagen, wurde $\alpha_{21} = \beta_{43} = 0$ gewählt. Dadurch vereinfachen sich die Ordnungsbedingungen wesentlich, und es werden nur vier Funktionsauswertungen pro Integrationsschritt benötigt.
Wir erhalten die 13 Ordnungsbedingungen:

(1) $\mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 = 1$

(2) $\mu_2\beta_2 + \mu_3\beta_3 + \mu_4\beta_4 + \mu_5\beta_5 = 1/2 - \gamma$

(3) $\mu_3\alpha_3^2 + \mu_4\alpha_4^2 + \mu_5\alpha_5^2 = 1/3$

(4) $\mu_3\beta_{32}\beta_{21} + \mu_4\beta_{42}\beta_{21} + \mu_5(\beta_{54}\beta_4 + \beta_{53}\beta_3 + \beta_{52}\beta_2) = \gamma^2 - \gamma + 1/6$

(5) $\mu_3\alpha_3^3 + \mu_4\alpha_4^3 + \mu_5\alpha_5^3 = 1/4$

(6) $\mu_3\beta_{21}\alpha_3\alpha_{32} + \mu_4\alpha_4(\alpha_{43}\beta_3 + \alpha_{42}\beta_2) + \mu_5\alpha_5(\alpha_{54}\beta_4 + \alpha_{53}\beta_3 + \alpha_{52}\beta_2) = 1/8 - \gamma/3$

(7) $\mu_5(\beta_{54}\alpha_4^2 + \beta_{53}\alpha_3^2) = 1/12 - \gamma/3$

(8) $\mu_4\alpha_3^2\alpha_{43}\alpha_4 + \mu_5\alpha_5(\alpha_{54}\alpha_4^2 + \alpha_{53}\alpha_3^2) = \gamma/4$

(9) $\mu_5(\beta_{54}\beta_{42}\beta_{21} + \beta_{53}\beta_{32}\beta_{21}) = -\gamma^3 + 3\gamma^2/2 - \gamma/2 + 1/24$

(10) $0 = \gamma^2 - 2\gamma/3 + 1/12$

(11) $\mu_5\beta_{54}\alpha_4^3 + \mu_5\beta_{53}\alpha_3^3 = -\gamma^2 + \gamma/4$

(12) $\mu_5\beta_{54}\alpha_4(\alpha_{43}\beta_3 + \alpha_{42}\beta_2) + \mu_5\beta_{53}\alpha_3\alpha_{32}\beta_2 = \gamma^3 - 5\gamma^2/6 + \gamma/8$

(13) $\mu_5\beta_{54}\alpha_4\alpha_{43}\alpha_3^2 = -\gamma^3 + \gamma^2/4$

Die Konvergenzbedingung liefert

$$\alpha = |1 - 4/\gamma + 3/\gamma^2 - 2/(3\gamma^3) + 1/(24\gamma^4)| < 1.$$

Aus (10) folgt $\gamma = 1/2$ oder $\gamma = 1/6$. Durch die Wahl $\gamma = 1/2$ folgt $\alpha = 1/3$. Außerdem erhalten wir dadurch auch ein A-stabiles Verfahren für "reine Differentialgleichungen" (siehe Skizze 1).

Wir setzen nun (×) $\beta_{32}\beta_2 = -\alpha_3^2/2$, $\beta_{42}\beta_2 = -\alpha_4^2/2$ und

(××) $1/12 + \mu_5(\beta_{54}\beta_4 + \beta_{53}\beta_3 + \beta_{52}\beta_2) = (1/3 - \alpha_5^2\,\mu_5)/2.$

Durch diese Wahl erreichen wir, daß (4) äquivalent zu (3) und (9) äquivalent zu (7) ist.

Als freie Parameter betrachten wir zunächst α_3, α_4, α_5 und β_2. Nach deren Wahl, die der Forderung aus Bemerkung 7.9 genügen sollte, ist durch (×) β_{32} und β_{42}, durch (7) und (11) $\mu_5\beta_{54}$ und $\mu_5\beta_{53}$ und schließlich durch (13) α_{43} bestimmt.

Betrachten wir weiterhin α_{53} und α_{54} als freie Parameter, so ergibt (3), (5) und (8) ein lineares Gleichungssystem zur Bestimmung von μ_3, μ_4 und μ_5:

$$\begin{pmatrix} \alpha_3^2 & \alpha_4^2 & \alpha_5^2 \\ \alpha_3^3 & \alpha_4^3 & \alpha_5^3 \\ 0 & \alpha_3^2\alpha_{43}\alpha_4 & \alpha_5(\alpha_{54}\alpha_4^2+\alpha_{53}\alpha_3^2) \end{pmatrix} \begin{pmatrix} \mu_3 \\ \mu_4 \\ \mu_5 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/4 \\ \gamma/4 \end{pmatrix}$$

Wählen wir schließlich noch α_{32}, α_{42}, β_5 und β_{52}, so ist durch (12) β_3, durch (××) β_4 und durch (6) α_{52} bestimmt. μ_2 erhalten wir aus (2) und μ_1 aus (1).
Die Koeffizienten $\hat{\mu}_1,\ldots,\hat{\mu}_5$ der eingebetteten Methode der Ordnung drei müssen den folgenden fünf linearen Gleichungen genügen:

(1) $\hat{\mu}_1+\ldots+\hat{\mu}_5 = 1$

(2) $\hat{\mu}_2\beta_2 + \hat{\mu}_3\beta_3 + \hat{\mu}_4\beta_4 + \hat{\mu}_5\beta_5 = 1/2 - \gamma$

(3) $\hat{\mu}_3\alpha_3^2 + \hat{\mu}_4\alpha_4^2 + \hat{\mu}_5\alpha_5^2 = 1/3$

(4) $\hat{\mu}_3\beta_{32}\beta_2 + \hat{\mu}_4\beta_{42}\beta_2 + \hat{\mu}_5(\beta_{54}\beta_4 + \beta_{53}\beta_3 + \beta_{52}\beta_2) = \gamma^2 - \gamma + 1/6$

(5) $\hat{\mu}_5(\beta_{54}\alpha_4^2 + \beta_{53}\alpha_3^2) = -\gamma^2 + \gamma/3$

Wegen (×) und (××) ist (4) äquivalent zu (5).
Wir können also ein $\hat{\mu}_i$ frei wählen.

Die freien Parameter der Methode DAE4SF wählten wir wie folgt:
$\alpha_3 = 1/2$, $\alpha_4 = 3/4$, $\alpha_5 = 1/2$, $\beta_2 = 1$, $\alpha_{53} = 1/4$, $\alpha_{54} = 1$, $\alpha_{32} = 1/4$, $\alpha_{42} = 1/8$, $\beta_{52} = -1/8$, $\beta_5 = 1/9$.

Mit $\hat{\mu}_3 = 0$ erhalten wir wie in a) ein eingebettetes Verfahren mit kleinem Abbrechfehler.

d) Das Verfahren DAE3NS

DAE3NS und die beiden folgenden Verfahren sind Methoden nach Ansatz B. Die Buchstaben NS in DAE3NS bedeuten, daß dieses Verfahren nicht gut für steife Differentialgleichungen geeignet ist, da der Ansatz B explizit für "reine Differentialgleichungen" ist.

DAE3NS hat Konvergenzordnung und Stufenzahl drei.

Nach Vereinfachung erhalten wir die sechs Ordnungsbedingungen:

(1) $\mu_1 + \mu_2 + \mu_3 = 1$

(2) $\mu_2\alpha_2 + \mu_3\alpha_3 = 1/2$

(3) $\mu_2\alpha_2^2 + \mu_3\alpha_3^2 = 1/3$

(4) $\mu_3\alpha_{32}\alpha_{21} = 1/6$

(5) $\alpha_{21} = 2\gamma$

(6) $\mu_3\beta_{32}\alpha_{21}^2 = -\gamma^2 + \gamma/3$

Die Konvergenzbedingung lautet

$$\alpha = |1 - 1/\gamma + (\mu_2\beta_{21} + \mu_3\beta_3)/\gamma^2 - \mu_3\beta_{32}\beta_{21}/\gamma^3| < 1.$$

Wegen Bemerkung 6.18 sind β_{21} und β_{31} frei wählbar.
Deshalb kann die Konvergenzbedingung $\alpha < 1$ für alle Methoden nach Ansatz B durch geeignete Wahl der β_{i1}, i=2,..,s immer erfüllt werden.
Wählen wir nun $\beta_{32} = 0$, so folgt aus (6) $\gamma = 1/3$.
Durch (5) ist α_{21} bestimmt und mit der Wahl $\alpha_{32} = 1$ folgt aus (4) $\mu_3 = 1/4$. μ_2 und α_3 sind dann durch (2) und (3) bestimmt, und μ_1 erhalten wir aus (1).
Wählen wir weiter $\beta_{21} = \beta_{31} = 8/27$, so gilt $\alpha = 0$.
Die Koeffizienten der eingebetteten Methode der Ordnung zwei müssen Gleichung (1) und (2) erfüllen. Daher kann z.B. $\hat{\mu}_3 = 0$ gewählt werden, und wir erhalten somit die Stufenzahl $\hat{s} = 2$ für die eingebettete Methode.

e) Das Verfahren DAE33NS

Es handelt sich um ein Verfahren der Konvergenzordnung drei und Stufenzahl s = 4. Die Konsistenzordnung ist jedoch 3 3 (siehe Bemerkung 7.8). Das eingebettete Verfahren hat Stufenzahl $\hat{s} = 4$ und Ordnung 2 2. Da auf diese Weise der Abbrechfehler der eingebetteten Methode in den z-Komponenten gegenüber dem Verfahren DAE3NS reduziert wird, hoffen wir, daß diese Methode mit größeren Schrittweiten als DAE3NS arbeiten kann. Diese Vermutung wird durch die numerischen Resultate in §8 und §9 bestätigt.
Für die Methode der Ordnung 3 3 sind die folgenden neun Ordnungsbedingungen zu erfüllen:

(1) $\mu_1 + \mu_2 + \mu_3 + \mu_4 = 1$

(2) $\mu_2\alpha_2 + \mu_3\alpha_3 + \mu_4\alpha_4 = 1/2$

(3) $\mu_2\alpha_2^2 + \mu_3\alpha_3^2 + \mu_4\alpha_4^2 = 1/3$

(4) $\mu_3\alpha_{32}\alpha_2 + \mu_4\alpha_{42}\alpha_2 + \mu_4\alpha_{43}\alpha_3 = 1/6$

(5) $\mu_3\alpha_{32}\alpha_2^2 + \mu_4\alpha_{42}\alpha_2^2 + \mu_4\alpha_{43}\alpha_3^2 + \mu_4\gamma w_{32}\alpha_2^2 = \gamma/3$

(6) $\mu_3 w_{32}\alpha_2^2 + \mu_4 w_{42}\alpha_2^2 + \mu_4 w_{43}\alpha_3^2 = 1 - 1/(3\gamma)$

(7) $\mu_2\alpha_2^3 + \mu_3\alpha_3^3 + \mu_4\alpha_4^3 + \gamma(\mu_3 w_{32}\alpha_2^3 + \mu_4 w_{42}\alpha_2^3 + \mu_4 w_{43}\alpha_3^3) = \gamma$

(8) $\mu_3\alpha_3\alpha_{32}\alpha_2 + \mu_4\alpha_4(\alpha_{43}\alpha_3 + \alpha_{42}\alpha_2) + \gamma\mu_4 w_{43}\alpha_3\alpha_{32}\alpha_2 = \gamma/2$

(9) $\mu_3\alpha_3\alpha_{32}\alpha_2^2 + \mu_4\alpha_4(\alpha_{43}\alpha_3^2 + \alpha_{42}\alpha_2^2) + \gamma\mu_4 w_{43}\alpha_3\alpha_{32}\alpha_2^2$
$+ \gamma\mu_4\alpha_{43} w_{32}\alpha_2^2 = \gamma^2$

Gleichung (9) ist äquivalent zu (8), falls wir (×) $\alpha_2 = 2\gamma$ und (××) $\alpha_4\alpha_3\alpha_2 = \alpha_4\alpha_3^2 + \gamma w_{32}\alpha_2^2$ wählen.

Als freie Parameter wählen wir α_2, α_3, α_4 und w_{43}.
(×) und (××) liefern dann die Werte γ und w_{32}.
Einsetzen von α_2·(6) in (7) ergibt zusammen mit (2) und (3) ein lineares Gleichungssystem zur Bestimmung von μ_2, μ_3 und μ_4:

$$\begin{pmatrix} \alpha_2 & \alpha_3 & \alpha_4 \\ \alpha_2^2 & \alpha_3^2 & \alpha_4^2 \\ \alpha_2^3 & \alpha_3^3 & \alpha_4^3 + w_{43}\gamma\alpha_3^2(\alpha_3-2\gamma) \end{pmatrix} \begin{pmatrix} \mu_2 \\ \mu_3 \\ \mu_4 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/3 \\ -2\gamma^2 + 5\gamma/3 \end{pmatrix}$$

w_{42} folgt nun aus (6) und μ_1 aus (1).
(4), (5) und (8) liefern ein lineares Gleichungssystem zur Bestimmung von α_{32}, α_{42} und α_{43}:

$$\begin{pmatrix} \mu_3\alpha_2 & \mu_4\alpha_2 & \mu_4\alpha_3 \\ \mu_3\alpha_2^2 & \mu_4\alpha_2^2 & \mu_4\alpha_3^2 \\ \mu_3\alpha_3\alpha_2 + \gamma\mu_4 w_{43}\alpha_3\alpha_2 & \mu_4\alpha_4\alpha_2 & \mu_4\alpha_4\alpha_3 \end{pmatrix} \begin{pmatrix} \alpha_{32} \\ \alpha_{42} \\ \alpha_{43} \end{pmatrix} = \begin{pmatrix} 1/6 \\ \gamma/3-\mu_4\gamma w_{32}\alpha_2^2 \\ \gamma/2 \end{pmatrix}$$

Durch die Wahl $\alpha_{21} = 1/2$, $\alpha_3 = 3/4$, $\alpha_4 = 1$ und $w_{43} = 20/9$ sind also alle Koeffizienten bis auf β_{i1}, i=2,3,4 durch die neun Bedingungen eindeutig bestimmt.
Die Wahl $\beta_2 = \beta_3 = 0$, $\beta_4 = 39/64$ impliziert
$\alpha = |1 - 1/\gamma + \mu_4\beta_4/\gamma^2| = 0$.
Die Koeffizienten $\hat{\mu}_1,\ldots,\hat{\mu}_4$ der eingebetteten Methode müssen den Ordnungsbedingungen $(\hat{1})$, $(\hat{2})$ und $(\hat{3})$ genügen:

$(\hat{1})$ $\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3 + \hat{\mu}_4 = 1$

$(\hat{2})$ $\hat{\mu}_2\alpha_2 + \hat{\mu}_3\alpha_3 + \hat{\mu}_4\alpha_4 = 1/2$

$(\hat{3})$ $\hat{\mu}_2\alpha_2^2 + \hat{\mu}_3\alpha_3^2 + \hat{\mu}_4\alpha_4^2 + \gamma(\hat{\mu}_3 w_{32}\alpha_2^2 + \hat{\mu}_4 w_{42}\alpha_2^2 + \hat{\mu}_4 w_{43}\alpha_3^2) = \gamma$

Durch obige Wahl ist hier (2) und (3) äquivalent, so daß wir zwei Koeffizienten frei wählen können.
Setzen wir $\hat{\mu}_4 = \mu_4$, so ist auch die eingebettete Methode konvergent mit $\hat{\alpha} = 0$. Weiterhin wählten wir $\hat{\mu}_1 = 0$.

f) Das Verfahren DAE34NS

Zur Herleitung von DAE34NS gehen wir den folgenden Weg:
Da die Ordnungsbedingungen, die zu Bäumen mit ausschließlich dünnen Knoten gehören, mit denen der expliziten Runge-Kutta Methoden übereinstimmen (siehe Proposition 6.16), können wir einen Koeffizientensatz einer schon bekannten Runge-Kutta Methode übernehmen. Dadurch sind die Koeffizienten μ_i, $\hat{\mu}_i$, α_{ij} für $j=1,..,i-1$ und $i=1,..,s$ bestimmt. Die noch zu bestimmenden Koeffizienten γ, γ_{ij}, $j=1,..,i-1$, $i=2,..,s$ werden so gewählt, daß die Ordnungsbedingungen, die zu Bäumen gehören, in denen dicke Knoten vorkommen, erfüllt sind.
Wir übernehmen nun den von Fehlberg vorgeschlagenen Koeffizientensatz RKF4 (siehe /4/, Methode RK4).
RKF4 ist ein explizites Runge-Kutta Verfahren der Ordnung vier mit Stufenzahl $s = 5$. Das Besondere ist, daß diese Methode einen sehr kleinen Abbrechfehler hat und in eine Methode der Ordnung fünf mit Stufenzahl $\hat{s} = 6$ eingebettet ist. Die Methode der Ordnung fünf wird also ausschließlich zur Schrittweitenkontrolle verwendet.
Der Vorteil, daß die Lösung durch die eingebettete Methode der Ordnung vier bestimmt wird, liegt darin, daß EST auf diese Weise eine gute Schätzung für den lokalen Fehler lie-

fert. Weiterhin können durch den kleinen Abbrechfehler größere Schrittweiten vorgeschlagen werden.
Wir übernehmen nun dieses Konzept zur Konstruktion einer Methode der Ordnung 3(4), d.h. eine Methode der Ordnung drei ist eingebettet in eine Methode der Ordnung vier, die nur zur Schrittweitensteuerung dient.
Wird die so entstehende Methode DAE34NS also auf Index-Eins Probleme angewendet, so hat sie die Konvergenzordnung drei, wird sie auf reine Differentialgleichungen angewendet, so hat sie die Konvergenzordnung vier.

Wir konstruieren zuerst die Methode der Ordnung vier mit $\hat{s} = 6$, die zur Schrittweitensteuerung dient:
Es sind insgesamt 18 Ordnungsbedingungen zu erfüllen (siehe (6.17a) bis (6.17r)). Durch Übernahme der Koeffizienten $\hat{\mu}_i$, α_{ij} für i=1,..,6 aus /4/ sind bereits die acht Bedingungen, die zu Bäumen mit ausschließlich dünnen Knoten gehören, erfüllt. Setzen wir nun

$$(7.10) \qquad \sum_{j=2}^{i} w_{ij} \alpha_j^2 = 2\alpha_i \quad \text{für } i=2,\ldots,6,$$

so fallen weitere sechs Bedingungen weg, denn:

 reduziert sich auf  wegen $1 = \sum \hat{\mu}_i w_{ij} \alpha_j^2 = 2\sum \hat{\mu}_i \alpha_i$,

 reduziert sich auf  ,

 reduziert sich auf  .

 reduziert sich auf  ,

 reduziert sich auf  ,

 reduziert sich auf  .

Es bleiben damit noch vier Ordnungsbedingungen, die zu den folgenden Bäumen gehören:

(7.11)  .

Diese vier Bedingungen, zusammen mit den fünf Gleichungen (7.10) ergeben ein lineares Gleichungssystem der Dimension neun mit den elf Unbekannten γ, w_{32}, w_{42}, w_{43}, w_{52}, w_{53}, w_{54}, w_{62}, w_{63}, w_{64}, w_{65}.

Die verbleibenden zwei Freiheitsgrade wollen wir nutzen, den Abbrechfehler des eingebetteten Verfahrens klein zu halten.

Die Koeffizienten der eingebetteten Methode müssen insgesamt sechs Ordnungsbedingungen erfüllen. Davon sind die vier Bedingungen, die zu den Bäumen mit ausschließlich dünnen Knoten gehören, trivialerweise erfüllt.

Die Bäume  bzw.  reduzieren sich wegen (7.10) jedoch auf  bzw.  . Damit ist das eingebettete Verfahren von Ordnung drei.

Betrachten wir nun dessen Abbrechfehler:

Dieser wird durch insgesamt zwölf Bäume bestimmt.

Da vier dieser Bäume ausschließlich dünne Knoten haben,

ist der dadurch erzeugte Abbrechfehler klein (siehe /4/). Vier weitere Bäume lassen sich wegen (7.10) auf andere reduzieren, so daß der Abbrechfehler letztendlich durch die vier Bäume (7.11) bestimmt wird.

Wir bezeichnen diese Abbrechfehler analog zu /4/ mit $T_1,..,T_4$, d.h.: $T_1 = 1 - \sum \mu_i w_{ij} \alpha_j^3$

$$T_2 = 1 - 4\sum \mu_i \alpha_{ij} w_{jk} \alpha_k^3$$

$$T_3 = 1 - 2\sum \mu_i \alpha_i \alpha_{ij} \alpha_j$$

$$T_4 = 1 - 8\sum \mu_i \alpha_{ij} w_{jk} \alpha_k \alpha_{kl} \alpha_l \ .$$

Weil in /4/ die Beziehung $2\sum_{k=2}^{j-1} \alpha_{jk}\alpha_k = \alpha_j^2$ für $j=3,..,6$ gilt, erhalten wir beispielsweise durch $w_{64} = 0$ und $\mu_3 w_{32} + \mu_4 w_{42} + \mu_5 w_{52} = 0$ eine gute Wahl für die zwei verbleibenden Freiheitsgrade. Damit ergibt sich:

$T_1 = 0.125$

$T_2 = 0.0896...$

$T_3 = 0.125$

$T_4 = 0.0384...$.

8. Testbeispiele

8.1. Konstruierte Testbeispiele

Zun Testen der entwickelten Verfahren betrachten wir zunächst drei kleinere Beispiele.

Wir verwenden in diesem Paragraphen die folgenden Abkürzungen:

Letzte I.Zeit: Zeitpunkt, zu dem das numerische Verfahren stoppte, weil ein Fehler aufgetreten ist, oder weil Letzte I.Zeit = TEND gilt.

CPU-Zeit: In Sekunden gemessene CPU-Zeit der Siemens 7570 Rechenanlage des Regionalen Rechenzentrums Kaiserslautern.

Erf. Schritte: Anzahl der erfolgreich durchgeführten Integrationsschritte.

Nicht erf. S.: Anzahl der Schritte, die verworfen wurden, weil der Fehlertest EST $\leq$ TOL nicht erfüllt war.

FCN-Aufr.ges.: Gesamtanzahl der Funktionsaufrufe von f und g, einschließlich derer zur numerischen Differentiation durch Berechnung der Differenzenquotienten. Zu beachten ist, daß die Verfahren nach Ansatz B nicht die ganze rechte Seite, sondern nur die algebraische Nebenbedingung g auswerten müssen.

FCN-Aufr.Ver.: Anzahl der rechte Seite-Aufrufe (f,g) des Verfahrens, ausschließlich derer zur numerischen Differentiation.

abs.Fehler (Letzte Z.) : absoluter Fehler der numerischen Lösung zum letzten Integrationszeitpunkt des Verfahrens.

Als Eingangsschrittweite wurde in den Beispielen dieses Abschnitts immer HI = 10^{-3} gewählt.

Beispiel 1:

$$y' = z \qquad , y(0) = 0$$
$$0 = y^2 + z^2 - 1 \quad , z(0) = 1$$

Die exakte Lösung des Problems lautet $y(x) = \sin(x)$

$z(x) = \cos(x)$.

Es gilt $\partial g/\partial z\ (y,z) = 0$ genau dann, wenn $z = 0$.
Es handelt sich daher um ein Index-Eins Problem, falls $x \leq c < \pi/2$. In $x = \pi/2$ tritt eine Singularität auf.

In jedem Integrationsschritt ist bei Integration mit einer Methode nach Ansatz A die Matrix

$$E := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} h\frac{\partial f}{\partial y} & h\frac{\partial f}{\partial z} \\ \frac{\partial g}{\partial y} & \frac{\partial g}{\partial z} \end{pmatrix} = \begin{pmatrix} 1 & -h \\ -2y & -2z \end{pmatrix}$$

zu invertieren.
Bei Integration mit sehr kleiner Schrittweite h ist diese Matrix in der Umgebung von $x = \pi/2$ schlecht konditioniert. Daher wird man erwarten, daß der Fehlertest EST $\leq$ TOL nicht mehr erfüllt ist. Dies hat jedoch zur Konsequenz, daß die

Schrittweite weiter verkleinert wird. h wird also sehr schnell so weit reduziert, daß $h \leq h_{min}$ gilt und das Programm stoppt. Auf diese Weise erhalten wir den Hinweis, daß die zu invertierende Matrix schlecht konditioniert bzw. singulär ist. Die Ursache dafür ist in der Singularität von $(\partial g/\partial z)$ zu suchen, was wiederum bedeutet, daß unser Problem nicht mehr vom Typ Index-Eins ist.

Bei Integration mit entsprechender Schrittweite kann es jedoch durchaus passieren, daß über eine solche Singularität hinwegintegriert wird, ohne daß das Programm dies bemerkt.

Zum Lösen des Problems mit einer Methode nach Ansatz B ist die Matrix $(\partial g/\partial z)$ zu invertieren. Die oben genannten Probleme können daher in gleicher Weise auftreten.

Um zu prüfen, wie die Verfahren auf solche Singularitäten reagieren, wurd integriert bis

a) TEND = 1

b) TEND = $\pi/2$, wobei $\pi/2$ auf 10 Stellen gerundet wurde,
d.h. TEND = 1.570796327

c) TEND = 2.

Die vollständigen numerischen Resultate aller hier und in den folgenden Beispielen behandelten Probleme sind in /21/ zu finden.

Wir geben hier nur einige repräsentative Auszüge wider:

a) TEND = 1

Verfahren	DAE4S	DAE4SF	DAE3S	DAE34NS	DAE33NS	DAE3NS	DASSL
TOL	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4
CPU-Zeit	0.0274	0.0146	0.0155	0.0150	0.0114	0.0117	0.0267
Letzte I.Zeit	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Erf. Schritte	21	17	28	17	19	22	21
Nicht erf. S.	14	1	0	0	0	0	0
FCN-Aufr.ges.	203	105	112	136	114	110	54
FCN Aufr.Ver.	161	71	56	102	76	66	34
Abs. Fehler y (letzte Z.)	1.0E-4	4.6E-5	9.3E-6	7.3E-5	1.1E-5	6.5E-6	1.1E-4
z	1.5E-4	3.7E-4	6.3E-6	9.7E-5	9.5E-6	5.4E-6	1.5E-4

b) TEND = 1.570796327

Verfahren	DAE4S	DAE4SF	DAE3S	DAE34NS	DAE33NS	DAE3NS	DASSL
TOL	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4
Letzte I.Zeit	1.5682	TEND	TEND	1.5624	TEND	TEND	TEND
CPU-Zeit	0.1053	0.0250	0.0255	0.1015	0.0125	0.0135	0.0312
Erf. Schritte	48	25	46	39	23	28	25
Nicht erf. S.	90	8	2	108	0	0	0
FCN-Aufr.ges.	699	174	186	855	138	140	66
FCN-Aufr.Ver.	600	124	94	775	92	84	44
Abs. Fehler y (letzte Z.)	2.5E-6	2.5E-5	9.9E-6	6.6E-7	2.6E-6	8.6E-7	6.8E-8
z	2.5E-4	3.6E-3	1.1E-3	4.5E-4	7.2E-5	3.3E-5	4.3E-4

c) TEND = 2

Verfahren	DAE4S	DAE4SF	DAE3S	DAE3ANS	DAE3BNS	DAE3NS	DASSL
TOL	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4
Letzte I.Zeit	1.5682		1.5720	1.5713	1.5653	2.0000	2.0000
CPU-Zeit	0.1041		0.0452	0.0725	0.0724	0.0210	0.0392
Erf. Schritte	48		46	35	23	40	29
Nicht erf. S.	90		47	63	143	6	1
FCN-Aufr.ges.	699		234	598	570	212	88
FCN-Aufr.Ver.	600		139	526	522	132	58
Abs. Fehler y (letzte Z.)	2.5E-6		9.8E-6	2.9E-5	3.4E-6	1.6E-4	1.9E-4
z	2.5E-4		9.8E-4	8.7E-3	2.1E-4	3.7E-4	4.2E-4

Das Verfahren DAE4SF scheiterte an Beispiel 1c) wegen einem Exponent-Overflow.

In allen Beispielen, in denen Letzte I.Zeit ≠ TEND gilt, erfolgte ein Programmstop, weil sich die Schrittweite sehr schnell so weit reduziert hatte, daß sie kleiner als h_{min} war.

Obige Resultate stimmen also gut mit den zu Beginn des Abschnitts beschriebenen Erwartungen überein.

Die Genauigkeit bzgl. der vorgegebenen Toleranz TOL ist aufgrund der Singularität in $\pi/2$ nicht immer ausreichend.

Ansonsten sind hier noch keine gravierenden Unterschiede der einzelnen Methoden zu erkennen.

Beispiel 2:

$$y_1' = 0.5\ z\ y_2^3 \quad , \ y_1(o) = 1$$
$$y_2' = y_2\ z/6 \quad , \ y_2(0) = 1$$
$$0 = z + 6y_1/y_2^3 \quad , \ z(0) = -6$$

Die exakte Lösung lautet $y_1(x) = \exp(-3x)$

$y_2(x) = \exp(-x)$

$z(x) = -6$.

Es handelt sich um ein Index-Eins Problem, da $(\partial g/\partial z) = 1$ für alle $x \in R$.

Es wurde integriert bis a) TEND = 0.5

b) TEND = 2.

Die Resultate im Fall a) entsprechen unseren Erwartungen. Es zeigt sich, daß die Einschrittverfahren dem Mehrschrittverfahren DASSL in der Genauigkeit etwas überlegen sind. Bzgl. der Rechenzeit sind sie bei den Toleranzen TOL = 10^{-2} und TOL = 10^{-4} ebenfalls vorzuziehen. Außerdem zeigt sich, daß die Methoden nach Ansatz A mehr Rechenzeit benötigen als die vergleichbaren Methoden nach Ansatz B.

Im Fall b) treten jedoch einige Schwierigkeiten auf:

Verfahren		DAE4S	DAE4SF	DAE3S	DAE3ANS	DAE3BNS	DAE3NS	DASSL
TOL		1.E-2	1.E-2	1.E-2	1.E-2	1.E-2	1.E-2	1.E-2
Letzte I.Zeit		1.9273	2.0000	1.9364	2.0000	2.0000	2.0000	2.0000
CPU-Zeit		0.1505	0.0291	0.0520	0.0225	0.0207	0.0197	0.0444
Erf. Schritte		67	21	31	20	28	29	23
Nicht erf. S.		45	0	20	0	0	1	0
FCN-Aufr.ges.		720	147	179	180	196	176	95
FCN-Aufr.Ver.		515	84	82	120	112	89	59
Abs.Fehler (letzte Z.)	y_1	1.6E-4	1.3E+8	2.3E-7	3.4E-4	8.3E-5	5.3E-5	3.4E-4
	y_2	1.4E-1	3.5E+3	1.4E-1	1.4E-2	3.4E-4	1.6E-4	1.7E-1
	z	8.8E+4	4.0E+4	1.4E+5	1.2E+0	2.5E-1	9.9E-2	5.5E+0

Verfahren		DAE4S	DAE4SF	DAE3S	DAE3ANS	DAE3BNS	DAE3NS	DASSL
TOL		1.E-6	1.E-6	1.E-6	1.E-6	1.E-6	1.E-6	1.E-6
Letzte I.Zeit		2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
CPU-Zeit		0.2929	0.2329	0.7721	0.1790	0.2492	0.2403	0.1503
Erf. Schritte		223	187	894	179	370	413	76
Nicht erf. S.		0	0	0	0	0	0	2
FCN-Aufr.ges.		1784	1309	4470	1611	2590	2478	327
FCN-Aufr.Ver.		1115	748	1788	1074	1480	1239	207
Abs.Fehler (letzte Z.)	y_1	5.4E-9	1.0E-9	3.E-10	2.9E-8	8.6E-9	4.8E-9	7.7E-8
	y_2	6.5E-6	1.2E-5	2.5E-7	5.4E-7	3.3E-9	4.7E-7	3.0E-5
	z	8.8E-4	1.6E-3	3.4E-5	3.8E-7	2.1E-5	5.1E-5	4.2E-3

Die Methoden nach Ansatz A versagen im Fall TOL = 10^{-2}.

Auch für TOL = 10^{-6} ist die Genauigkeit, besonders in der

z-Komponente, nicht immer ausreichend.

Die CPU-Zeit von DAE34NS und DASSL ist vergleichbar, obwohl DASSL erheblich weniger Funktionsauswertungen benötigt.

Jedoch ist DAE34NS aufgrund der guten Genauigkeit vorzuziehen.

Beispiel 3:

$$
\begin{aligned}
y_1' &= z_1 && , \; y_1(0) = 0 \\
y_2' &= -0.5\, z_2^{1/4} && , \; y_2(0) = 1 \\
0 &= y_1^2 + z_1^2 - y_2^4 / z_2 && , \; z_1(0) = 1 \\
0 &= z_2 - y_2^4 && , \; z_2(0) = 1
\end{aligned}
$$

Die exakte Lösung lautet

$$
\begin{aligned}
y_1(x) &= \sin(x) \\
y_2(x) &= \exp(-0.5x) \\
z_1(x) &= \cos(x) \\
z_2(x) &= \exp(-2x) \quad .
\end{aligned}
$$

Es handelt sich um ein Index-Eins Problem für $x \leq c < \pi/2$, da $(\partial g/\partial z)\ (x=\pi/2) = 0$.

Wir integrierten das Problem bis a) TEND = 1

b) TEND = $\pi/2$, wobei $\pi/2$ wie in Beispiel 1b) auf 10 Stellen gerundet wurde.

b) TEND = 1.570796327

Verfahren		DAE4S	DAE4SF	DAE3S	DAE34NS	DAE33NS	DAE3NS	DASSL
TOL		1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4
Letzte I.Zeit		0.5078	TEND	TEND	1.5303	TEND	TEND	TEND
CPU-Zeit		0.3556	0.0619	0.0700	0.1461	0.0650	0.0575	0.0674
Erf. Schritte		14	26	44	28	49	53	31
Nicht erf. S.		186	3	4	62	1	1	0
FCN-Aufr.ges.		875	217	268	595	395	373	120
FCN-Aufr.Ver.		814	113	92	479	199	161	76
Abs.Fehler (letzte Z.)	y_1	1.5E-4	3.3E-5	1.3E-5	8.6E-5	2.8E-7	4.1E-7	7.9E-7
	y_2	1.7E-5	1.4E-6	6.4E-6	3.0E-5	3.6E-6	1.6E-6	1.3E-5
	z_1	2.5E-3	4.0E-3	9.8E-4	1.4E-4	2.3E-4	1.6E-4	8.7E-4
	z_2	4.9E-4	1.9E-6	2.5E-6	1.2E-5	1.4E-6	5.7E-7	6.0E-6

DAE4S versagt bei TOL = 10^{-2} und TOL = 10^{-4} sowohl im Fall a) wie in b).

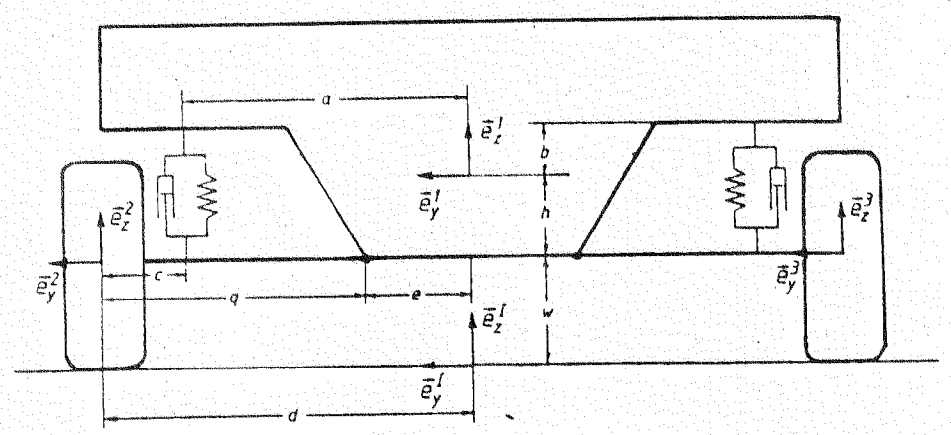
Wie schon in Beispiel 1 können wir erkennen, daß DAE34NS empfindlich auf Singularitäten reagiert.

Die Methoden DAE3NS und DAE33NS liefern in allen Beispielen gleich gute Resultate. Der Mehraufwand zur Herleitung von DAE33NS hat sich also nicht bezahlt gemacht.

8.2. Beispiel Fahrzeugachse

Um in der Fahrzeugentwicklung schon frühzeitig Aussagen über das Fahrverhalten und während des Betriebs auftretende Kräfte machen zu können, werden in der Automobilindustrie mechanische Systeme oftmals simuliert.
In /18/ wurde ein Lastfall an einem einfachen Modell einer Fahrzeughinterachse gerechnet, der dem Einfahren in eine Kurve entspricht.
Die Achse (siehe Skizze 2 unten) ist aus drei Massen aufgebaut: Fahrzeugaufbau und zwei Räder, die durch Federn, Dämpfer und Gelenke miteinander verbunden sind. Die Reifen werden durch ein System aus horizontalen und vertikalen Federn und Dämpfer modelliert.

Skizze 2 (aus /18/):
Einfaches Modell einer Fahrzeugachse.



Die Modellierung dieses Problems führt auf ein nichtlineares DAE-System der Form

$$M(q)\ddot{q} = f(q,\dot{q},t) + G(q)\lambda \quad , \quad q(o) = q_o \tag{8.1a}$$

$$0 = \Phi(q) \quad , \quad \lambda(0) = \lambda_o \; . \tag{8.1b}$$

Die algebraische Nebenbedingung (8.1b) repräsentiert die Zwangskräfte an das mechanische System.
Wie wir bereits in §2 gesehen haben, handelt es sich um ein Index-Drei Problem.
Um (8.1) numerisch sinnvoll bearbeiten zu können, müssen wir das Problem in ein Index-Eins oder Index-Null Problem transformieren. Dazu differenzieren wir die algebraische Nebenbedingung (8.1b) zweimal und erhalten

$$M(q)\ddot{q} = f(q,\dot{q},t) + G(q)\lambda \tag{8.2a}$$

$$0 = \ddot{\Phi}(q) = G^t\ddot{q} + (D_q G^t)\dot{q} \; . \tag{8.2b}$$

Gleichung (8.2b) ist äquivalent zu (8.1b), wenn die Anfangswerte den Bedingungen $\Phi(t=0) = 0$ und $\dot{\Phi}(t=0) = 0$ genügen.
Nun gehen wir von einer Ruhelage des Systems aus, d.h. $\dot{q}(0) = 0$. Außerdem wurden konsistente Anfangswerte $\Phi(q_o) = 0$ gewählt. Daher sind die oben genannten Bedingungen an die Anfangswerte implizit erfüllt.
Zu beachten ist noch, daß auch λ_o konsistent gewählt werden muß, d.h. (8.2b) muß für $t = 0$ erfüllt sein.
Somit erhalten wir ein Index-Eins Problem, das numerisch bearbeitet werden kann.

Bemerkung 8.3:

Wie wir wissen, erhält man ein Index-Null Problem, d.h. ein gewöhnliches AWP, falls (8.2b) nochmals differenziert und nach $\dot{\lambda}$ aufgelöst wird.
Das ist jedoch nur möglich, falls λ wie in (8.1) linear vorkommt. Außerdem wird das entstehende Differentialgleichungssystem unnötig kompliziert, so daß es vorteilhafter ist, das Index-Eins Problem (8.2) zu lösen.

Nehmen wir an, daß die Wahl der konsistenten Anfangsbedingungen und die Wahl der Ruhelage mit kleinen Fehlern verbunden ist, d.h. es gilt $\Phi(t=0) = \delta$ und $\dot{\Phi}(t=0) = \varepsilon$.
Aus (8.2b) folgt $\Phi(t) = \varepsilon\, t + \delta$, d.h. die Gleichung $\Phi(q) = 0$ ist *linear instabil*, da der Fehler in Φ linear mit der Zeit t wächst.
Wir führen nun sogenannte *Baumgarte-Koeffizienten* (siehe /1/) α und β ein und ersetzen (8.2b) durch die äquivalente Gleichung

(8.4b) $\quad 0 = \ddot{\Phi} + \alpha\, \dot{\Phi} + \beta\, \Phi \quad .$

Natürlich ändert sich dadurch der Index des Systems nicht.
Die Eigenwerte der Differentialgleichung (8.4b) sind gegeben durch $\lambda_{1/2} = \pm\left((\alpha/2)^2 - \beta \right)^{1/2} - \alpha/2$.
Setzen wir $\beta = (\alpha/2)^2$, $\alpha > 0$, so ist (8.4b) *asymptotisch stabil*.

Insgesamt erhalten wir nach der Transformation des Problems in ein autonomes System erster Ordnung ein Index-Eins System der Form

$$\text{(8.5a)} \quad \dot{y} = f(y,z) \;,\; y(0) = y_0 \qquad \text{mit} \qquad y = \begin{pmatrix} t \\ q \\ \dot{q} \end{pmatrix} \;,\; z = \lambda$$

$$\text{(8.5b)} \quad 0 = g(y,z) \;,\; z(0) = z_0$$

Unter den insgesamt 23 Gleichungen sind vier algebraische Nebenbedingungen. Das FORTRAN-Unterprogramm FACHSE zur Berechnung von f und g ist in /21/ zu finden.
Das System wurde mit TOL = 10^{-2}, TOL = 10^{-4} und TOL = 10^{-6} bis zum Zeitpunkt TEND = 30 integriert.
Als Baumgarte-Koeffizienten wurde $\alpha = 10$ und $\beta = 25$ gewählt.
Aus dem in /18/ (siehe auch Skizze 3 und 4 unten) dargestellten Lösungsverhalten und den dort berechneten Eigenwerten der Jacobi-Matrix des Systems, die bis zu den Werten $-8 \pm 75i$ oszillieren, erkennt man, daß es sich um ein steifes DAE-System handelt. Dies spiegelt sich gut in den mit den expliziten Verfahren nach Ansatz B gewonnenen Ergebnissen wider:

Verfahren	DAE4S	DAE4SF	DAE3S	DAE3ANS	DAE3BNS	DAE3NS	DASSL
TOL	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4	1.E-4
Letzte I.Zeit	30.000	30.000	30.000	30.000	30.000	30.000	30.000
CPU-Zeit	5.765	2.220	3.767	24.537	25.294	23.112	6.429
Erf. Schritte	115	49	91	904	1244	1193	558
Nicht erf. S.	20	4	16	542	349	308	15
FCN-Aufr.ges.	3300	1335	2291	28926	34635	31634	1909
FCN-Aufr.Ver.	655	208	198	8134	6023	4195	1161

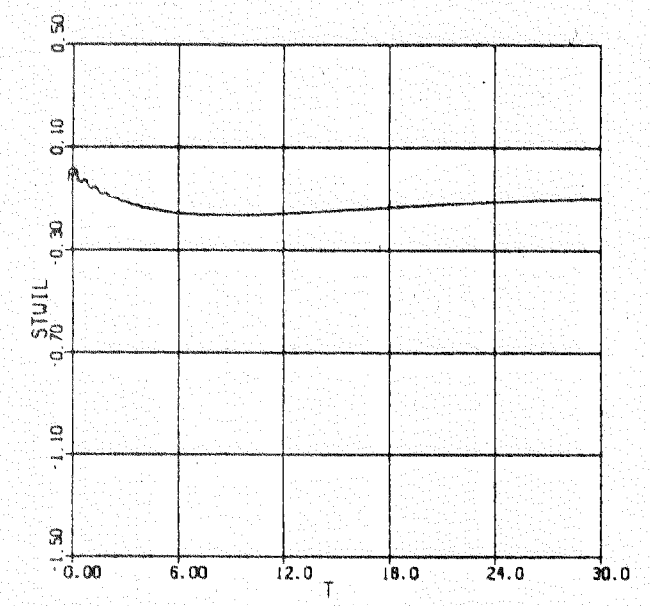
Alle anderen Resultate sowie einige Werte zum Zeitpunkt $t = 30$ sind in /21/ zu finden.

Die berechneten Kräfte, Sturzwinkel und Aufbauverschiebungen sind gut mit den in /18/ angegebenen zu vergleichen.

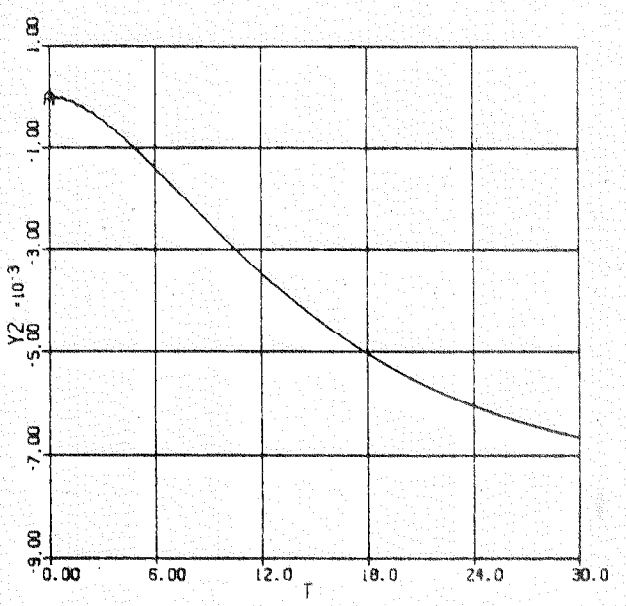
Wie wir erkennen können, arbeiten die impliziten Verfahren nach Ansatz A, insbesondere DAE4SF für die Toleranzen $TOL = 10^{-2}$ und $TOL = 10^{-4}$ sehr gut.

In den Skizzen 3 und 4 stellen wir zwei typische Zeitverläufe der Zustandsvariablen dar:

Skizze 3 (aus /18/):
Sturzwinkel Rad links: ϕ_2



Skizze 4 (aus /18/):
Aufbauverschiebung in z-Richtung: z_1



9. Probleme in Nicht-Normalform

9.1. Erweiterung der Problemklasse

Bisher waren wir in der Anwendung der in §6 und §7 entwickelten Verfahren auf Index-Eins Probleme in der Normalform

$$(9.1)\quad \begin{aligned} y' &= f(y,z) \\ 0 &= g(y,z) \qquad \text{beschränkt.} \end{aligned}$$

Wir erweitern nun die zulässige Problemklasse auf Index-Eins Probleme der Form

$$(9.2)\quad A\,y' = f(y)\,,\ y(x_0) = y_0\,,$$

wobei die (n,n)-Matrix A singulär ist.
In dieser Form ist keine explizite Trennung der gekoppelten Differentialgleichungen und algebraischen Nebenbedingungen gegeben. Diese Trennung können wir jedoch sehr einfach durch eine Transformation herbeiführen. Dazu wenden wir die *Singulärwertzerlegung* (SWZ) nach Golub und Reinsch /7/ auf A an und erhalten:

$$(9.3)\quad U\,A\,V\,V^{-1}y' = U\,f(V\,V^{-1}y)\,,\ V^{-1}y(x_0) = V^{-1}y_0.$$

U und V sind reguläre Householdermatrizen, und es gilt

$$U\,A\,V = \left(\begin{array}{c|c} D & 0 \\ \hline 0 & 0 \end{array}\right) = \left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right),\quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r\,.$$

Die numerische Rangbestimmung von A kann auf diese Weise sehr einfach über die Anzahl r der $\sigma_i \neq 0$ bzw. $\sigma_i \geq c$, wobei c von der relativen Maschinengenauigkeit abhängt, erfolgen.

Wir bezeichnen nun die ersten r Komponenten von $V^{-1}y$ mit $\tilde{y}$, die Komponenten r+1,..,n mit $\tilde{z}$ und erhalten nach Invertierung von D

$$(9.4)\quad \tilde{y}' = D^{-1}\,(U\,f(\,V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}))_{1,..,r}\quad ,\qquad \begin{pmatrix}\tilde{y}(x_o)\\ \tilde{z}(x_o)\end{pmatrix} = V^{-1}y_o$$

$$0 = (\,U\,f(\,V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix})\,)_{r+1,..,n}$$

wobei $(w)_{1,..,r}$ die ersten r Komponenten des Vektors w bezeichne.

Nun ist es a priori sehr schwer zu entscheiden, ob (9.2) ein Index-Eins Problem ist. In /7/ ist ansatzweise ein Algorithmus zur Bestimmung des Index vorgeschlagen.

Wir haben in §2 und Abschnitt 8.1 gesehen, daß die numerische Behandlung von Problemen vom Typ Index größer Eins einige Schwierigkeiten mit sich bringt. Im allg. wird die Schrittweitenkontrolle derart versagen, daß die Schrittweite sukzessive verkleinert wird, bis $h \leq h_{min}$ gilt.

Falls ein Versagen der Schrittweitenkontrolle auf diese Weise eintritt, erhalten wir damit den Hinweis, daß der Index von (9.4) größer Eins ist.

Zuzüglich zur einmaligen Berechnung der Singulärwertzerlegung von A entsteht in jeder Auswertung der rechten Seite ein merklicher Mehraufwand.

Zur Auswertung der Fuktion f kommen zwei Matrix-Vektor Produkte der Form $V\cdot\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}$ bzw. $U\cdot f$ und r Divisionen der Form $(1/\sigma_i)\;(U\,f\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix})_i$, i=1,..,r hinzu. Die ursprüngliche Lösung y des Problems (9.2) ergibt sich schließlich aus $y = V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}$.

Zur Lösung von (9.2) mit expliziten Verfahren nach Ansatz B ist dieser Mehraufwand unumgänglich, da eine Trennung der Differentialgleichungen und der algebraischen Nebenbedingungen nötig ist.

Diese Trennung kann jedoch bei Anwendung der impliziten Methoden nach Ansatz A vermieden werden. Dazu formulieren wir die Verfahrensklasse

$$(9.5) \qquad a_i = y_o + \sum_{j=1}^{i-1} \alpha_{ij} k_j \ , \ i = 1,\ldots,s$$

$$A\, k_i = h\, f(a_i) + h \sum_{j=1}^{i} \gamma_{ij} (D_y f)_o k_j \ , \ i=1,\ldots,s$$

$$y_1 = y_o + \sum_{i=1}^{s} \mu_i k_i \ .$$

(9.5) entspricht den bekannten ROW-Methoden zur Lösung von gewöhnlichen AWP, falls wir in (9.5) $A\, k_i$ durch k_i ersetzen.

Lemma 9.6:

Sei $\begin{pmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{pmatrix}$ die durch Anwendung einer impliziten Methode nach Ansatz A auf das Problem (9.4) erzeugte Lösung im Zeitpunkt $x_o + h.\ y_1$ sei die durch (9.5) erzeugte numerische Lösung von (9.2).

Wenn die Koeffizienten α_{ij}, γ_{ij}, γ, μ_i, $i=1,\ldots,s$, $j=1,\ldots,i-1$ der Methode nach Ansatz A und der Methode (9.5) übereinstimmen, dann gilt:

$$y_1 = V \begin{pmatrix} \tilde{y}_1 \\ \tilde{z}_1 \end{pmatrix} .$$

Beweis

Anwendung der Methode nach Ansatz A auf (9.4) liefert:

$$\begin{pmatrix}\tilde{a}_i\\ \tilde{b}_i\end{pmatrix} = \begin{pmatrix}\tilde{y}_o\\ \tilde{z}_o\end{pmatrix} + \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix}\tilde{l}_j\\ \tilde{k}_j\end{pmatrix}$$

$$\tilde{l}_i = h\, D^{-1}(U\, f(V\begin{pmatrix}\tilde{a}_i\\ \tilde{b}_i\end{pmatrix}))_{1,\ldots,r} + h \sum_{j=1}^{i} \gamma_{ij} \Big[\Big(D_{\tilde{y}}(D^{-1}(U\, f(V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}))_{1,\ldots,r}\Big)_{(\tilde{y}_o,\tilde{z}_o)} \cdot \tilde{l}_j + \Big(D_{\tilde{z}}(D^{-1}(U\, f(V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}))_{1,\ldots,r}\Big)_{(\tilde{y}_o,\tilde{z}_o)} \cdot \tilde{k}_j \Big]$$

$$0 = \Big(U\, f(V\begin{pmatrix}\tilde{a}_i\\ \tilde{b}_i\end{pmatrix})\Big)_{r+1,\ldots,n} + \sum_{j=1}^{i} \gamma_{ij} \Big[\Big(D_{\tilde{y}}(U\, f(V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}))_{r+1,\ldots,n}\Big)_{(\tilde{y}_o,\tilde{z}_o)} \cdot \tilde{l}_j + \Big(D_{\tilde{z}}(U\, f(V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix}))_{r+1,\ldots,n}\Big)_{(\tilde{y}_o,\tilde{z}_o)} \cdot \tilde{k}_j \Big]$$

$\leftrightarrow$

$$U^{-1} \left(\begin{array}{c|c} D & 0\\ \hline 0 & 0\end{array}\right) V^{-1}\, V \begin{pmatrix}\tilde{l}_i\\ \tilde{k}_i\end{pmatrix} = h\, f(V\begin{pmatrix}\tilde{a}_i\\ \tilde{b}_i\end{pmatrix}) + h \sum_{j=1}^{i} \gamma_{ij} \Big[\Big(D_{(\tilde{y},\tilde{z})} f\, (V\begin{pmatrix}\tilde{y}\\ \tilde{z}\end{pmatrix})\Big)_{(\tilde{y}_o,\tilde{z}_o)} \cdot \begin{pmatrix}\tilde{l}_j\\ \tilde{k}_j\end{pmatrix} \Big]$$

$\leftrightarrow$

$$A\, V \begin{pmatrix}\tilde{l}_i\\ \tilde{k}_i\end{pmatrix} + h\, f(V\begin{pmatrix}\tilde{a}_i\\ \tilde{b}_i\end{pmatrix}) + h \sum_{j=1}^{i} \gamma_{ij} \Big[(D_y f)_{(V\begin{pmatrix}\tilde{y}_o\\ \tilde{z}_o\end{pmatrix})} \cdot V \Big) \cdot \begin{pmatrix}\tilde{l}_j\\ \tilde{k}_j\end{pmatrix} .$$

Wegen $y_o = V \begin{pmatrix} \tilde{y}_o \\ \tilde{z}_o \end{pmatrix}$ folgt sukzessive für i=1,..,s:

$$a_i = V \begin{pmatrix} \tilde{a}_i \\ \tilde{b}_i \end{pmatrix} \quad \text{und} \quad k_i = V \begin{pmatrix} \tilde{l}_i \\ \tilde{k}_i \end{pmatrix} \quad .$$

Damit ist die Behauptung bewiesen.

Mit dem obigen Lemma ist somit gezeigt, daß die Ordnungs- und Konvergenzbedingungen der Methoden (9.5) äquivalent zu denen der impliziten Methoden nach Ansatz A sind.

Die Implementierung einer Methode nach (9.5) erfolgt analog zu der der bekannten ROW-Methoden, d.h.:

$$(A - h\,\gamma(D_y f)_o)\,(k_i + \sum_{j=1}^{i-1} \tilde{\gamma}_{ij} k_j) = h\,f(a_i) + A \sum_{j=1}^{i-1} \tilde{\gamma}_{ij} k_j$$

$$i=1,..,s \; ; \; \tilde{\gamma}_{ij} = \gamma_{ij}/\gamma \; .$$

Da A singulär ist, kann der Fall eintreten, daß bei Integration mit sehr kleiner Schrittweite h die zu invertierende Matrix $E = (A - h\,\gamma(D_y f)_o)$ schlecht konditioniert ist. Dies kann zu einem Scheitern des Fehlertests $EST \leq TOL$ führen, und es tritt der bekannte Effekt auf, daß h sukzessive reduziert wird, bis $h \leq h_{min}$ gilt.
Dieser Effekt ist unerwünscht und tritt bei Integration von Index-Eins Problemen der transformierten Form (9.4) mit den Methoden nach Ansatz A oder B nicht auf.
Daher kann es numerisch trotz Mehraufwand durchaus sinnvoll sein, das Problem (9.2) auf die Form (9.4) zu transformieren.

Eine weitere Folge für die Methode (9.5) ist, daß die Schrittweitensteuerung aufgrund der fehlenden Separation der Differentialgleichungen und algebraischen Nebenbedingungen nicht, wie in Abschnitt 7.1 beschrieben ist, gesplittet werden kann. Die für ein Verfahren p-ter Ordnung übliche Schrittweitensteuerung

$$(9.7)\quad h_{neu} = 0.9\, h_{alt} \min\left((TOL/EST_y)^{1/p},\ (TOL/EST_z)^{1/(p-1)} \right)$$

wird ersetzt durch

$$(9.8)\quad h_{neu} = 0.9\, h_{alt}\, (TOL/EST)^{1/p} \quad \text{mit } EST = \max(EST_y, EST_z).$$

Gehen wir von einem erfolgreichen Schritt aus, d.h. $EST \leq TOL$, so schlägt (9.8) eine etwas kleinere Schrittweite als (9.7) vor. Wir können also erwarten, daß die Lösung von (9.2) einige Integrationsschritte mehr als die Lösung von (9.4) benötigt.

Die Aufrufliste der Methoden (9.5) ist analog zu der in Abschnitt 7.1 beschriebenen Aufrufliste der Methoden nach Ansatz A. Sie ist lediglich um die Eingabematrix A ergänzt, und die geforderte Eingabe M der Anzahl der Differentialgleichungen entfällt.

Als erstes Testbeispiel behandelten wir das folgende Problem:

$$(9.9)\quad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_1 + y_2 \\ 2y_1 + 5 \end{pmatrix}, \quad \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} -1 \\ 4 \end{pmatrix}.$$

Die exakte Lösung lautet

$$y_1(x) = (3\exp(x) - 5)/2$$
$$y_2(x) = (3\exp(x) + 5)/2\ .$$

Wir testen in diesem und den folgenden Beispielen nur noch die Methoden DAE4SF, DAE34NS (mit bzw. ohne SWZ) und DASSL.

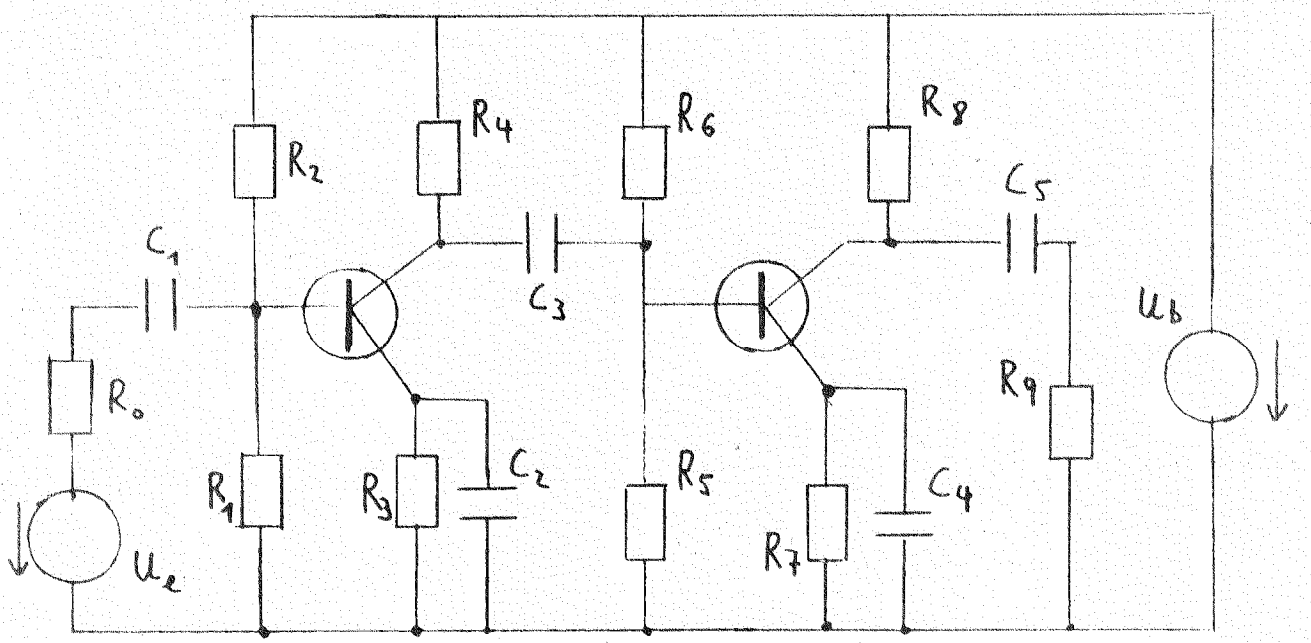
Verfahren		DAE4SF ohne SWZ	DAE4SF mit SWZ	DAE34NS mit SWZ	DASSL ohne SWZ
TOL		1.E-4	1.E-4	1.E-4	1.E-4
Letzte Int.Zeit		2.00	2.00	2.00	2.00
CPU-Zeit		0.0158	0.0260	0.0273	0.0329
Erfolg. Schritte		18	18	18	27
Nicht erf. Schr.		0	0	0	1
FCN-Aufr. ges.		108	108	144	73
FCN-Aufr. Verf.		72	72	108	49
Rel. Fehler (letzte Zeit)	y_1	8.5E-4	8.4E-4	1.7E-5	4.2E-4
	y_2	5.4E-4	5.4E-4	1.1E-5	2.7E-4

Trotz der geringen Anzahl von Integrationsschritten macht sich schon ein deutlicher Unterschied in der CPU-Zeit bei Berechnung des ursprünglichen Systems und des analog (9.4) transformierten Systems bemerkbar.

9.2. Beispiel Verstärker

In diesem Abschnitt berechnen wir das Eingangs- Ausgangsverhalten eines zweistufigen Verstärkers, dessen Modell uns die Herren Prof.Dr. H.J. Oberle und Prof. Dr. K. Glashoff von der Universität Hamburg zur Verfügung stellten.

Skizze 5: Schaltbild eines zweistufigen Verstärkers



Durch Anwendung der Kirchhoffschen Sätze ergeben sich die Knotengleichungen:

(1) $U_e/R_o - U_1/R_o + (\dot{U}_2 - \dot{U}_1)\cdot C_1 = 0$

(2) $(\dot{U}_1 - \dot{U}_2)\cdot C_1 - U_2(1/R_1 + 1/R_2) + U_b/R_2 - f(U_2 - U_3)$
$+ \alpha\, f(U_2 - U_3) = 0$

$$(3)\quad f(U_2 - U_3) - U_3/R_3 - \dot{U}_3 C_2 = 0$$

$$(4)\quad -\alpha\, f(U_2 - U_3) - U_4/R_4 + U_b/R_4 - (\dot{U}_4 - \dot{U}_5)\, C_3 = 0$$

$$(5)\quad (\dot{U}_4 - \dot{U}_5)\, C_3 - U_5(1/R_5 + 1/R_6) + U_b/R_6 - f(U_5 - U_6) + \alpha\, f(U_5 - U_6) = 0$$

$$(6)\quad f(U_5 - U_6) - U_6/R_7 - \dot{U}_6\, C_4 = 0$$

$$(7)\quad -\alpha\, f(U_5 - U_6) - U_7/R_8 + U_b/R_8 - (\dot{U}_7 - \dot{U}_8)\, C_5 = 0$$

$$(8)\quad (\dot{U}_7 - \dot{U}_8)\, C_5 - U_8/R_9 = 0$$

Der Verstärker wird also durch ein Index-Eins Gleichungssystem der Form

(9.10) $C\,\dot{U} = F(U, U_e)$ beschrieben.

$U = (U_1, \ldots, U_8)$ bezeichnet die Knotenspannungen, U_8 ist die Ausgangsspannung, C ist eine von den gewählten Kapazitäten C_i der Kondensatoren abhängige singuläre Matrix, F ist abhängig von den gewählten Widerständen R_i und Transistorkennlinien f, von der Betriebsspannung U_b sowie von dem Eingangssignal $U_e(t)$.

Die (8,8)-Matrix C ist von der Gestalt

$$C = \begin{pmatrix} -C_1 & C_1 & & & & & & \\ C_1 & -C_1 & & & & & & \\ & & -C_2 & & & & & \\ & & & -C_3 & C_3 & & & \\ & & & C_3 & -C_3 & & & \\ & & & & & -C_4 & & \\ & & & & & & -C_5 & C_5 \\ & & & & & & C_5 & -C_5 \end{pmatrix}$$

und hat somit den Rang 5.

Das FORTRAN-Unterprogramm FVER zur Berechnung von f ist in /21/ widergegeben.

Das Beispiel wurde mit verschiedenen Eingangssignalen $U_e(t)$ gerechnet.

a) $U_e(t) \equiv 0.1$

Zunächst erfordert die Bearbeitung des Problems die Bestimmung konsistenter Anfangswerte U(0). Dazu transformierten wir (9.10) durch eine SWZ der Matrix C analog zu (9.3) in ein Problem der Form

$$\text{(9.11a)} \quad y' = f(y,z)$$
$$\text{(9.11b)} \quad 0 = g(y,z) \quad , \quad \begin{bmatrix} y \\ z \end{bmatrix} = V^{-1} \begin{bmatrix} U_1 \\ \vdots \\ U_8 \end{bmatrix} .$$

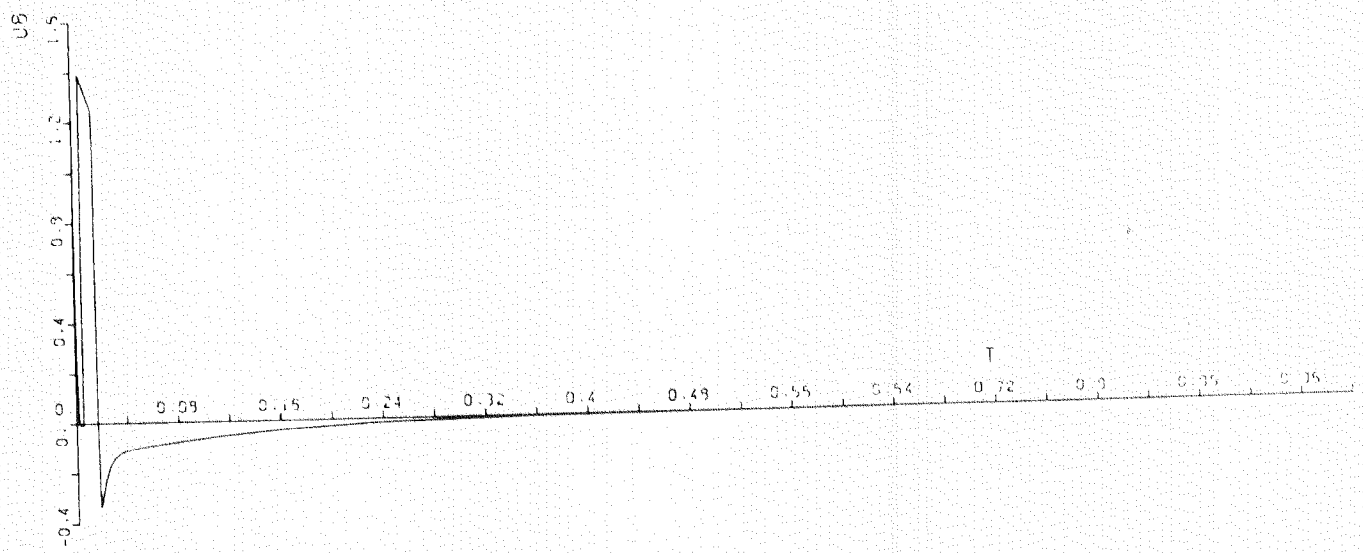
Die Anfangswerte y_o können beliebig gewählt werden.
Die übrigen Anfangswerte z_o werden mit Hilfe von (9.11b) durch ein Newtonverfahren berechnet. Auf diese Weise erhalten wir $U_o = V \begin{bmatrix} y_o \\ z_o \end{bmatrix}$.

Nun kann das Problem direkt oder nach Transformation in (9.11) mit den bekannten Methoden bearbeitet werden.
Da durch den vorliegenden Verstärker keine Gleichspannungen verstärkt werden, geht die Ausgangsspannung nach kurzer Einschwingphase asymptotisch gegen Null. Also handelt es sich um ein steifes DAE-System. Das ist auch sehr schön an den Resultaten der expliziten Methode DAE34NS und an dem in Skizze 6 dargestellten Lösungsverhalten zu erkennen.

Verfahren	DAE4SF mit SWZ	DAE34NS mit SWZ	DAE4SF ohne SWZ	DASSL ohne SWZ
TOL	1.E-4	1.E-4	1.E-4	1.E-4
Letzte Int.Zeit	1.000	1.000	1.000	1.000
CPU-Zeit	1.199	24.211	0.772	0.807
Erfolg. Schritte	97	1940	108	198
Nicht erf. Schr.	17	1054	20	15
FCN-Aufr. ges.	1215	32430	1356	810
FCN-Aufr. Verf.	439	16910	492	466

Skizze 6:

Ausgangsspannung U_8 für $U_e \equiv 0.1$:



Es bestätigt sich auch die Erwartung aus Abschnitt 9.1, daß die Bearbeitung des nichttransformierten Problems mit DAE4SF einige mehr Integrationsschritte benötigt als die Bearbeitung des transformierten Problems mit DAE4SF.

b) $U_e(t) = 0.1 \sin(2\pi \cdot 100 \cdot t)$

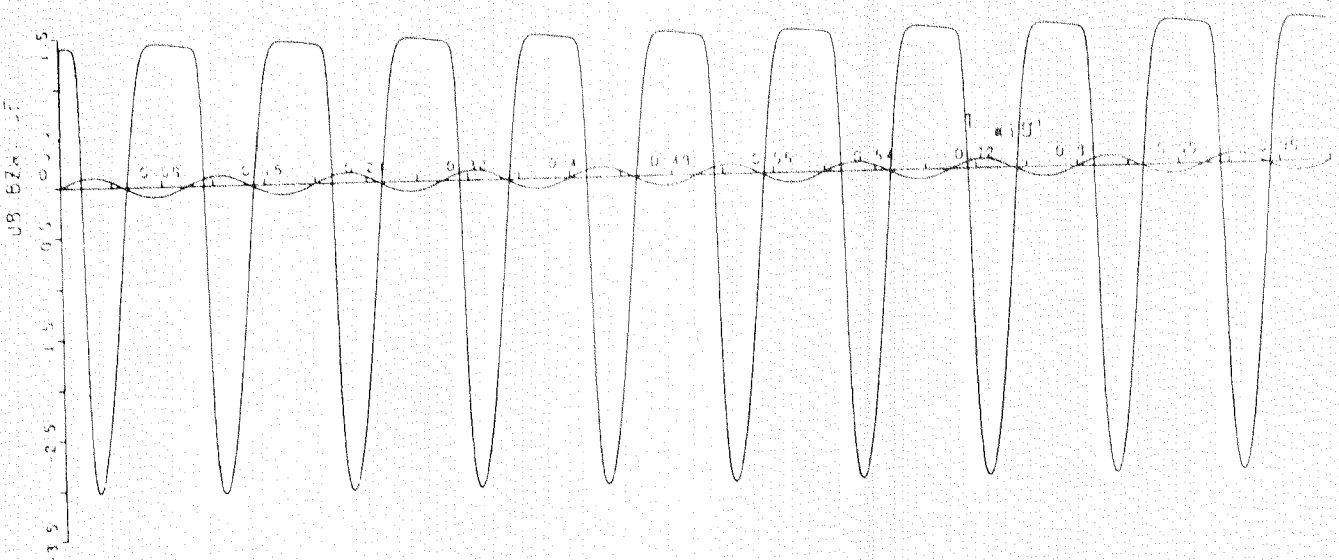
Zur Bestimmung konsistenter Anfangswerte gehen wir hier wie folgt vor: wie in a) bestimmen wir konsistente Anfangswerte für das Problem mit $U_e \equiv 0$. Nun integrieren wir das Problem mit $U_e \not\equiv 0$ eine gewisse Zeit bei der hohen Genauigkeitsforderung TOL = 10^{-8}, bis die Einschwingphase vorüber ist. Die resultierenden Werte $U_1,\ldots,U_8$ können nun als Anfangswerte für das ursprüngliche Problem b) gewählt werden. Dadurch sparen wir uns die numerische Behandlung der sehr empfindlichen Einschwingphase bei $U_e = 0.1 \sin(2\pi \cdot 100 \cdot t)$, die durch die beliebige Wahl konsistenter Anfangswerte entstehen würde.

In den Beispielen a), b) und c) wurde die Eingangsschrittweite immer HI = 10^{-9} gewählt. Bei größerem HI erfolgt sonst ein Exponentation-Overflow beim Aufruf der rechten Seite F, weil F extrem nichtlineare Terme enthält. HI darf auch nicht kleiner gewählt werden, da sonst die in Abschnitt 9.1 beschriebene Matrix E beim Aufruf von DAE4SF (ohne SWZ) singulär wird.

Wir erhalten folgende Resultate und den unten skizzierten Lösungsverlauf der Ausgangsspannung U_8:

Verfahren	DAE4SF mit SWZ	DAE34NS mit SWZ	DAE4SF ohne SWZ	DASSL ohne SWZ
TOL	1.E-4	1.E-4	1.E-4	1.E-4
Letzte Int.Zeit	0.1000	0.1000	0.1000	0.1000
CPU-Zeit	11.941	13.335	6.687	5.043
Erfol. Schritte	734	1000	750	1009
Nicht erf. Schr.	133	141	119	180
FCN-Aufr. ges.	9942	15706	10107	5512
FCN-Aufr. Verf.	3335	6706	3357	2608

Skizze 7: Eingangsspannung $U_e = 0.1 \sin(2\pi \cdot 100 \cdot t)$ und Ausgangsspannung U_8.



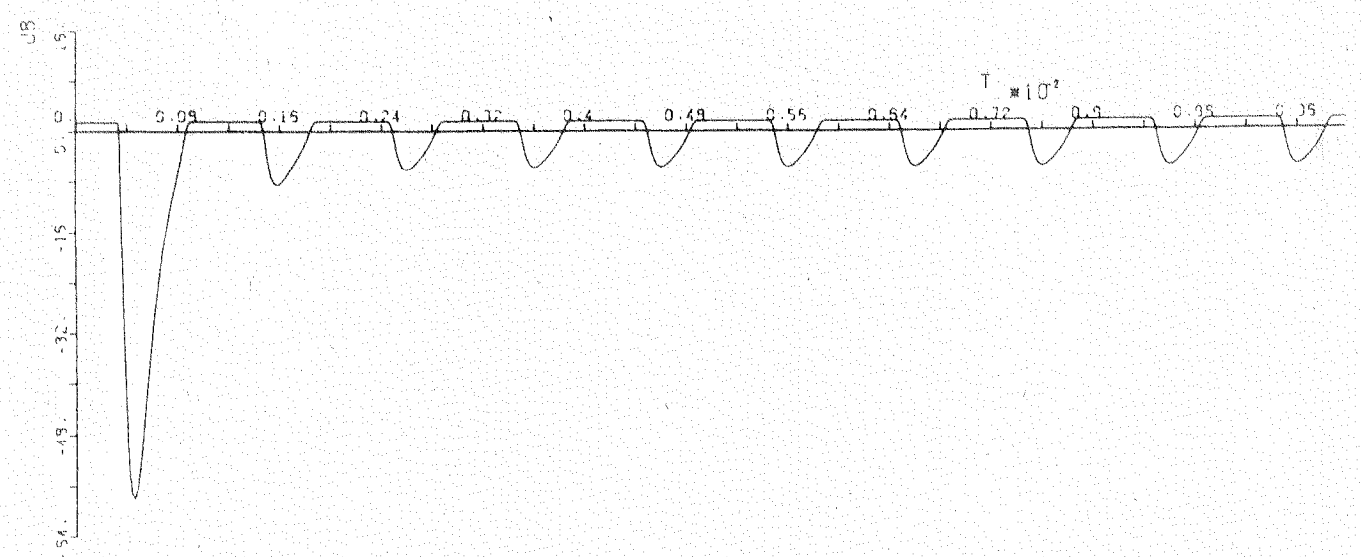
Obwohl die Gesamtanzahl der rechte Seite Aufrufe von DAE34NS um 50% höher ist als die von DAE4SF (mit SWZ), unterscheidet sich die verbrauchte CPU-Zeit nicht wesentlich. Hier macht sich bemerkbar, daß bei der Berechnung mit DAE4SF in jedem Schritt die volle (8,8)-Matrix LU-zerlegt werden muß, während bei der Berechnung mit DAE34NS pro Schritt nur eine (3,3)-Matrix LU-zerlegt werden muß.
DAE4SF (ohne SWZ) arbeitet in allen Beispielen sehr gut. Bei der Integration mit kleineren Toleranzen TOL = 10^{-5} oder TOL = 10^{-6} steigt die Rechenzeit aller Einschrittverfahren im Unterschied zu DASSL stark an. Erst hier macht sich ein Vorteil von DAE34NS gegenüber DAE4SF (mit SWZ) bemerkbar.

c) $U_e = 0.1 \sin(2\pi \cdot 1000 \cdot t)$

Die Verstärkung des Eingangssignals mit der Frequenz 1000 Hz liefert ähnliche numerische Resultate wie in b). Diese sind in der Arbeit /21/ zu finden.

Skizze 8:

Ausgangsspannung U_8 für $U_e = 0.1 \sin(2\pi \cdot 1000 \cdot t)$



10. Zusammenfassung

Die in /17/ auf Index-Eins DAE's der Form

$$(10.1)\quad y' = f(y,z)$$
$$0 = g(y,z)$$

übertragene Butcherreihentheorie wurde noch einmal entwickelt und widergegeben. Mit Hilfe dieser Theorie konnten wir ebenfalls analog zu /17/ Ordnungs- und Konvergenzbedingungen für einen semi-impliziten Lösungsansatz herleiten.
Diesem stellten wir einen neuen semi-expliziten Lösungsansatz für DAE's gegenüber, für den wir dann in ähnlicher Weise Ordnungs- und Konvergenzbedingungen herleiteten.
Mit Hilfe dieser Bedingungen berechneten wir verschiedene Koeffizientensätze für explizite und implizite Verfahren dritter und vierter Ordnung.
Die mit einer Schrittweitensteuerung versehenen Methoden implementierten wir ähnlich wie die bekannten ROW-Methoden und testeten sie schließlich.
Neben einigen kleineren Testbeispielen rechneten wir zwei Beispiele aus der Technik. Das mathematische Modell des Lastfalls einer Fahrzeughinterachse ist stellvertretend für die bei der Simulation mechanischer Systeme auftretenden oft steifen DAE-Systeme.
Es zeigt sich, daß die impliziten Methoden konkurrenzfähig sind und weiter studiert werden sollten.
Auch in der Elektrotechnik können viele Systeme, wie beispielsweise der hier behandelte Verstärker, durch DAE's

beschrieben werden.
Das entstehende DAE-System hat die Form

(10.2) $A\,y' = f(y)$, wobei A singulär ist.

Mit Hilfe einer Singulärwertzerlegung kann (10.2) transformiert werden in (10.1). Jedoch erweist es sich als effizienter, (10.2) direkt mit einer leicht modifizierten Version der impliziten Verfahren zu bearbeiten.
Wir zeigten, daß beide Vorgehensweisen mathematisch äquivalent sind. In einigen speziellen Fällen kann die letztere zu numerischen Schwierigkeiten führen.

Die Vorteile der expliziten Methoden konnten wir leider mit den hier gerechneten Beispielen nicht deutlich herausstellen. Sie treten insbesondere bei der Integration nichtsteifer großer DAE-Systeme mit einer geringen Anzahl von algebraischen Nebenbedingungen auf.

Abschließend können wir sagen, daß mit den hier entwickelten Verfahren gute Alternativen zur Lösung von DAE's vorgestellt wurden.

11. Literatur

/1/ Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Comput. Math. Appl. Mech. Engrg. 1 (1972) 1-16.

/2/ Butcher, J. C.: Coefficients for the study of Runge-Kutta Integration Processes. J. Austral. Math. Soc. 3 (1963) 185-201.

/3/ Deuflhard, P., Hairer, E., Zugck, J.: One step and extrapolation methods for differential algebraic equations. Institut für Andewandte Mathematik, Universität Heidelberg, Preprint 318, Juni 1985.

/4/ Fehlberg, E.: Klassische Runge-Kutta Formeln vierter und niedriger Ordnung mit Schrittweitenkontrolle und ihre Anwendung auf Wärmeleitungsprobleme. Computing 6 (1970) 61-71.

/5/ Gantmacher, F. R.: The Theorie of Matrices, Vol.2, Chelsea, New York, 1964.

/6/ Gear, C. W., Petzold, L. R.: ODE methods for the solution of differential/algebraic systems. SIAM J. Numer. Anal. 21 (1984) 716-728.

/7/ Golub, G. H., Reinsch, C.: Singular value decomposition and least squares solutions. Numer. Math. 14(5) (1970) 403-420.

/8/ Hairer, E., Wanner, G.: On the Butcher group and general mulivalue methods. Computing 13 (1974) 1-15.

/9/ Kaps, P., Rentrop, P.: Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations. Numer. Math. 33 (1979) 55-68.

/10/ Kaps, P., Wanner, G.: A study of Rosenbrock-type methods of high order. Numer. Math. 38 (1981) 279-298.

/11/ Petzold, L., Lötstedt, P.: Numerical solution of nonlinear differential equations with algebraic constraints II: practical implications. SIAM J. Sci. Stat. Comp. 7 (1986) 720-733.

/12/ Petzold, L.R.: A description of DASSL: A differential algebraic system solver. SAND 82-8637, Sandia National Laboratories, Livermore, CA, 1982.

/13/ Petzold, L.: Differential/algebraic equations are not ODE's. SIAM J. Sci. Stat. Comput. 3 (1982) 367-384.

/14/ Petzold, L.R.: Order results for implicit Runge-Kutta Methods applied to differential/algebraic systems. SIAM J. Numer. Anal. 23 (1986) 837-852.

/15/ Rentrop, P.: Spezielle Verfahren zur numerischen Lösung von Differentialgleichungsmodellen aus der Technik. Vorlesung an der Universität Kaiserslautern im SS 1986.

/16/ Rheinboldt, W.C.: Differential-algebraic systems as differential equations on manifolds. Math. of Comput. 43 (1984) 473-482.

/17/ Roche, M.: Rosenbrock methods for differential algebraic equations. Dept. de Mathematique, Universität Genf, September 1986.

/18/ Senger, K.-H.: Vergleich linearer und nichtlinearer Mehrkörperformalismen und Programme am Beispiel eines einfachen Modells einer Fahrzeughinterachse. Institut für Dynamik der Flugsysteme, DFVLR Oberpfaffenhofen, Bericht IB515-85/02, August 1985.

/19/ Sincovec, R., Erisman, A., Yip, E., Epton, M.: Analysis of descriptor systems using numerical algorithms. IEEE Trans. Aut. Cont. 26 (1981) 139-147.

/20/ Sincovec, R., Yip, E., Epton, M., Manke, J., Erisman, A., Dembart, B. and Lu, P.: Solvability of large-scale descriptor systems. System Engrg. for Power: Organizational Forms for large scale systems, vol.1, L.H. Fink and T.A. Trygar, Editors, CONF-790904-P2. Springfield, VA: NTIS, Oct. 1979.

/21/ Steinebach, G.: Semi-implizite Einschrittverfahren zur numerischen Lösung Differential-Algebraischer Gleichungen technischer Modelle. Fachbereich Mathematik, Universität Kaiserslautern, Diplomarbeit 1987.

/22/ Stoer, J., Bulirsch, R.: Einführung in die numerische Mathematik II. Springer-Verlag: Berlin, Heidelberg, New York 1978.

/23/ Wanner, G.: On the choice of γ for singly-implicit RK or Rosenbrock methods. BIT 20 (1980) 102-106.

12. Anhang

12.1. Koeffizientensätze

a) DAE3S

$\gamma = 0.4$

$\alpha_{21} = 0.75$

$\alpha_{31} = 0.75$

$\alpha_{32} = 0$

$\mu_1 = 0.407407407407407$

$\mu_2 = 0.120527306967984$

$\mu_3 = 0.472065285624607$

$\tilde{\gamma}_{ij} := \gamma_{ij}/\gamma$

$\tilde{\gamma}_{21} = 1.9921875$

$\tilde{\gamma}_{31} = -2.081711542553192$

$\tilde{\gamma}_{32} = -0.251063829787234$

$\hat{\mu}_1 = 0.3761659144637868$

$\hat{\mu}_2 = 0.1238340855362132$

$\hat{\mu}_3 = 0.5$

b) DAE4S

Koeffizientensatz siehe /17/.

c) DAE4SF

$\gamma = 0.5$

$\alpha_{21} = 0.$

$\alpha_{31} = 0.25$

$\alpha_{32} = 0.25$

$\alpha_{41} = 0.0625$

$\alpha_{42} = 0.125$

$\alpha_{43} = 0.5625$

$\alpha_{51} = 1.200810185185185$

$\alpha_{52} = -1.950810185185185$

$\alpha_{53} = 0.25$

$\alpha_{54} = 1.$

$\tilde{\gamma}_{21} = 2.$

$\tilde{\gamma}_{31} = 0.23148148148148$

$\tilde{\gamma}_{32} = -0.75$ $\quad$ $\tilde{\gamma}_{51} = -6.0034722222222$

$\tilde{\gamma}_{41} = 0.96875$ $\quad$ $\tilde{\gamma}_{52} = 3.6516203703703$

$\tilde{\gamma}_{42} = -0.8125$ $\quad$ $\tilde{\gamma}_{53} = 9.5$

$\tilde{\gamma}_{43} = -1.125$ $\quad$ $\tilde{\gamma}_{54} = -7.9259259259259$

$\mu_1 = 0.5388888888888$ $\quad$ $\hat{\mu}_1 = 0.4523148148148$

$\mu_2 = -0.1314814814814$ $\quad$ $\hat{\mu}_2 = -0.1560185185185$

$\mu_3 = -0.2$ $\quad$ $\hat{\mu}_3 = 0.$

$\mu_4 = 0.5925925925925$ $\quad$ $\hat{\mu}_4 = 0.5037037037037$

$\mu_5 = 0.2$ $\quad$ $\hat{\mu}_5 = 0.2$

d) DAE3NS

$\gamma = 1/3$

$\alpha_{21} = 2/3$ $\quad$ $\tilde{\gamma}_{21} = -10/9$ $\quad$ $\mu_1 = 1/4$

$\alpha_{31} = -1/3$ $\quad$ $\tilde{\gamma}_{31} = 17/27$ $\quad$ $\mu_2 = 1/2$

$\alpha_{32} = 1$ $\quad$ $\tilde{\gamma}_{32} = -3$ $\quad$ $\mu_3 = 1/4$

$\hat{\mu}_1 = 1/4$ $\quad$ $\hat{\mu}_2 = 3/4$ $\quad$ $\hat{\mu}_3 = 0$

e) DAE33NS

$\gamma = 1/4$

$\alpha_{21} = 1/2$ $\quad$ $\alpha_{43} = 1/4$ $\quad$ $\tilde{\gamma}_{41} = -265/144$

$\alpha_{31} = 9/8$ $\quad$ $\tilde{\gamma}_{42} = 11/6$

$\alpha_{32} = -3/8$ $\quad$ $\tilde{\gamma}_{21} = -2$ $\quad$ $\tilde{\gamma}_{43} = -14/9$

$\alpha_{41} = 1/2$ $\quad$ $\tilde{\gamma}_{31} = -21/4$

$\alpha_{42} = 1/4$ $\quad$ $\tilde{\gamma}_{32} = 9/4$

$\mu_1 = 14/117$ $\quad \hat{\mu}_1 = 0$

$\mu_2 = 37/39$ $\quad \hat{\mu}_2 = 17/13$

$\mu_3 = -44/117$ $\quad \hat{\mu}_3 = -8/13$

$\mu_4 = 4/13$ $\quad \hat{\mu}_4 = 4/13$

f) DAE34NS

$\gamma = 1/8$

$\alpha_{21} = 1/4$ $\quad \tilde{\gamma}_{21} = -2$

$\alpha_{31} = 3/32$ $\quad \tilde{\gamma}_{31} = -1.5$

$\alpha_{32} = 9/32$ $\quad \tilde{\gamma}_{32} = -1.5$

$\alpha_{41} = 1932/2197$ $\quad \tilde{\gamma}_{41} = -12.82476103777$

$\alpha_{42} = -7200/2197$ $\quad \tilde{\gamma}_{42} = 23.70505234410$

$\alpha_{43} = 7296/2197$ $\quad \tilde{\gamma}_{43} = -18.26490669094$

$\alpha_{51} = 439/216$ $\quad \tilde{\gamma}_{51} = -27.77893518518$

$\alpha_{52} = -8$ $\quad \tilde{\gamma}_{52} = 58.5$

$\alpha_{53} = 3680/513$ $\quad \tilde{\gamma}_{53} = -44.07407407407$

$\alpha_{54} = -845/4104$ $\quad \tilde{\gamma}_{54} = 0.978009259259$

$\alpha_{61} = -8/27$ $\quad \tilde{\gamma}_{61} = 2.066912615740$

$\alpha_{62} = 2$ $\quad \tilde{\gamma}_{62} = -7.309027777777$

$\alpha_{63} = -3544/2565$ $\quad \tilde{\gamma}_{63} = 7.013304093567$

$\alpha_{64} = 1859/4104$ $\quad \tilde{\gamma}_{64} = -3.347751431530$

$\alpha_{65} = -11/40$ $\quad \tilde{\gamma}_{65} = 1.7875$

$\mu_1 = 25/216$ $\quad \mu_4 = 2197/4104$ $\quad \hat{\mu}_1 = 16/135$ $\quad \hat{\mu}_5 = -9/50$

$\mu_2 = 0$ $\quad \mu_5 = -1/5$ $\quad \hat{\mu}_2 = 0$ $\quad \hat{\mu}_6 = 2/55$

$\mu_3 = 1408/2565$ $\quad \hat{\mu}_3 = 6656/12825$

$\hat{\mu}_4 = 28561/56430$