

INSTITUT FÜR ANGEWANDTE UND
NUMERISCHE MATHEMATIK

TECHNISCHE UNIVERSITÄT WIEN

Report Nr. 124/98

**Steuerungsmaßnahmen bei numerischer Lösung
singulärer Anfangswertprobleme**

W. Auzinger
P. Kofler
E. Weinmüller

Mit der teilweisen Unterstützung des FWF, Projekt P12507-MAT.

Für den Inhalt verantwortlich:

Dr. Winfried Auzinger, Dr. Peter W. Kofler und Dr. Ewa B. Weinmüller, Wien.

Verlag:

Institut für Angewandte und Numerische Mathematik, Technische Universität Wien.

Alle Rechte

bei den Autoren Dr. Winfried Auzinger, Dr. Peter Kofler und Dr. Ewa B. Weinmüller, Wien.

Inhaltsverzeichnis

Kurzfassung	2
1 Einleitung	3
2 Grundlagen	6
2.1 Singuläre Anfangswertaufgaben	6
2.2 Testbeispiele	8
2.3 Numerische Lösung von Anfangswertaufgaben	14
2.4 Hard- und Softwarespezifikation	16
3 Schätzung des lokalen Fehlers	18
3.1 Runge-Kutta Paare	18
3.1.1 Runge-Kutta 5(4) Paar (DOPRI5)	19
3.1.2 Runge-Kutta 3(2) Paar	29
3.1.3 Runge-Kutta 8(7) Paar (DOPRI8)	29
3.1.4 Runge-Kutta 4(5) Paar	29
3.1.5 Runge-Kutta 5(4) Paar mit „exakter“ Fehlerschätzung	33
3.1.6 Zusammenfassung	38
3.2 Richardson-Extrapolation	39
3.2.1 Basismethode explizites Runge-Kutta 4	40
3.2.2 Basismethode implizites Runge-Kutta 5 (Radau IIA)	45
3.2.3 Zusammenfassung	49
3.3 Ordnung des lokalen Fehlers der Verfahren	49
3.3.1 Explizites Runge-Kutta 4	50
3.3.2 Implizites Runge-Kutta 5 (Radau IIA)	55
3.3.3 Ordnungskurven	58
3.3.4 Ordnung bezüglich x und h	60
3.3.5 Zusammenfassung	63
3.4 „Exakte“ Fehlerschätzung	65
3.4.1 Explizites Runge-Kutta 4	65
3.4.2 Zusammenfassung	69
4 Struktur des lokalen Fehlers	70
4.1 Ordnung des lokalen Fehlers ERK 4	70
4.1.1 Lineare Probleme mit konstanter Koeffizientenmatrix	72
4.1.2 Lineare Probleme mit variabler Koeffizientenmatrix	75
4.1.3 Zusammenfassung	79
4.2 Ordnung des lokalen Fehlers IRK 4	80
4.2.1 Lineare Probleme mit konstanter Koeffizientenmatrix	81
4.2.2 Zusammenfassung	82

5	Schätzung des globalen Fehlers	83
5.1	Der globale Fehler der Verfahren	83
5.1.1	Explizites Runge-Kutta 4	83
5.1.2	Implizites Runge-Kutta 4	86
5.1.3	Zusammenfassung	88
5.2	Zadunaisky-Schätzung	88
5.2.1	Basismethode explizites Runge-Kutta 4	90
5.2.2	Basismethode explizites Runge-Kutta 7	100
5.2.3	Ordnung des Fehlers des Zadunaisky-Schätzers für ERK 4	100
5.2.4	Basismethode implizites Runge-Kutta 4 (Gauss)	102
5.2.5	Zusammenfassung	108
6	Zusammenfassung	110
	Tabellenverzeichnis	111
	Abbildungsverzeichnis	111
	Literatur	114

Kurzfassung

Wir betrachten nichtlineare Anfangswertaufgaben für Systeme gewöhnlicher Differentialgleichungen erster Ordnung mit einer Singularität erster Art. Basierend auf [28], welches das analytische Hintergrundwissen über die betrachtete Problemklasse liefert, werden verschiedene Methoden der Schätzung des lokalen und globalen Diskretisierungsfehlers von Runge-Kutta Verfahren sowie verschiedene Gitterwahlstrategien experimentell untersucht.

Es zeigt sich, daß in der Nähe der Singularität die Ordnung des lokalen Fehlers fast immer reduziert ist, wodurch die Schrittweitenwahl basierend auf der Schätzung des lokalen Fehlers nicht sehr zuverlässig ist. Im Gegensatz dazu funktionieren Methoden zur Schätzung des globalen Diskretisierungsfehlers ausgesprochen gut. Der globale Fehler wird gut wiedergegeben, und man erhält bereits auf groben Gittern eine relativ genaue Schätzung, woraus ein zur Integration singulärer Anfangswertaufgaben geeigneter Algorithmus hergeleitet werden kann.

1 Einleitung

Mathematische Modelle zahlreicher Anwendungen in der Chemie, der Physik und der Mechanik haben die Form von Randwertproblemen für Systeme partieller Differentialgleichungen. Beispiele dafür sind die Modelle nach Ginzburg-Landau, Helmholtz und Thomas-Fermi¹. Gleichungen vom Ginzburg-Landau Typ ergeben sich unter anderen in Problemen der Supraleitung, [45, 47, 49, 50, 55], bei der Modellierung von unidirektionalen Ring-Lasern und in der Laser-Hydrodynamik, [46, 54]. Thomas-Fermi Gleichungen kommen vor allem in statistischen Modellen von Atomen und Molekülen, sowie Modellen für isolierte neutrale Atome vor, [48, 53].

Falls geometrische Symmetrien vorhanden sind, kann man häufig Systeme partieller Differentialgleichungen auf Systeme gewöhnlicher Differentialgleichungen transformieren, wobei sich dabei typischerweise Singularitäten im Differentialoperator ergeben. Das Interesse an der numerischen Lösung dieser transformierten Systeme ist durch ihren großen Anwendungsbereich zu erklären. Daher gibt es bereits umfangreiche Literatur, in der verschiedene Näherungsmethoden auf ihre Tauglichkeit zur Behandlung singulärer Probleme untersucht werden.

Trotzdem existiert noch keine speziell für singuläre Systeme entwickelte professionelle Software. Dies liegt am Mangel von Erkenntnissen auf dem Gebiet der Fehlerschätzung und anderer Steuermechanismen für die zur Lösung singulärer Probleme verwendeten numerischen Verfahren.

Die Problemklasse, die dieser Arbeit zugrunde liegt, besteht aus nichtlinearen singulären Randwertaufgaben erster Ordnung,

$$y'(x) = x^{-\alpha} f(x, y(x)), \quad 0 < x \leq 1, \quad (1.1a)$$

$$B_0 \cdot y(0) + B_1 \cdot y(1) = \beta, \quad (1.1b)$$

$$y \in \mathbf{C}[0, 1], \quad (1.1c)$$

mit $\alpha > 0$, wobei y und f vektorwertige Funktionen der Dimension n , B_0 und B_1 konstante, reelle $(m \times n)$ -Matrizen mit $m \leq n$ sind, und β ein konstanter, reeller, m -dimensionaler Vektor ist. In Abhängigkeit von α unterscheidet man zwischen einer schwachen Singularität ($\alpha < 1$), einer Singularität erster Art ($\alpha = 1$) und einer Singularität zweiter Art ($\alpha > 1$).

Zur numerischen Lösung von Randwertproblemen zweiter Ordnung mit schwacher Singularität wurden Dreipunkt Differenzenverfahren in [4, 5] und Spline-Funktionen auf nicht äquidistanten Gittern in [51, 52] studiert.

Randwertprobleme mit einer Singularität erster Art wurden bereits von zahlreichen Autoren untersucht. In [37] wurde an einer speziellen skalaren Differentialgleichung zweiter Ordnung gezeigt, daß Kollokationsmethoden zur Lösung singulärer Randwertaufgaben eingesetzt werden können. Allgemeiner haben de Hoog und Weiss in [20] die numerische Lösbarkeit von linearen Systemen erster Ordnung mittels Kollokation studiert. Es zeigte sich, daß die klassischen Stabilitäts- und Konvergenzresultate der

¹Eine kurze Liste der neuesten Publikationen auf diesen Gebieten befindet sich am Ende des Literaturverzeichnisses, um das rege Interesse an der Lösung dieser Probleme zu dokumentieren.

Kollokationsmethoden, vgl. z.B. [36], vom Spektrum der Koeffizientenmatrix abhängen, und nicht ohne Einschränkungen gelten.

In [21] bewiesen de Hoog und Weiss, daß Differenzenverfahren konvergieren und die erwarteten Konvergenzordnungen aufweisen, falls die Problemdata entsprechend glatt sind. Systeme zweiter Ordnung wurden in [6, 26, 32, 37] (skalare Probleme) und [41, 42] untersucht. Es zeigte sich, daß die Konvergenz der Verfahren ebenfalls vom Spektrum der Koeffizientenmatrix abhängt und eine Reduktion der klassischen Konvergenzgeschwindigkeit zu erwarten ist. Die Singularität wirkt sich auch auf beschleunigte Algorithmen negativ aus, da gewisse Problemdata die Konvergenzgeschwindigkeit herabsetzen können und Unglattheiten im Defekt bewirken, [1, 11] (Iterierte Defekt-Korrektur).

Die Verwendung von Schießverfahren, vgl. [27], zur numerischen Lösung von singulären Problemen wurde noch nicht sehr eingehend untersucht. Allerdings wurde die Lösung von singulären Anfangswertaufgaben in [23, 29] mit Adams-Methoden und in [24] mit Runge-Kutta Verfahren studiert.

Spezielle Modelle wurden auch mit anderen Methoden gelöst, [17, 34, 35] (Taylorreihenentwicklung), [2, 3] (Approximation mittels Bessel- und Greenfunktionen) und [15, 16] (monotone Diskretisierungsmethoden).

Randwertprobleme erster Ordnung mit einer Singularität zweiter Art wurden von de Hoog und Weiss in [22] bezüglich ihrer analytischen Eigenschaften und in [25] bezüglich der numerische Lösbarkeit diskutiert.

In dieser Arbeit beschäftigen wir uns mit folgendem Typ von nichtlinearen Anfangswertaufgaben für ein System gewöhnlicher Differentialgleichungen erster Ordnung mit einer Singularität erster Art:

$$y'(x) - \frac{M(x)}{x} \cdot y(x) = f(x, y(x)), \quad 0 < x \leq 1, \quad (1.2a)$$

$$B_0 \cdot y(0) = \beta, \quad (1.2b)$$

$$y \in \mathbf{C}[0, 1]. \quad (1.2c)$$

Dabei sind y und f vektorwertige Funktionen der Dimension n . $M(x)$ ist eine stetige, reelle $(n \times n)$ -Matrix, B_0 eine konstante, reelle $(m \times n)$ -Matrix, $m \leq n$, und β ein konstanter, reeller m -dimensionaler Vektor. Es wird vorausgesetzt, daß f stetig bezüglich x und Lipschitz-stetig bezüglich y auf einem geeignet definierten Gebiet ist.

Weiters betrachten wir Systeme zweiter Ordnung der Form

$$y''(x) - \frac{A_1(x)}{x} \cdot y'(x) - \frac{A_0(x)}{x^2} \cdot y(x) = f(x, y(x)), \quad 0 < x \leq 1, \quad (1.3a)$$

$$B_0 \cdot y(0) = \beta, \quad (1.3b)$$

$$y \in \mathbf{C}[0, 1], \quad (1.3c)$$

wobei y und f vektorwertige Funktionen der Dimension n sind. $A_0(x)$ und $A_1(x)$ sind reelle $(n \times n)$ -Matrizen, B_0 und B_1 sind konstante, reelle $(m \times n)$ -Matrizen, $m \leq 2n$, und β ist ein konstanter, reeller m -dimensionaler Vektor. Weiters wird vorausgesetzt, daß $A_0(x)$ und $A_1(x)$ stetig auf $[0, 1]$ sind und daß f stetig bezüglich x und Lipschitz-stetig

bezüglich y auf einem geeignet definierten Gebiet ist. Die Existenz und Eindeutigkeit der Lösungen dieser Probleme wurde bereits in [28] abgehandelt.

Das Ziel dieser Arbeit ist es, verschiedene Gitterstrategien an singulären Anfangswertaufgaben zu testen, um einen effizienten Algorithmus zur Integration solcher Probleme formulieren zu können.

Die Arbeit ist wie folgt organisiert: In Abschnitt 2 wird das relevante analytische Hintergrundwissen über singuläre Anfangswertprobleme und die grundlegende Theorie der numerischen Lösung von Anfangswertaufgaben und die relevante Information über die verwendete Soft- und Hardware zusammengetragen.

In Abschnitt 3 wird die Tauglichkeit von Schrittweitensteuerungen basierend auf der Schätzung des lokalen Fehlers mit Hilfe von Runge-Kutta Paaren und Richardson-Extrapolation untersucht. Um den Einfluß der Singularität auf den lokalen Fehler besser zu verstehen, wird in Abschnitt 4 die Struktur des lokalen Fehlers mit Hilfe des Computeralgebrasystems MAPLE² untersucht.

In Abschnitt 5 werden Methoden zur Schätzung des globalen Diskretisierungsfehlers, die Zadunaisky-Methode, [44], und eine mögliche Vereinfachung nach Stetter, [40], getestet. Basierend auf den Ergebnissen der vorherigen Abschnitte wird in Abschnitt 6 ein zur Integration singulärer Anfangswertaufgaben geeigneter Algorithmus hergeleitet.

²Verwendet wurde MAPLE V RELEASE 3.0 für Microsoft Windows 3.1, Copyright 1981-1994 Waterloo Maple Software.

2 Grundlagen

2.1 Singuläre Anfangswertaufgaben

Systeme erster Ordnung

Wir betrachten nichtlineare Anfangswertaufgaben für ein System gewöhnlicher Differentialgleichungen erster Ordnung mit einer Singularität erster Art,

$$y'(x) - \frac{M(x)}{x} \cdot y(x) = f(x, y(x)), \quad 0 < x \leq 1, \quad (2.1a)$$

$$B_0 \cdot y(0) = \beta, \quad (2.1b)$$

$$M(0) \cdot y(0) = 0, \quad (2.1c)$$

wobei wir annehmen, daß die Eigenwerte der Matrix $M := M(0)$ gleich Null oder ihre Realteile negativ sind. Dabei ist die Bedingung (2.1c) nach [28] bzw. [30] *notwendig und hinreichend* für $y \in \mathbf{C}[0, 1]$, wobei wir mit $\mathbf{C} := \mathbf{C}[0, 1]$ die Menge aller stetigen Abbildungen der Menge $[0, 1]$ in \mathbb{C} bezeichnen.

Man beachte, daß die rechte Seite von (2.1a) nicht Lipschitz-stetig in y gleichmäßig in x auf $[0, 1]$ ist, weil die Lipschitz-Konstante von $1/x$ abhängt. Somit kann der Existenz- und Eindeutigkeitssatz von Picard-Lindelöf nicht angewendet werden, um die Existenz und Eindeutigkeit der Lösung dieser Anfangswertprobleme zu zeigen.

Im Fall linearer Probleme mit konstanter Koeffizientenmatrix M kann man auch ohne die Lösung von (2.1a) explizit zu kennen den Wert $y'(0)$ berechnen. Unter den gegebenen Bedingungen gilt $y \in \mathbf{C}^1[0, 1]$, daher muß

$$\lim_{x \rightarrow 0} y'(x) < \infty$$

gelten. Dieser Grenzwert wird durch lokale Taylorentwicklung berechnet, und $y'(0)$ kann mittels

$$y'(0) = (I_n - M)^{-1} \cdot f(0) \quad (2.2)$$

berechnet werden, da die Inverse von $I_n - M$ laut Voraussetzung existiert.

Systeme zweiter Ordnung

Wir betrachten nun die entsprechenden nichtlinearen Anfangswertprobleme zweiter Ordnung,

$$y''(x) - \frac{A_1(x)}{x} \cdot y'(x) - \frac{A_0(x)}{x^2} \cdot y(x) = f(x, y(x)), \quad 0 < x \leq 1, \quad (2.3a)$$

$$B_0 \cdot y(0) = \beta, \quad (2.3b)$$

$$A_0(0) \cdot y(0) = 0, \quad y'(0) = 0, \quad (2.3c)$$

wobei wir annehmen, daß die Eigenwerte der Matrix $M := M(0)$ gleich Null oder ihre Realteile negativ sind. Dabei ist die Bedingung (2.3c) nach [28] bzw. [30] *notwendig und hinreichend* für $y \in \mathbf{C}[0, 1]$.

Für lineare Probleme mit konstanter Koeffizientenmatrix folgt daraus $y \in \mathbf{C}^2[0, 1]$, weswegen wiederum

$$\lim_{x \rightarrow 0} y''(x) < \infty$$

gelten muß. Aus den Taylorentwicklungen von $y(x)$ und $y'(x)$ um die Stelle $x = 0$ liefert der Übergang $x \rightarrow 0$

$$A_0 \cdot y(0) = 0, \quad (A_0 + A_1) \cdot y'(0) = 0,$$

damit dieser Grenzwert existiert. Diese Bedingungen sind erfüllt, da entweder A_0 regulär und $y(0) = 0, y'(0) = 0$ ist, oder $y(0) = \gamma_1, y'(0) = 0$ und γ_1 im Kern von A_0 gilt. Daher gilt

$$y''(0) = \left(I_n - A_1 - \frac{A_0}{2} \right)^{-1} \cdot f(0), \quad (2.4)$$

falls $(I_n - A_1 - A_0/2)$ regulär ist, was in den für die Anwendungen relevanten Fällen der Fall ist.

Mit

$$z(x) := x y'(x)$$

und

$$z'(x) = y'(x) + x y''(x) = \frac{I_n + A_1(x)}{x} \cdot z(x) + \frac{A_0(x)}{x} \cdot y(x) + x f(x, y(x))$$

ergibt sich aus (2.3a) das folgende System erster Ordnung der Dimension $2n$:

$$\begin{pmatrix} y(x) \\ z(x) \end{pmatrix}' = \frac{1}{x} \begin{pmatrix} 0 & I_n \\ A_0(x) & I_n + A_1(x) \end{pmatrix} \cdot \begin{pmatrix} y(x) \\ z(x) \end{pmatrix} + x \begin{pmatrix} 0 \\ f(x, y(x)) \end{pmatrix}.$$

Mit

$$\begin{aligned} v(x) &= \begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix} := \begin{pmatrix} y(x) \\ z(x) \end{pmatrix}, \\ M(x) &:= \begin{pmatrix} 0 & I_n \\ A_0(x) & I_n + A_1(x) \end{pmatrix}, \\ \overset{\circ}{f}(x, v(x)) &:= \begin{pmatrix} 0 \\ f(x, v_1(x)) \end{pmatrix}, \end{aligned} \quad (2.5)$$

kann das letzte System wie folgt geschrieben werden:

$$v'(x) = \frac{1}{x} M(x) \cdot v(x) + x \overset{\circ}{f}(x, v(x)), \quad 0 < x \leq 1. \quad (2.6)$$

Diese Differentialgleichung entspricht bis auf die Dimension dem Problem (2.1a) und für die Lösung $y(x)$ des Problems gilt

$$y(x) = v_1(x), \quad y'(x) = v_1'(x) = \frac{v_2(x)}{x}.$$

2.2 Testbeispiele

Als Testbeispiele betrachten wir ausgewählte singuläre Anfangswertaufgaben zweiter Ordnung, (2.3), deren Lösungen explizit bekannt sind. Die Anfangswertaufgaben werden zur numerischen Lösung auf Systeme erster Ordnung (2.6) transformiert. Die angegebenen Werte λ_1 und λ_2 entsprechen den Eigenwerten der Matrix M , bzw. $M(0)$, vgl. (2.5).

1. Wir beginnen mit linearen Anfangswertaufgaben mit konstanter Koeffizientenmatrix.

- (a) Wie Beispiel 2 in [12], Seite 26, bzw. Beispiel 2 aus [13], Seite 139, wobei die Eigenwerte auf $\lambda_1 = \lambda_2 = 0$ und die rechte Seite $f(x)$ leicht modifiziert wurden,

$$y''(x) = \frac{-1}{x}y'(x) + c x^{k-2}e^{-\alpha x}(k^2 - \alpha x(1 + 2k) + \alpha^2 x^2), \quad 0 < x \leq 1,$$

$$y(0) = 0, \quad y'(0) = 0,$$

bzw.

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot v(x) + \begin{pmatrix} 0 \\ c x^{k-1} e^{-\alpha x} (k^2 - \alpha x(1 + 2k) + \alpha^2 x^2) \end{pmatrix},$$

$$v(0) = 0, \quad v'(0) = 0,$$

mit $c = \left(\frac{\alpha}{k}\right)^k e^k$, $k = 4$ und $\alpha = 8$. Die allgemeine Lösung ist $y(x) = c_1 + c_2 \ln(x) + c x^k e^{-\alpha x}$, und die exakte Lösung $y(x) = c x^k e^{-\alpha x}$, bzw.

$$v(x) = \begin{pmatrix} c x^k e^{-\alpha x} \\ c(k - \alpha)x^{k-1} e^{-\alpha x} \end{pmatrix},$$

deren erste drei Ableitungen bei Null verschwinden.

- (b) Beispiel 2 aus [41], Seite 460,

$$y''(x) = \frac{-2}{x}y'(x) - n^2 \cos(nx) - \frac{2}{x}n \sin(nx), \quad 0 < x \leq 1,$$

$$y(0) = 2, \quad y'(0) = 0,$$

mit $n = 3$, und $\lambda_1 = 0$, $\lambda_2 = -1$. Die Lösungsschar ist $y(x) = 1 + c_1 + c_2/x + \cos(nx)$, und die exakte Lösung $y(x) = 1 + \cos(nx)$. Das transformierte Problem

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \cdot v(x) + \begin{pmatrix} 0 \\ -x n^2 \cos(nx) - 2n \sin(nx) \end{pmatrix},$$

$$v(0) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad v'(0) = 0,$$

hat die gerade Funktion $v(x) = (1 + \cos(nx), -nx \sin(nx))^T$ als Lösung.

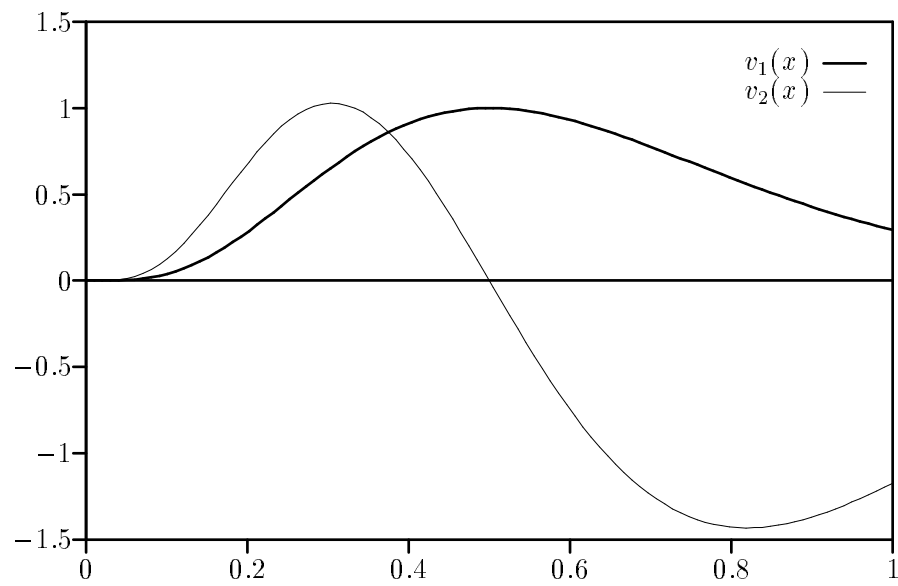


Abbildung 2.1: Die Lösungskomponenten von Beispiel 1a

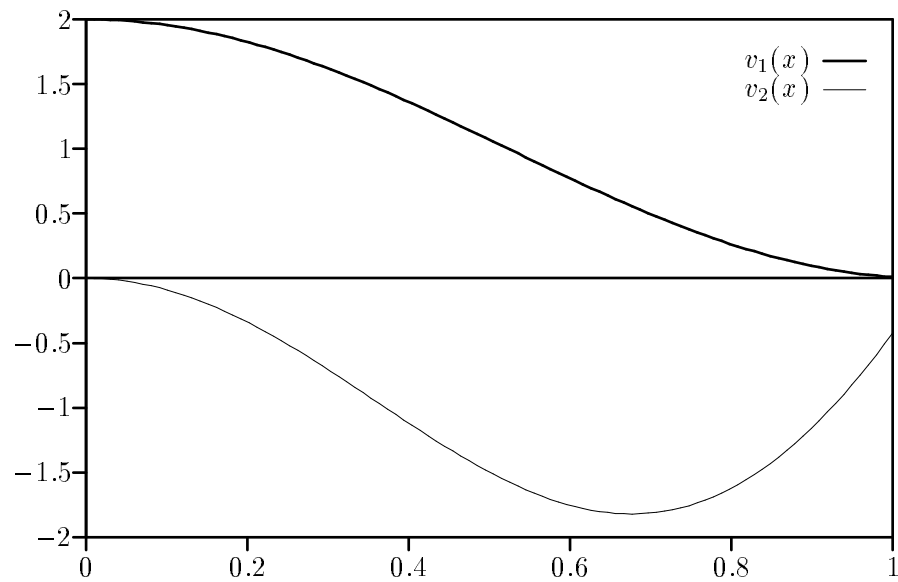


Abbildung 2.2: Die Lösungskomponenten von Beispiel 1b

- (c) In Beispiel 3 aus [11], Seite 13, wurden die Eigenwerte auf $\lambda_1 = -1$, $\lambda_2 = -2$ geändert und die Lösung $y(x)$ etwas schwieriger gemacht:

$$y''(x) = \frac{-4}{x}y'(x) + \frac{-2}{x^2}y(x) + e^{kx}(k^2x^2 + 8xk + 12), \quad 0 < x \leq 1,$$

$$y(0) = 0, \quad y'(0) = 0,$$

für $k = 2$ ergibt sich die Lösungsschar $y(x) = c_1/x + c_2/x^2 + x^2e^{kx}$. Die Lösung ist $y(x) = x^2e^{kx}$. Das transformierte Problem

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} \cdot v(x) + e^{kx} \begin{pmatrix} 0 \\ k^2x^3 + 8x^2k + 12x \end{pmatrix},$$

$$v(0) = 0, \quad v'(0) = 0,$$

hat die Lösung $v(x) = (x^2e^{kx}, (2 + kx)x^2e^{kx})^T$.

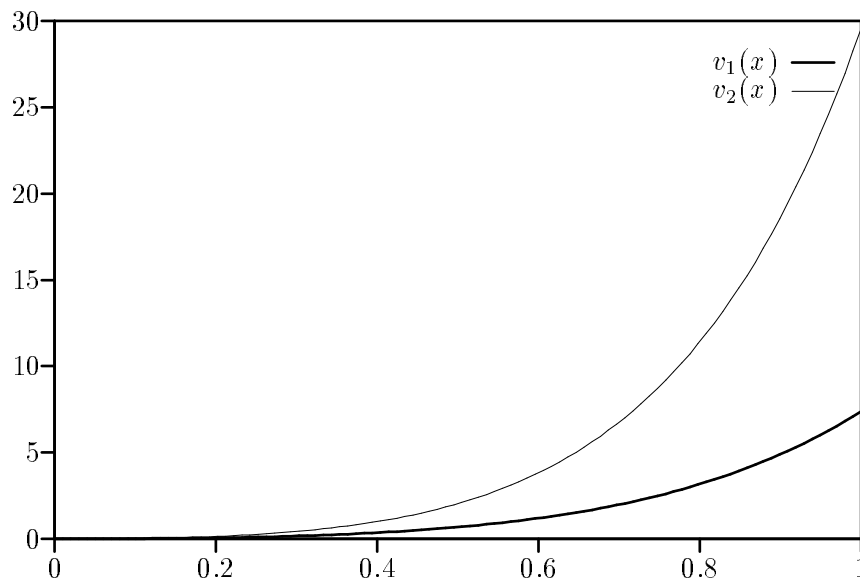


Abbildung 2.3: Die Lösungskomponenten von Beispiel 1c

2. Die nächsten Testbeispiele sind lineare Anfangswertaufgaben mit variabler Koeffizientenmatrix M .

- (a) Wie Beispiel 2 in [12], Seite 26, bzw. Beispiel 2 aus [13], Seite. 139, wobei die Eigenwerte auf $\lambda_1 = \lambda_2 = 0$ modifiziert wurden,

$$y''(x) = \frac{-1}{x}y'(x) + \alpha^2 y(x) + cx^{k-2}e^{-\alpha x}(k^2 - \alpha x(1 + 2k)), \quad 0 < x \leq 1,$$

$$y(0) = 0, \quad y'(0) = 0,$$

bzw.

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot v(x) + x \begin{pmatrix} 0 \\ \alpha^2 v_1(x) + cx^{k-2}e^{-\alpha x}(k^2 - \alpha x(1 + 2k)) \end{pmatrix},$$

$$v(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v'(0) = 0,$$

mit $c = \left(\frac{\alpha}{k}\right)^k e^k$, $k = 4$ und $\alpha = 2$. Die Lösungsschar ist $y(x) = c_1 I_0(\alpha x) + c_2 K_0(\alpha x) + cx^k e^{-\alpha x}$, wobei $I_0(x)$ die modifizierte Besselfunktion 0. Ordnung und $K_0(x)$ die dazugehörige Neumannfunktion 0. Ordnung ist. Die exakte Lösung ist, wie in Beispiel 1a, $y(x) = cx^k e^{-\alpha x}$.

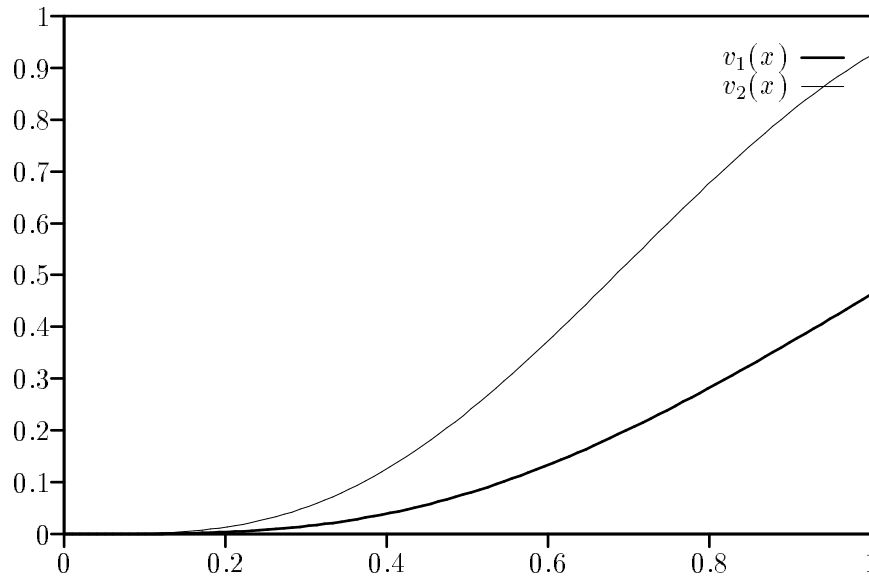


Abbildung 2.4: Die Lösungskomponenten von Beispiel 2a

(b) Mit $A = B = 1$, $\lambda_1 = \lambda_2 = 0$, ist die exakte Lösung von

$$y''(x) = \frac{-\cosh(x)}{x}y'(x) + \frac{Bx(4 + \cosh(x)) + B^2x^2}{x^2}y(x) + 2A(1 + \cosh(x))e^{Bx}, \quad 0 < x \leq 1,$$

$$y(0) = 0, \quad y'(0) = 0,$$

$y(x) = Ax^2e^{Bx}$, bzw. für

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ Bx(4 + \cosh(x)) + B^2x^2 & 1 - \cosh(x) \end{pmatrix} \cdot v(x) + x \begin{pmatrix} 0 \\ 2A(1 + \cosh(x))e^{Bx} \end{pmatrix}, \quad 0 < x \leq 1,$$

$$v(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v'(0) = 0,$$

$$v(x) = (Ax^2e^{Bx}, A(2 + Bx)x^2e^{Bx})^T.$$

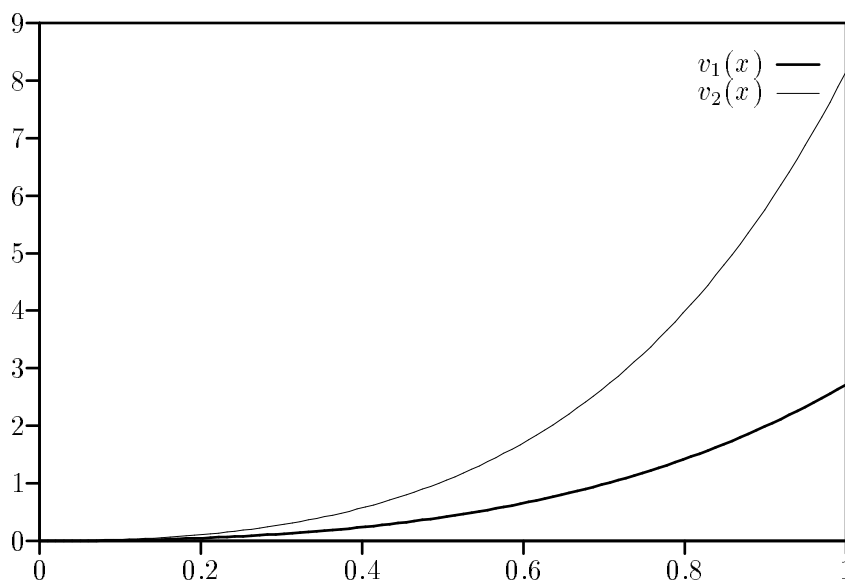


Abbildung 2.5: Die Lösungskomponenten von Beispiel 2b

3. Die dritte Gruppe der Testbeispiele bilden nichtlineare Anfangswertaufgaben. Wir wählen ein repräsentatives Beispiel aus:

- (a) Die „Emden-Differentialgleichung“, Beispiel 2 in [20], Seite 215, bzw. Beispiel 2 in [21], Seite 807, oder Beispiel 7.8 in [37], Seite 33, mit $\lambda_1 = 0$, $\lambda_2 = -1$,

$$y''(x) = \frac{-2}{x}y'(x) - y^5(x), \quad 0 < x \leq 1,$$

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

hat die exakte Lösung $y(x) = \frac{1}{\sqrt{1+x^2/3}}$. Für $v(x) = (y(x), xy'(x))^T$, wird

$$v'(x) = \frac{1}{x} \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} \cdot v(x) - x \begin{pmatrix} 0 \\ v_1^5(x) \end{pmatrix}, \quad 0 < x \leq 1,$$

$$v(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad v'(0) = 0,$$

durch die gerade Funktion

$$v(x) = \begin{pmatrix} \frac{1}{\sqrt{1+x^2/3}} \\ -\frac{x^2}{3\sqrt{1+x^2/3}} \end{pmatrix}$$

gelöst.

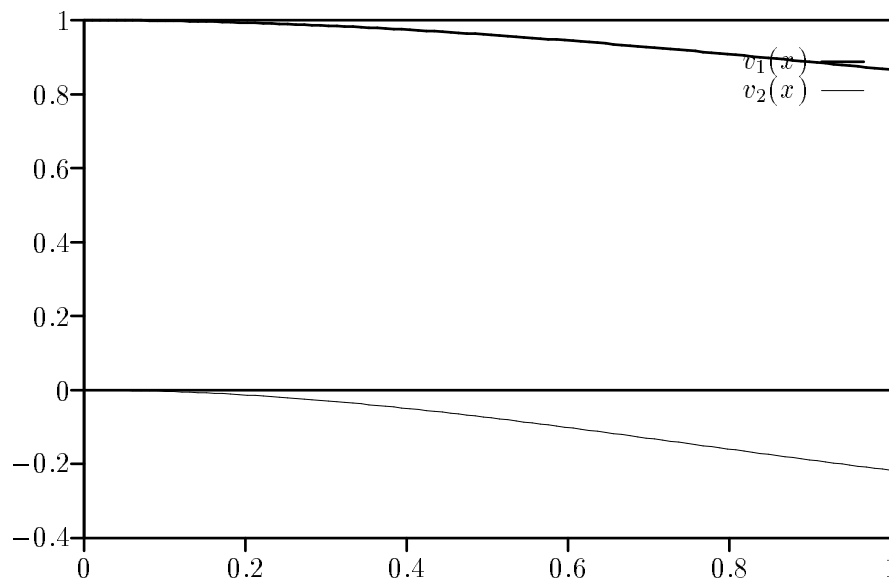


Abbildung 2.6: Die Lösungskomponenten von Beispiel 3a

2.3 Numerische Lösung von Anfangswertaufgaben

Es sei $n \in \mathbb{N}$. Wir definieren das Diskretisierungsgitter Δ_n mit den Gitterpunkten $x_j \in [a, b]$, $0 \leq j \leq n$, welche verschieden und aufsteigend geordnet sind. Wir setzen weiters $x_0 := a$ und $x_n := b$. Das heißt

$$\Delta_n := \{x_0, x_1, \dots, x_{n-1}, x_n\}$$

mit

$$a = x_0 < x_1 < \dots < x_{n-1} < x_n = b.$$

Sind die Abstände $x_{j+1} - x_j$ konstant für alle j , so spricht man von einem äquidistanten Gitter. Das Gitter wird dann mit $\Delta = \Delta_h$, $h := 1/n$, bezeichnet.

Wir betrachten die analytische Lösung $y(x)$ des klassischen Anfangswertproblems erster Ordnung

$$y'(x) = F(x, y(x)), \quad a \leq x \leq b, \quad (2.7a)$$

$$y(a) = b_0, \quad (2.7b)$$

wobei y und F vektorwertige Funktionen der Dimension n sind, und b_0 ein konstanter Vektor der Dimension n ist, und wir voraussetzen, daß F Lipschitz-stetig bezüglich y gleichmäßig in x auf $[a, b]$ ist. Mit y_Δ bezeichnen wir den $(n+1)$ -dimensionalen Vektor mit den Näherungen y_j für $y(x_j)$, $0 \leq j \leq n$, also die numerische Lösung. Dieser Vektor

$$y_\Delta := (y_0, y_1, \dots, y_n) \in \mathbb{R}^{n+1}$$

wird auch als diskrete Gitterfunktion bezeichnet, weil er jedem Punkt x_j des Gitters Δ_n einen Wert y_j zuordnet. Das Diskretisierungsgitter muß dabei nicht notwendigerweise äquidistant sein. Der Vektor der exakten Lösungswerte

$$y(\Delta) := (y(a), y(x_1), \dots, y(x_{n-1}), y(b))$$

ist ebenfalls eine Gitterfunktion.

Für jedes x_j definieren wir den globalen Fehler als

$$\varepsilon_j := y_j - y(x_j), \quad 0 \leq j \leq n,$$

bzw. als Gitterfunktion ε_Δ .

Da sowohl y_Δ als auch $y(\Delta)$ Elemente des \mathbb{R}^{n+1} sind, definieren wir ihren Abstand $\delta(y_\Delta, y(\Delta))$ als

$$\delta(y_\Delta, y(\Delta)) := |y_\Delta - y(\Delta)| = \max_{0 \leq j \leq n} (|\varepsilon_j|),$$

welches der Maximumnorm bezüglich des Gitters entspricht.

Wir beschränken uns auf Einschrittverfahren, welche (2.7) schrittweise lösen. Dabei wird $y'(x)$ durch einen einfachen Differenzenquotienten diskretisiert. Die rechte Seite von (2.7a) wird durch eine passende Inkrementfunktion $\Phi(x, y, h)$ approximiert. Das führt zu der Einschrittrekursion

$$y_0 := b_0, \quad (2.8a)$$

$$y_{j+1} := y_j + h_j \Phi(x_j, y_j, h_j), \quad 0 \leq j \leq n-1. \quad (2.8b)$$

Den Fehler, der im Schritt von x_j zu x_{j+1} gemacht wird, nennt man den lokalen Fehler in x_{j+1} ,

$$e_{j+1} := y_{j+1} - u_j(x_{j+1}), \quad 0 \leq j \leq n-1, \quad (2.9)$$

wobei $u_j(x)$ die exakte Lösung der Anfangswertaufgabe

$$\begin{aligned} u'_j(x) &= F(x, u_j(x)), & x_j \leq x \leq x_{j+1}, \\ u_j(x_j) &= y_j, \end{aligned}$$

ist.

Natürlich wird die Gleichung (2.8b) von den exakten Werten der Lösung $y(\Delta)$ nicht erfüllt. Setzt man $y(\Delta)$ in (2.8b) ein, so nennt man das entstehende Residuum den lokalen Diskretisierungsfehler,³ der durch τ_{j+1} ,

$$\tau_{j+1} := y(x_j) + h_j \Phi(x_j, y(x_j), h_j) - y(x_{j+1}), \quad 0 \leq j \leq n-1,$$

definiert ist.

Für die betrachteten Einschrittverfahren existiert bereits eine umfassende klassische Theorie bezüglich Konsistenz, Stabilität und Konvergenz. Die Ergebnisse findet man in der einschlägigen Literatur, wie z.B. in [14] und [18]. Wir konzentrieren unsere Untersuchungen auf Runge-Kutta Verfahren, die wir im allgemeinen mit RK abkürzen werden. Ein explizites s -stufiges Runge-Kutta Verfahren (ERK) ist ein Einschrittverfahren mit

$$y_{j+1} := y_j + h_j \sum_{i=1}^s b_i k_i, \quad (2.10)$$

wobei die k_i rekursiv mittels

$$\begin{aligned} k_1 &:= F(x_j, y_j), \\ k_2 &:= F(x_j + c_2 h_j, y_j + h_j a_{21} k_1), \\ &\vdots \\ k_s &:= F(x_j + c_s h_j, y_j + h_j \sum_{m=1}^{s-1} a_{sm} k_m), \end{aligned}$$

definiert sind. Das heißt, es werden s dazwischenliegende Approximationen an Punkten x_j und $x_j + c_i h_j$, $2 \leq i \leq s$, zur Berechnung von $\Phi(x_j, y_j, h_j)$ verwendet. Je nach s und den Werten der Skalare a_{im} , b_i und c_i unterscheidet man unterschiedliche Verfahren.

Man definiert die Ordnung des Runge-Kutta Verfahrens als p , wenn für ein genügend glattes Problem (2.7) die Abschätzung

$$|y(a+h) - y_1| = |e_1(h)| \leq K h^{p+1} \quad (2.11)$$

mit $K \in \mathbb{R}$ gilt, d.h. wenn der lokale Fehler für $h \rightarrow 0$ mit der Ordnung $p+1$ verschwindet.

³Dieser Fehler, in der englischen Literatur als „local truncation error“ bezeichnet, ist nicht zu verwechseln mit dem lokalen Fehler (2.9).

In FORTRAN 90, vgl. Abschnitt 2.4, gestaltet sich die Berechnung von (2.10) besonders übersichtlich.⁴ Ist $\mathbf{a}(:, :)$ die Matrix der a_{im} , $\mathbf{b}(:)$, $\mathbf{c}(:)$ die Vektoren der b_i , c_i , $\mathbf{w}(:)$ ein Hilfsvektor und $\mathbf{k}(:, :)$ die Matrix der k_i , dann berechnet Dormand, [8], y_{j+1} mittels des in Abb. 2.7 angegebenen Codes, der für Systeme beliebiger Dimension geeignet ist.

```

k(1,:) = F(x, y)           ! 1. k berechnen
DO i = 2, s                ! alle Stufen durchlaufen
  w = y + h*MATMUL(a(i,1:i-1), k(1:i-1,:))
  k(i,:) = F(x + c(i)*h, w) ! i. k berechnen
END DO
y = y + h*MATMUL(b, k)    ! neue Naeherung fuer y

```

Abbildung 2.7: FORTRAN 90 Code für ein explizites Runge-Kutta Verfahren

Ein implizites s -stufiges Runge-Kutta Verfahren (IRK) ist ein Einschrittverfahren mit (2.10), wobei die k_i Lösung des nichtlinearen $s \times n$ -dimensionalen Gleichungssystems

$$\begin{aligned}
 k_1 &:= F(x_j + c_1 h_j, y_j + h_j \sum_{m=1}^s a_{1m} k_m), \\
 &\vdots \\
 k_s &:= F(x_j + c_s h_j, y_j + h_j \sum_{m=1}^s a_{sm} k_m),
 \end{aligned}$$

mit $a_{im} \neq 0$ für $i \geq m$ sind. Diese Lösung existiert nach dem Fixpunktsatz von Banach für entsprechend kleine h_j , vgl. Satz 7.2, Abschnitt II.7, [18]. Die Berechnung von y_{j+1} ist komplizierter als für explizite Verfahren, da das nichtlineare Gleichungssystem $G(k) = 0$ mit Newton-Iteration gelöst werden muß.

Implizite Runge-Kutta Verfahren zeichnen sich durch spezielle Stabilitätseigenschaften wie z.B. A-Stabilität aus. Weiters haben implizite Verfahren mit $c_1 \neq 0$, also Radau bzw. Gauss Verfahren, den Vorteil, daß keine Auswertung von $F(x_0, y_0)$ notwendig ist. Diese ist für $a = x_0 = 0$ problematisch, da wegen $1/x$ in $F(x, y)$ diese nicht automatisch durchgeführt werden kann, sondern durch den Benutzer gesondert bereitgestellt werden muß, vgl. Abschnitt 2.1, Formeln (2.2) bzw. (2.4).

2.4 Hard- und Softwarespezifikation

Grundsätzlich wurden die Testroutinen in FORTRAN 90 codiert, da die vektorielle Syntax die Programmierung sehr erleichtert, vgl. Abb. 2.7, bzw. [31]. Verwendet wurde dazu der ESSENTIAL LAHEY FORTRAN 90 COMPILER RELEASE 3.00A⁵ (ELF90), der über Internet frei verfügbar ist. ELF90 läuft unter MS-DOS und erzeugt 32-Bit Code. In manchen Fällen mußte jedoch auch auf FORTRAN 77 zurückgegriffen werden, wenn benötigte Bibliotheken nur dort verfügbar waren, bzw. Codes aus älteren Büchern, z.B.

⁴Man vergleiche dazu auch die entsprechende Algorithmen in [33].

⁵Copyright 1994-1997 Lahey Computer Systems.

[19], verwendet wurden. Generell wurde der Fortran Datentyp `REAL(kind = 8)` bzw. `DOUBLE PRECISION` verwendet, um die Rechenfehler möglichst klein zu halten.

Sämtliche Tests wurden auf PCs mit Intel Prozessoren durchgeführt. Die verwendete Arithmetik ist binär mit Round-to-Nearest Rundung. Der Datentyp `DOUBLE PRECISION` entspricht der Gleitkommaarithmetik

$$(2, 53, 1023, -1022, false). \quad (2.12)$$

Der maximale dezimale Wertebereich von `DOUBLE PRECISION` für normierte Zahlen der verwendeten Maschine ist daher

$$2.23 \cdot 10^{-308} \leq |x| \leq 1.8 \cdot 10^{308}$$

und für subnormale Zahlen

$$9.88 \cdot 10^{-324} \leq |x| \leq 2.23 \cdot 10^{-308}.$$

Der maximale relative Rundungsfehler `EPS` ist somit

$$\text{EPS} = \frac{2^{1-53}}{2} = 2^{-53} = 1.11 \cdot 10^{-16},$$

weil Round-to-Nearest Rundung ausgeführt wird.

3 Schätzung des lokalen Fehlers

Zur Schrittweitensteuerung mit Hilfe des lokalen Fehlers wird üblicherweise wie folgt vorgegangen. Zuerst schätzt man den lokalen Fehler des letzten Schrittes zu y_j , man berechnet also eine Näherung d_j für den lokalen Fehler e_j . Aufgrund der Größenordnung von d_j im Vergleich mit einer vorgegebenen Toleranz T wird entschieden, ob der letzte Schritt angenommen oder verworfen wird. Wird er angenommen, so wird mit Hilfe von d_j die Schrittweite für den nächsten Schritt vorhergesagt, andernfalls wird der Schritt mit der eben berechneten Schrittweite wiederholt. Dadurch wird versucht den relativen Fehler knapp unter der Toleranz zu halten, wodurch unnötig kleine Schrittweiten vermieden werden. Dabei wird vorausgesetzt, daß der lokale und der globale Fehler korreliert sind, da man ja prinzipiell an den globalen Fehler Genauigkeitsanforderungen stellt. Dies ist für reguläre Probleme fast immer der Fall, daher ist die hier beschriebene Methode die meist verwendete.

3.1 Runge-Kutta Paare

Dieser Abschnitt basiert auf [8], Abschnitt 5.3, bzw. [18], Abschnitt II.4. Die Idee ist, zwei verschiedene Näherungslösungen y_j und \hat{y}_j mit zwei ERK der Ordnungen p und $q > p$ zu berechnen, wobei man mit dem gleichen Wert y_{j-1} startet. Man definiert

$$d_j := y_j - \hat{y}_j.$$

Setzt man darin (2.9) ein, so folgt

$$d_j = y_j - \hat{y}_j = u_{j-1}(x_j) + e_j - u_{j-1}(x_j) - \hat{e}_j = e_j - \hat{e}_j. \quad (3.1)$$

Es gilt $d_j \approx e_j$, wenn $|e_j| \gg |\hat{e}_j|$ ist, d.h. wenn die Schrittweite klein genug ist, daß die zusätzlichen h^{q-p} , vgl. (2.11), \hat{e}_j sehr klein machen. Unter geeigneten Voraussetzungen und für $|d_j| \leq T$ kann die nächste Schrittweite dann mit

$$h_{j+1} := 0.9 h_j \left(\frac{T}{|d_j|} \right)^{\frac{1}{p+1}} \quad (3.2)$$

vorausgesagt werden, wobei 0.9 ein klassischer „Sicherheitsfaktor“ ist.

In unseren Fällen gilt $q = p + 1$. Aus Gründen der Effizienz verwendet man Runge-Kutta Paare mit den selben Skalaren a_{im} und c_i , $1 \leq m \leq i - 1$, $2 \leq i \leq s$, sogenannte „embedded Runge-Kutta pairs“. Dadurch reduziert sich der Aufwand auf s zusätzliche Multiplikationen und Additionen gegenüber einem einfachen Runge-Kutta Verfahren, vgl. [8], Abschnitt 5.6, bzw. [18] Abschnitt II.4.

Obwohl d_j eine Schätzung von e_j ist, nimmt man natürlich den genaueren Wert \hat{y}_j für die weiteren Berechnungen, was mit lokaler Extrapolation bezeichnet wird. Wir schreiben für ein Runge-Kutta Paar mit den Ordnungen p und q kurz $q(p)$ Paar. Prinzipiell gilt ein $q(p)$ Paar als Verfahren mit Ordnung p , da für diese die Schätzung des Fehlers berechnet wird.

Für die folgenden Abschnitte wurde Dormands Programm `RKmbed.f90`, [8], Seite 293, leicht modifiziert, wobei vor allem die Ein- und Ausgabe allgemeiner gestaltet wurde.

Sind $\mathbf{b}(\cdot)$, $\mathbf{b1}(\cdot)$ die Vektoren der Koeffizienten \hat{b}_i , b_i zu den Ordnungen q und p , dann berechnet man $|d_j|$, im Code \mathbf{d} , mit

```
 $\mathbf{d} = \mathbf{h} * \text{MAXVAL}(\text{ABS}(\text{MATMUL}(\mathbf{b1} - \mathbf{b}, \mathbf{k})))$ 
```

Die Berechnung von h_{j+1} erfolgt genau wie in (3.2).

3.1.1 Runge-Kutta 5(4) Paar (DOPRI5)

Für dieses Runge-Kutta Paar wurden die Koeffizienten aus [8], Seite 84, Abb. 5.4, gewählt. Es sind zwei sieben-stufige ERK mit Ordnungen $p = 4$ und $q = 5$, welche auch mit DOPRI5 für Dormand-Prince, [7], bezeichnet werden. Dieses Runge-Kutta Paar dient uns als Basismethode.

Mit der Startschrittweite $h_1 := 0.1$ wurden Testprobleme mit 3 verschiedenen Toleranzen T für den lokalen Fehler, $T = 1\text{E}-6$, $1\text{E}-9$ und $1\text{E}-12$, gelöst. Die logarithmisch skalierten Graphen zeigen jeweils das von RK45.F90 automatisch gewählte Gitter (Karo), die Maximumnorm des globalen Fehlers, $|\varepsilon_j|$, $0 \leq j \leq n$, (dicke Linie), die Schätzung des lokalen Fehlers, $|d_j|$, (punktierte Linie) und die Maximumnorm des exakten lokalen Fehlers, $|e_j|$, (dünne Linie), falls die allgemeine Lösung bekannt ist. Dies trifft prinzipiell nur auf die linearen Beispiele mit konstanter Koeffizientenmatrix zu.

Zum besseren Überblick über die gewählten Schrittweiten ergänzt eine x_j zu h_j Grafik die Testreihe. Dabei wurde in jedem Punkt x_j die Schrittweite als y -Wert aufgetragen, mit welcher der Punkt erreicht wurde, also $x_j - x_{j-1}$.

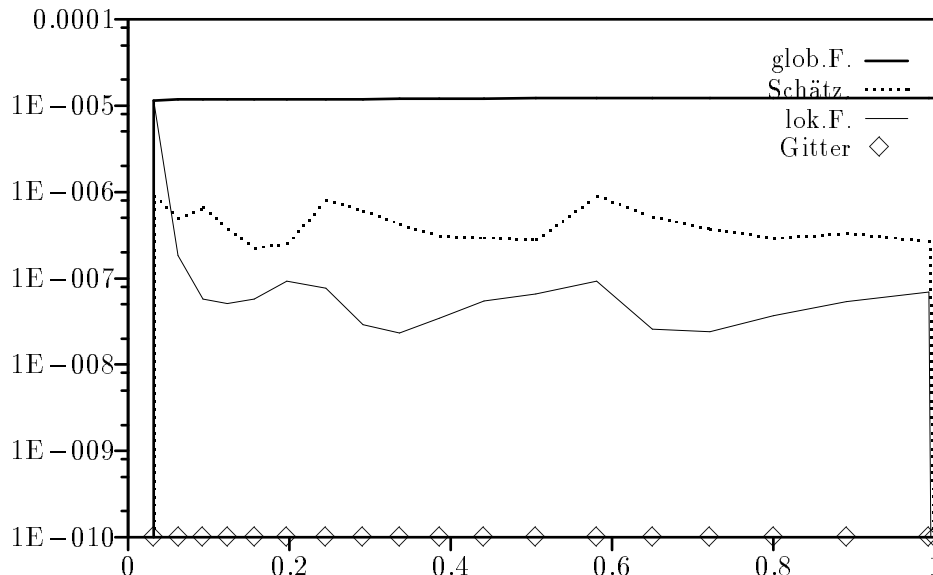
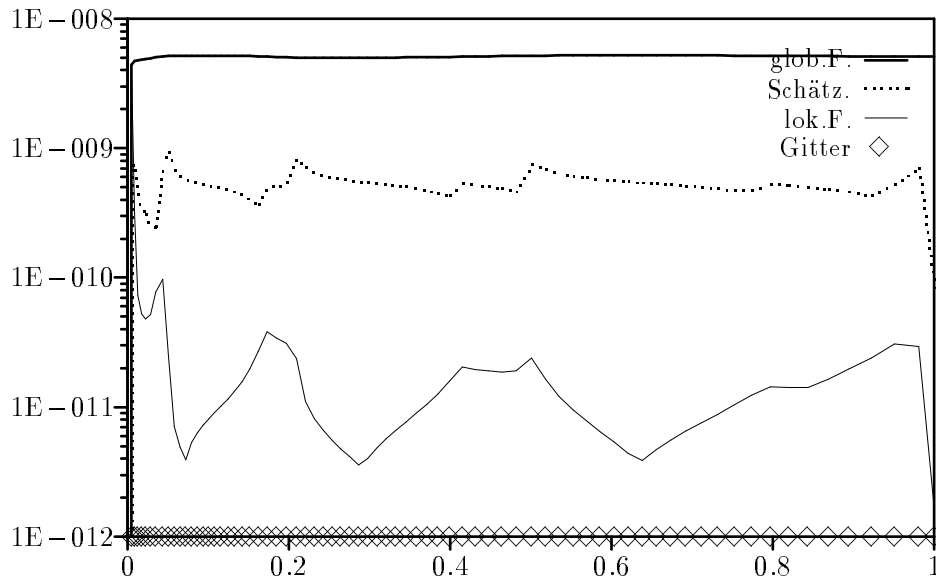
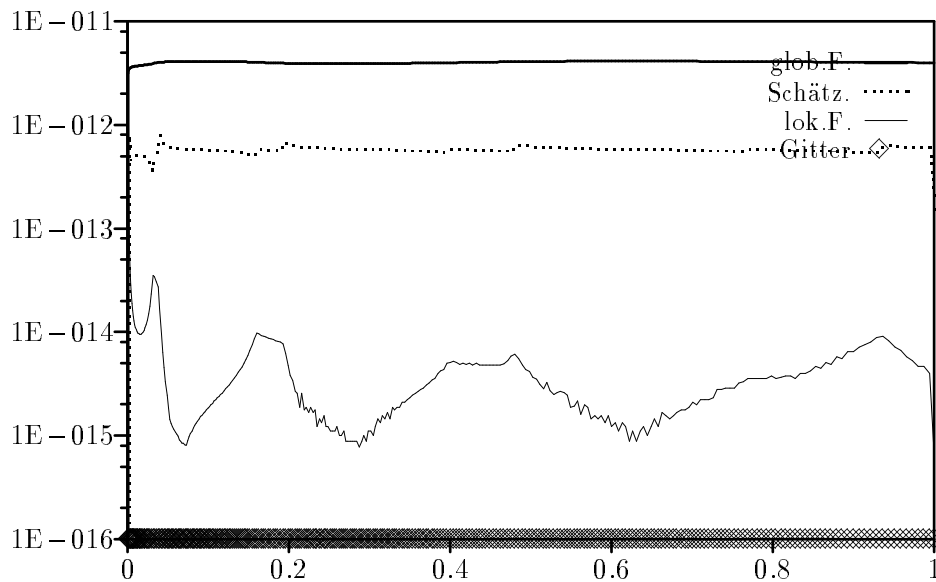
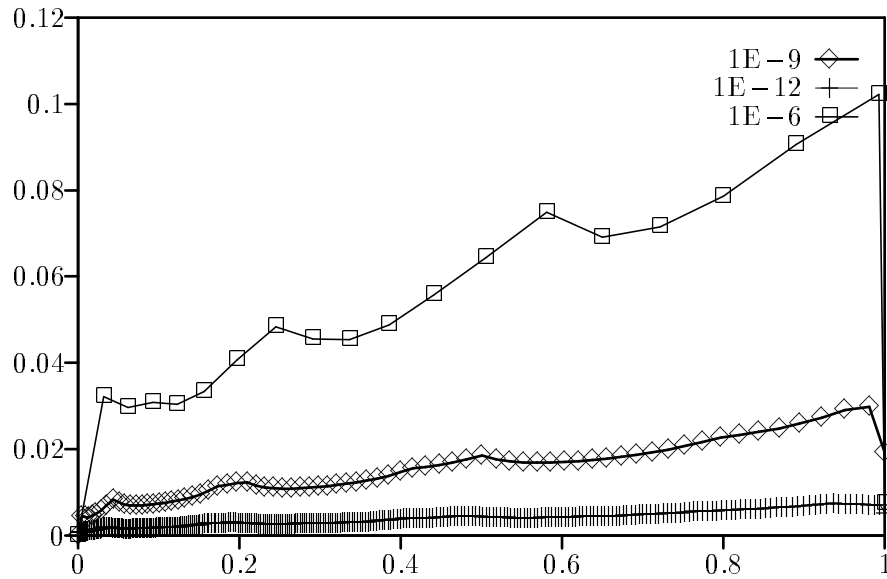


Abbildung 3.1: Fehler 5(4) Paar in Beispiel 1a, Toleranz $1\text{E}-6$

Abbildung 3.2: Fehler 5(4) Paar in Beispiel 1a, Toleranz $1E-9$ Abbildung 3.3: Fehler 5(4) Paar in Beispiel 1a, Toleranz $1E-12$

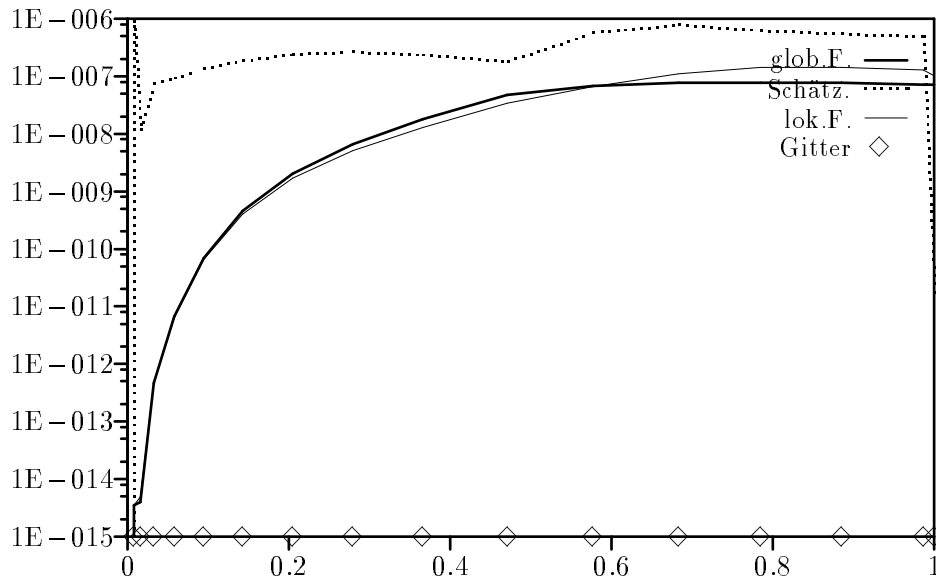
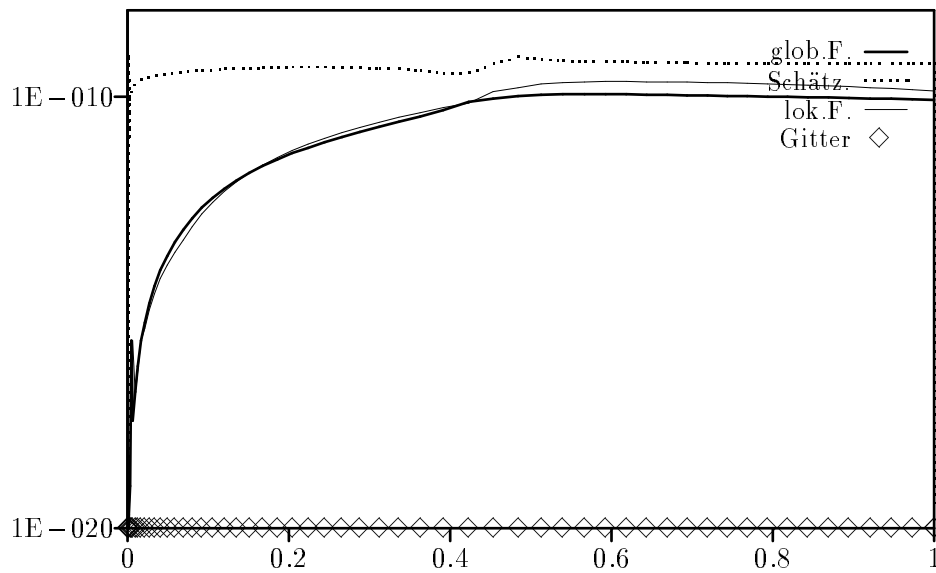
Abbildung 3.4: $x_j h_j$ -Plot 5(4) Paar für Beispiel 1a

5(4) Paar für Beispiel 1a: In Abb. 3.1 folgt die Schätzung des lokalen Fehlers diesem qualitativ relativ gut, nur die erste Spitze wird um eine Zehnerpotenz unterschätzt. Weiters, vgl. Abb. 3.2 und 3.3, wird der lokale Fehler bei steigender Genauigkeitsanforderung stärker überschätzt. Der lokale Fehler nimmt bei der Singularität mit einer Spitze den größten Wert an. Der globale Fehler nimmt jedoch bei kleinerer lokaler Toleranz entsprechend von etwa $1E-5$ über $1E-8$ auf $1E-11$ ab.

5(4) Paar für Beispiel 1b: Im Gegensatz zu Beispiel 1a zeigt sich hier ein etwas anderes Verhalten. In Abb. 3.5 ist eine markante Spitze in der Schätzung des lokalen Fehlers zu erkennen. Daraus resultierend wurden in Abb. 3.7 am Anfang des Intervalls viele kleine Schritte durchgeführt, obwohl die Lösung dort glatt ist. Der lokale Fehler selber zeigt keine Spitze, wie etwa in Abb. 3.1-3.3.

5(4) Paar für Beispiel 2a: Abb. 3.8 bis 3.10 zeigen durchwegs markante Spitzen im lokalen Fehler am Beginn des Intervalls, die von der Schätzung mit bis zu zwei Zehnerpotenzen unterschätzt werden. Weg von der Singularität ist die Schätzung des lokalen Fehlers zwar zu hoch, folgt aber dem Verlauf des lokalen Fehlers gut. Der globale Fehler nimmt entsprechend den lokalen Toleranzen ab, vgl. Beispiel 1a.

5(4) Paar für Beispiel 2b: Wie in Beispiel 1b zeigt die Schätzung des lokalen Fehlers in den Abb. 3.12 und 3.13 ausgeprägte Spitzen am Anfang des Intervalls. Obwohl die Lösung dort nur kleine höhere Ableitungen besitzt, wird eine kleine Schrittweite gewählt, vgl. Abb. 3.14. Der globale Fehler verbessert sich entsprechend der höheren lokalen Genauigkeit, auffallend ist jedoch der geringe Wert zwischen 0.1 und 0.2 in Abb. 3.13.

Abbildung 3.5: Fehler 5(4) Paar in Beispiel 1b, Toleranz $1E-6$ Abbildung 3.6: Fehler 5(4) Paar in Beispiel 1b, Toleranz $1E-9$

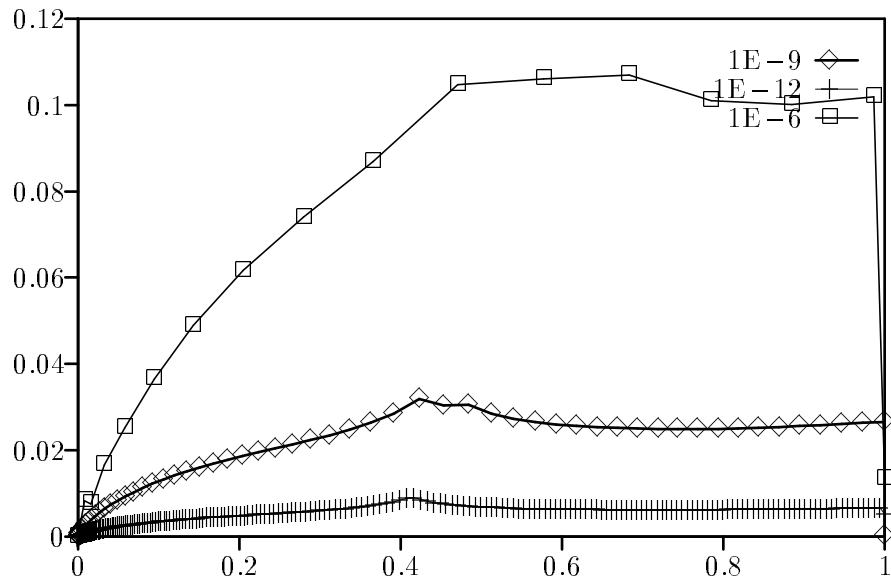


Abbildung 3.7: $x_j h_j$ -Plot 5(4) Paar für Beispiel 1b

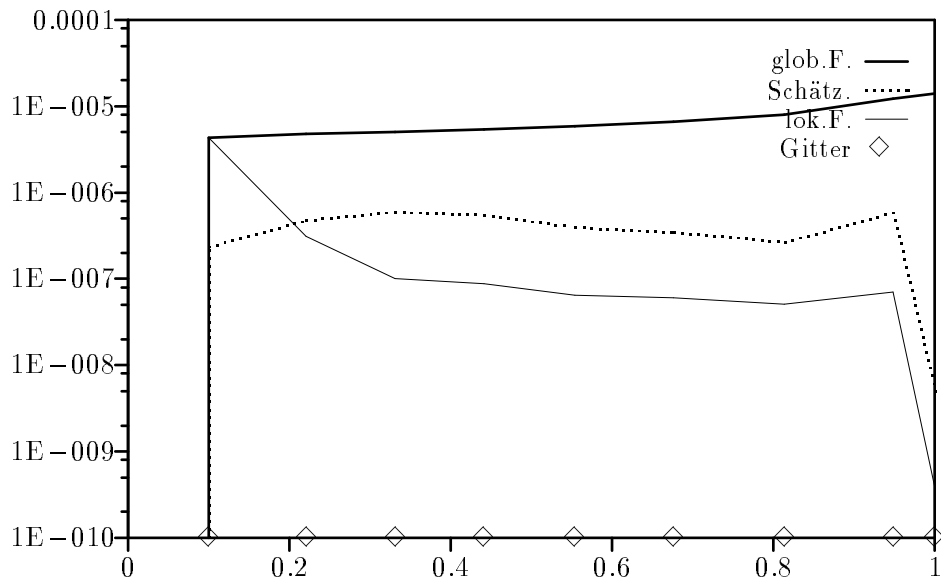
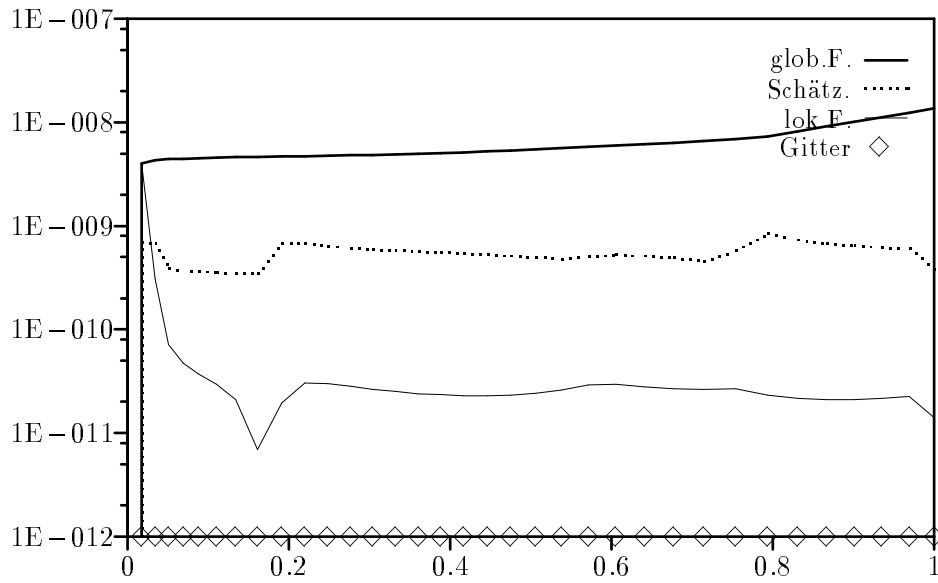
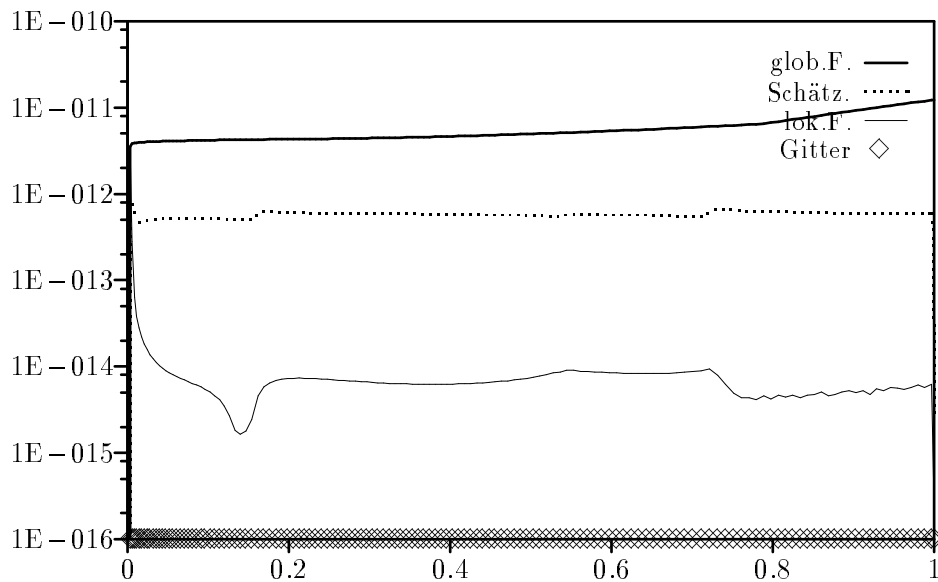


Abbildung 3.8: Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-6$

Abbildung 3.9: Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-9$ Abbildung 3.10: Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-12$

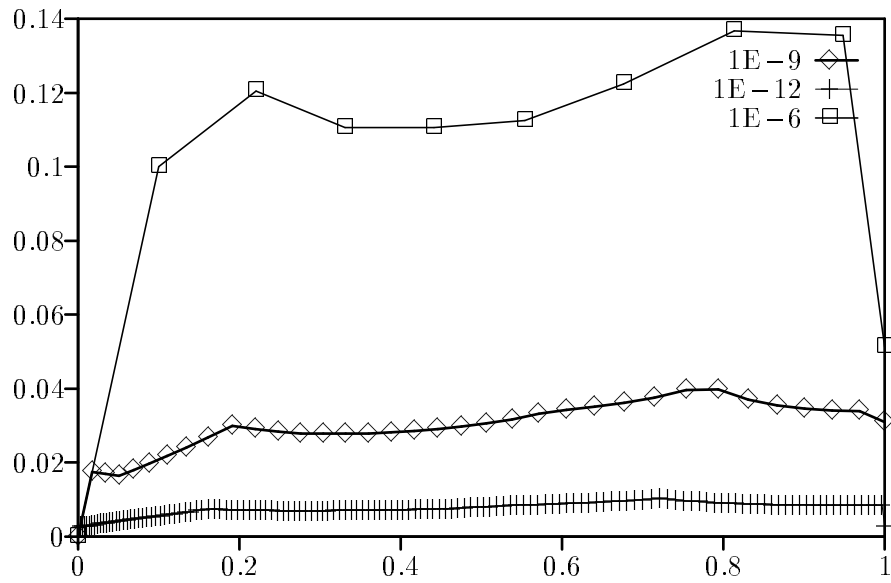


Abbildung 3.11: $x_j h_j$ -Plot 5(4) Paar für Beispiel 2a

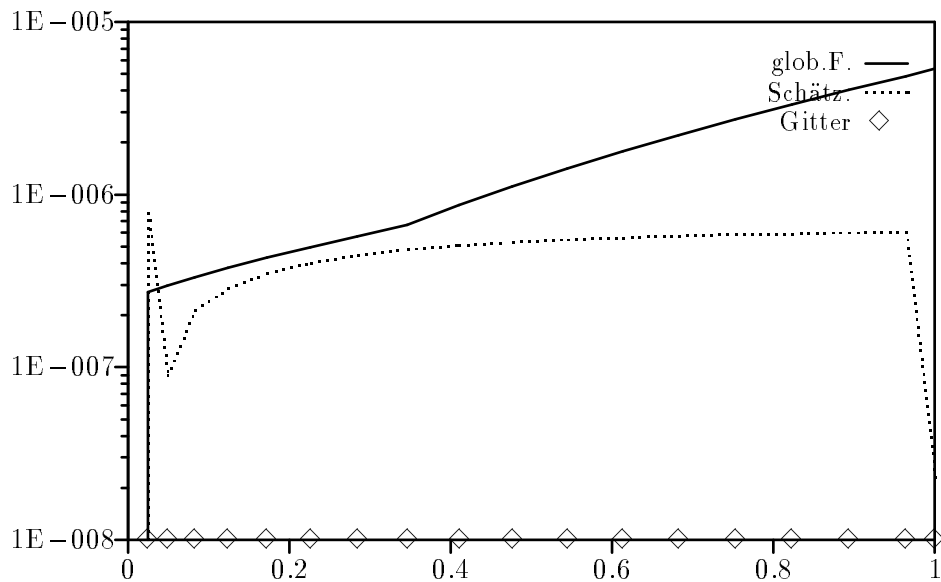
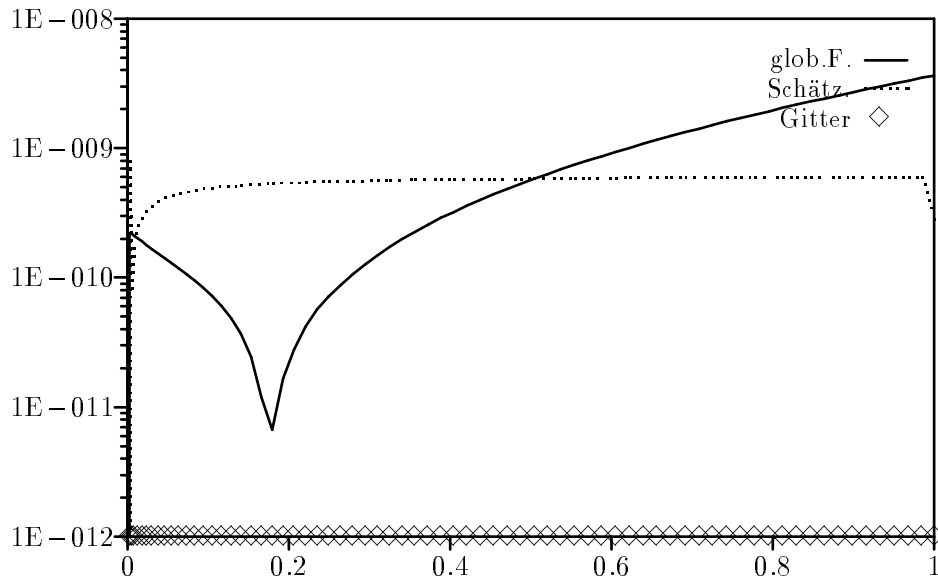
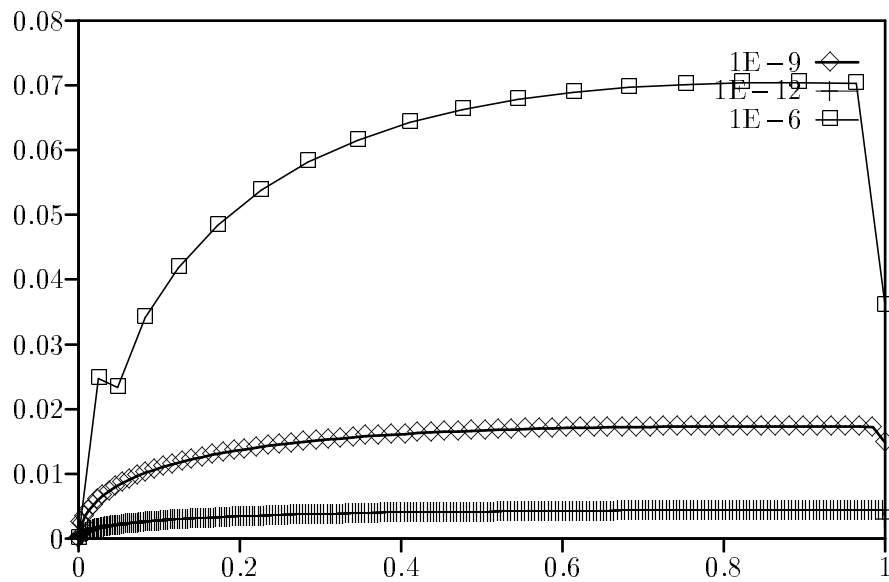
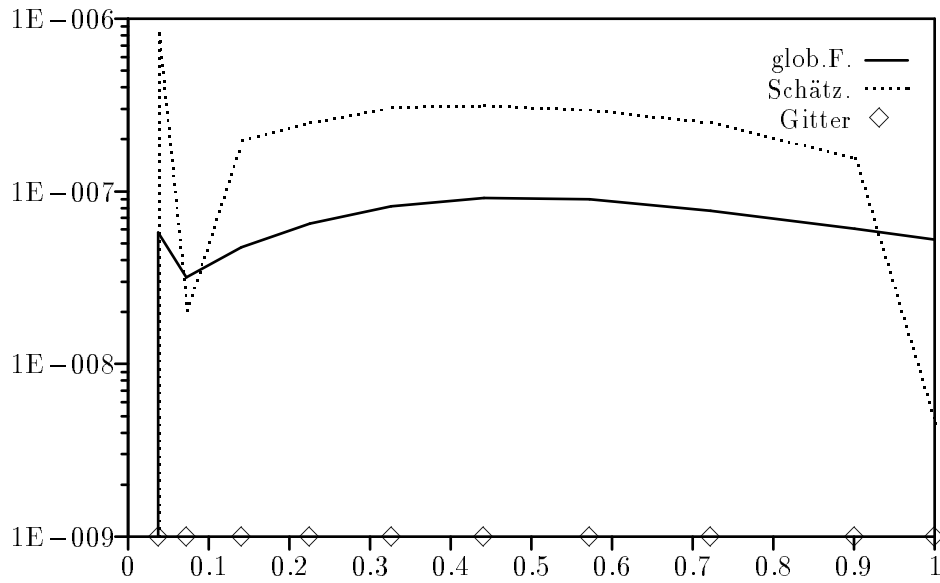
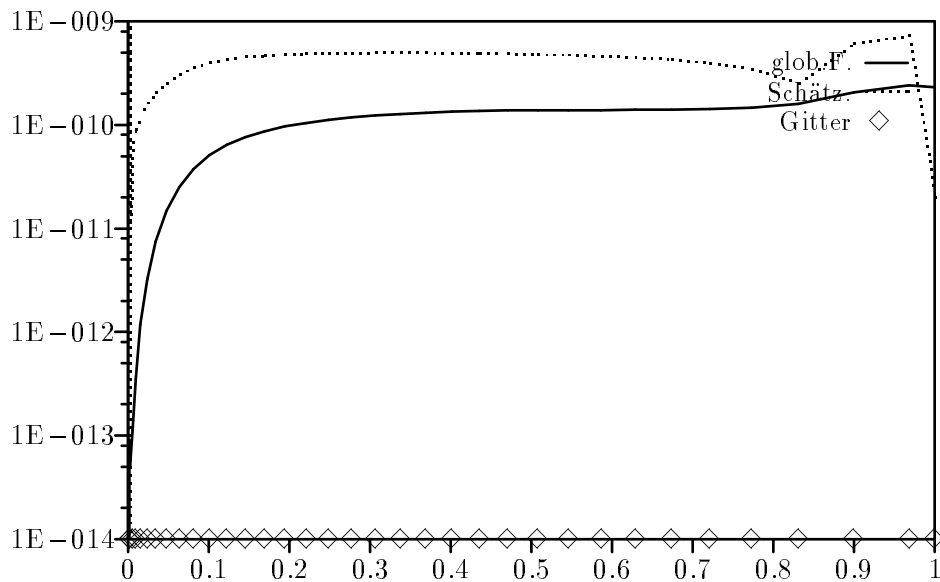
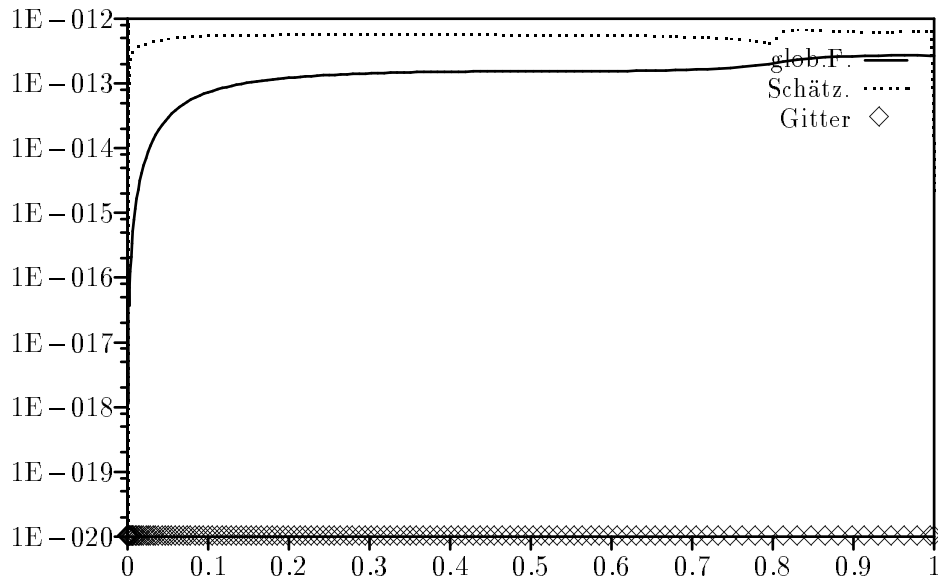
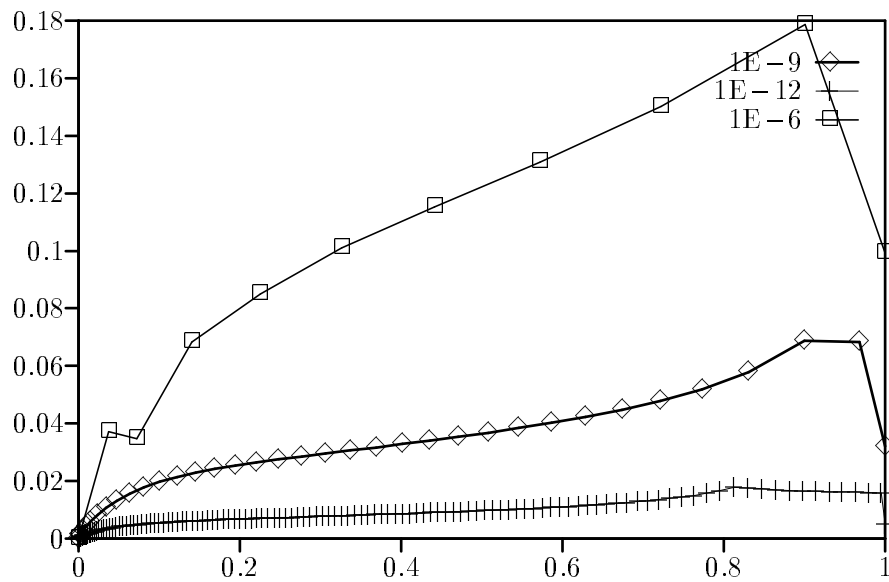


Abbildung 3.12: Fehler 5(4) Paar in Beispiel 2b, Toleranz $1E-6$

Abbildung 3.13: Fehler 5(4) Paar in Beispiel 2b, Toleranz $1E-9$ Abbildung 3.14: $x_j h_j$ -Plot 5(4) Paar für Beispiel 2b

Abbildung 3.15: Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-6$ Abbildung 3.16: Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-9$

Abbildung 3.17: Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-12$ Abbildung 3.18: $x_j h_j$ -Plot 5(4) Paar für Beispiel 3a

5(4) Paar für Beispiel 3a: In Abb. 3.15 ist am Beginn der Integration eine stark ausgeprägte Spitze in der lokalen Fehlerschätzung zu erkennen, die weniger stark sichtbar auch in den Abb. 3.16 und 3.17 vorhanden ist. Die globalen Fehler nehmen mit dem Verkleinern der lokalen Toleranzen entsprechend ab. In Abb. 3.18 ist, wie auch schon in den Beispielen 1b und 2b, die Wahl der kleinen Schrittweiten am Beginn des Intervalls zu erkennen.

3.1.2 Runge-Kutta 3(2) Paar

Um zu überprüfen, welchen Einfluß die Genauigkeit des Verfahrens hat, wurde ein weiteres Runge-Kutta Paar mit Koeffizienten aus [8], Seite 79, Abb. 5.1, gewählt. Dies sind zwei drei-stufige ERK mit $p = 2$ und $q = 3$. Zum Vergleich mit Abschnitt 3.1.1 wurden einige Testbeispiele numerisch gelöst. Die von DOPRI5 gewählten Schrittweiten bewegten sich etwa in $[0.01, 0.1]$. Bei gleichen Schrittweiten lieferte das 3(2) Paar nach (2.11) daher um 10^2 bis 10^4 mal schlechtere Lösungen. Daher wurden die Toleranzen für den lokalen Fehler auf $T = 1E-3$ und $1E-6$ reduziert.

Wir geben hier eine kurze Zusammenfassung der in [30] ausführlich durchgeführten Tests: Die Ergebnisse bezüglich der Schrittweitenwahl für die lokale Fehlertoleranz von $1E-3$ konnten daher mit $1E-6$ von DOPRI5 verglichen werden. Es zeigte sich, daß der lokale Fehler um das bis zu 1000-fache überschätzt wird und ebenfalls die Tendenz für eine Spitze am Beginn des Intervalls zeigt. Der globale Fehler nimmt bei fallenden Toleranzen jeweils um drei Zehnerpotenzen ab, wie erwartet. Alle Beispiele zeigten im Vergleich mit DOPRI5 qualitativ das selbe Verhalten, am Anfang sind die Schrittweiten klein und werden dann größer.

3.1.3 Runge-Kutta 8(7) Paar (DOPRI8)

Weiters wurden einige Testbeispiele mit einem 8(7) Paar gelöst, um den Einfluß der höheren Genauigkeit zu testen. Dazu wurde ein 13-stufiges Runge-Kutta Paar mit Ordnungen $p = 7$ und $q = 8$ gewählt, [18], Seite 195, Abb. 6.4. Von den durchgeführten Tests mit lokalen Toleranzen $1E-9$ und $1E-12$, sind die mit $T = 1E-9$ mit denen zu $T = 1E-6$ von DOPRI5 vergleichbar.

Die Ergebnisse der in [30] durchgeführten Tests zeigen gleich hohe und teilweise sogar fast um eine Zehnerpotenz deutlicher ausgeprägte Spitzen im lokalen Fehler an der Singularität, wie die entsprechenden Ergebnisse von DOPRI5. Im Verlaufe der Integration ist die Überschätzung des lokalen Fehlers ebenfalls im Bereich von ein bis zwei Zehnerpotenzen. Der globale Fehler verhält sich dafür noch geradliniger und variiert kaum entlang des Intervalls, und nimmt für die Reduktion von T entsprechend ab.

3.1.4 Runge-Kutta 4(5) Paar

Das verwendete Runge-Kutta Paar ist das gleiche wie in Abschnitt 3.1.1. Um die Überschätzung des lokalen Fehlers, die sich in diesen Abschnitten gezeigt hat, zu überprüfen, folgen wir, im Gegensatz zu Abschnitt 3.1.1, der der Lösung y_j des ERK zur Ordnung p , wofür die Schätzung des lokalen Fehlers eigentlich berechnet wird.

Getestet wurden zwei lineare Beispiele mit konstanter Koeffizientenmatrix, Beispiel 1a und 1b, und eines mit variabler Koeffizientenmatrix, 2a. Für nichtlineare Probleme kann dieser Test nicht durchgeführt werden, da der exakte lokale Fehler nicht bekannt ist, und daher der Grad der möglichen Überschätzung nicht festgestellt werden kann.

4(5) Paar für Beispiel 1a: In Abb. 3.19 ist deutlich die Spitze im lokalen Fehler und das Versagen der lokalen Fehlerschätzung des 4(5) Paares in diesem Bereich zu erkennen. Der lokale Fehler wird um eine Zehnerpotenz unterschätzt, wie auch schon in Abb. 3.1 für DOPRI5. Weg von der Singularität folgt die lokale Fehlerschätzung den Fehler sehr genau. Mit steigender Ordnung bzw. kleiner werdender Schrittweite nimmt die Ausprägung der Spitze im lokalen Fehler an der Singularität zu, vgl. Abb. 3.20.

4(5) Paar für Beispiel 1b: In Abb. 3.21 und 3.22 liegen die lokalen Fehler des 4(5) Paares teilweise bis zu einer Zehnerpotenz über dem Schätzer. In Abb. 3.21 ist die Spitze im lokalen Fehler an der Singularität gut zu erkennen, während sie in Abb. 3.22 so schmal ist, daß sie auf der Grafik nicht zu sehen ist.

4(5) Paar für Beispiel 2a: Die Abb. 3.23 und 3.24 lassen folgendes erkennen. Am Beginn der Integration hat der lokale Fehler eine Spitze, welche von der lokalen Fehlerschätzung des 4(5) Paares bis zu einer Zehnerpotenz unterschritten wird, vgl. Abschnitt 3.1.1, Abb. 3.8 und 3.9. Ab etwa $x = 0.2$ gibt die Schätzung den lokalen Fehler gut wieder.

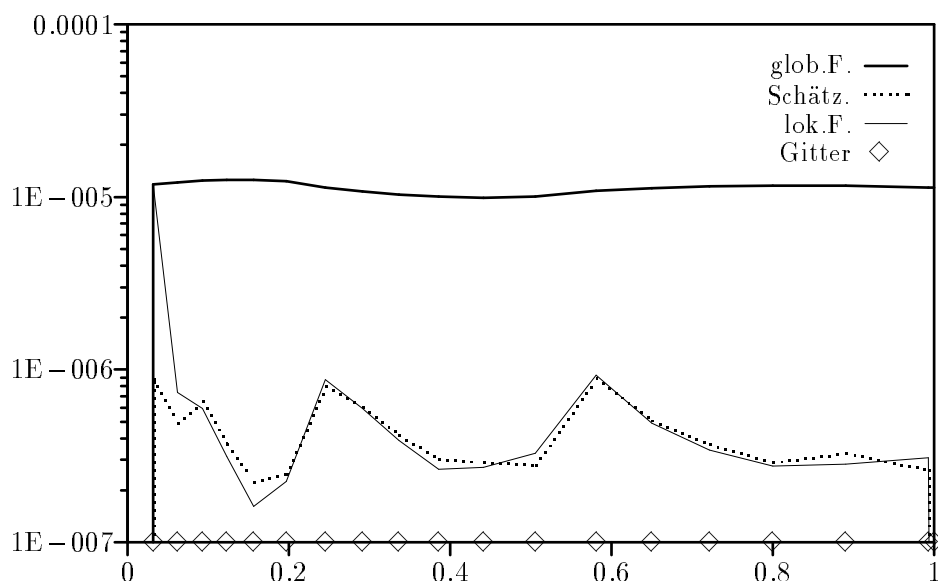
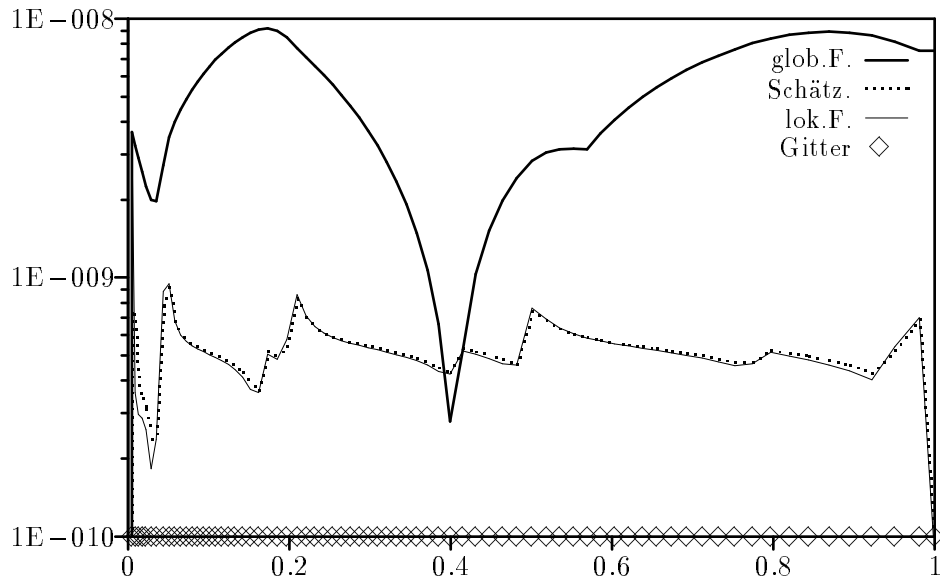
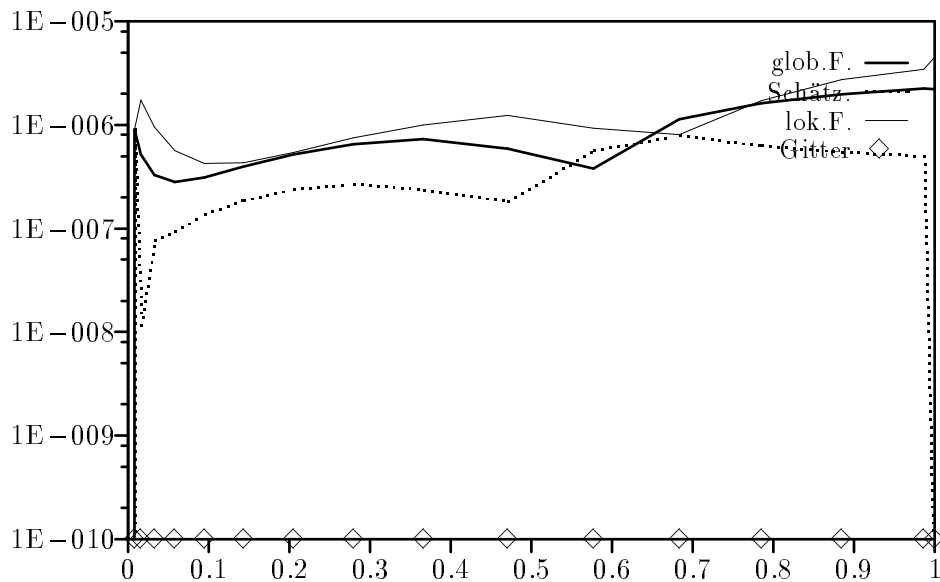
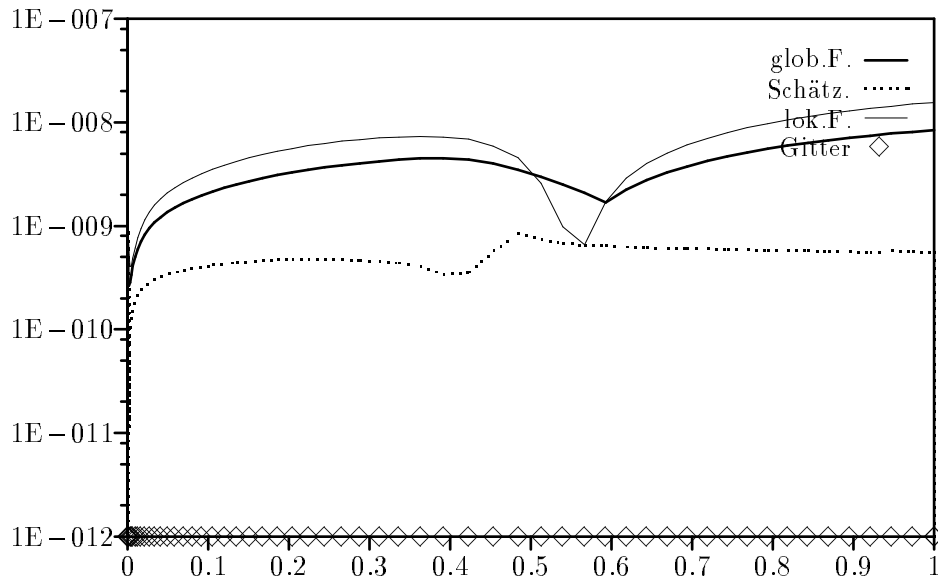
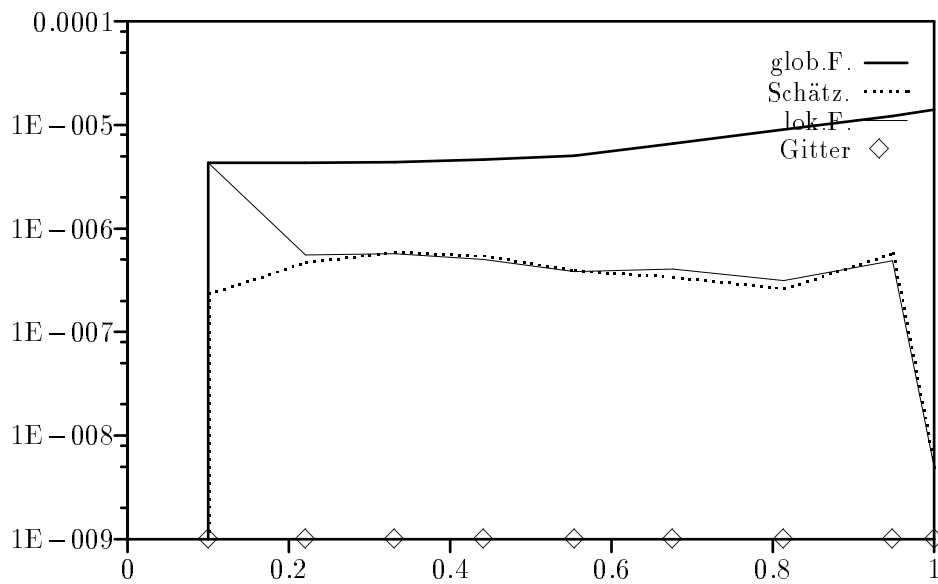
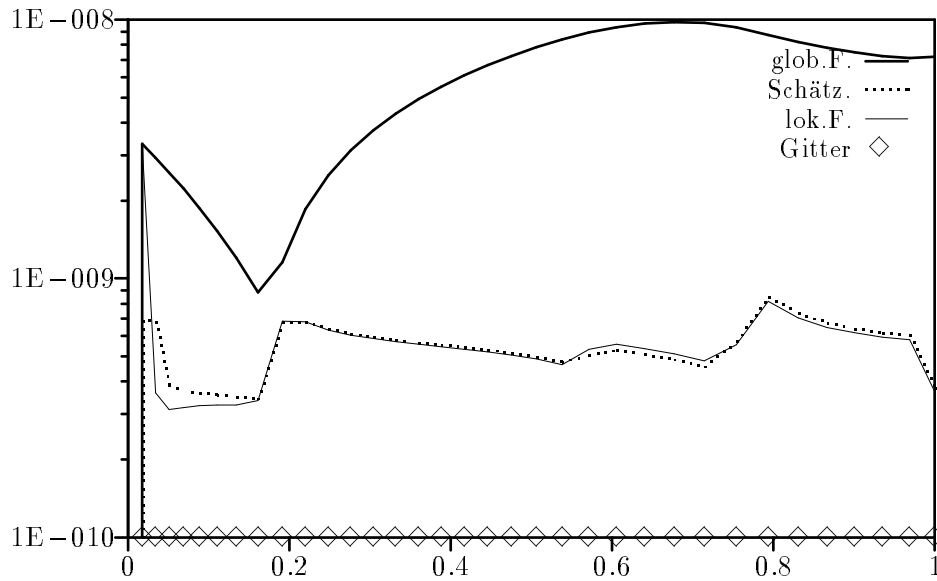


Abbildung 3.19: Fehler 4(5) Paar in Beispiel 1a, Toleranz $1E-6$

Abbildung 3.20: Fehler 4(5) Paar in Beispiel 1a, Toleranz $1E-9$ Abbildung 3.21: Fehler 4(5) Paar in Beispiel 1b, Toleranz $1E-6$

Abbildung 3.22: Fehler 4(5) Paar in Beispiel 1b, Toleranz $1E-9$ Abbildung 3.23: Fehler 4(5) Paar in Beispiel 2a, Toleranz $1E-6$

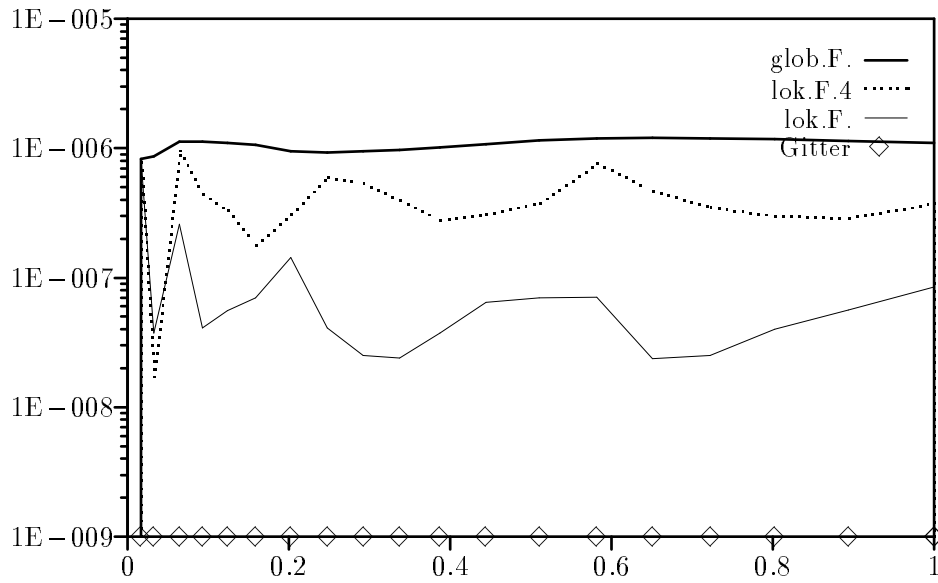
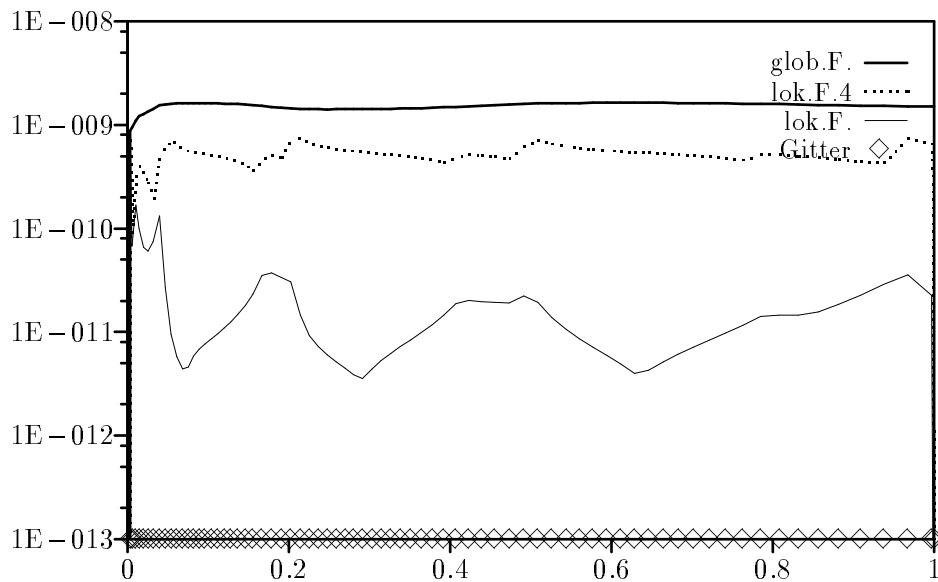
Abbildung 3.24: Fehler 4(5) Paar in Beispiel 2a, Toleranz $1E-9$

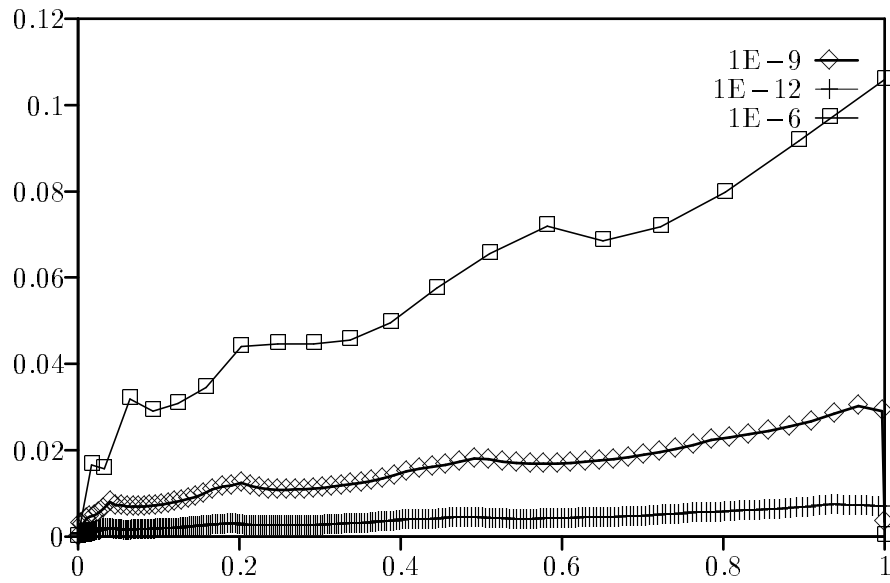
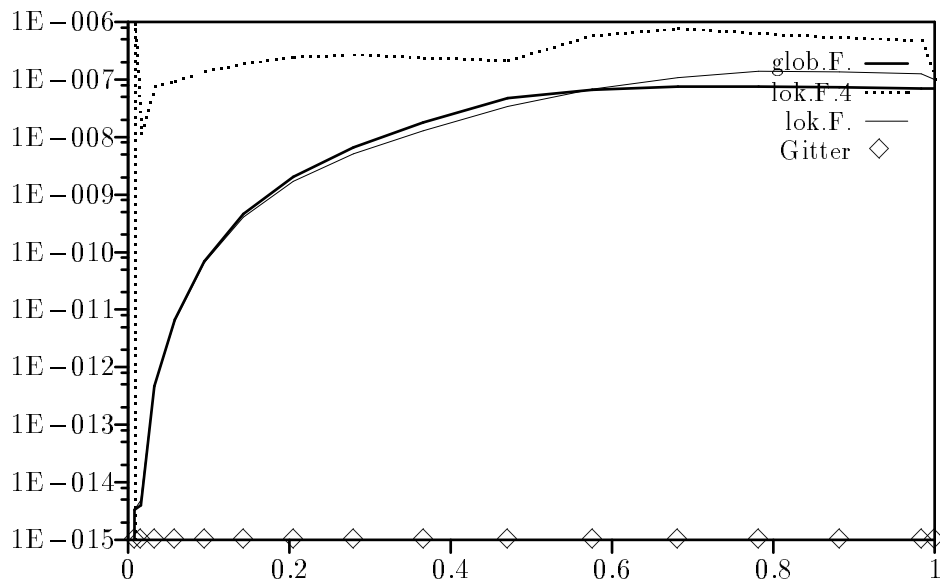
3.1.5 Runge-Kutta 5(4) Paar mit „exakter“ Fehlerschätzung

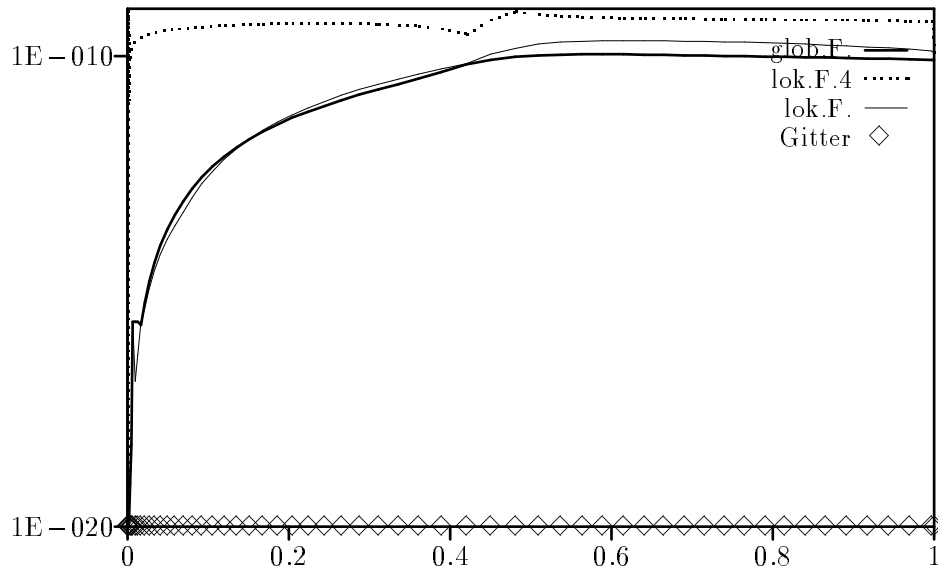
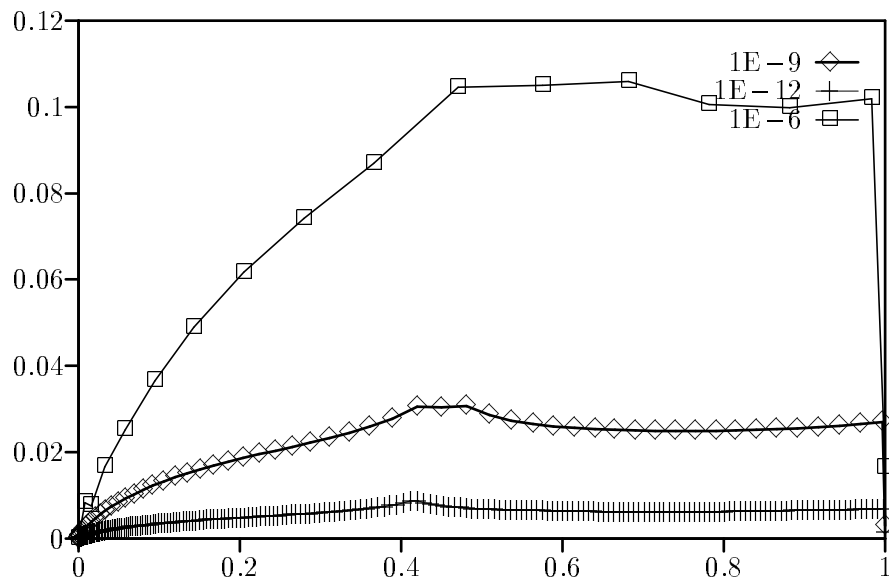
Um die Qualität der lokalen Fehlerschätzungen des Abschnitts 3.1.1 weiter zu untersuchen, gehen wir wie folgt vor. Wir berechnen den Schritt von y_{j-1} zu y_j und dessen exakten lokalen Fehler e_j . Der Schritt mit dem höheren Verfahren liefert \hat{y}_j , mit dem wie zuvor weiter gerechnet wird. Somit wird der Fall simuliert, daß $d_j = y_j - \hat{y}_j$ eine sehr genaue Schätzung des lokalen Fehlers e_j ist. Somit werden mögliche Probleme, die durch die Fehlerschätzung entstehen, ausgegrenzt. Getestet wurden die Beispiele 1a, 1b und 2a, da für diese der exakte lokale Fehler bekannt ist.

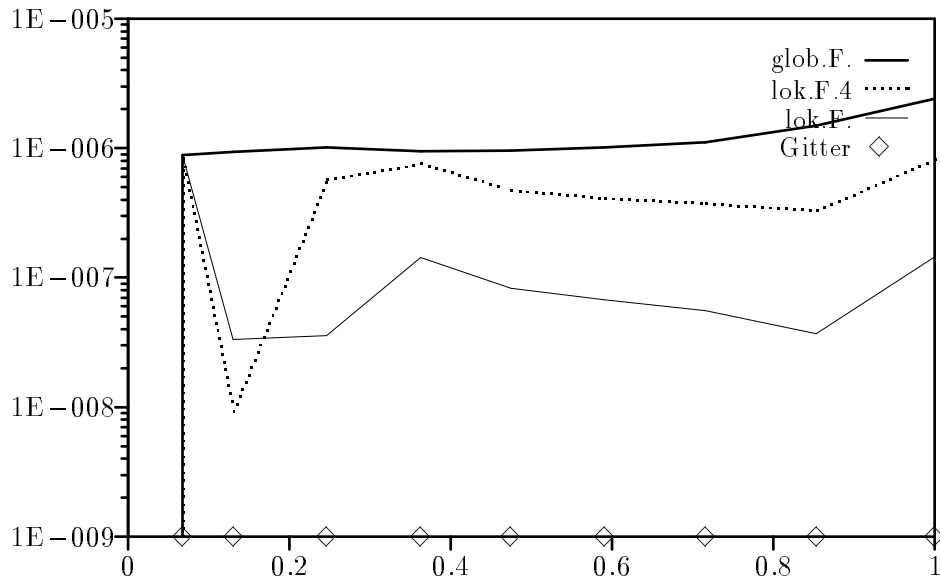
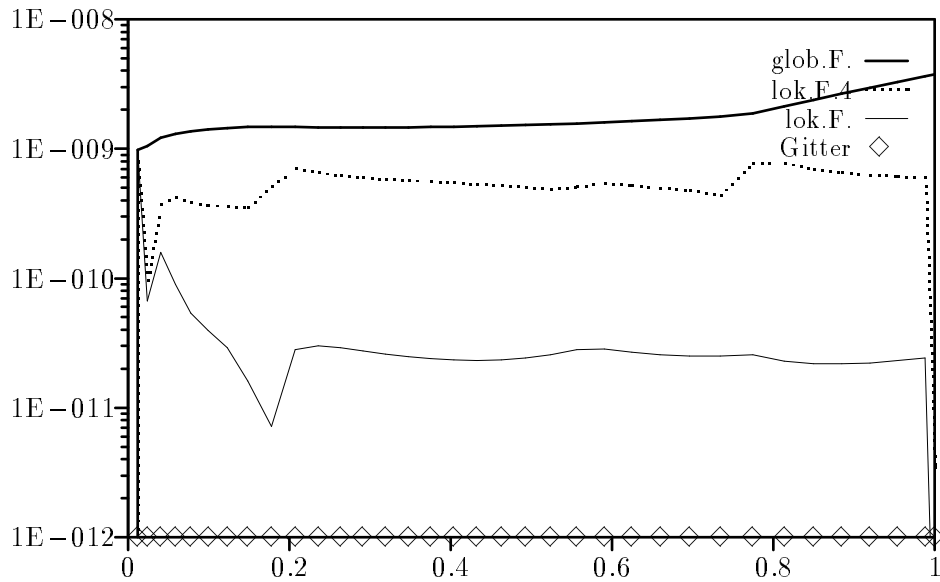
Die folgenden Grafiken zeigen das gewählte Gitter (Karo), die Maximumnorm des globalen Fehlers (dicke Linie), die Schätzung des lokalen Fehlers d_j , die in diesem Fall gleich der Maximumnorm des lokalen Fehlers e_j der Lösung y_j des Verfahrens 4. Ordnung ist, (punktierte Linie) und die Maximumnorm des lokalen Fehlers \hat{e}_j (dünne Linie), d.h. den lokalen Fehler der Lösung \hat{y}_j .

5(4) Paar mit „exakter“ Fehlerschätzung für Beispiel 1a: Im Vergleich von Abb. 3.25 mit Abb. 3.1 zeigt sich folgendes: Die Spitze des lokalen Fehlers bei der Singularität ist um eine Zehnerpotenz niedriger. Die lokale Fehlerschätzung folgt der Spitze genau, während sie in Abb. 3.1 diese um fast eine Zehnerpotenz unterschätzt hat. Weg von der Singularität wird der lokale Fehler um eine Zehnerpotenz überschätzt. Ab $x = 0.2$ verlaufen beide Fehlerschätzungen gleich, d.h. das 5(4) Paar in Abschnitt 3.1.1 hat dort den exakten lokalen Fehler des Verfahrens 4. Ordnung gut wiedergegeben. Der globale Fehler nimmt ebenfalls um drei Zehnerpotenzen ab, wenn man die Toleranz reduziert und verläuft geradlinig, ist jedoch fast um eine Zehnerpotenz kleiner als der von DOPRI5. Die ersten beiden Schritte, vgl. Abb. 3.27, wurden kürzer gewählt als in Abb. 3.4. Sonst stimmen beide x_j zu h_j Grafiken überein.

Abbildung 3.25: Fehler 5(4) Paar exakt in Beispiel 1a, Toleranz $1E-6$ Abbildung 3.26: Fehler 5(4) Paar exakt in Beispiel 1a, Toleranz $1E-9$

Abbildung 3.27: $x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 1aAbbildung 3.28: Fehler 5(4) Paar exakt in Beispiel 1b, Toleranz $1E-6$

Abbildung 3.29: Fehler 5(4) Paar exakt in Beispiel 1b, Toleranz $1E-9$ Abbildung 3.30: $x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 1b

Abbildung 3.31: Fehler 5(4) Paar exakt in Beispiel 2a, Toleranz $1E-6$ Abbildung 3.32: Fehler 5(4) Paar exakt in Beispiel 2a, Toleranz $1E-9$

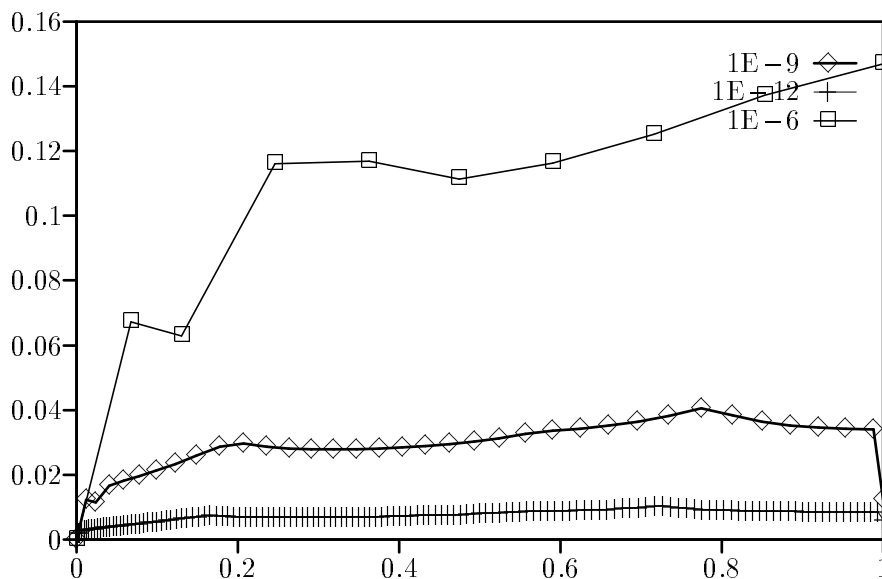


Abbildung 3.33: $x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 2a

5(4) Paar mit „exakter“ Fehlerschätzung für Beispiel 1b: In den Abb. 3.28, 3.29 und 3.30 sind keine Unterschiede zu den Abb. 3.5, 3.6 und 3.7 aus Abschnitt 3.1.1 zu erkennen. Das heißt, daß in diesem Beispiel die Schätzung des lokalen Fehlers durch DOPRI5 ausreichend genau war.

5(4) Paar mit „exakter“ Fehlerschätzung für Beispiel 2a: In Abb. 3.31 zeigt der lokale Fehler ebenfalls eine Spitze an der Singularität, diese wird jedoch von der Fehlerschätzung im Gegensatz zu Abb. 3.8 aufgelöst. Bei 0.1 hat die Fehlerschätzung einen Einschnitt und unterschätzt den lokalen Fehler. Ab etwa 0.4 verlaufen lokaler Fehler und Fehlerschätzung wie für DOPRI5. Der globale Fehler verläuft analog, ist in den Abb. 3.31 und 3.32 jedoch um fast eine Zehnerpotenz kleiner. Abb. 3.33 zeigt im Vergleich mit 3.11, daß am Beginn kürzere Schrittweiten gewählt wurden.

3.1.6 Zusammenfassung

In fast allen Fällen hat der lokale Fehler eine typische Zacke am Beginn des Intervalls. Meist wird diese Zacke umso steiler, je kleiner die Schrittweite gewählt wurde, bzw. je höher die Ordnung des Verfahrens war. Will man den lokalen Fehler mit T beschränken, so muß man am Anfang der Integration viele kleine Schritte machen, obwohl dies von der Struktur der Lösung nicht gerechtfertigt sein muß. Dies führt zu einem unnötig hohen Aufwand.

Zusätzlich wird der lokale Fehler nahe der Singularität durch die Runge-Kutta Paare teilweise **unterschätzt**, vgl. Abschnitt 3.1.1, d.h. es werden weniger Schritte gemacht, als eigentlich notwendig, vgl. Abschnitt 3.1.5. Genügend weit weg von der Singularität wird der Fehler überschätzt, dies liegt aber daran, daß der genaueren Lösung der beiden ERK Verfahren gefolgt wird. Abschnitt 3.1.4 zeigt, daß in diesem Bereich die Fehlerschätzung sehr genau ist und den lokalen Fehler gut wiedergibt.

Möchte man traditionell dem lokalen Fehler folgen, so ist dies mit Runge-Kutta Paaren **nicht** möglich, da deren Schätzung bei der Singularität nicht dem lokalen Fehler folgt.

Nach [24] ist es möglich, daß bei singulären Problemen die Ordnung eines ERK Verfahrens bis auf zwei reduziert wird. Im Fall von Runge-Kutta Paaren könnten sich in beiden Verfahren unterschiedliche Ordnungsreduktionen einstellen, schlimmstenfalls auf die gleiche Ordnung. Dadurch ist $|\hat{e}_j| \ll |e_j|$ nicht mehr gegeben, woraus in (3.1) $d_j \not\approx e_j$, und ev. sogar $|d_j| \ll |e_j|$ folgt, d.h. der lokale Fehler wird **unterschätzt**. Dies scheint für DOPRI5 teilweise der Fall zu sein. Das 3(2) Paar zeigt keine solche Auswirkungen, da kleine Ordnungen nicht stark reduziert werden können. Überraschenderweise scheint DOPRI8 auch nicht von Ordnungsreduktionen betroffen zu sein.

Der globale Fehler steht bei den getesteten Beispielen im Zusammenhang mit der lokalen Fehlertoleranz. Wird diese um drei Zehnerpotenzen reduziert, so nimmt der globale Fehler bis auf einige Ausnahmen ebenfalls um drei Zehnerpotenzen ab. Die Größenordnung befindet ist von 1/10 bis 10 der Größenordnung der lokalen Toleranz.

3.2 Richardson-Extrapolation

Neben Schätzung des lokalen Fehlers mittels Runge-Kutta Paaren ist die Methode der Richardson-Extrapolation weit verbreitet. Man startet dabei mit einem Verfahren der Ordnung p von y_{j-1} zu y_j . Dann halbiert man die Schrittweite und führt mit $h_{j-1}/2$ zwei Schritte von y_{j-1} zu \hat{y}_j durch. In den Gleichungen für die lokalen Fehler dieser Schritte,

$$\begin{aligned} y_j - u_{j-1}(x_j) &\approx ch_j^{p+1}, \\ \hat{y}_j - u_{j-1}(x_j) &\approx 2c \left(\frac{h_j}{2}\right)^{p+1}, \end{aligned}$$

kann c durch y_j , \hat{y}_j und h_j ausgedrückt und in der ersten Zeile eingesetzt werden. Nach [18], Satz 4.1, ist

$$d_j := \frac{2^p}{2^p - 1}(y_j - \hat{y}_j) \quad (3.3)$$

eine sinnvolle Schätzung des lokalen Fehlers e_j . Für die weiteren Berechnungen wird dann natürlich \hat{y}_j verwendet.

Für die Tests wurde das Dormands Programm `RKmbed.f90` weiter modifiziert. Statt einem Schritt des Runge-Kutta Paares werden jetzt drei, einer mit `y1`, entsprechend y_j , und zwei mit `y2` zu $h/2$, entsprechend \hat{y}_j , durchgeführt. Dann berechnet sich die Schätzung (3.3) von $|d_j|$ mit

```
fk = 2.0D0**p/(2.0D0**p-1.0D0) ! Faktor von p fuer delta
...
d = MAXVAL(ABS(y1-y2))*fk ! lokale Fehlerschaetzung
```

Die Berechnung von h_{j+1} erfolgt unverändert mit (3.2).

3.2.1 Basismethode explizites Runge-Kutta 4

Um diesen Abschnitt mit Abschnitt 3.1.1 vergleichen zu können wählen wir ein Runge-Kutta Verfahren 4. Ordnung. Der Einfachheit halber wurde gleich die niedere Methode von DOPRI5 gewählt, also das sieben-stufige ERK Verfahren 4. Ordnung aus [8], Seite 84, Abb. 5.4, mit den Koeffizienten b_i . Mit der Startschrittweite $h_1 := 0.1$ wurden die Probleme 1a, 2a und 3a mit 3 verschiedenen Toleranzen T , $T = 1E-6$, $1E-9$ und $1E-12$, numerisch gelöst.

In den folgenden Abbildungen werden das Gitter, die Maximumnorm des globalen Fehlers, die Schätzung des lokalen Fehlers und die Maximumnorm des lokalen Fehlers logarithmisch skaliert gezeigt.

Richardson-Extrapolation für ERK 4 für Beispiel 1a: In den Abb. 3.34 und 3.35 folgt die lokale Fehlerschätzung exakt den Konturen des lokalen Fehlers, selbst am Beginn des Intervalls, überschätzt diesen aber um mehr als eine Zehnerpotenz. Die globalen Fehler nehmen von 3.34 auf 3.35 um 2.5 Zehnerpotenzen ab und verhalten sich nicht so geradlinig wie in den Abb. 3.1 bis 3.3. Die Wahl der Schrittweiten, Abb. 3.36, zeigt keine Unterschiede zu Abb. 3.4.

Richardson-Extrapolation für ERK 4 für Beispiel 2a: Die Schätzung stellt den lokalen Fehler in den Abb. 3.37 und 3.38 gut dar. Der lokale Fehler zeigt die typische Spitze. Die globalen Fehler nehmen mit 2.5 Zehnerpotenzen ab. Abb. 3.39 zeigt im Vergleich mit Abb. 3.11 für $T = 1E-6$ bei 0.1 eine deutliche Verkürzung der Schrittweite, obwohl die Schrittweiten generell etwas größer gewählt werden.

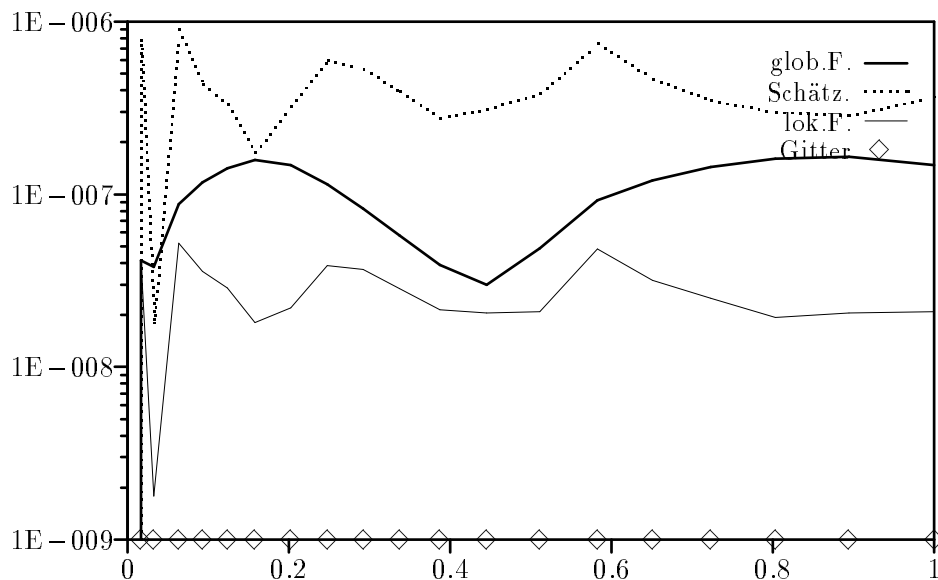


Abbildung 3.34: Fehler Richardson für ERK 4 in Beispiel 1a, Toleranz $1E-6$

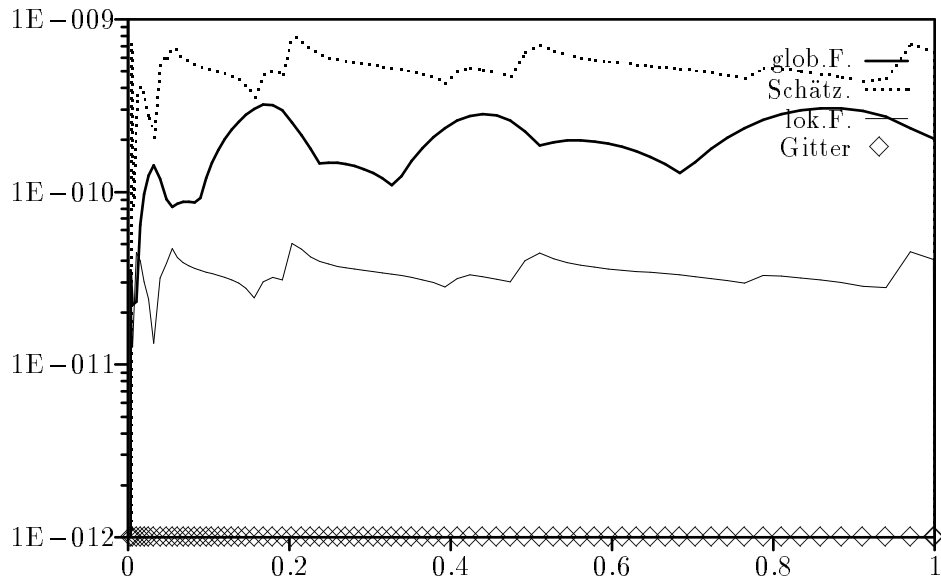


Abbildung 3.35: Fehler Richardson für ERK 4 in Beispiel 1a, Toleranz 10^{-9}

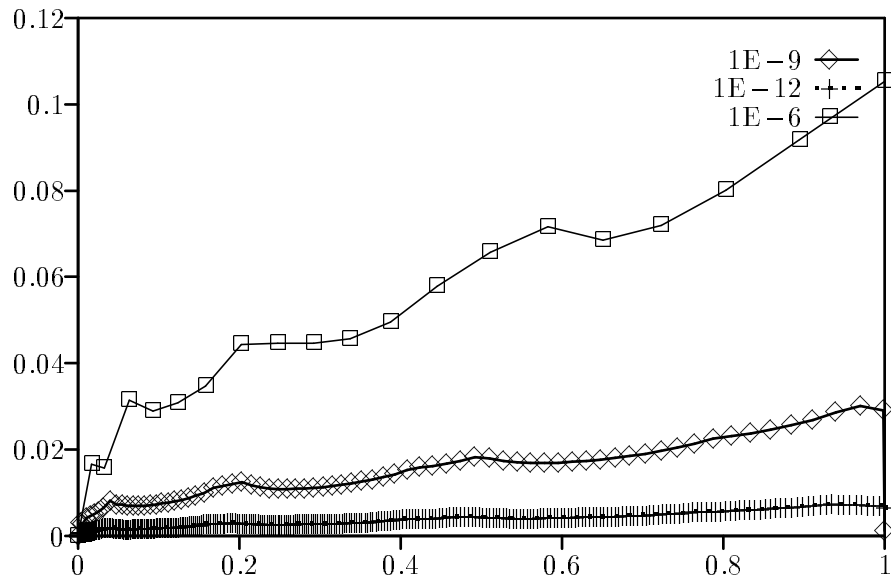
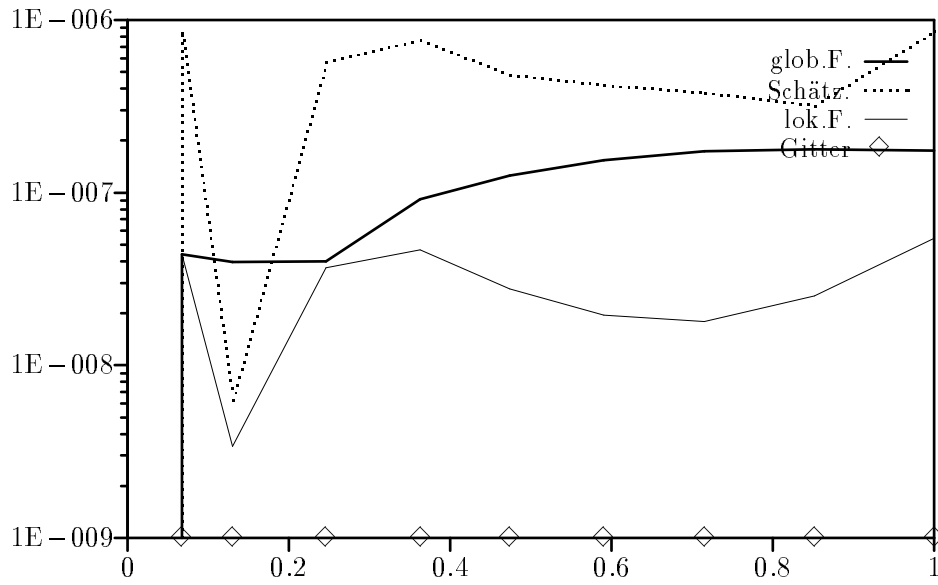
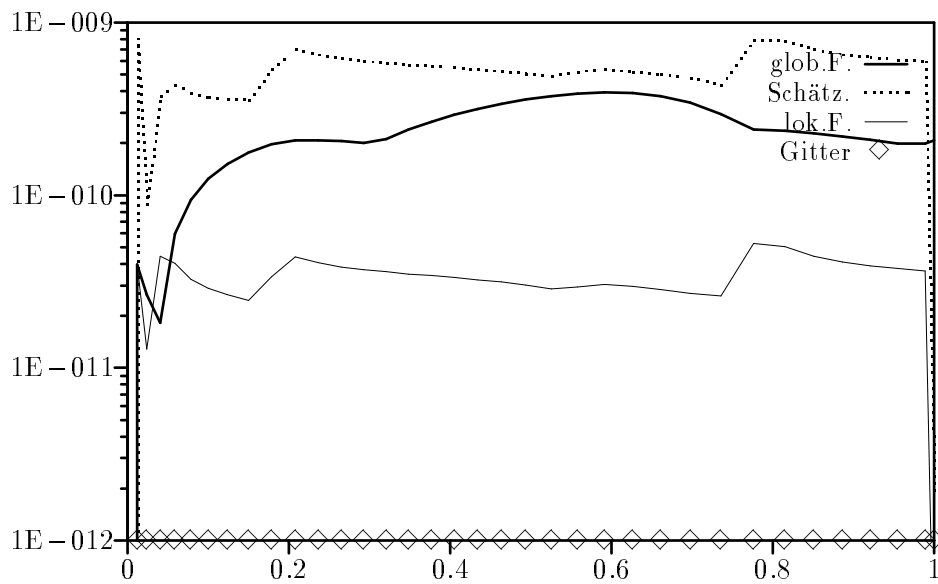
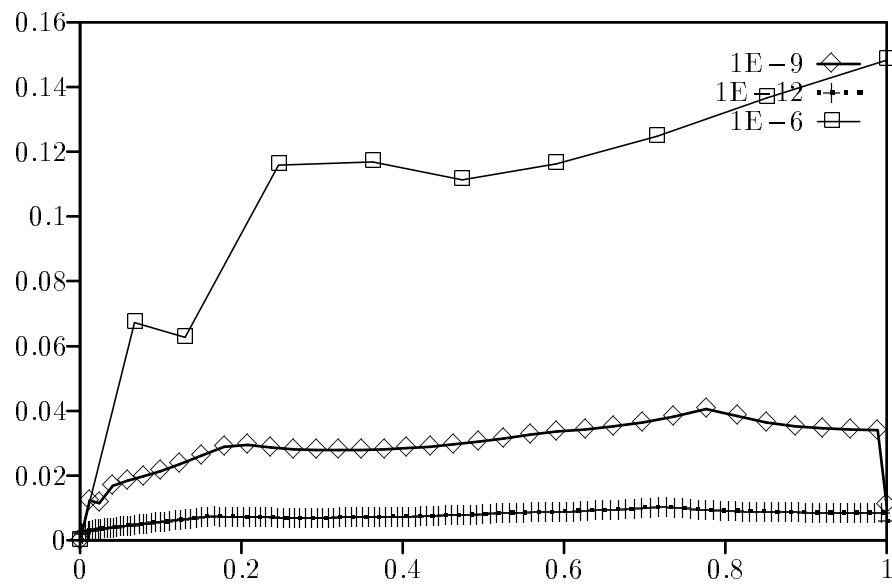
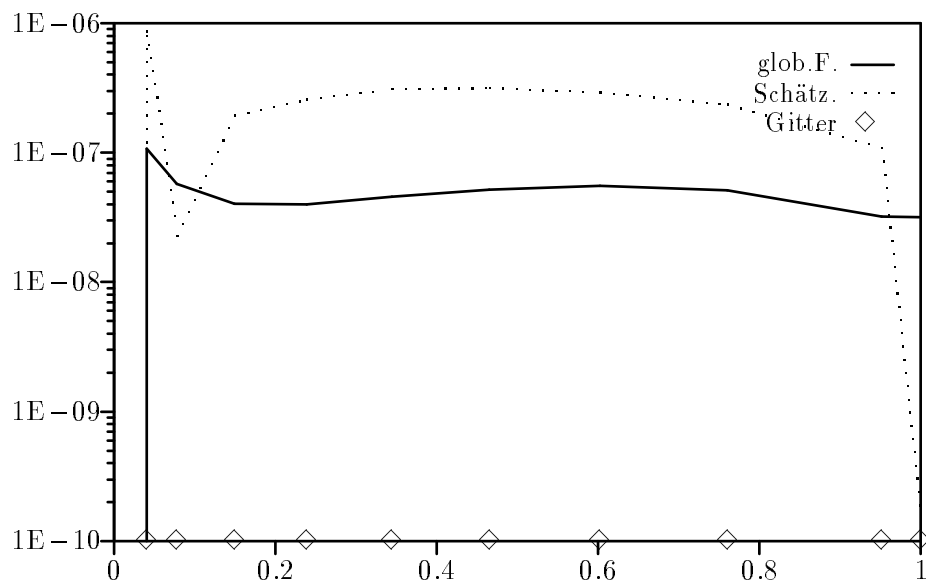
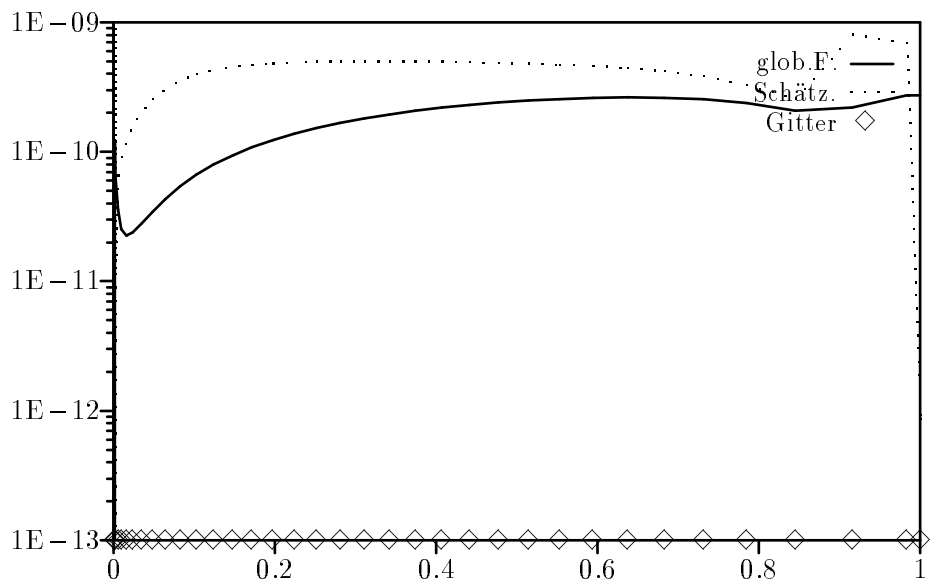
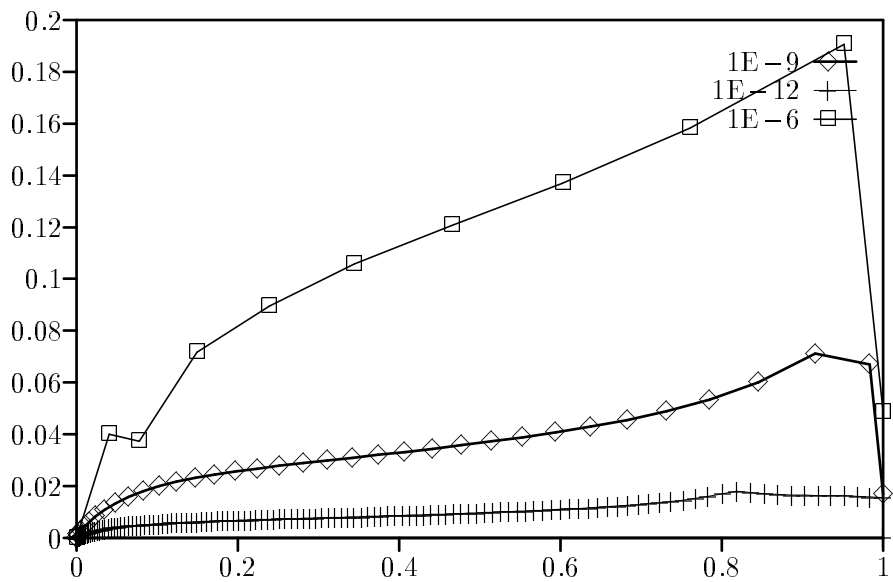


Abbildung 3.36: $x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 1a

Abbildung 3.37: Fehler Richardson für ERK 4 in Beispiel 2a, Toleranz $1E-6$ Abbildung 3.38: Fehler Richardson für ERK 4 in Beispiel 2a, Toleranz $1E-9$

Abbildung 3.39: $x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 2aAbbildung 3.40: Fehler Richardson für ERK 4 in Beispiel 3a, Toleranz $1E-6$

Abbildung 3.41: Fehler Richardson für ERK 4 in Beispiel 3a, Toleranz $1E-9$ Abbildung 3.42: $x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 3a

Richardson-Extrapolation für ERK 4 für Beispiel 3a: Wie in den Abb. 3.15 und 3.16 zeigt die lokale Fehlerschätzung in den Abb. 3.40 und 3.41 nahe der Singularität große Spitzen, wodurch die Schrittweiten klein gewählt werden. Abb. 3.42 stimmt mit 3.18 überein, viele kleine Schritte am Anfang, die mit der Glattheit der Lösung nicht übereinstimmen.

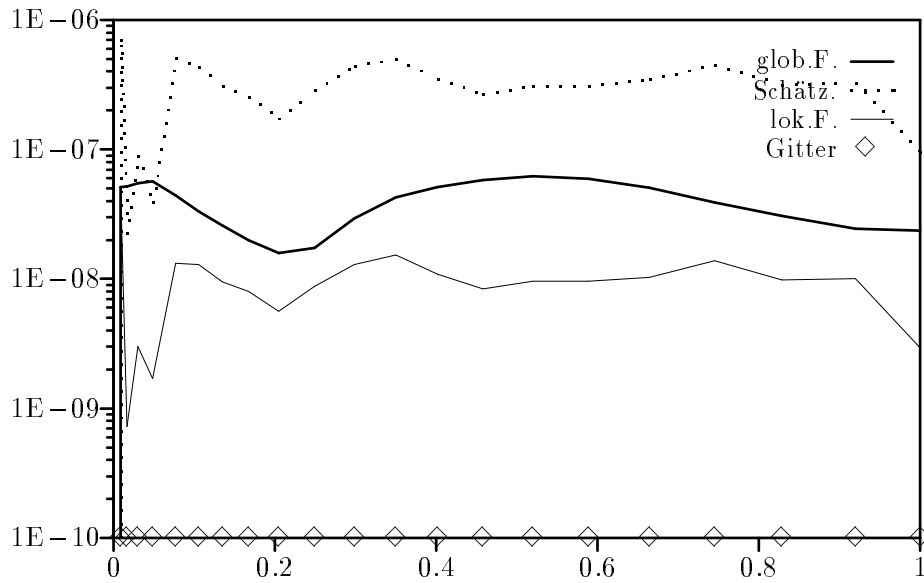
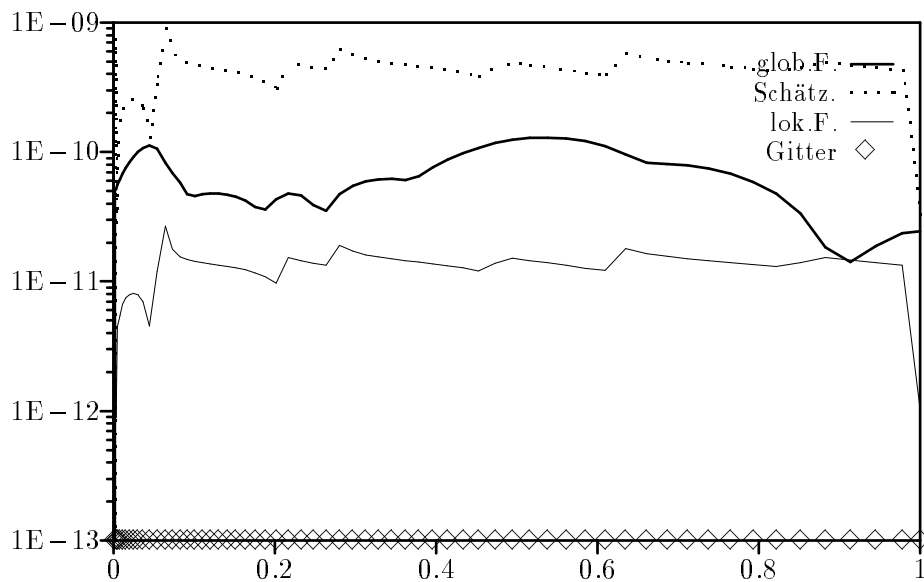
Motiviert durch das gute Verhalten von impliziten Verfahren bei steifen Differentialgleichungen, vgl. [19], die teilweise auch Ordnungsreduktionen zeigen, werden weiters implizite Runge-Kutta Verfahren untersucht. Wir gehen dabei genau wie für explizite Verfahren vor, die Schätzung des lokalen Fehlers erfolgt mittels (3.3), die Berechnung der nächsten Schrittweite mit (3.2) und lokale Extrapolation wird verwendet. Das nichtlineare $s \times n$ -dimensionale Gleichungssystem $G(k) = 0$, mit $k = (k_1, \dots, k_s)^T$ wird mit dem Newton-Verfahren gelöst, wobei die Fehlerschranke $1\text{E}-14$ beträgt, um mögliche Beeinflussungen der Ergebnisse aus der Iteration gering zu halten. Das heißt, sowohl das Residuum $G(k^{(i)})$, als auch das Newton-Inkrement $\Delta k^{(i)}$ müssen bezüglich der Maximumnorm des $s \times n$ kleiner als $1\text{E}-14$ sein, damit die i . Iteration als Lösung angenommen wird. Als Startwert für die Komponenten $k_m^{(0)}$, $1 \leq m \leq s$, wird jeweils die Lösung des letzten Schrittes verwendet. Da in dieser Phase auf den Aufwand keine Rücksicht genommen wird, wird die Jacobi-Matrix, welche für das jeweilige Beispiel bereitgestellt werden muß, in jeder Iteration neu ausgewertet.

Getestet wurde mit einem FORTRAN 77 Programm, da im Zuge des Newton-Verfahrens Routinen zur Lösung eines linearen Gleichungssystems benötigt werden, welche z.B. in LAPACK für FORTRAN 77 zur Verfügung stehen.

3.2.2 Basismethode implizites Runge-Kutta 5 (Radau IIA)

Es wurde ein implizites Radau IIA Verfahren aus [19], Seite 74, Abb. 5.6, ausgewählt. Die Ordnung p beträgt 5 und dieser Abschnitt wird mit Abschnitt 3.2.1 verglichen. Mit der Startschrittweite $h_1 := 0.1$ wurden die Probleme 1a und 2a mit 3 Toleranzen $T = 1\text{E}-6$, $1\text{E}-9$ und $1\text{E}-12$, gelöst. Die Abbildungen zeigen die Maximumnorm des globalen Fehlers als dicke Linie, die Schätzung des lokalen Fehlers als punktierte Linie und bei den linearen Anfangswertaufgaben mit konstanter Koeffizientenmatrix die Maximumnorm des lokalen Fehlers als dünne Linie. Weiters ergänzen x_j zu h_j Grafiken die Ergebnisse.

Richardson-Extrapolation für IRK 5 für Beispiel 1a: Die Abb. 3.43 und 3.44 stimmen fast mit den Abb. 3.34 und 3.35 überein, der lokale Fehler deutet am Beginn der Integration eine leichte Zacke an. Die lokale Fehlerschätzung folgt qualitativ dem Verlauf des lokalen Fehlers, selbst am Beginn des Intervalls, nur überschätzt sie diesen um zwei Zehnerpotenzen, was an der lokalen Extrapolation liegt. Die globalen Fehler verbessern sich von Abb. 3.43 zu 3.44 um jeweils drei Zehnerpotenzen, wie für DOPRI5, vgl. Abb. 3.1, 3.2 und 3.3, sind jedoch um zwei Zehnerpotenzen kleiner als dort. Die Wahl der Schrittweiten, vgl. Abb. 3.45, erfolgt in der Nähe der Singularität kleiner, als in den Abb. 3.4 bzw. 3.36.

Abbildung 3.43: Fehler Richardson für IRK 5 in Beispiel 1a, Toleranz $1E-6$ Abbildung 3.44: Fehler Richardson für IRK 5 in Beispiel 1a, Toleranz $1E-9$

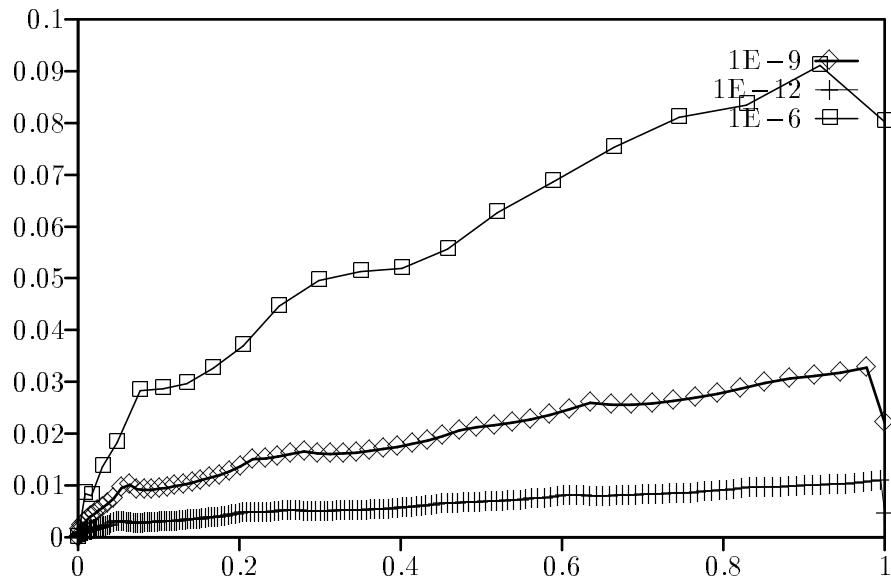


Abbildung 3.45: $x_j h_j$ -Plot Richardson für IRK 5 für Beispiel 1a

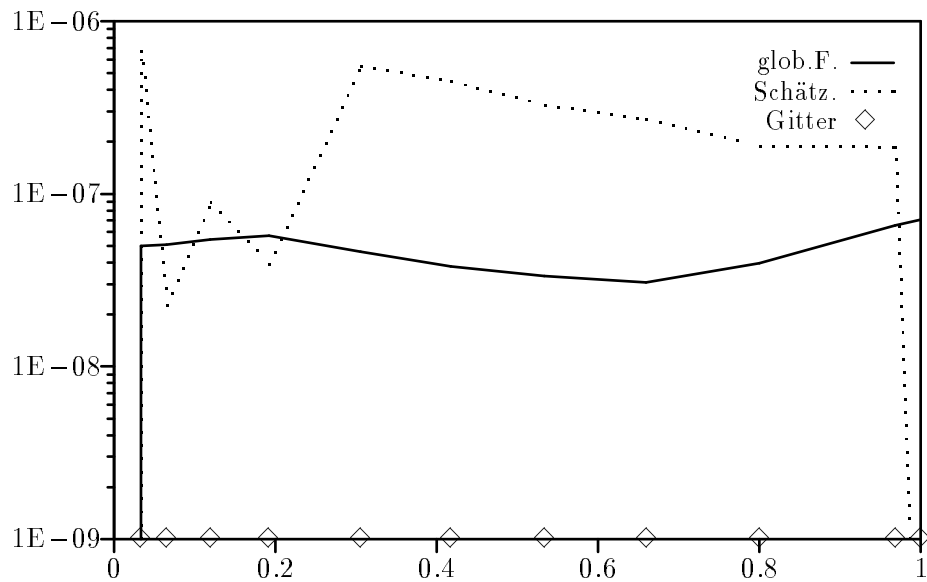
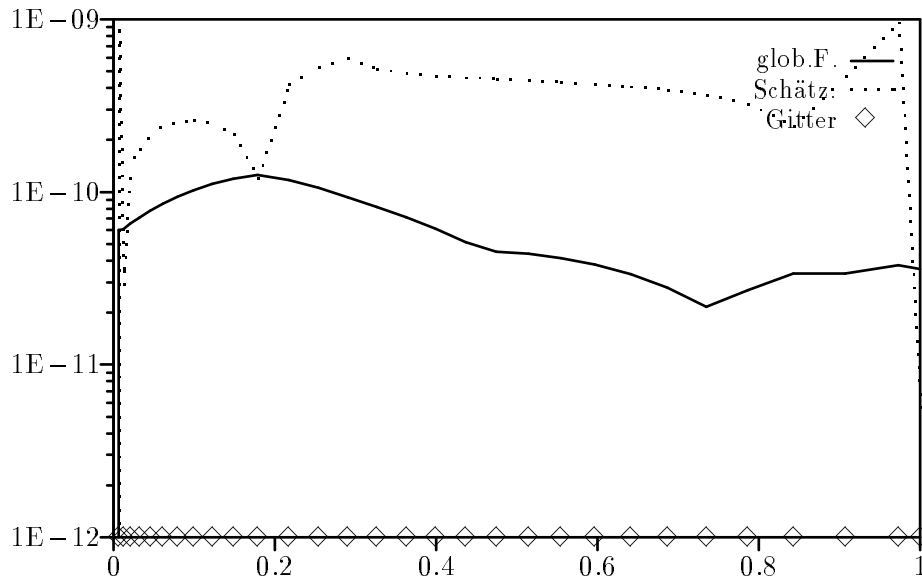
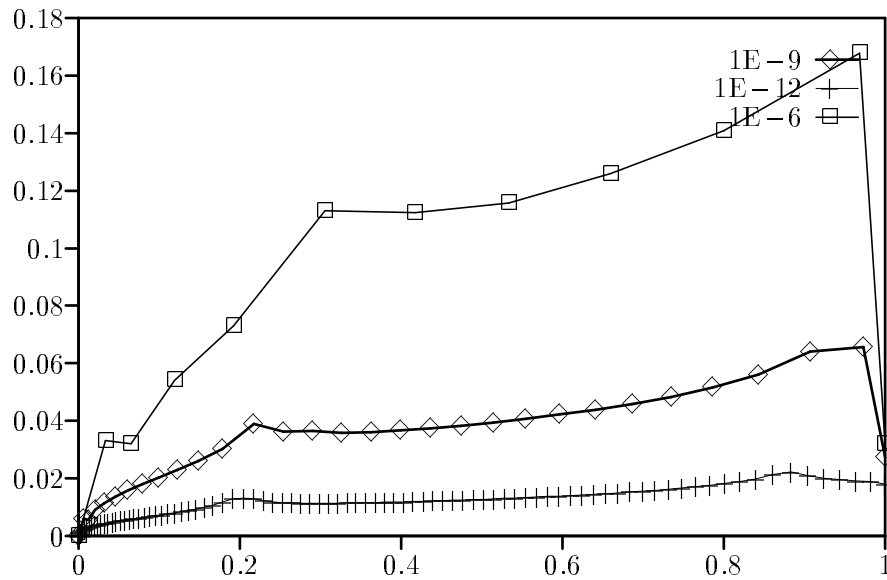


Abbildung 3.46: Fehler Richardson für IRK 5 in Beispiel 2a, Toleranz $1E-6$

Abbildung 3.47: Fehler Richardson für IRK 5 in Beispiel 2a, Toleranz $1E-9$ Abbildung 3.48: $x_j h_j$ -Plot Richardson für IRK 5 für Beispiel 2a

Richardson-Extrapolation für IRK 5 für Beispiel 2a: Schon in den Abb. 3.46 und 3.47 im Vergleich mit 3.37 und 3.38 fällt auf, daß das IRK mehr Schritte am Beginn der Integration durchführt, vgl. auch Abb. 3.48. Dadurch scheint die Spitze in der lokalen Fehlerschätzung von zwei auf 1.5 Zehnerpotenzen abgeschwächt zu werden, trotzdem zeigt die lokale Fehlerschätzung nahe der Singularität große Spitzen. Ab $x = 0.2$ verlaufen dann die Fehlerschätzungen der Abb. 3.46, 3.47 bzw. 3.37, 3.38 synchron. Die globalen Fehler nehmen um drei Zehnerpotenzen ab. Die Schrittweiten, vgl. Abb. 3.48, werden am Beginn des Intervalls viel kleiner gewählt, als in Abb. 3.39. Ab $x = 0.3$ jedoch sind sie etwas größer.

3.2.3 Zusammenfassung

Alle Beispiele zeigen im lokalen Fehler markante Spitzen an der Singularität. Die Schätzung des lokalen Fehlers durch Richardson-Extrapolation gibt das Verhalten des lokalen Fehlers sehr gut wieder. Dies mag daran liegen, daß zur Schätzung das selbe Verfahren verwendet wird, wie zur Berechnung, was für Runge-Kutta Paare nicht der Fall war. Eine Unterschätzung des lokalen Fehlers an der Singularität konnte nicht beobachtet werden. Weg von der Singularität wird der Fehler überschätzt, was an der lokalen Extrapolation liegt, d.h. weil mit der genaueren Lösung \hat{y} weiter gerechnet wird, während die Fehlerschätzung für y berechnet wurde.

Sollte es zu Ordnungsreduktionen von p auf $\tilde{p} < p$ kommen, so wird der lokale Fehler **unterschätzt**, da statt dem richtigen Wert $2^{\tilde{p}}/(2^{\tilde{p}} - 1)$ der kleinere Wert $2^p/(2^p - 1)$ in die Fehlerschätzung eingeht, was Unterschiede bis zu 20% ausmachen könnte. Es ist daher zu vermuten, daß die Fehlerschätzung mit Richardson-Extrapolation ebenfalls den lokalen Fehler an der Singularität unterschätzt, da die Schrittweiten, vgl. die $x_j h_j$ -Plotts, fast mit denen aus Abschnitt 3.1.1 übereinstimmen.

Bei impliziten Verfahren werden am Beginn der Integration kleinere Schritte als in den Abschnitten 3.1.1 bzw. 3.2.1 gewählt. Weg von der Singularität erfolgt die Wahl der Schrittweiten größer, was für ein implizites Verfahren zu erwarten war. Von den klassischen Möglichkeiten, die Schrittweite an der Singularität über den lokalen Fehler zu steuern, scheinen implizite Methoden den expliziten deutlich überlegen zu sein.

3.3 Ordnung des lokalen Fehlers der Verfahren

In den Zusammenfassungen 3.1.6 und 3.2.3 wurde der negative Einfluß möglicher Ordnungsreduktionen im lokalen Fehler angesprochen. In diesem Abschnitt untersuchen wir daher die Ordnung des lokalen Fehlers an bestimmten Punkten x^* im Intervall $[0, 1]$. Dazu starten wir bei x^* von der exakten Lösung $y_0(h_1) := y(x^*)$ und machen einen Schritt mit h_1 . Der globale Fehler $\varepsilon_1(h_1) := y_1(h_1) - y(x^* + h_1)$ ist in diesem Fall auch der lokale Fehler $e_1(h_1)$. Analog erhalten wir für h_2 $e_1(h_2) := y_1(h_2) - y(x^* + h_2)$. Die Ordnung wird dann folgendermaßen geschätzt: Für ein Verfahren p . Ordnung postulieren wir für den lokalen Fehler die Form

$$e_1(h) \approx c h^{p+1}, \quad (3.4)$$

wobei c die sogenannte Fehlerkonstante ist, vgl. K in (2.11). Daraus folgt

$$e_1(h_1) \approx c h_1^{p+1}$$

und

$$e_1(h_2) \approx c h_2^{p+1},$$

woraus nach beiderseitiger Division die Schätzung für $p + 1$,

$$p + 1 \approx \frac{\ln\left(\frac{|e_1(h_1)|}{|e_1(h_2)|}\right)}{\ln\left(\frac{|h_1|}{|h_2|}\right)} \quad (3.5)$$

folgt. Die Fehlerkonstante c erhält man danach als

$$c \approx \frac{e_1(h_1)}{h_1^{p+1}}. \quad (3.6)$$

Dieser Zugang ermöglicht es, für jede Komponente der Lösung die erhaltene Ordnung und Fehlerkonstante zu untersuchen.

In den vorliegenden Untersuchungen wird mit der Schrittweite $h_1 = 1/5$ begonnen. Weitere Schrittweiten ergeben sich durch sukzessives Halbieren von h_1 . Es werden bis zu 17 Halbierungen durchgeführt, der kleinste h -Wert ist $h_{18} = 1/5 \cdot 2^{-17} \approx 1.53 \cdot 10^{-6}$. In den meisten Fällen ist man dann im Bereich der Rechengenauigkeit, so daß sich der Einfluß der Rundungsfehler bemerkbar macht und das asymptotische Fehlerverhalten verfälscht. Sämtliche Tabellen zeigen nur den relevanten Bereich der h -Werte, d.h. jenen Bereich, in dem der Rechenfehlereinfluß noch nicht stark sichtbar ist.

Die folgenden Tabellen zeigen die Ordnung des lokalen Fehlers der ersten Komponente der Testbeispiele. Die erste Komponente der Fehlerkonstante c wird ebenfalls tabelliert. Stabilisiert sie sich, so zeigt dies, daß die Schätzung von p korrekt ist.

3.3.1 Explizites Runge-Kutta 4

Wir beginnen mit dem ERK der niederen Ordnung von DOPRI5 mit $p = 4$. Die für klassische Probleme erwartete lokale Ordnung ist daher 5. Wir messen die Ordnung an den Stellen $x^* = 0, 0.0001, 0.01, 0.1, 0.3$ und $x^* = 0.5$, um einen Überblick über das Verhalten des lokalen Fehlers im Intervall $[0, 1]$ zu erhalten. Die entsprechenden Ordnungen und Fehlerkonstanten sind dann $p_{0.0}, c_{0.0}, \dots, p_{0.5}, c_{0.5}$.

Ordnung lokaler Fehler von ERK 4 für Beispiel 1a: Tabelle 3.1 zeigt eine klare Ordnungsreduktion an der Singularität von $p = 5$ auf $p_{0.0} = 4$. Mit wachsendem x^* nehmen die Ordnungen für große Schrittweiten kontinuierlich zu, bis bei $x^* = 0.5$ mit $h = 0.1$ sich die erwartete Ordnung $4.9 \approx 5$ einstellt.

Außer an der Singularität ist im oberen Teil von Tabelle 3.1 ein Zunehmen der Ordnung mit wachsender Gitterfeinheit zu erkennen. Für kleine Werte h sieht man den wachsenden Einfluß des Rechenfehlers, mit zunehmendem x^* . Während sich für $x^* = 0.0$ und 0.0001 die Ordnungen bis zur 16. Verfeinerung einstellen, beginnt $p_{0.01}$ schon bei der 11. abzubröckeln. Ab $x^* = 0.1$ stellen sich Ordnungen nur bis zur 7. bzw. 8.

Verfeinerung der Schrittweite ein. Im unteren Teil der Tabelle, also weg von der Singularität, vor allem in den letzten zwei Spalten, beobachtet man, wie sich die klassischen Konvergenzordnungen einstellen.

h	$p_{0.0}$	$c_{0.0}$	$p_{0.0001}$	$c_{0.0001}$	$p_{0.01}$	$c_{0.01}$
1/5	2.48	$3.1 \cdot 10^{-01}$	2.47	$3.0 \cdot 10^{-01}$	0.62	$2.1 \cdot 10^{-03}$
$1/5 \cdot 2^{-1}$	3.83	$7.1 \cdot 10^{+01}$	3.84	$7.1 \cdot 10^{+01}$	4.13	$6.9 \cdot 10^{+01}$
$1/5 \cdot 2^{-2}$	4.09	$1.5 \cdot 10^{+01}$	4.10	$1.5 \cdot 10^{+01}$	5.04	$1.0 \cdot 10^{+02}$
$1/5 \cdot 2^{-3}$	4.11	$1.6 \cdot 10^{+01}$	4.14	$1.8 \cdot 10^{+01}$	7.90	$4.0 \cdot 10^{+07}$
$1/5 \cdot 2^{-4}$	4.08	$1.4 \cdot 10^{+01}$	4.14	$1.7 \cdot 10^{+01}$	1.56	$3.4 \cdot 10^{-05}$
$1/5 \cdot 2^{-5}$	4.04	$1.2 \cdot 10^{+01}$	4.17	$2.0 \cdot 10^{+01}$	3.88	$-4.4 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	4.02	$1.0 \cdot 10^{+01}$	4.27	$3.7 \cdot 10^{+02}$	4.50	$-1.5 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	4.01	$9.7 \cdot 10^{+01}$	4.51	$1.8 \cdot 10^{+02}$	4.75	$-8.0 \cdot 10^{+02}$
$1/5 \cdot 2^{-8}$	4.00	$9.3 \cdot 10^{+01}$	5.18	$2.1 \cdot 10^{+04}$	4.87	$-1.9 \cdot 10^{+02}$
$1/5 \cdot 2^{-9}$	4.00	$9.0 \cdot 10^{+01}$	6.07	$2.3 \cdot 10^{+07}$	4.93	$-3.1 \cdot 10^{+02}$
$1/5 \cdot 2^{-10}$	4.00	$8.9 \cdot 10^{+01}$	1.97	$-1.4 \cdot 10^{-08}$	4.97	$-4.0 \cdot 10^{+03}$
$1/5 \cdot 2^{-11}$	4.00	$8.8 \cdot 10^{+01}$	3.88	$-6.4 \cdot 10^{+00}$	4.99	$-5.2 \cdot 10^{+03}$
$1/5 \cdot 2^{-12}$	4.00	$8.7 \cdot 10^{+01}$	4.42	$-1.4 \cdot 10^{+02}$	5.45	$-5.1 \cdot 10^{+05}$
$1/5 \cdot 2^{-13}$	4.00	$8.7 \cdot 10^{+01}$	4.70	$-2.6 \cdot 10^{+03}$	1.00	$-1.3 \cdot 10^{-16}$
$1/5 \cdot 2^{-14}$	4.00	$8.7 \cdot 10^{+01}$	4.84	$-1.3 \cdot 10^{+04}$	0.00	$-1.6 \cdot 10^{-21}$
$1/5 \cdot 2^{-15}$	4.00	$8.7 \cdot 10^{+01}$	4.92	$-3.3 \cdot 10^{+05}$		
$1/5 \cdot 2^{-16}$	4.00	$8.7 \cdot 10^{+01}$	4.96	$-5.4 \cdot 10^{+05}$		

h	$p_{0.1}$	$c_{0.1}$	$p_{0.3}$	$c_{0.3}$	$p_{0.5}$	$c_{0.5}$
1/5	5.97	$-4.0 \cdot 10^{+02}$	4.49	$2.2 \cdot 10^{-01}$	4.47	$1.1 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	4.99	$-4.2 \cdot 10^{+01}$	3.77	$-4.3 \cdot 10^{-01}$	4.90	$3.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	3.49	$4.6 \cdot 10^{-01}$	4.56	$-4.6 \cdot 10^{+00}$	5.00	$4.0 \cdot 10^{+00}$
$1/5 \cdot 2^{-3}$	4.46	$1.7 \cdot 10^{+00}$	4.80	$-1.1 \cdot 10^{+00}$	5.01	$4.2 \cdot 10^{+00}$
$1/5 \cdot 2^{-4}$	4.75	$6.1 \cdot 10^{+01}$	4.90	$-1.7 \cdot 10^{+00}$	5.01	$4.1 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	4.88	$1.1 \cdot 10^{+01}$	4.95	$-2.2 \cdot 10^{+00}$	5.00	$4.1 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	4.94	$1.6 \cdot 10^{+01}$	4.98	$-2.6 \cdot 10^{+00}$	4.98	$3.5 \cdot 10^{+00}$
$1/5 \cdot 2^{-7}$	4.97	$1.9 \cdot 10^{+01}$	4.93	$-1.8 \cdot 10^{+00}$	5.04	$5.2 \cdot 10^{+00}$
$1/5 \cdot 2^{-8}$	4.96	$1.8 \cdot 10^{+01}$	2.00	$-1.4 \cdot 10^{-09}$		

Tabelle 3.1: Ordnung lokaler Fehler ERK 4 für Beispiel 1a an $x^* = 0, \dots, 0.5$

h	$p_{0.0}$	$c_{0.0}$	$p_{0.0001}$	$c_{0.0001}$	$p_{0.01}$	$c_{0.01}$
1/5	2.11	$-1.7 \cdot 10^{-02}$	2.12	$-1.7 \cdot 10^{-02}$	2.66	$-2.8 \cdot 10^{-02}$
$1/5 \cdot 2^{-1}$	2.03	$-1.4 \cdot 10^{-02}$	2.04	$-1.4 \cdot 10^{-02}$	2.92	$-5.2 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	2.00	$-1.3 \cdot 10^{-02}$	2.03	$-1.4 \cdot 10^{-02}$	3.32	$-1.7 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	2.00	$-1.3 \cdot 10^{-02}$	2.05	$-1.5 \cdot 10^{-02}$	3.78	$-9.4 \cdot 10^{+00}$
$1/5 \cdot 2^{-4}$	2.00	$-1.3 \cdot 10^{-02}$	2.10	$-1.9 \cdot 10^{-02}$	4.19	$-5.7 \cdot 10^{+01}$
$1/5 \cdot 2^{-5}$	2.00	$-1.3 \cdot 10^{-02}$	2.19	$-3.1 \cdot 10^{-02}$	4.51	$-2.8 \cdot 10^{+01}$
$1/5 \cdot 2^{-6}$	2.00	$-1.3 \cdot 10^{-02}$	2.36	$-8.2 \cdot 10^{-01}$	4.72	$-9.8 \cdot 10^{+02}$
$1/5 \cdot 2^{-7}$	2.00	$-1.3 \cdot 10^{-02}$	2.63	$-4.7 \cdot 10^{+00}$	4.85	$-2.2 \cdot 10^{+02}$
$1/5 \cdot 2^{-8}$	2.00	$-1.3 \cdot 10^{-02}$	3.02	$-7.3 \cdot 10^{+01}$	4.96	$-4.9 \cdot 10^{+03}$
$1/5 \cdot 2^{-9}$	2.00	$-1.3 \cdot 10^{-02}$	3.47	$-2.6 \cdot 10^{+02}$	3.75	$-3.7 \cdot 10^{-01}$
$1/5 \cdot 2^{-10}$	1.99	$-1.3 \cdot 10^{-02}$	3.93	$-1.2 \cdot 10^{+04}$		
$1/5 \cdot 2^{-11}$	1.99	$-1.3 \cdot 10^{-02}$	4.31	$-4.5 \cdot 10^{+06}$		
$1/5 \cdot 2^{-12}$	1.99	$-1.3 \cdot 10^{-02}$	4.61	$-8.5 \cdot 10^{+07}$		
$1/5 \cdot 2^{-13}$	2.00	$-1.3 \cdot 10^{-02}$				
$1/5 \cdot 2^{-14}$	2.00	$-1.3 \cdot 10^{-02}$				
$1/5 \cdot 2^{-15}$	1.99	$-1.2 \cdot 10^{-02}$				
$1/5 \cdot 2^{-16}$	1.99	$-1.2 \cdot 10^{-02}$				

h	$p_{0.1}$	$c_{0.1}$	$p_{0.3}$	$c_{0.3}$	$p_{0.5}$	$c_{0.5}$
1/5	4.04	$-4.2 \cdot 10^{-01}$	2.72	$-1.5 \cdot 10^{-04}$	5.26	$6.2 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	4.39	$-9.4 \cdot 10^{-01}$	4.37	$-6.8 \cdot 10^{-02}$	5.21	$5.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	4.63	$-1.9 \cdot 10^{-01}$	4.72	$-1.9 \cdot 10^{-02}$	5.13	$4.3 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	4.79	$-3.5 \cdot 10^{+00}$	4.86	$-3.2 \cdot 10^{-01}$	5.07	$3.5 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	4.89	$-5.3 \cdot 10^{+00}$	4.93	$-4.4 \cdot 10^{-01}$	5.04	$3.0 \cdot 10^{-02}$
$1/5 \cdot 2^{-5}$	4.94	$-6.9 \cdot 10^{+00}$	4.96	$-5.1 \cdot 10^{-01}$	5.02	$2.8 \cdot 10^{-02}$
$1/5 \cdot 2^{-6}$	5.00	$-9.6 \cdot 10^{+00}$	4.39	$-1.8 \cdot 10^{-03}$	5.00	$2.3 \cdot 10^{-02}$

Tabelle 3.2: Ordnung lokaler Fehler ERK 4 für Beispiel 1b an $x^* = 0, \dots, 0.5$

Ordnung lokaler Fehler von ERK 4 für Beispiel 1b: Bei $x^* = 0$ stellen sich in Tabelle 3.2 drastische Ordnungsreduktionen ein. Die Ordnung des lokalen Fehler fällt bis auf $p_{0.0} = 2$, also um drei h -Potenzen.

Ab $x^* = 0.0001$ zeigt sich das bis jetzt typische Verhalten der Ordnung. Mit wachsender Entfernung zur Singularität nimmt die Ordnung zu, und mit feiner werdendem h ebenfalls. Zuletzt stellen sich die klassischen Ordnungen, $p_{0.5}$, mit relativ kleinen Fehlerkonstanten $c_{0.5}$ ein und die Rundungsfehler machen sich immer früher bemerkbar.

h	$p_{0.0}$	$c_{0.0}$	$p_{0.0001}$	$c_{0.0001}$	$p_{0.01}$	$c_{0.01}$
$1/5$	4.12	$5.6 \cdot 10^{-01}$	4.12	$5.6 \cdot 10^{-01}$	4.40	$6.8 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	4.12	$5.7 \cdot 10^{-01}$	4.13	$5.8 \cdot 10^{-01}$	4.81	$1.7 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	4.08	$5.0 \cdot 10^{-01}$	4.10	$5.2 \cdot 10^{-01}$	5.95	$5.3 \cdot 10^{+01}$
$1/5 \cdot 2^{-3}$	4.05	$4.4 \cdot 10^{-01}$	4.08	$4.8 \cdot 10^{-01}$	3.95	$3.3 \cdot 10^{-02}$
$1/5 \cdot 2^{-4}$	4.02	$4.0 \cdot 10^{-01}$	4.08	$5.0 \cdot 10^{-01}$	3.24	$-1.4 \cdot 10^{-04}$
$1/5 \cdot 2^{-5}$	4.01	$3.7 \cdot 10^{-01}$	4.13	$6.4 \cdot 10^{-01}$	4.21	$-2.0 \cdot 10^{-02}$
$1/5 \cdot 2^{-6}$	4.00	$3.5 \cdot 10^{-01}$	4.25	$1.2 \cdot 10^{-01}$	4.60	$-1.8 \cdot 10^{-01}$
$1/5 \cdot 2^{-7}$	4.00	$3.5 \cdot 10^{-01}$	4.51	$6.6 \cdot 10^{+00}$	4.79	$-6.6 \cdot 10^{+00}$
$1/5 \cdot 2^{-8}$	4.00	$3.4 \cdot 10^{-01}$	5.18	$8.4 \cdot 10^{+02}$	4.89	$-1.3 \cdot 10^{+00}$
$1/5 \cdot 2^{-9}$	4.00	$3.4 \cdot 10^{-01}$	5.98	$4.4 \cdot 10^{+05}$	4.94	$-2.0 \cdot 10^{+00}$
$1/5 \cdot 2^{-10}$	4.00	$3.4 \cdot 10^{-01}$	2.03	$-9.3 \cdot 10^{-10}$	4.97	$-2.5 \cdot 10^{+00}$
$1/5 \cdot 2^{-11}$	4.00	$3.4 \cdot 10^{-01}$	3.88	$-2.6 \cdot 10^{-03}$	5.01	$-3.7 \cdot 10^{+01}$
$1/5 \cdot 2^{-12}$	4.00	$3.4 \cdot 10^{-01}$	4.42	$-5.5 \cdot 10^{+00}$	5.06	$-6.1 \cdot 10^{+01}$
$1/5 \cdot 2^{-13}$	4.00	$3.4 \cdot 10^{-01}$	4.70	$-1.0 \cdot 10^{+01}$	2.00	$-4.4 \cdot 10^{-13}$
$1/5 \cdot 2^{-14}$	4.00	$3.4 \cdot 10^{-01}$	4.84	$-5.3 \cdot 10^{+02}$	0.00	$-6.6 \cdot 10^{-23}$
$1/5 \cdot 2^{-15}$	4.00	$3.4 \cdot 10^{-01}$	4.92	$-1.3 \cdot 10^{+02}$	-1.00	$4.0 \cdot 10^{-28}$
$1/5 \cdot 2^{-16}$	4.00	$3.4 \cdot 10^{-01}$	4.96	$-2.1 \cdot 10^{+02}$		

h	$p_{0.1}$	$c_{0.1}$	$p_{0.3}$	$c_{0.3}$	$p_{0.5}$	$c_{0.5}$
$1/5$	4.54	$8.0 \cdot 10^{-03}$	5.48	$-6.7 \cdot 10^{-01}$	5.19	$-2.6 \cdot 10^{-02}$
$1/5 \cdot 2^{-1}$	1.63	$-9.7 \cdot 10^{-06}$	5.42	$-5.9 \cdot 10^{-01}$	5.16	$-2.5 \cdot 10^{-02}$
$1/5 \cdot 2^{-2}$	3.97	$-1.0 \cdot 10^{-03}$	5.31	$-4.2 \cdot 10^{-01}$	5.10	$-2.1 \cdot 10^{-02}$
$1/5 \cdot 2^{-3}$	4.52	$-8.3 \cdot 10^{-02}$	5.19	$-2.7 \cdot 10^{-02}$	5.06	$-1.7 \cdot 10^{-02}$
$1/5 \cdot 2^{-4}$	4.76	$-2.4 \cdot 10^{-02}$	5.11	$-1.9 \cdot 10^{-02}$	5.03	$-1.5 \cdot 10^{-02}$
$1/5 \cdot 2^{-5}$	4.88	$-4.3 \cdot 10^{-01}$	5.05	$-1.4 \cdot 10^{-02}$	5.02	$-1.5 \cdot 10^{-02}$
$1/5 \cdot 2^{-6}$	4.94	$-6.0 \cdot 10^{-01}$	5.15	$-2.5 \cdot 10^{-02}$	5.33	$-8.9 \cdot 10^{-01}$
$1/5 \cdot 2^{-7}$	4.97	$-7.3 \cdot 10^{-01}$			1.80	$-1.1 \cdot 10^{-11}$

Tabelle 3.3: Ordnung lokaler Fehler ERK 4 für Beispiel 2a an $x^* = 0, \dots, 0.5$

Ordnung lokaler Fehler von ERK 4 für Beispiel 2a: Das Verhalten des lokalen Fehlers unterscheidet sich für lineare Anfangswertaufgaben mit variabler Koeffizientenmatrix kaum von den früheren Beispielen. Wie in Beispiel 1a, vgl. Tabelle 3.1, reduziert sich die Ordnung in Tabelle 3.3 auf 4, und die Fehlerkonstanten $c_{0.0}$ ändern sich mit kleiner werdendem h nicht. Entfernt man sich von der Singularität, so stellt sich immer mehr das klassische Verhalten ein.

h	$p_{0.0}$	$c_{0.0}$	$p_{0.0001}$	$c_{0.0001}$	$p_{0.01}$	$c_{0.01}$
1/5	1.19	$-3.5 \cdot 10^{-03}$	1.20	$-3.6 \cdot 10^{-03}$	1.92	$-1.7 \cdot 10^{-03}$
$1/5 \cdot 2^{-1}$	2.59	$-8.8 \cdot 10^{-02}$	2.59	$-8.9 \cdot 10^{-02}$	2.99	$-2.0 \cdot 10^{-02}$
$1/5 \cdot 2^{-2}$	2.83	$-1.8 \cdot 10^{-02}$	2.84	$-1.9 \cdot 10^{-02}$	3.41	$-7.3 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	2.92	$-2.5 \cdot 10^{-02}$	2.94	$-2.7 \cdot 10^{-02}$	3.79	$-2.9 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	2.96	$-3.0 \cdot 10^{-02}$	3.00	$-3.4 \cdot 10^{-01}$	4.16	$-1.5 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	2.98	$-3.3 \cdot 10^{-01}$	3.05	$-4.4 \cdot 10^{-01}$	4.49	$-7.8 \cdot 10^{+01}$
$1/5 \cdot 2^{-6}$	2.99	$-3.4 \cdot 10^{-01}$	3.11	$-6.4 \cdot 10^{-01}$	4.71	$-2.8 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	2.99	$-3.5 \cdot 10^{-01}$	3.21	$-1.2 \cdot 10^{-01}$	4.84	$-6.8 \cdot 10^{+02}$
$1/5 \cdot 2^{-8}$	2.99	$-3.6 \cdot 10^{-01}$	3.37	$-3.8 \cdot 10^{+00}$	4.92	$-1.1 \cdot 10^{+02}$
$1/5 \cdot 2^{-9}$	2.99	$-3.6 \cdot 10^{-01}$	3.62	$-2.6 \cdot 10^{+00}$	4.96	$-1.5 \cdot 10^{+02}$
$1/5 \cdot 2^{-10}$	2.99	$-3.6 \cdot 10^{-01}$	3.95	$-4.4 \cdot 10^{+02}$	4.98	$-1.9 \cdot 10^{+02}$
$1/5 \cdot 2^{-11}$	2.99	$-3.6 \cdot 10^{-01}$	4.29	$-1.1 \cdot 10^{+03}$	6.17	$-1.1 \cdot 10^{+07}$
$1/5 \cdot 2^{-12}$	2.99	$-3.6 \cdot 10^{-01}$	4.58	$-1.8 \cdot 10^{+04}$	0.00	$-2.7 \cdot 10^{-20}$
$1/5 \cdot 2^{-13}$	2.99	$-3.6 \cdot 10^{-01}$	4.77	$-1.3 \cdot 10^{+05}$		
$1/5 \cdot 2^{-14}$	2.99	$-3.6 \cdot 10^{-01}$	4.87	$-4.6 \cdot 10^{+06}$		
$1/5 \cdot 2^{-15}$	2.99	$-3.6 \cdot 10^{-01}$	4.93	$-9.2 \cdot 10^{+06}$		
$1/5 \cdot 2^{-16}$	2.99	$-3.6 \cdot 10^{-01}$	4.78	$-1.3 \cdot 10^{+05}$	0.58	$6.8 \cdot 10^{-16}$

h	$p_{0.1}$	$c_{0.1}$	$p_{0.3}$	$c_{0.3}$	$p_{0.5}$	$c_{0.5}$
1/5	3.60	$-4.0 \cdot 10^{-01}$	4.36	$-1.0 \cdot 10^{-01}$	4.59	$-1.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	4.21	$-1.6 \cdot 10^{-01}$	4.68	$-2.2 \cdot 10^{-01}$	4.80	$-2.2 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	4.56	$-4.5 \cdot 10^{+00}$	4.84	$-3.5 \cdot 10^{+00}$	4.90	$-3.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	4.76	$-9.7 \cdot 10^{+00}$	4.92	$-4.7 \cdot 10^{+00}$	4.95	$-3.6 \cdot 10^{+00}$
$1/5 \cdot 2^{-4}$	4.87	$-1.6 \cdot 10^{+00}$	4.96	$-5.6 \cdot 10^{+00}$	4.97	$-4.0 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	4.93	$-2.1 \cdot 10^{+00}$	4.98	$-6.2 \cdot 10^{+00}$	4.98	$-4.3 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	4.96	$-2.5 \cdot 10^{+00}$	4.98	$-6.6 \cdot 10^{+00}$	5.01	$-4.9 \cdot 10^{+00}$
$1/5 \cdot 2^{-7}$	4.98	$-2.8 \cdot 10^{+00}$	5.08	$-1.2 \cdot 10^{+00}$	4.24	$-3.5 \cdot 10^{-02}$
$1/5 \cdot 2^{-8}$	4.98	$-2.8 \cdot 10^{+00}$	2.80	$-1.0 \cdot 10^{-07}$	2.00	$-3.6 \cdot 10^{-09}$

Tabelle 3.4: Ordnung lokaler Fehler ERK 4 für Beispiel 2b an $x^* = 0, \dots, 0.5$

Ordnung lokaler Fehler von ERK 4 für Beispiel 2b: Tabelle 3.4 zeigt eine Ordnungsreduktion an der Singularität um zwei h -Potenzen. Bereits bei $x^* = 0.0001$ und sehr kleinen Schrittweiten nähert sich die Ordnung der erwarteten 5. Mit wachsendem x^* stellt sich die klassischen Ordnungen für immer größere h ein, bis bei 0.5 für $h = 0.05$ $p = 4.9 \approx 5$ gilt.

h	$p_{0.0}$	$c_{0.0}$	$p_{0.0001}$	$c_{0.0001}$	$p_{0.01}$	$c_{0.01}$
$1/5$	3.30	$-1.4 \cdot 10^{-02}$	3.31	$-1.4 \cdot 10^{-02}$	3.77	$-2.1 \cdot 10^{-02}$
$1/5 \cdot 2^{-1}$	2.42	$-1.9 \cdot 10^{-03}$	2.43	$-1.9 \cdot 10^{-03}$	3.32	$-7.5 \cdot 10^{-02}$
$1/5 \cdot 2^{-2}$	2.11	$-7.5 \cdot 10^{-03}$	2.14	$-8.0 \cdot 10^{-03}$	3.46	$-1.1 \cdot 10^{-02}$
$1/5 \cdot 2^{-3}$	2.02	$-5.5 \cdot 10^{-03}$	2.08	$-6.5 \cdot 10^{-03}$	3.82	$-4.3 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	2.00	$-5.0 \cdot 10^{-03}$	2.11	$-7.4 \cdot 10^{-03}$	4.21	$-2.3 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	2.00	$-4.9 \cdot 10^{-03}$	2.20	$-1.1 \cdot 10^{-03}$	4.52	$-1.1 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	2.00	$-4.8 \cdot 10^{-03}$	2.36	$-3.0 \cdot 10^{-03}$	4.72	$-3.7 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	2.00	$-4.8 \cdot 10^{-03}$	2.63	$-1.7 \cdot 10^{-02}$	4.85	$-8.2 \cdot 10^{+01}$
$1/5 \cdot 2^{-8}$	2.00	$-4.8 \cdot 10^{-03}$	3.02	$-2.7 \cdot 10^{-01}$	4.97	$-2.0 \cdot 10^{+01}$
$1/5 \cdot 2^{-9}$	2.00	$-4.8 \cdot 10^{-03}$	3.47	$-9.6 \cdot 10^{+01}$	1.00	$-5.6 \cdot 10^{-12}$
$1/5 \cdot 2^{-10}$	1.99	$-4.8 \cdot 10^{-03}$	3.93	$-4.6 \cdot 10^{+03}$	0.00	$1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-11}$	2.00	$-4.8 \cdot 10^{-03}$	4.33	$-1.9 \cdot 10^{+04}$	0.00	$1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-12}$	2.00	$-4.8 \cdot 10^{-03}$	5.16	$-7.7 \cdot 10^{+08}$	0.00	$1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-13}$	2.00	$-4.8 \cdot 10^{-03}$	0.00	$-1.1 \cdot 10^{-16}$	0.00	$-1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-14}$	2.00	$-4.8 \cdot 10^{-03}$	0.00	$-1.1 \cdot 10^{-16}$	0.00	$1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-15}$	2.02	$-6.7 \cdot 10^{-03}$	0.00	$-1.1 \cdot 10^{-16}$	0.00	$-1.1 \cdot 10^{-16}$

h	$p_{0.1}$	$c_{0.1}$	$p_{0.3}$	$c_{0.3}$	$p_{0.5}$	$c_{0.5}$
$1/5$	4.78	$-2.0 \cdot 10^{-02}$	4.91	$-5.3 \cdot 10^{-02}$	4.85	$-1.1 \cdot 10^{-03}$
$1/5 \cdot 2^{-1}$	4.75	$-1.8 \cdot 10^{-02}$	4.92	$-5.5 \cdot 10^{-02}$	4.90	$-1.3 \cdot 10^{-03}$
$1/5 \cdot 2^{-2}$	4.82	$-2.3 \cdot 10^{-02}$	4.95	$-6.0 \cdot 10^{-02}$	4.95	$-1.4 \cdot 10^{-03}$
$1/5 \cdot 2^{-3}$	4.89	$-3.0 \cdot 10^{-02}$	4.97	$-6.4 \cdot 10^{-02}$	4.97	$-1.6 \cdot 10^{-03}$
$1/5 \cdot 2^{-4}$	4.94	$-3.7 \cdot 10^{-01}$	4.98	$-6.8 \cdot 10^{-02}$	4.98	$-1.6 \cdot 10^{-03}$
$1/5 \cdot 2^{-5}$	4.95	$-3.9 \cdot 10^{-01}$	5.03	$-8.4 \cdot 10^{-02}$	4.70	$-3.9 \cdot 10^{-03}$
$1/5 \cdot 2^{-6}$	5.55	$-1.2 \cdot 10^{+00}$			2.58	$-1.9 \cdot 10^{-09}$
$1/5 \cdot 2^{-7}$	1.58	$-9.3 \cdot 10^{-11}$				

Tabelle 3.5: Ordnung lokaler Fehler ERK 4 für Beispiel 3a an $x^* = 0, \dots, 0.5$

Ordnung lokaler Fehler von ERK 4 für Beispiel 3a: Im diesem nichtlinearen Fall, vgl. Tabelle 3.5, reduziert sich die Ordnung bei $x^* = 0$ auf 2, nach [24] die maximale mögliche Ordnungsreduktion. Wie in allen anderen Beispielen nimmt die Ordnung mit wachsendem x^* und wachsender Genauigkeit zu.

Neben den expliziten Verfahren muß auch untersucht werden, ob bei impliziten Verfahren Ordnungsreduktionen im lokalen Fehler an $x^* = 0$ auftreten. Wir gehen dabei genau wie für explizite Methoden vor.

3.3.2 Implizites Runge-Kutta 5 (Radau IIA)

Wir verwenden das implizite Radau IIA Verfahren aus Abschnitt 3.2.2 mit der lokalen Ordnung 6. Wir messen die Ordnung des lokalen Fehlers an den Stellen $x^* = 0$ und 0.5.

Ordnung lokaler Fehler von IRK 5 für Beispiel 1a: Tabelle 3.6 zeigt wie auch Tabelle 3.1 eine klare Ordnungsreduktion an der Singularität von $p = 6$ auf $p_{0.0} = 4$,

und die Fehlerkonstanten sind etwa gleich groß. Bei $x^* = 0.5$ stellt sich die erwartete Ordnung 5.7 bzw. $5.9 \approx 6$ ein. Der lokale Fehler $e_{0.5}$ ist um bis zu fünf Zehnerpotenzen genauer als $e_{0.0}$. Während sich an der Singularität für sämtliche Gitterverfeinerungen eine Ordnung einstellt, machen sich weg von der Singularität schon ab h_7 die Rundungsfehler bemerkbar, sobald der lokale Fehler im Rundungsfehlerbereich EPS liegt.

h	$e_{0.0}$	$p_{0.0}$	$c_{0.0}$	$e_{0.5}$	$p_{0.5}$	$c_{0.5}$
1/5	$5.6 \cdot 10^{-01}$	6.094	$1.0 \cdot 10^{+03}$	$9.7 \cdot 10^{-04}$	7.148	$9.7 \cdot 10^{+01}$
$1/5 \cdot 2^{-1}$	$8.3 \cdot 10^{-03}$	1.425	$2.2 \cdot 10^{-02}$	$6.9 \cdot 10^{-06}$	8.091	$8.5 \cdot 10^{+02}$
$1/5 \cdot 2^{-2}$	$3.0 \cdot 10^{-04}$	2.973	$-2.2 \cdot 10^{+00}$	$2.5 \cdot 10^{-09}$	3.818	$-2.3 \cdot 10^{-04}$
$1/5 \cdot 2^{-3}$	$3.9 \cdot 10^{-04}$	3.585	$-2.1 \cdot 10^{+01}$	$1.7 \cdot 10^{-10}$	5.466	$-1.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$3.2 \cdot 10^{-05}$	3.807	$-5.7 \cdot 10^{+02}$	$4.0 \cdot 10^{-11}$	5.778	$-4.0 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	$2.3 \cdot 10^{-07}$	3.907	$-9.5 \cdot 10^{+02}$	$7.3 \cdot 10^{-13}$	5.919	$-8.2 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	$1.5 \cdot 10^{-08}$	3.954	$-1.2 \cdot 10^{+02}$	$1.2 \cdot 10^{-15}$		
$1/5 \cdot 2^{-7}$	$1.0 \cdot 10^{-09}$	3.977	$-1.4 \cdot 10^{+02}$	0.0		
$1/5 \cdot 2^{-8}$	$6.4 \cdot 10^{-10}$	3.988	$-1.5 \cdot 10^{+02}$	$1.1 \cdot 10^{-16}$	0.000	$-1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-9}$	$4.0 \cdot 10^{-11}$	3.994	$-1.6 \cdot 10^{+02}$	$1.1 \cdot 10^{-16}$		
$1/5 \cdot 2^{-10}$	$2.5 \cdot 10^{-13}$	3.997	$-1.6 \cdot 10^{+02}$	0.0		
$1/5 \cdot 2^{-11}$	$1.5 \cdot 10^{-14}$	3.998	$-1.7 \cdot 10^{+02}$	$1.1 \cdot 10^{-16}$	0.000	$-1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-12}$	$9.9 \cdot 10^{-15}$	3.999	$-1.7 \cdot 10^{+02}$	$1.1 \cdot 10^{-16}$	0.000	$-1.1 \cdot 10^{-16}$
$1/5 \cdot 2^{-13}$	$6.2 \cdot 10^{-16}$	3.999	$-1.7 \cdot 10^{+02}$	$1.1 \cdot 10^{-16}$		$2.7 \cdot 10^{-21}$
$1/5 \cdot 2^{-14}$	$3.8 \cdot 10^{-17}$	3.999	$-1.7 \cdot 10^{+02}$	$2.2 \cdot 10^{-16}$		

Tabelle 3.6: Ordnung lokaler Fehler IRK 5 für Beispiel 1a an $x^* = 0, 0.5$

Ordnung lokaler Fehler von IRK 5 für Beispiel 1b: In Tabelle 3.7 zeigen sich im Gegensatz zu Tabelle 3.2 keine Ordnungsreduktionen. Sowohl an $x^* = 0$ als auch 0.5 ist die Ordnung 6 und die Fehlerkonstanten sind nicht groß.

h	$e_{0.0}$	$p_{0.0}$	$c_{0.0}$	$e_{0.5}$	$p_{0.5}$	$c_{0.5}$
1/5	$9.2 \cdot 10^{-05}$	5.981	$-1.4 \cdot 10^{-01}$	$8.5 \cdot 10^{-05}$	5.695	$-8.2 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$1.4 \cdot 10^{-07}$	5.995	$-1.4 \cdot 10^{-01}$	$1.6 \cdot 10^{-07}$	5.846	$-1.1 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$2.3 \cdot 10^{-09}$	5.998	$-1.4 \cdot 10^{-01}$	$2.8 \cdot 10^{-09}$	5.922	$-1.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$3.6 \cdot 10^{-10}$	5.999	$-1.4 \cdot 10^{-01}$	$4.7 \cdot 10^{-10}$	5.961	$-1.6 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$5.6 \cdot 10^{-12}$	5.988	$-1.4 \cdot 10^{-01}$	$7.6 \cdot 10^{-12}$	5.991	$-1.9 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$8.8 \cdot 10^{-14}$			$1.1 \cdot 10^{-14}$		

Tabelle 3.7: Ordnung lokaler Fehler IRK 5 für Beispiel 1b an $x^* = 0, 0.5$

Ordnung lokaler Fehler von IRK 5 für Beispiel 2a: Die Ordnung an der Singularität wird auf 4 reduziert, vgl. Tabelle 3.8, wie auch schon im expliziten Fall, vgl. Tabelle 3.3. Weg von der Singularität stellt sich das klassische Verhalten ein, wobei sich nach 6 Verfeinerungen bereits wieder die Rundungsfehler bemerkbar machen.

h	$e_{0.0}$	$p_{0.0}$	$c_{0.0}$	$e_{0.5}$	$p_{0.5}$	$c_{0.5}$
$1/5$	$3.1 \cdot 10^{-03}$	3.006	$-4.0 \cdot 10^{-01}$	$4.4 \cdot 10^{-05}$	5.631	$3.8 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$3.9 \cdot 10^{-04}$	3.594	$-1.5 \cdot 10^{-01}$	$8.8 \cdot 10^{-07}$	5.833	$6.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$3.2 \cdot 10^{-05}$	3.810	$-2.9 \cdot 10^{-01}$	$1.5 \cdot 10^{-09}$	5.920	$7.8 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$2.3 \cdot 10^{-07}$	3.907	$-4.2 \cdot 10^{+00}$	$2.5 \cdot 10^{-11}$	5.961	$9.1 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$1.5 \cdot 10^{-08}$	3.954	$-5.2 \cdot 10^{+00}$	$4.1 \cdot 10^{-12}$	5.988	$1.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$1.0 \cdot 10^{-09}$	3.977	$-5.8 \cdot 10^{+00}$	$6.5 \cdot 10^{-14}$	5.873	$5.7 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$6.4 \cdot 10^{-10}$	3.988	$-6.2 \cdot 10^{+00}$	$1.1 \cdot 10^{-16}$	2.000	$1.1 \cdot 10^{-11}$
$1/5 \cdot 2^{-7}$	$4.0 \cdot 10^{-11}$	3.994	$-6.5 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	1.000	$1.7 \cdot 10^{-14}$
$1/5 \cdot 2^{-8}$	$2.5 \cdot 10^{-13}$	3.997	$-6.6 \cdot 10^{+00}$	$1.3 \cdot 10^{-17}$	-1.000	$-1.0 \cdot 10^{-20}$
$1/5 \cdot 2^{-9}$	$1.5 \cdot 10^{-14}$	3.998	$-6.7 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	0.000	$-2.7 \cdot 10^{-17}$
$1/5 \cdot 2^{-10}$	$9.9 \cdot 10^{-15}$	3.999	$-6.7 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	0.000	$2.7 \cdot 10^{-17}$
$1/5 \cdot 2^{-11}$	$6.2 \cdot 10^{-16}$	3.999	$-6.7 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	-0.584	$1.2 \cdot 10^{-19}$
$1/5 \cdot 2^{-12}$	$3.8 \cdot 10^{-17}$	3.999	$-6.8 \cdot 10^{+00}$	$4.1 \cdot 10^{-16}$	0.584	$-1.3 \cdot 10^{-14}$
$1/5 \cdot 2^{-13}$	$2.4 \cdot 10^{-19}$	3.999	$-6.8 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	0.000	$-2.7 \cdot 10^{-17}$
$1/5 \cdot 2^{-14}$	$1.5 \cdot 10^{-20}$	3.999	$-6.8 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	0.000	$2.7 \cdot 10^{-17}$
$1/5 \cdot 2^{-15}$	$9.4 \cdot 10^{-21}$	3.999	$-6.8 \cdot 10^{+00}$	$2.7 \cdot 10^{-17}$	1.000	$4.5 \cdot 10^{-11}$

Tabelle 3.8: Ordnung lokaler Fehler IRK 5 für Beispiel 2a an $x^* = 0, 0.5$

h	$e_{0.0}$	$p_{0.0}$	$c_{0.0}$	$e_{0.5}$	$p_{0.5}$	$c_{0.5}$
$1/5$	$4.0 \cdot 10^{-03}$	4.635	$-7.0 \cdot 10^{+00}$	$1.1 \cdot 10^{-05}$	6.277	$-2.7 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$1.6 \cdot 10^{-05}$	4.345	$-3.6 \cdot 10^{+00}$	$1.4 \cdot 10^{-07}$	6.151	$-2.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$8.0 \cdot 10^{-06}$	4.181	$-2.2 \cdot 10^{-01}$	$2.0 \cdot 10^{-09}$	6.080	$-1.6 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$4.4 \cdot 10^{-07}$	4.093	$-1.6 \cdot 10^{-01}$	$3.0 \cdot 10^{-11}$	6.041	$-1.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$2.6 \cdot 10^{-09}$	4.047	$-1.3 \cdot 10^{-01}$	$4.5 \cdot 10^{-12}$	6.016	$-1.2 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$1.5 \cdot 10^{-10}$	4.023	$-1.1 \cdot 10^{-01}$	$7.1 \cdot 10^{-14}$	6.000	$-1.1 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$9.6 \cdot 10^{-11}$	4.011	$-1.0 \cdot 10^{-01}$	$1.1 \cdot 10^{-16}$		
$1/5 \cdot 2^{-7}$	$6.0 \cdot 10^{-12}$	4.005	$-1.0 \cdot 10^{-01}$	0.0		
$1/5 \cdot 2^{-8}$	$3.7 \cdot 10^{-13}$	4.003	$-1.0 \cdot 10^{-01}$	$5.5 \cdot 10^{-16}$	-1.000	$-4.3 \cdot 10^{-19}$
$1/5 \cdot 2^{-9}$	$2.3 \cdot 10^{-15}$	4.001	$-1.0 \cdot 10^{-01}$	$1.1 \cdot 10^{-16}$		
$1/5 \cdot 2^{-10}$	$1.4 \cdot 10^{-16}$	4.000	$-1.0 \cdot 10^{-01}$	0.0		
$1/5 \cdot 2^{-11}$	$9.0 \cdot 10^{-17}$	4.000	$-1.0 \cdot 10^{-01}$	$5.5 \cdot 10^{-16}$	-1.000	$5.4 \cdot 10^{-20}$
$1/5 \cdot 2^{-12}$	$5.6 \cdot 10^{-18}$	4.000	$-1.0 \cdot 10^{-01}$	$1.1 \cdot 10^{-16}$	1.000	$-2.2 \cdot 10^{-12}$
$1/5 \cdot 2^{-13}$	$3.5 \cdot 10^{-19}$	4.000	$-1.0 \cdot 10^{-01}$	$5.5 \cdot 10^{-16}$	-1.584	$-2.7 \cdot 10^{-24}$
$1/5 \cdot 2^{-14}$	$2.2 \cdot 10^{-21}$	4.000	$-1.0 \cdot 10^{-01}$	$1.6 \cdot 10^{-16}$	0.584	$1.2 \cdot 10^{-13}$
$1/5 \cdot 2^{-15}$	$1.3 \cdot 10^{-22}$	4.000	$-1.0 \cdot 10^{-01}$	$1.1 \cdot 10^{-16}$	-0.584	$9.8 \cdot 10^{-19}$

Tabelle 3.9: Ordnung lokaler Fehler IRK 5 für Beispiel 2b an $x^* = 0, 0.5$

Ordnung lokaler Fehler von IRK 5 für Beispiel 2b: Beispiel 2b unterscheidet sich in diesem Fall kaum von Beispiel 2a. Tabelle 3.9 zeigt eine Ordnungsreduktion an der Singularität um zwei h -Potenzen. Bei $x^* = 0.5$ stellen sich die klassischen Ordnungen ein. Zu beachten ist, daß das explizite Runge-Kutta Verfahren hier auf $\tilde{p} = 3$ reduziert wurde, vgl. Tabelle 3.4.

h	$e_{0.0}$	$p_{0.0}$	$c_{0.0}$	$e_{0.5}$	$p_{0.5}$	$c_{0.5}$
$1/5$	$3.8 \cdot 10^{-08}$	6.884	$-2.5 \cdot 10^{-04}$	$6.2 \cdot 10^{-07}$	5.771	$-6.7 \cdot 10^{-03}$
$1/5 \cdot 2^{-1}$	$3.2 \cdot 10^{-10}$	6.362	$-7.5 \cdot 10^{-04}$	$1.1 \cdot 10^{-09}$	5.861	$-8.3 \cdot 10^{-03}$
$1/5 \cdot 2^{-2}$	$4.0 \cdot 10^{-12}$	6.114	$-3.6 \cdot 10^{-04}$	$1.9 \cdot 10^{-11}$	5.922	$-1.0 \cdot 10^{-03}$
$1/5 \cdot 2^{-3}$	$5.7 \cdot 10^{-14}$			$3.2 \cdot 10^{-12}$	5.935	$-1.0 \cdot 10^{-03}$
$1/5 \cdot 2^{-4}$	0.0			$5.3 \cdot 10^{-14}$	5.584	$-2.2 \cdot 10^{-04}$

Tabelle 3.10: Ordnung lokaler Fehler IRK 5 für Beispiel 3a an $x^* = 0, 0.5$

Ordnung lokaler Fehler von IRK 5 für Beispiel 3a: In diesem Fall, vgl. Tabelle 3.10, können keine eindeutigen Aussagen über die Ordnung des lokalen Fehlers an $x^* = 0$ gemacht werden, da dieser zu schnell im Rundungsfehlerbereich liegt. Dagegen ist, wie in allen anderen Beispielen, bei 0.5 die klassische Ordnung mit adäquaten Fehlerkonstanten zu beobachten. Es zeigt sich bei $x^* = 0.5$ $p_{0.5} \approx 6$ mit relativ kleinen Fehlerkonstanten $c_{0.5}$.

3.3.3 Ordnungskurven

Weiters soll festgestellt werden, wie schnell sich die Ordnung von der Ordnungsreduktion an der Singularität erholt. Dazu berechnen wir die Ordnung des lokalen Fehlers mit bestimmten Verfahren für jeden Punkt eines feinen Gitters, welches unabhängig von der Schrittweite gewählt ist.

Ordnungskurven lokaler Fehler von ERK 4 für Beispiel 2b: Wir verwenden die 3/8-Regel der Ordnung 4, vgl. [18]. In Abb. 3.49 stellt sich wie erwartet die Ordnung 2 an der Singularität ein, vgl. Tabelle 3.13 bzw. Abschnitt 4.1.2. Mit wachsender Entfernung von der Singularität nimmt die Ordnung immer mehr zu, bis der klassische Wert $p = 5$ erreicht wird. Je kleiner die Schrittweite ist, desto schneller geht die Ordnung gegen den klassischen Wert, was auch schon in Abschnitt 3.3.5 beobachtet wurde. Alle Beispiele mit konstanter Koeffizientenmatrix verhalten sich analog, vgl. [30]

Ordnungskurven lokaler Fehler von IRK 4 für Beispiel 2b: Auch die impliziten Verfahren, in der Abb. 3.50 wurde das Verfahren von Gauss 4. Ordnung verwendet, zeigen die gleichen Ergebnisse. Der einzige Unterschied zu der Abb. 3.49 liegt darin, daß die Ordnungsreduktion an der Singularität drei statt zwei beträgt.

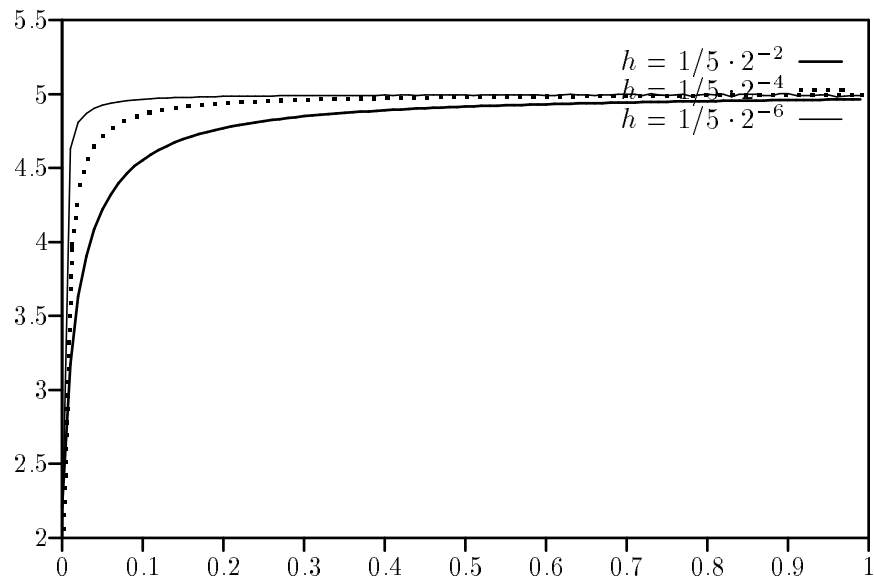


Abbildung 3.49: Ordnungskurven lokaler Fehler ERK 4 für Beispiel 2b

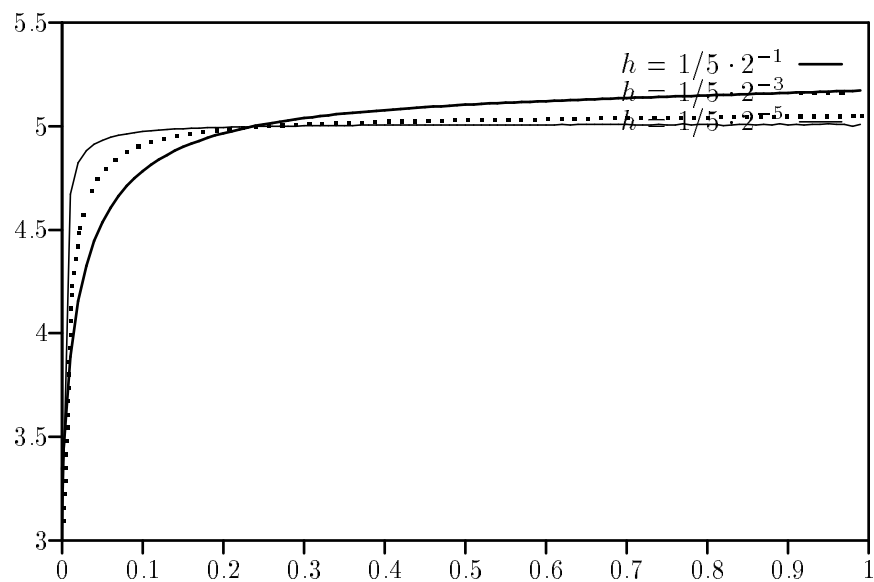


Abbildung 3.50: Ordnungskurven lokaler Fehler IRK 4 für Beispiel 2b

3.3.4 Ordnung bezüglich x und h

Es zeigt sich, daß die Ordnung des lokalen Fehlers um die Singularität von x und h abhängig ist, also $p = p(x, h)$. Besonders interessant ist daher das kleinste $x =: x^*$, von wo aus mit Methoden der lokalen Fehlerschätzung gearbeitet werden kann. Wird dieses x^* zu nahe bei Null gewählt, so wird das Problem durch die lokalen Methoden nicht effektiv und verläßlich integriert, vgl. z.B. Abschnitte 3.1.6 und 3.2.3. Fordern wir eine untere Grenze p_{\min} für die Ordnung, z.B. $p_{\min} = 4.9$, so kann theoretisch ein $x^* = x^*(h, p_{\min})$ berechnet werden, mit $p(x, h) \geq p_{\min}$ für $x \geq x^*(h, p_{\min})$.

Aus den Abbildungen des vorigen Abschnitts ist zu vermuten, daß die Ordnung an einer Stelle x von der Anzahl der Schritte abhängt, die bis dahin durchgeführt wurden. Um nun zusätzliche Informationen über $x^*(h, p_{\min})$ zu erhalten führen wir folgenden Test durch. Mit gewissen Schrittweiten wird das Testbeispiel gelöst, wobei in jedem Schritt wieder von der exakten Lösung gestartet wird. Nach jedem Schritt wird die Ordnung $p(x, h)$ der aktuellen Stelle x berechnet und mit p_{\min} verglichen. Ist $p > p_{\min}$, so sind wir fertig und die Anzahl der Schritte, die bis jetzt gemacht wurde, wird ausgegeben. Danach wird die Schrittweite halbiert und der Test wird wiederholt.

Die Ergebnisse dieses Tests für das Beispiel 1c sind in Tabelle 3.12 gelistet, wobei wieder die 3/8-Regel 4. Ordnung zur Integration verwendet wurde. Die einzelnen Spalten zeigen jeweils die Anzahl k der notwendigen Schritte und die Ordnung des lokalen Fehlers an der Stelle $x = kh$. Es zeigt sich, daß in diesem Fall $x^*(h)$ fast in linearem Zusammenhang mit h steht,

$$x^*(h, p_{\min}) \approx kh, \quad (3.7)$$

wobei k natürlich von p_{\min} abhängt.

Das Ergebnis der analogen Tests aller anderen Testbeispiele ist in Tabelle 3.11 dargestellt. Dabei sind zwei Typen von Beispielen zu unterscheiden. Die meisten Beispiele, 1a, 2a und 2b, zeigen das gleiche Verhalten, wie Beispiel 1c, vgl. Tabelle 3.12. Das heißt für die ersten, groben h -Werte stellt sich unter Umständen keine Ordnung ein, danach ist der lineare Zusammenhang selbst bis zu sehr kleinen Schrittweiten gut zu erkennen. Im Vergleich der Beispiele 1a und 1c ist zu vermuten, daß mit höherem Exponent x^k der Lösung die Ordnung sich schneller normalisiert.

p_{\min}	4.9	4.95	4.975	4.9875	4.99372	4.996875
Beispiel 1a	12	22	44	90	150 – 160	160 – 175
Beispiel 1b	11 – 7	16 – 8	43 – 15	37 – 15	49 – 15	59 – 16
Beispiel 1c	15	30	58	113 – 123	146 – 161	161 – 180
Beispiel 2a	12	23	44	89	140 – 160	160 – 175
Beispiel 2b	15	30	58	100 – 120	120 – 160	120 – 160
Beispiel 3a	8 – 3	11 – 5	21 – 11	53 – 11	53 – 11	53 – 11

Tabelle 3.11: Anzahl Schritte bis zum Erreichen der Ordnung für alle Beispiele

Alle anderen Beispiele zeigen für geeignet kleine Schrittweiten abfallende Folgen von Schritten, die notwendig sind, um die Ordnung zu erreichen. D.h. die Ordnung erholt sich in diesen Beispielen mit kleiner werdender Schrittweite schneller, als in den

Beispielen der Art 1c.

p_{\min}		4.9		4.95		4.975		4.9875		4.99375
h	k	p	k	p	k	p	k	p	k	p
$1/5 \cdot 2^{-1}$	10	4.92								
$1/5 \cdot 2^{-2}$	12	4.91	18	4.951						
$1/5 \cdot 2^{-3}$	13	4.90	23	4.953	36	4.9758				
$1/5 \cdot 2^{-4}$	14	4.90	26	4.952	44	4.9752	71	4.98774		
$1/5 \cdot 2^{-5}$	15	4.90	28	4.952	51	4.9756	91	4.98753	148	4.996660
$1/5 \cdot 2^{-6}$	15	4.90	29	4.951	54	4.9751	102	4.98774	161	4.994700
$1/5 \cdot 2^{-7}$	15	4.90	29	4.950	56	4.9752	123	4.98935	145	5.000474
$1/5 \cdot 2^{-8}$	15	4.90	30	4.952	57	4.9750	110	4.99237	153	5.000884
$1/5 \cdot 2^{-9}$	15	4.90	30	4.951	58	4.9756	109	4.98780	131	5.001527
$1/5 \cdot 2^{-10}$	15	4.90	30	4.951	58	4.9754	124	4.99815	124	4.998151
$1/5 \cdot 2^{-11}$	15	4.90	30	4.951	58	4.9751	109	4.98821	129	5.000000
$1/5 \cdot 2^{-12}$	15	4.90	30	4.951	58	4.9754	120	4.99376	120	4.993755
$1/5 \cdot 2^{-13}$	15	4.90	30	4.951	58	4.9754	109	4.99009	127	4.997346
$1/5 \cdot 2^{-14}$	15	4.90	30	4.951	59	4.9752	104	4.98792	127	4.999554
$1/5 \cdot 2^{-15}$	15	4.90	30	4.951	58	4.9751	97	4.98789	131	4.994107
$1/5 \cdot 2^{-16}$	15	4.90	30	4.951	58	4.9751	115	4.99236	124	4.998329
$1/5 \cdot 2^{-17}$	15	4.90	30	4.951	59	4.9750	113	4.98944	124	4.996657
$1/5 \cdot 2^{-18}$	15	4.90	30	4.951	58	4.9751	116	4.99555	116	4.995553
$1/5 \cdot 2^{-19}$	15	4.90	30	4.951	59	4.9752	115	4.99002	154	5.003217
$1/5 \cdot 2^{-20}$	15	4.90	30	4.951	59	4.9752	107	4.98942	134	5.004237
$1/5 \cdot 2^{-21}$	15	4.90	30	4.951	58	4.9752	107	4.98956	160	5.000000
$1/5 \cdot 2^{-22}$	15	4.90	30	4.951	59	4.9755	131	4.98868	140	4.994580
$1/5 \cdot 2^{-23}$	15	4.90	30	4.951	58	4.9752	122	4.98769	124	4.995819
$1/5 \cdot 2^{-24}$	15	4.90	30	4.951	59	4.9754	118	4.98814	121	4.998055
$1/5 \cdot 2^{-25}$	15	4.90	30	4.951	58	4.9752	103	4.98973	146	4.996580

Tabelle 3.12: Anzahl Schritte bis zum Erreichen der Ordnung in Beispiel 1c

Nach Tabelle 3.11 scheint die Wahl $x^* := 30h$ zur numerischen Berechnung sinnvoll zu sein. Wir wollen dies aber anhand weiterer analytischer Überlegungen mit dem Computeralgebrasystem MAPLE belegen. Ein Programm liefert uns die Ordnung an der Stelle $x_0 = kh$ mit Hilfe von Extrapolation, also nicht die exakte Ordnung, da diese im Laufe eines Schrittes variiert.

```

> ordkurve := proc(k,h)
> # gibt Ordnungskurve der 1. Komponente fuer h an der Stelle h*k zurueck
> # global v(v)...Loesung
> # global F(x,z)...rechte Seite
> local a,v0,k1,k2,k3,k4,lokf,w;
> # lokal w ... Platzhalter fuer h und h/2
> v0 := v(a);
> if (a=0) then k1 := vector([0,0]);
> else k1 := F(a,v0);
> fi;
> k2 := F(a+c2*w,evalm(v0+w*a21*k1));
> k3 := F(a+c3*w,evalm(v0+w*(a31*k1+a32*k2)));
> k4 := F(a+c4*w,evalm(v0+w*(a41*k1+a42*k2+a43*k3)));
> evalm(v0 + w*(b1*k1+b2*k2+b3*k3+b4*k4));

```

```

> lokf := evalm(" - v(a+w)):
> subs(w=h,lokf[1]):
> subs(w=h/2,lokf[1]):
> ln(abs(""/"))/ln(2):
> expand(simplify(subs(a = k*h,simplify(")))):
> end:

```

Wir machen zusätzliche Voraussetzungen, damit MAPLE besser zurecht kommt.

```

> assume(k, positive):
> assume(h, AndProp(RealRange(1e-16,0.1),positive)):

```

Zuerst untersuchen wir einige Testbeispiele mit der 3/8-Regel. Beispiel 1a:

```

> g := x -> 2^4*x^4*exp(4-8*x):
> l1 := 0: l2 := 0:
> y := x -> g(x):
> f := x -> D(D(g))(x)-(l1+l2-1)*D(g)(x)/x+l1*l2*g(x)/x^2:
> v := x -> vector([y(x), x*D(y)(x)]):
> F := (x,z) -> vector([z[2]/x,((-l1*l2)*z[1]+(l1+l2)*z[2])/x+x*f(x)]:
> series(ordkurve(k,h),h,1): # Entwicklung nach h, konstanter Term
> convert(",polynom):

```

$$6 + \frac{\ln(6k^{\sim} + 1)}{\ln(2)} + \frac{\ln(2k^{\sim} + 1)}{\ln(2)} + \frac{\ln(405k^{\sim}e^4 + 810e^4k^{\sim 2} + 72e^4)}{\ln(2)} \\ - \frac{\ln(k^{\sim} + 1)}{\ln(2)} - \frac{\ln(1620k^{\sim}e^4 + 6480e^4k^{\sim 2} + 144e^4)}{\ln(2)} - \frac{\ln(3k^{\sim} + 2)}{\ln(2)}$$

```

> evalf(solve("=4.9,k));

```

$$10.07559817, -0.596347715, -0.098186215 + 0.07939216262 I, \\ -0.098186215 - 0.07939216262 I$$

```

> evalf(solve("=4.95,k));

```

$$20.89737052, -0.59750090, -0.099472262 + 0.07959901578 I, \\ -0.099472262 - 0.07959901578 I$$

Die negativen und komplexen Werte für k scheiden aus, somit bleiben 10.07 und 20.9. Diese Ergebnisse stimmen mit Tabelle 3.11, mit den Werten 12 und 22 etwa überein. Eine genauere Berechnung über den linearen Term der Ordnungskurve ist aus Gründen der Komplexität mit MAPLE nicht mehr möglich.

```

> series(ordkurve(k,h),h,2): # Entwicklung nach h, linearer Term
> convert(",polynom):
> evalf(solve("=4.9,k));
Error, (in solve/scalar) integer too large in context

```

Beispiel 1c in gekürzter Form:

```

> g := x -> x^2*exp(2*x):
> l1 := -1: l2 := -2: ...
> series(ordkurve(k,h),h,1): # Entwicklung nach h, konstanter Term
> convert(",polynom):
> evalf(solve("=4.9,k));

```

$$13.84698686, -0.5809271330$$

```

> evalf(solve("=4.95,k));

```

$$28.27225316, -0.5821300230$$

Die Werte 13.85 und 28.27 liegen nahe an den Werten 15 und 30 aus Tabelle 3.11.

Beispiel 2b mit variabler Koeffizientenmatrix,

```
> y := x -> x^2*exp(x): ...
> series(ordkurve(k,h),h,1): # Entwicklung nach h, konstanter Term
> convert(",polynom):
> evalf(solve("=4.9,k));
13.84698686, -.5809271330

> evalf(solve("=4.95,k));
28.27225316, -.5821300230
```

liefert die gleichen Werte nahe bei 15 und 30, wie Beispiel 1c.

Für allgemeine Beispiele mit konstanter Koeffizientenmatrix gilt ebenfalls

```
> g := 'g':
> l1 := 'l1': l2 := 'l2': ...
> series(ordkurve(k,h),h,3): # Entwicklung nach h, hoehere Terme
> convert(",polynom):
> simplify(limit(",h=0));
(3 ln(2) + ln(2 k~ + 1) + ln(6 k~ + 1) - ln(3 k~ + 2) - ln(k~ + 1)) / (ln(2))

> evalf(solve("=4.9,k));
13.84698648, -.5809271375

> evalf(solve("=4.95,k));
28.27225101, -.5821300045
```

unabhängig von der Wahl der Eigenwerte λ_1 und λ_2 . Die Berechnung ist jedoch etwas schwieriger, da wegen der Terme $\ln(1/h)$ der konstante Term der Entwicklung nach h nicht direkt mit `series` ausgerechnet werden kann. Der Grenzübergang $h \rightarrow 0$ kann aber durchgeführt werden. Diese ersten Terme der Entwicklung, 14 und 28, liefern offensichtlich eine gute Näherung für das exakte x^* ,

```
> evalf(solve(ordkurve(k,h)=4.9,k));
```

bzw.

```
> evalf(solve(ordkurve(k,h)=4.95,k));
```

3.3.5 Zusammenfassung

Neben DOPRI5 wurden auch die lokalen Fehler anderer expliziter und impliziter Runge-Kutta Verfahren untersucht. Tabelle 3.13 zeigt die Ordnungen verschiedener expliziter Verfahren, die sich bei $x^* = 0$ eingestellt haben. Dabei sind D2 und D3 die beiden Runge-Kutta Verfahren aus dem 3(2) Paar aus Abschnitt 3.1.2 mit Ordnungen $p = 2$ und $q = 3$. DP4, DP5 und DP7 und DP8 sind jeweils die beiden Verfahren der Paare DOPRI5 und DOPRI8, vgl. Abschnitte 3.1.1 und 3.1.3, 3/8R die 3/8-Regel, GI die Formel von Gills 4. Ordnung und RK bezeichnet das klassische Runge-Kutta Verfahren mit dem Butcher-Array

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

Verfahren	D2	D3	DP4	RK	3/8R	GI	DP5	DP7	DP8
klass. Ord.	3	4	5	5	5	5	6	8	9
Beispiel 1a	4	4	4	5	4	5	4	5	5
Beispiel 1b	2	4	2	4	4	4	6	8	?
Beispiel 1c	2	2	2	2	2	2	?	?	?
Beispiel 2a	4	4	4	5	4	5	4	5	5
Beispiel 2b	2	2	3	3	2	3	3	4	4
Beispiel 3a	2	4	2	4	4	4	4	4	4

Tabelle 3.13: Ordnungen lokaler Fehler ERK aller Beispiele an $x^* = 0$

Tabelle 3.14 zeigt die Ordnungen verschiedener impliziter Verfahren, die sich bei $x^* = 0$ eingestellt haben. Dabei stehen RIIA3 und RIIA5 für die rechtsseitigen Radau Formeln 3. und 5. Ordnung aus [19], Seite 74, Abb. 5.5 und 5.6, und G4 bzw. G6 für die Gauss Formeln 4. und 6. Ordnung, [19], Seite 72, Abb. 5.1 und 5.2.

Verfahren	RIIA3	G4	RIIA5	G6
klass. Ord.	4	5	6	7
Beispiel 1a	4	4	4	4
Beispiel 1b	4	4	6	6
Beispiel 1c	3	3	4	?
Beispiel 2a	4	4	4	4
Beispiel 2b	3	3	4	4
Beispiel 3a	4	4	6	6

Tabelle 3.14: Ordnungen lokaler Fehler IRK aller Beispiele an $x^* = 0$

Man sieht, daß die Singularität beträchtliche Ordnungsreduktionen verursachen kann. Bis zu vier h -Potenzen sind möglich und der nach [24] theoretisch schlechteste Fall, Reduktion auf Ordnung 2, tritt ebenfalls ein. Jedoch geschieht dies nicht systematisch, sondern die Ergebnisse variieren je nach Beispiel und verwendetem Verfahren. So können z.B. die Ordnungen von DP4 und DP5 auf den gleichen Wert reduziert werden, dies muß aber nicht immer der Fall sein. Dies unterstreicht die Problematik, die in Abschnitt 3.1.6 angesprochen wurde.

Entfernt man sich von der Singularität, so steigt die Ordnung gegen die klassische, wobei die Ordnung des lokalen Fehlers um die Singularität von x^* und h abhängig ist, also $p = p(x^*, h)$. Im Bereich bis $x^* = 0.1$ ist zusätzlich zu bemerken, daß sich mit kleiner werdendem h die Ordnung ebenfalls verbessert. Mit wachsendem x^* stellt sich die klassischen Ordnungen für immer größere h ein. Das heißt, daß sich weg von der Singularität immer die klassische Ordnung mit passenden Fehlerkonstanten einstellen, die Fehlerkonstanten nicht allzu groß sind und sich alle Verfahren wie erwartet verhalten.

Allerdings wächst auch der Einfluß des Rechenfehlers, der sich im frühzeitigen Abbröckeln der Ordnungen zeigt. Dies mag daran liegen, daß durch die Ordnungsreduktion die lokale Lösung weniger genau ist, und daher kleinere Schrittweiten zulässig sind als bei höherer Genauigkeit. Für Beispiel 1b mit DP8 konnte keine Ordnung beobachtet werden.

Es macht jedoch den Eindruck, daß die impliziten Verfahren etwas weniger stark an Ordnung verlieren. Im Vergleich mit Tabelle 3.13 kommen in Tabelle 3.14 keine Ordnungsreduktionen auf zwei vor.

3.4 „Exakte“ Fehlerschätzung

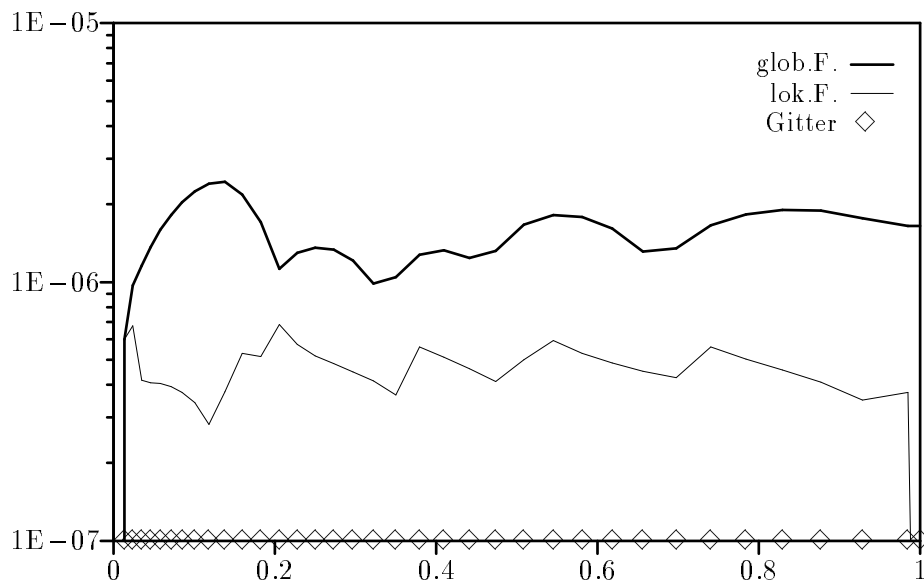
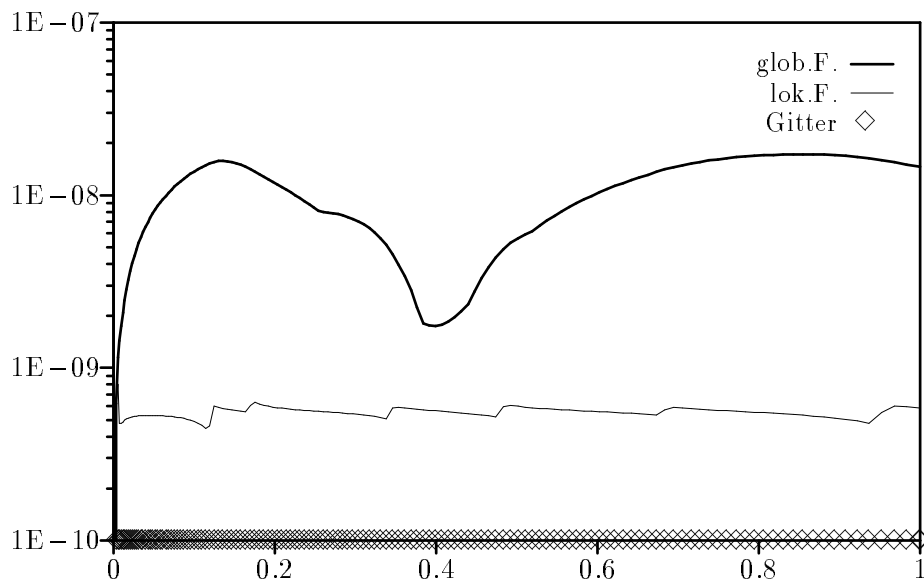
Zum Vergleich mit den Abschnitten 3.1 und 3.2 wird der exakte lokale Fehler als Fehlerschätzung herangezogen, vgl. auch Abschnitt 3.1.5, um die bestmögliche Variante auszuloten, die mit einer geeigneten Schätzung des lokalen Fehlers erreichbar ist. Man setzt dazu $d_j := \hat{e}_j$ und berechnet h_{j+1} nach Vorschrift (3.2). Dadurch werden mögliche Probleme in der Fehlerschätzung, die durch Ordnungsreduktionen bedingt sind, ausgegrenzt. Eine bessere Variante wäre aber, die nächste Schrittweite mittels exakter lokaler Ordnung zu schätzen. Daher kann die Schrittweite trotz exakter Schätzung teilweise zu klein gewählt werden, jedoch nie zu groß, da immer mit dem exakten lokalen Fehler des letzten Schrittes korrigiert wird.

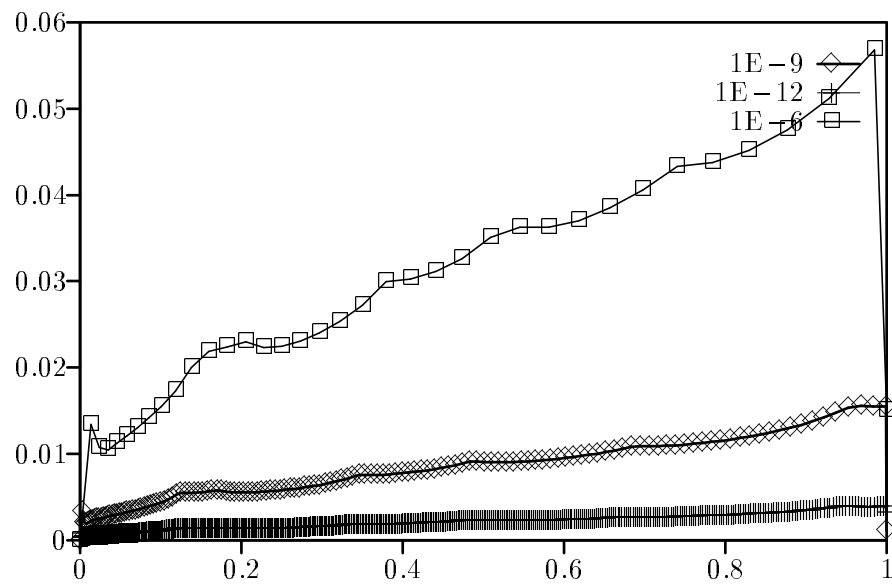
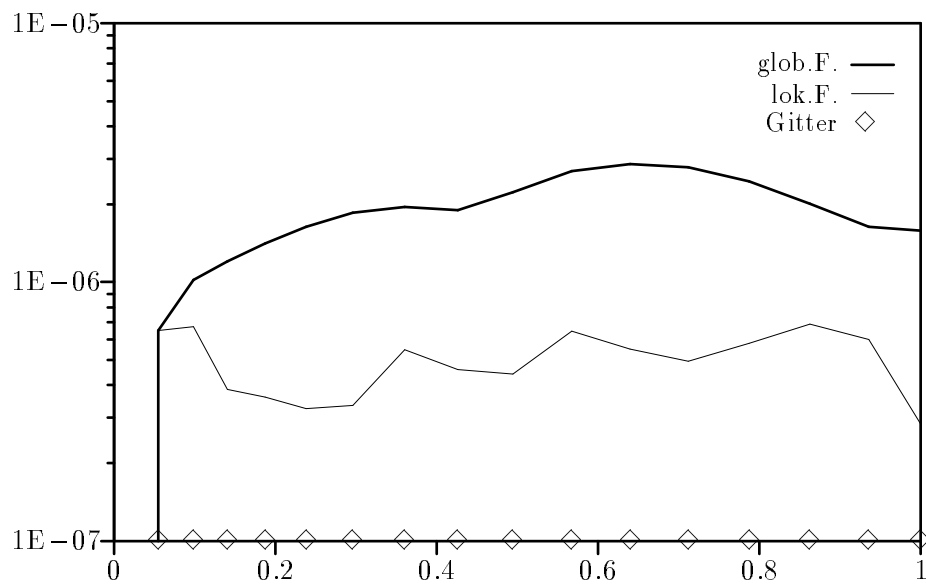
3.4.1 Explizites Runge-Kutta 4

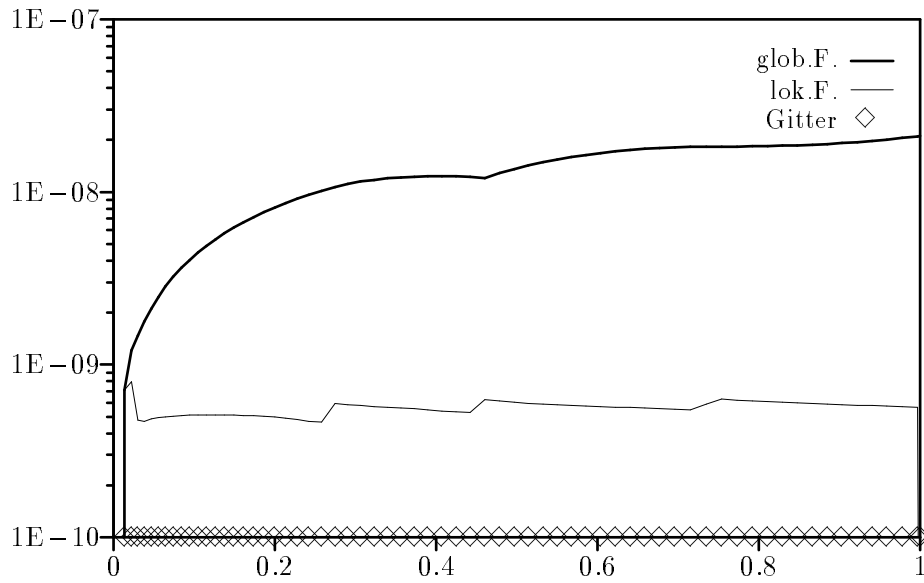
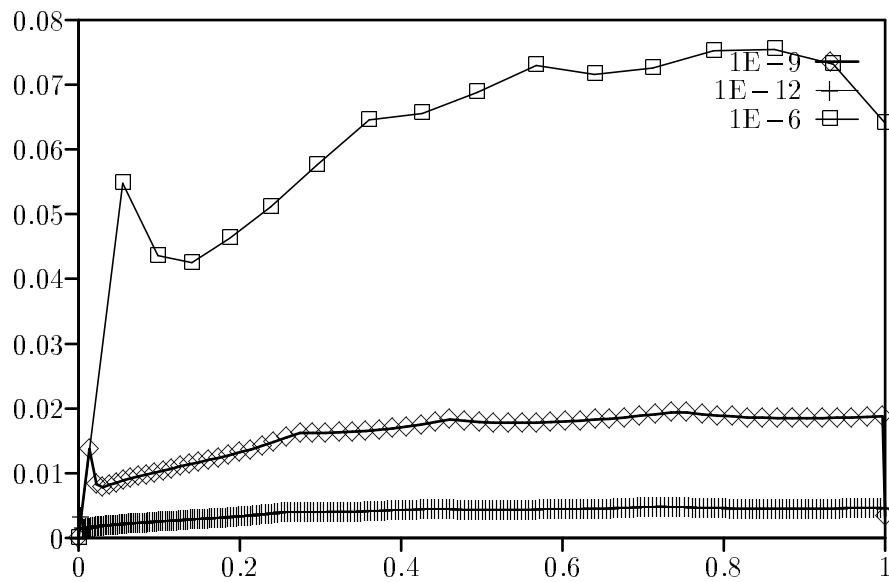
Wir wählen als Basisfunktion das klassische Runge-Kutta Verfahren aus Abschnitt 3.3.5. Mit der Ordnung 4 ist es vergleichbar mit DOPRI5, Abschnitt 3.1.1 und Abschnitt 3.2.1. Die Abbildungen zeigen jeweils Gitter, globale Fehler und lokale Fehler zu den lokalen Toleranzen $T = 1\text{E}-6$, $1\text{E}-9$ und $1\text{E}-12$.

„Exakte“ Fehlerschätzung für ERK 4 für Beispiel 1a: Die Abb. 3.51 und 3.52 zeigen folgendes: Der lokale Fehler zeigt keine ausgeprägte Spitze an der Singularität und verläuft mehr oder weniger geradlinig an der vorgegebenen Toleranz entlang. Der globale Fehler reduziert sich mit kleiner werdendem T um jeweils zwei Zehnerpotenzen, ist jedoch größer als in Abschnitt 3.2.1, und verläuft nicht so geradlinig, wie in den Abb. 3.1, 3.2 und 3.3. Die Schrittweiten, vgl. Abb. 3.53 werden prinzipiell kleiner gewählt, als in den entsprechenden Abbildungen der Runge-Kutta Paare, bzw. Richardson-Extrapolation, Abb. 3.4 bzw. 3.36.

„Exakte“ Fehlerschätzung für ERK 4 für Beispiel 2a: In Abb. 3.54 und 3.55 zeigt der lokale Fehler keine Spitze an der Singularität. Für kleine Schrittweiten verläuft er geradlinig unter der Toleranz. Der globale Fehler nimmt mit wachsender Genauigkeit um je zwei Zehnerpotenzen ab, ist jedoch größer als in den Abb. 3.8, 3.9 und 3.10, und wächst im Verlauf der Integration. Die gewählten Schrittweiten, vgl. Abb. 3.56 sind viel kleiner als von DOPRI5, Abb. 3.11, bzw. Richardson-Extrapolation, Abb. 3.39.

Abbildung 3.51: Fehler ERK 4 exakt in Beispiel 1a, Toleranz 10^{-6} Abbildung 3.52: Fehler ERK 4 exakt in Beispiel 1a, Toleranz 10^{-9}

Abbildung 3.53: $x_j h_j$ -Plot ERK 4 exakt für Beispiel 1aAbbildung 3.54: Fehler ERK 4 exakt in Beispiel 2a, Toleranz $1E-6$

Abbildung 3.55: Fehler ERK 4 exakt in Beispiel 2a, Toleranz $1E-9$ Abbildung 3.56: $x_j h_j$ -Plot ERK 4 exakt für Beispiel 2a

3.4.2 Zusammenfassung

In den Abschnitten 3.1.6 und 3.2.3 zeigte der lokale Fehler eine deutliche Spitze an der Singularität. Selbst in Abschnitt 3.1.5 war teilweise eine Zacke zu erkennen, obwohl dort der exakte lokale Fehler e_j als Schätzung verwendet wurde. Im Gegensatz dazu zeigt der lokale Fehler $\hat{e}_j = d_j$ in diesem Fall **keine** ausgeprägte Spitze an der Singularität. Die gewählten Schrittweiten sind durchwegs kleiner als bei den anderen Verfahren. Dies mag daran liegen, daß in der Schrittweitenwahl zwar der exakte lokale Fehler, aber nicht dessen Ordnung eingeht. Soll die Schrittweite vergrößert werden, so bedingt die höhere Ordnung p statt \tilde{p} in (3.2) einen kleineren Wert für h_{j+1} , als möglich wäre.

In Abschnitt 3.3.5 hat sich gezeigt, daß die Ordnung des lokalen Fehlers von expliziten Runge-Kutta Verfahren bis auf zwei fallen kann. Nach Abschnitt 3.1.6 und 3.2.3 unterschätzen dann sowohl Runge-Kutta Paare als auch Richardson-Extrapolation den lokalen Fehler in der Nähe der Singularität. Es liegt daher nahe, die Ordnung des lokalen Fehlers selbst zu bestimmen und bei der Schätzung des lokalen Fehlers zu verwenden. Dies ist natürlich mit zusätzlichem Rechenaufwand verbunden, und kann mittels einer Art „dreifachen Richardson Extrapolation“ realisiert werden, wobei analog zu (3.4) ein $3n$ -dimensionales Gleichungssystem für $u_{j-1}(x_j)$, c und p aufgelöst werden muß. Nachdem (3.4) nur asymptotisch gilt, erhalten wir Näherungswerte \tilde{p} , \hat{y}_j und \tilde{c} . Die nächste Schrittweite h_{j+1} wird mit $\hat{p} := \min_{1 \leq i \leq n}(\tilde{p}_i)$ bestimmt.

Die Experimente in [30] zeigen, daß die Methode der „dreifachen Extrapolation“ geeignet ist, um Ordnungen zu schätzen. So werden Ordnungsreduktionen an der Singularität erkannt und die Fehlerkonstanten haben die gleiche Größenordnung wie die exakten, und sind im allgemeinen nicht zu groß. Weg von der Singularität ist die Schätzung der Ordnung noch genauer und gibt teilweise die exakte Ordnung auf die ersten Stellen exakt wieder. Bei der Schrittweitenwahl durch „dreifache Extrapolation“ zeigt der lokale Fehler am Beginn des Intervalls eine mehr oder weniger ausgeprägte Spitze und verhält sich wesentlich unsystematischer als mit Runge-Kutta Paaren, bzw. Richardson-Extrapolation.

Die Schätzung gibt das qualitative Verhalten des lokalen Fehlers gut wieder und folgt auch bei der Singularität dem lokalen Fehler genau, weil Ordnungsreduktionen erkannt werden. Dadurch ist die Schätzung des lokalen Fehlers exakter, daher werden am Anfang kleinere Schritte gewählt, allerdings größer als für die „exakte“ Fehlerschätzung. Weg von der Singularität verläuft die Schrittweitensteuerung wie für Richardson-Extrapolation. Leider ist diese Methode aufgrund des hohen Aufwandes und einer gewissen Unsystematik nicht für die Anwendung geeignet.

4 Struktur des lokalen Fehlers

Nachdem in Abschnitt 3 verschiedene Methoden zur Schätzung des lokalen Fehlers experimentell untersucht worden sind, wird in diesem Abschnitt der lokale Fehler an der Singularität analytisch untersucht. Er ergänzt somit Abschnitt 3.3. Verwendet wird dabei wieder das Computeralgebrasystem MAPLE. Der Leser sollte in diesem Abschnitt mit der MAPLE Syntax vertraut sein.

4.1 Ordnung des lokalen Fehlers ERK 4

Wir untersuchen den lokalen Fehler der Testbeispiele und vergleichen die Ergebnisse mit Abschnitt 3.3.5. Dabei berechnen wir den exakten lokalen Fehler der 3/8-Regel für die singulären 2-dimensionalen Testbeispiele an der Singularität mit MAPLE und entwickeln ihn in h -Potenzen. Der erste Koeffizient der nicht verschwindet gibt den Vektor der Ordnungen des lokalen Fehlers der beiden Komponenten an der Stelle $x = 0$ an.

Zur Manipulation von vektorwertigen Funktionen benötigt man das Paket für die Lineare Algebra (`with(linalg):`). Dann wird die Position, wo der lokale Fehler berechnet werden soll,

```
> x0 := 0:
```

und das Runge-Kutta Schema definiert.

```
> a21 := 1/3: a31 := -1/3: a32 := 1: a41 := 1: a42 := -1: a43 := 1:
> b1 := 1/8: b2 := 3/8: b3 := 3/8: b4 := 1/8:
> c1 := 0: c2 := a21: c3 := a31+a32: c4 := a41+a42+a43:
```

Das MAPLE Programm `calclok` übernimmt die Berechnung des lokalen Fehlers, vgl. Abb. 4.1. Zuerst werden die k_i für (2.10) berechnet. Dabei ist zu beachten, daß $k_1 = F(x_j, y_j)$ in diesem Fall $F(x_0, y_0)$ ist, was wegen $x_0 = 0$ **nicht** automatisch berechnet werden kann. Es gilt jedoch nach Abschnitt 2.1 für lineare Probleme die Formel (2.2),

$$y'(0) = F(0, y(0)) = (I_n - M(0))^{-1} \cdot f(0).$$

Nachdem alle Testbeispiele transformierte Systeme zweiter Ordnung sind, bedeutet das

$$v'(0) = F(0, v(0)) = (I_n - M(0))^{-1} \cdot 0 \cdot f(0) = 0,$$

also müssen `vd0` und `vd1` für die betrachteten Beispiele stets Null gewählt werden. Für Systeme erster Ordnung, die nicht durch Transformation aus (2.3) entstehen, ist dies nicht der Fall, und die Berechnung von $v'(0)$ kann vor allem im nichtlinearen Fall Probleme bereiten.

Nach der Berechnung der k_i wird (2.10) ausgewertet, die exakte Lösung wird abgezogen und die Differenz wird nach Potenzen von h entwickelt, wobei die Länge der Entwicklung mittels `sum` angegeben werden kann.

Zur Überprüfung der Beispiele braucht nur noch die Lösung $\mathbf{v}(\mathbf{x})$, die rechte Seite $\mathbf{F}(\mathbf{x}, \mathbf{v}(\mathbf{x}))$ und der Startwert `vd0` definiert zu werden.


```

> calclok := proc(vd0,vd1,sum)
> # Global F(x,z[.]) ... rechte Seite als Funktion in x und z[.].
> # Global x0 ... Startposition
> # Global v[.](x) ... exakte Loesung
> # Global v0[.] ... Startwert
> # vd0, vd1 ... Werte von F(0,v0)[1] und F(0,v0)[2].
> # sum ... Anzahl der Folgenglieder fuer series
> local k1,k2,k3,k4:
> k1 := vector([vd0,vd1]): # F(x0,v0);
> k2 := F(x0+c2*h,evalm(v0+h*a21*k1)):
> k3 := F(x0+c3*h,evalm(v0+h*(a31*k1+a32*k2))):
> k4 := F(x0+c4*h,evalm(v0+h*(a41*k1+a42*k2+a43*k3))):
> evalm(v0 + h*(b1*k1+b2*k2+b3*k3+b4*k4)):
> evalm(" - v(x0+h)");
> map(series,"h,sum):
> map(expand,"):
> end:

```

Abbildung 4.1: MAPLE Programm zur Berechnung des lokalen Fehlers von ERK

Wir beginnen mit den linearen Anfangswertaufgaben mit konstanter Koeffizientenmatrix, Beispiel 1a:

```

> v := x -> vector([2^4*x^4*exp(4-8*x), 2^4*x^4*exp(4-8*x)*(4-8*x)]):
> F := (x,z) -> vector([z[2]/x, 2^4*x^3*exp(4-8*x)*((64*x-72)*x+16)]):
> v0 := v(x0):
> calclok(0,0,7);

```

$$\left[-\frac{64}{27}e^4h^4 + \frac{856}{27}e^4h^5 - \frac{4736}{27}e^4h^6 + O(h^7) - \frac{320}{27}e^4h^5 + \frac{512}{3}e^4h^6 + O(h^7) \right]$$

liefert eine Entwicklung, die mit h^4 startet, wie es auch in der Spalte der 3/8-Regel (3/8R) von Tabelle 3.13 zu sehen ist.

Beispiel 1b:

```

> v := x -> vector([1+cos(3*x), -3*x*sin(3*x)]):
> F := (x,z) -> vector([z[2]/x, -z[2]/x -3^2*x*cos(3*x)-3*2*sin(3*x)]):
> v0 := v(x0):
> calclok(0,0,10);

```

$$\left[-\frac{1}{4}h^4 + \frac{43}{160}h^6 - \frac{339}{4480}h^8 + O(h^{10}) \frac{1}{4}h^4 - \frac{53}{80}h^6 + \frac{75}{224}h^8 + O(h^{10}) \right]$$

Beispiel 2a:

```

> v := x -> vector([(x/2)^4*exp(4-2*x), (x/2)^4*exp(4-2*x)*(4-2*x)]):
> F := (x,z) -> vector([z[2]/x, 4*x*z[1]+(x/2)^3/2*exp(4-2*x)*(16-18*x)]):
> v0 := v(x0):
> calclok(0,0,7);

```

$$\left[-\frac{1}{108}e^4h^4 + \frac{107}{3456}e^4h^5 - \frac{737}{15552}e^4h^6 + O(h^7) - \frac{5}{432}e^4h^5 + \frac{11}{432}e^4h^6 + O(h^7) \right]$$

Beispiel 2b:

```

> v := x -> vector([x^2*exp(x), x^2*exp(x)*(2+x)]):
> F := (x,z) -> vector([z[2]/x, (z[1]*(x*(4+cosh(x))+x^2)+
> z[2]*(1-cosh(x)))/x+2*x*exp(x)*(1+cosh(x))]):
> v0 := v(x0):
> calclok(0,0,6);

```

$$\left[-\frac{1}{12}h^2 - \frac{7}{12}h^3 - \frac{19}{48}h^4 - \frac{175}{1296}h^5 + O(h^6) - \frac{5}{12}h^3 - \frac{5}{6}h^4 - \frac{415}{1296}h^5 + O(h^6) \right]$$

In diesem und in Beispiel 2a ist zu sehen, daß die verschiedenen Komponenten des Systems durchaus verschiedene Ordnungen im lokalen Fehler haben können. Ein Effekt, der bis jetzt nicht untersucht wurde, da in Abschnitt 3 immer nur die Ordnung des lokalen Fehlers der ersten Komponente betrachtet wurde.

Beispiel 3a:

```
> v := x -> vector([1/sqrt(1+x^2/3), -x^2/3/sqrt(1+x^2/3)^3]);
> F := (x,z) -> vector([z[2]/x, -z[2]/x - x*z[1]^5]);
> v0 := v(x0);
> calclok(0,0,10);

$$\left[ -\frac{1}{24}h^4 + \frac{5}{432}h^6 - \frac{35}{10368}h^8 + O(h^{10}), \frac{7}{48}h^4 - \frac{35}{144}h^6 + \frac{335}{2592}h^8 + O(h^{10}) \right]$$

```

Alle Beispiele zeigen die Ordnungen, die auch schon experimentell gezeigt werden konnten. Diese Übereinstimmung bestätigt zwar die numerisch gewonnenen Ergebnisse, gibt aber keine neuen Einblicke.

Immer noch ist nicht klar, warum manche Testbeispiele Ordnungsreduktionen bis auf 2 erfahren und andere nicht. Dies kann nur anhand eines allgemeinen Runge-Kutta Verfahrens untersucht werden. Das heißt, bei der Definition des Runge-Kutta Schemas der Ordnung 4 werden nur die c_i definiert,

```
> a21:='a21': a31:='a31': a32:='a32': a41:='a41': a42:='a42': a43:='a43':
> b1:='b1': b2:='b2': b3:='b3': b4:='b4':
> c1 := 0: c2 := a21: c3 := a31+a32: c4 := a41+a42+a43:
```

die anderen Koeffizienten bleiben variabel.

4.1.1 Lineare Probleme mit konstanter Koeffizientenmatrix

Nachdem die Struktur der Lösung bekannt ist, betrachten wir allgemeine Anfangswertprobleme zweiter Ordnung mit konstanter Koeffizientenmatrix, wobei jedoch in drei Fälle unterschieden werden muß. Erstens, beide Eigenwerte λ_1, λ_2 des transformierten Systems sind negativ. Zweitens, $\lambda_1 < 0, \lambda_2 = 0$ und drittens $\lambda_1 = \lambda_2 = 0$. Wir beginnen mit Fall 1, $\lambda_1 \leq \lambda_2 < 0$.

Wir definieren zuerst die Lösung des Problems zweiter Ordnung $\mathbf{y}(\mathbf{x})$. Da nach Lemma [30] $y(x) = y_p(x)$ gilt, ist $\mathbf{y}(\mathbf{x}) = \mathbf{g}(\mathbf{x})$, wobei $\mathbf{g}(\mathbf{x})$ die Lösung des inhomogenen Problems bezeichnet.

```
> y := x -> g(x);
> f := x -> D(D(g))(x) - (l1+l2-1)*D(g)(x)/x + l1*l2*g(x)/x^2:
```

$\mathbf{f}(\mathbf{x})$ entspricht $f(x)$ in (2.3a), also der Inhomogenität, die verbleibt, wenn der Differentialoperator auf die partikuläre Lösung angewendet wird, wobei l1 und l2 die beiden Eigenwerte λ_1 und λ_2 sind. Das transformierte Problem erster Ordnung ist dann

```
> v := x -> vector([y(x), x*D(y)(x)]);
> F := (x,z) -> vector([z[2]/x, (-l1*l2*z[1] + (l1+l2)*z[2])/x + x*f(x)];
> v0 := v(x0):
```

und der lokale Fehler berechnet sich ebenfalls mit Hilfe von `calclok`,

```
> lok1 := calclok(0,0,5):
```

Da dieser Ausdruck ziemlich komplex ist, bearbeiten wir ihn stückweise, d.h. wir untersuchen die einzelnen Koeffizienten der h -Potenzen des lokalen Fehlers in den Kom-

ponenten $v_1(x)$ und $v_2(x)$.

> simplify(subs(h=0,lok1[i]));

0

für $i = 1, 2$.

> collect(coeff(lok1[1],h),D(g)(0));

$$\left(-\frac{b_3 a_{32} l_1}{a_{31} + a_{32}} - \frac{b_3 a_{32} l_2}{a_{31} + a_{32}} + \frac{b_3 a_{32}}{a_{31} + a_{32}} + \frac{b_3 a_{32} l_1 l_2}{a_{31} + a_{32}} - \frac{b_4 a_{42} l_1}{a_{41} + a_{42} + a_{43}} \right. \\ - \frac{b_4 a_{43} a_{32} l_1^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_4 a_{43} a_{32} l_1 l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ + \frac{b_4 a_{43} a_{32} l_1^2 l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_4 a_{43} a_{32} l_2^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ + \frac{b_4 a_{43} a_{32} l_1 l_2^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_4 a_{43} a_{31} l_1}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ - \frac{b_4 a_{43} a_{31} l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} + \frac{b_4 a_{43} a_{31}}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ + \frac{b_4 a_{43} a_{31} l_1 l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} + \frac{b_4 a_{43} a_{32}}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ \left. - \frac{b_4 a_{42} l_2}{a_{41} + a_{42} + a_{43}} + \frac{b_4 a_{42}}{a_{41} + a_{42} + a_{43}} + \frac{b_4 a_{42} l_1 l_2}{a_{41} + a_{42} + a_{43}} - 1 \right) D(g)(0)$$

Die 2. Komponente des Koeffizienten von h in lok1 hat die analoge Gestalt. Wenn $f \in \mathbf{C}$ ist, so gilt nach [30] $g'(0) = y'_p(0) = y'(0) = 0$, woraus $D(g)(0) = g'(0) = 0$ folgt.

Die ersten Terme, die nicht verschwinden sind

> collect(coeff(lok1[1],h^2),D(D(g))(0));

$$\left(-\frac{1}{2} + \frac{1}{2} \frac{b_4 a_{43} a_{31} l_1 l_2}{a_{41} + a_{42} + a_{43}} - \frac{b_4 a_{43} a_{31} l_1}{a_{41} + a_{42} + a_{43}} - \frac{b_4 a_{43} a_{31} l_2}{a_{41} + a_{42} + a_{43}} \right. \\ + 2 \frac{b_4 a_{43} a_{31}}{a_{41} + a_{42} + a_{43}} + \frac{1}{2} \frac{b_4 a_{43} a_{32} l_1 l_2}{a_{41} + a_{42} + a_{43}} - \frac{b_4 a_{43} a_{32} l_1}{a_{41} + a_{42} + a_{43}} \\ - \frac{b_4 a_{43} a_{32} l_2}{a_{41} + a_{42} + a_{43}} + 2 \frac{b_4 a_{43} a_{32}}{a_{41} + a_{42} + a_{43}} + \frac{1}{2} \frac{b_3 a_{32} a_{21} l_1 l_2}{a_{31} + a_{32}} \\ + \frac{1}{2} \frac{b_4 a_{43} a_{32} a_{21} l_1^2 l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_4 a_{43} a_{32} a_{21} l_1^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ - 2 \frac{b_4 a_{43} a_{32} a_{21} l_1 l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_3 a_{32} a_{21} l_1}{a_{31} + a_{32}} + 2 \frac{b_3 a_{32} a_{21}}{a_{31} + a_{32}} \\ + 2 \frac{b_4 a_{43} a_{32} a_{21} l_1}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} + 2 \frac{b_4 a_{42} a_{21}}{a_{41} + a_{42} + a_{43}} \\ + \frac{1}{2} \frac{b_4 a_{43} a_{32} a_{21} l_1 l_2^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} - \frac{b_4 a_{43} a_{32} a_{21} l_2^2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} \\ + 2 \frac{b_4 a_{43} a_{32} a_{21} l_2}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} + \frac{1}{2} \frac{b_4 a_{42} a_{21} l_1 l_2}{a_{41} + a_{42} + a_{43}} \\ \left. - \frac{b_4 a_{42} a_{21} l_1}{a_{41} + a_{42} + a_{43}} - \frac{b_4 a_{42} a_{21} l_2}{a_{41} + a_{42} + a_{43}} - \frac{b_3 a_{32} a_{21} l_2}{a_{31} + a_{32}} \right) D^{(2)}(g)(0)$$

bzw. ein entsprechend analoger Term für `lok1[2]`, d.h. die Entwicklung des lokalen Fehlers startet mit h^2 , wobei man diese Aussage nicht sofort auf Systeme erster Ordnung erweitern kann. Dies entspricht auch den Ergebnissen von De Hoog und Weiss, [24], daß Ordnung zwei für singuläre Anfangswertaufgaben gesichert ist

Alle weiteren Koeffizienten haben die Form

$$\begin{aligned} &> \text{collect}(\text{coeff}(\text{lok1}[i], h^3), D(D(D(g))))(0); \\ &\quad \frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43}, l_1, l_2)}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(3)}(g)(0) \end{aligned}$$

$$\begin{aligned} &> \text{collect}(\text{coeff}(\text{lok1}[i], h^4), D(D(D(D(g))))(0)); \\ &\quad \frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43}, l_1, l_2)}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(4)}(g)(0) \end{aligned}$$

$i = 1, 2$, wobei die Terme der zweiten Komponente im allgemeinen mehr Summanden enthalten. Es ist überraschend, daß jeweils nur die k . Ableitung von $y(x) = y_p(x)$ bei Null in den Koeffizienten von h^k eingeht.

Im zweiten Fall, $\lambda_1 < \lambda_2 = 0$, besteht die Lösung $y(x)$ zusätzlich aus einer Konstanten,

$$\begin{aligned} &> y := x \rightarrow c + g(x); \\ &> f := x \rightarrow D(D(g))(x) - (1-1)*D(g)(x)/x; \\ &> v := x \rightarrow \text{vector}([y(x), x*D(y)(x)]); \\ &> F := (x, z) \rightarrow \text{vector}([z[2]/x, (1*z[2])/x + x*f(x)]); \\ &> v0 := v(x0); \end{aligned}$$

Nach der Berechnung der Potenzreihe des lokalen Fehlers,

$$> \text{lok2} := \text{calclok}(0, 0, 5);$$

zeigt sich wie in Fall 1, daß die konstanten Terme und die Koeffizienten von h wegen $y'(0) = 0$ verschwinden. Die weiteren Glieder der Potenzreihe des lokalen Fehlers erfüllen wieder

$$\begin{aligned} &> \text{collect}(\text{coeff}(\text{lok2}[i], h^2), D(D(g)))(0); \\ &\quad \frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43}, l_1)}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(2)}(g)(0) \end{aligned}$$

$$\begin{aligned} &> \text{collect}(\text{coeff}(\text{lok2}[i], h^3), D(D(D(g))))(0); \\ &\quad \frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43}, l_1)}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(3)}(g)(0) \end{aligned}$$

$$\begin{aligned} &> \text{collect}(\text{coeff}(\text{lok2}[i], h^4), D(D(D(D(g))))(0)); \\ &\quad \frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43}, l_1)}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(4)}(g)(0) \end{aligned}$$

etc., $i = 1, 2$.

Für den Fall $\lambda_1 = \lambda_2 = 0$ wird mit

$$\begin{aligned} &> y := x \rightarrow c + g(x); \\ &> f := x \rightarrow D(D(g))(x) + D(g)(x)/x; \\ &> v := x \rightarrow \text{vector}([y(x), x*D(y)(x)]); \\ &> F := (x, z) \rightarrow \text{vector}([z[2]/x, x*f(x)]); \\ &> v0 := v(x0); \\ &> \text{lok3} := \text{calclok}(0, 0, 5); \end{aligned}$$

die Potenzreihe des lokalen Fehlers berechnet. Wieder startet die Potenzreihe mit h^2 , deren Koeffizienten im Vergleich mit Fall 1 relativ einfach sind, vgl.

```
> collect(coeff(lok3[1],h^2),D(D(g))(0));
```

$$\left(-\frac{1}{2} + 2 \frac{b_3 a_{32} a_{21}}{a_{31} + a_{32}} + 2 \frac{b_4 a_{42} a_{21}}{a_{41} + a_{42} + a_{43}} + 2 \frac{b_4 a_{43} a_{31}}{a_{41} + a_{42} + a_{43}} + 2 \frac{b_4 a_{43} a_{32}}{a_{41} + a_{42} + a_{43}} \right) D^{(2)}(g)(0)$$

und es gilt

```
> collect(coeff(lok3[i],h^3),D(D(D(g)))(0));
```

$$\frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43})}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(3)}(g)(0)$$

```
> collect(coeff(lok3[i],h^4),D(D(D(D(g))))(0));
```

$$\frac{u(b_1, \dots, b_4, a_{21}, \dots, a_{43})}{(a_{41} + a_{42} + a_{43})(a_{31} + a_{32})} D^{(4)}(g)(0)$$

etc., für $i = 1, 2$.

Zusammenfassend für alle drei Fälle zeigt sich, daß der lokale Fehler (2.9) eines beliebigen 4-stufiges Runge-Kutta Verfahren an der Stelle h die Form

$$e_1 = u_2 y^{(2)}(0) h^2 + u_3 y^{(3)}(0) h^3 + u_4 y^{(4)}(0) h^4 + O(h^5) \quad (4.1)$$

hat, wobei die u_i nur von den Koeffizienten des Runge-Kutta Verfahrens und den Eigenwerten des Problems abhängen. Dies erklärt, warum die Ordnung für Beispiel 1a nicht unter 4 fallen kann, da $y(x) = cx^4 e^{-\alpha x}$ ist. Man muß jedoch betonen, daß dieses Ergebnis nur für transformierte Systeme zweiter Ordnung gilt, da für allgemeine Systeme $y'(0)$ nicht Null zu sein braucht.

4.1.2 Lineare Probleme mit variabler Koeffizientenmatrix

Für Beispiele mit variablen Koeffizienten $A_i(x) = A_i(0) + x * C_i(x)$, $i = 1, 2$, gehen wir genauso vor wie für konstante A_i . Im Fall, daß $\lambda_1 \leq \lambda_2 < 0$ ist, definieren wir das allgemeine Problem mit

```
> y := x -> Phi(x) + g(x):
> f := x -> D(D(g))(x) - (11+12-1)*D(g)(x)/x + 11*12*g(x)/x^2
> -C1(x)*D(g)(x) - C0(x)/x*g(x):
```

wobei $g(x)$ die Lösung des inhomogenen Problems und $\Phi(x)$ die Lösung des homogenen Problems bezeichnet. Das transformierte Problem erster Ordnung ist dann

```
> v := x -> vector([y(x), x*D(y)(x)]):
> F := (x,z) -> vector([z[2]/x, ((-11*12+x*C0(x))*z[1]+
> (11+12+x*C1(x))*z[2])/x+x*f(x)]):
> v0 := v(x0):
```

und der lokale Fehler berechnet sich wieder mittels `calclok`,

```
> lok1 := calclok(0,0,5):
```

Im Gegensatz zu den Anfangswertaufgaben mit konstanter Koeffizientenmatrix ist der konstante Term der Entwicklung des lokalen Fehlers nicht Null,

```
> collect(subs(h=0,lok1[1]),Phi(0));
```

$$\left(-\frac{b_4 a_4^3 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} - \frac{b_3 a_3^2 l_1 l_2}{(a_3^2 + a_3^1) a_2^1} \right. \\ - \frac{b_4 a_4^3 a_3^2 l_1^2 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1) a_2^1} - \frac{b_4 a_4^2 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3) a_2^1} \\ \left. - \frac{b_4 a_4^3 a_3^2 l_1 l_2^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1) a_2^1} \right) \Phi(0)$$

bzw. ein analoger Term in der zweiten Komponente. Allerdings folgt nach [30], daß sowohl $g(x)$ als auch $\Phi(x)$ keine konstanten Terme haben, also daß $\Phi(0) = 0$ ist und die Potenzreihe des lokalen Fehlers mit h beginnt.

```
> coeff(lok1[1],h):
> collect("Phi(0)":
> collect("D(g)(0)":
```

$$\left(\frac{b_4 a_4^3 C_0(0) a_3^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} + \frac{b_4 a_4^3 a_3^2 l_2 C_0(0)}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} \right. \\ + \frac{b_4 a_4^3 a_3^2 l_1 C_0(0)}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} + \frac{b_4 a_4^3 C_0(0) a_3^1}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} \\ - \frac{b_4 a_4^3 C_1(0) a_3^2 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1) a_2^1} + \frac{b_4 a_4^2 C_0(0)}{a_4^1 + a_4^2 + a_4^3} \\ - \frac{b_4 a_4^3 C_1(0) a_3^2 a_3^1 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1) a_2^1} + \frac{b_3 a_3^2 C_0(0)}{a_3^2 + a_3^1} \left. \right) \Phi(0) \\ + \left(-\frac{b_4 a_4^3 a_3^2 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} - \frac{b_4 a_4^2 l_2}{a_4^1 + a_4^2 + a_4^3} \right. \\ - \frac{b_4 a_4^3 a_3^2 l_2^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} + \frac{b_4 a_4^2}{a_4^1 + a_4^2 + a_4^3} \\ + \frac{b_4 a_4^3 a_3^1}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} - \frac{b_3 a_3^2 l_1}{a_3^2 + a_3^1} - \frac{b_3 a_3^2 l_2}{a_3^2 + a_3^1} - 1 \\ + \frac{b_3 a_3^2 l_1 l_2}{a_3^2 + a_3^1} + \frac{b_4 a_4^3 a_3^1 l_1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} + \frac{b_4 a_4^2 l_1 l_2}{a_4^1 + a_4^2 + a_4^3} \\ + \frac{b_4 a_4^3 a_3^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} + \frac{b_4 a_4^3 a_3^2 l_1^2 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} \\ - \frac{b_4 a_4^3 a_3^1 l_1}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} - \frac{b_4 a_4^3 a_3^1 l_2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} \\ - \frac{b_4 a_4^3 a_3^2 l_1^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} - \frac{b_4 a_4^2 l_1}{a_4^1 + a_4^2 + a_4^3} + \frac{b_3 a_3^2}{a_3^2 + a_3^1} \\ \left. + \frac{b_4 a_4^3 a_3^2 l_1 l_2^2}{(a_4^1 + a_4^2 + a_4^3)(a_3^2 + a_3^1)} \right) D(g)(0) - D(\Phi)(0)$$

Wegen $\lambda_1 \leq \lambda_2 < 0$ ist $R = 0$, woraus nach [30] folgt, daß sowohl $g(x)$ als auch $\Phi(x)$ auch keine linearen Glieder besitzen, also $\Phi(0) = 0$ und $g'(0) = \Phi'(0) = 0$, woraus das Verschwinden des h -Termes im lokalen Fehler folgt. Der lokale Fehler startet daher mit h^2 , vgl. [24].

Um die Abhängigkeit der weiteren Koeffizienten von $g(x)$ bzw. $\Phi(x)$ und deren Ableitungen bei Null zu untersuchen, dient das MAPLE Programm `testabl`. Dabei wird der entsprechende Koeffizient des lokalen Fehlers zuerst nach den möglichen Werten von $g^{(i)}(0)$ und $\Phi^{(i)}(0)$ sortiert, dann wird überprüft, für welche i , $i = 0, \dots, 5$, diese vorkommen. Das Ergebnis sind zwei Vektoren mit Wahrheitswerten, je nach dem, ob der Koeffizient von $g^{(i)}(0)$, bzw. $\Phi^{(i)}(0)$ Null ist. Bemerkenswert ist, daß MAPLE sogar zur Untersuchung von Problemen mit variabler Koeffizientenmatrix in allgemeiner Form benutzt werden kann. Dies unterstreicht die Mächtigkeit der symbolischen Software als Werkzeug zur Analyse.

Für den bereits behandelten Fall, die erste Komponente des Koeffizienten zu h in `lok1`, liefert

```
> testabl(lok1,1,1);
[ true false true true true true ]
[ false false true true true true ]
```

d.h. dieser Koeffizient enthält $g'(0)$, $\Phi(0)$ und $\Phi'(0)$, wie erwartet. Für die zweite Komponente gilt dies analog. Für die Glieder zweiter und höherer Ordnung gilt

```
> testabl(lok1,i,2);
[ false false false true true true ]
[ false true false true true true ]

> testabl(lok1,i,3);
[ false false false false true true ]
[ false true true false true true ]
```

für $i = 1, 2$. Somit hängt der k . Koeffizient des lokalen Fehlers in Potenzen von h von $\Phi(0)$, $\Phi^{(k)}(0)$ und $g(0)$, $g'(0)$, \dots , $g^{(k)}(0)$ ab.

Wir betrachten nun den Fall $\lambda_1 < \lambda_2 = 0$,

```
> y := x -> Phi(x) + c + g(x):
> f := x -> D(D(g))(x)-(11-1)*D(g)(x)/x-C1(x)*D(g)(x)-C0(x)/x*g(x):
> v := x -> vector([y(x), x*D(y)(x)]):
> F := (x,z) -> vector([z[2]/x, ((x*C0(x))*z[1]+
(11+x*C1(x))*z[2])/x+x*f(x)]):
> v0 := v(x0):
> lok2 := calclok(0,0,5):
```

In diesem Fall gilt

```
> subs(h=0,lok2[i]);
0
```

und

```
> testabl(lok2,i,1);
[ true false true true true true ]
[ false false true true true true ]
```

Unter der Zusatzbedingung $A'_0(0) \cdot y(0) = 0$ folgt nach [30], daß der lokale Fehler mit h^2 startet.

Weiters gilt

```
> testabl(lok2,i,2);
[ false false false true true true ]
[ false true false true true true ]
```

$i = 1, 2$, analog zu Fall $\lambda_1 \leq \lambda_2 < 0$.

Im dritten Fall, $\lambda_1 = \lambda_2 = 0$,

```
> y := x -> Phi(x) + c + g(x):
> f := x -> D(D(g))(x)+D(g)(x)/x-C1(x)*D(g)(x)-C0(x)/x*g(x):
> v := x -> vector([y(x), x*D(y)(x)]):
> F := (x,z) -> vector([z[2]/x,C0(x)*z[1]+C1(x)*z[2]+x*f(x)]):
> v0 := v(x0):
```

startet die Entwicklung des lokalen Fehlers

```
> lok3 := calclok(0,0,5):
```

wegen

```
> subs(h=0,lok3[i]);
0
```

und

```
> testabl(lok3,1,1);
[ false false true true true true ]
[ false false true true true true ]
```

```
> testabl(lok3,2,1);
[ false false true true true true ]
[ false false true true true true ]
```

ebenfalls mit h^2 . Im Unterschied zu den Fällen 1 und 2 stimmen hier die beiden Komponenten der lokalen Fehlerterme nicht überein,

```
> testabl(lok3,1,2);
[ true false false true true true ]
[ false true false true true true ]
```

```
> testabl(lok3,2,2);
[ false false false true true true ]
[ false true false true true true ]
```

```
> testabl(lok3,1,3);
[ true false false false true true ]
[ false true true false true true ]
```

```
> testabl(lok3,2,3);
[ false false false false true true ]
[ false true true false true true ]
```


Der lokale Fehler (2.9) eines beliebigen 4-stufigen Runge-Kutta Verfahrens hat für ein transformiertes System zweiter Ordnung mit variabler Koeffizientenmatrix $M(x)$ an der Stelle h die Form

$$\begin{aligned} e_1 = & u_2(\Phi(0), \Phi^{(2)}(0), g(0), g'(0), g^{(2)}(0))h^2 + \\ & u_3(\Phi(0), \Phi^{(3)}(0), g(0), \dots, g^{(3)}(0))h^3 + \\ & u_4(\Phi(0), \Phi^{(4)}(0), g(0), \dots, g^{(4)}(0))h^4 + O(h^5), \end{aligned} \quad (4.2)$$

wobei die u_i weiters von den Koeffizienten des Runge-Kutta Verfahrens und den Eigenwerten von $M(0)$ abhängen.

Ist nun $\Phi(x) \equiv 0$, was durch die Vorgabe von $y(0) = y'(0) = 0$ erreicht wird, so vereinfacht sich der Ausdruck für e_1 dementsprechend. Dies ist auch in den beiden Testbeispielen 2a und 2b der Fall. Der lokale Fehler hängt an der Singularität nur von $g(x)$ ab, welches wegen $g(x) = y(x)$ bekannt ist. Somit ist klar, warum wegen $y(x) = cx^4e^{-\alpha x}$ die Ordnung in Beispiel 2a nicht unter 4 fallen kann.

4.1.3 Zusammenfassung

Aus den beiden vorigen Abschnitten geht hervor, daß der Effekt der Ordnungsreduktion durch den Grad der Nullstelle $x = 0$ der Lösung verwischt wird. Es wurden zusätzlich Testbeispiele, wie in Abschnitt 3.3 getestet. Die Ordnungen verschiedener Verfahren, die sich bei $x^* = 0$ eingestellt haben, entsprechen immer der Heuristik. Aufgrund der analytischen Überlegungen betrachten wir nur lineare Anfangswertaufgaben. Im Unterschied zu Abschnitt 3.3.5 ist für diese jetzt durchaus eine Systematik der Ordnungsreduktionen zu erkennen. Eigentlich wird die Ordnung an der Singularität fast immer auf 2 reduziert, d.h. der schlechteste Fall ist zu erwarten.

Sollte $x = 0$ eine Nullstelle der Lösung mit Grad höher als zwei sein, so erhält man unter Umständen eine höhere Ordnung aufgrund der speziellen Struktur der Lösung, welches einen Sonderfall darstellt. Für lineare Beispiele mit variabler Koeffizientenmatrix ist dies dann der Fall, wenn $y(0) = y'(0) = 0$ die Lösung des homogenen Problems verschwinden läßt, und somit der Fall $y(0) = \Phi(0) + g(0) = 0$ aber $\Phi(0) \neq 0, g(0) \neq 0$ ausgeschlossen bleibt.

Zu klären bleibt noch die Frage, warum das klassische RK-Verfahren im Fall des Beispiels 2b Ordnung 3 liefert, vgl. Tabelle 3.13. In diesem Beispiel gilt $\lambda_1 = \lambda_2 = 0$, wodurch die Terme der Potenzreihenentwicklung des lokalen Fehlers einfacher werden, bedingt durch die einfachere Differentialgleichung. In diesen Fällen ist „zufällig“ der Koeffizient u_2 in der Entwicklung (4.1) bzw. (4.2) Null. Dies ist auch konsistent mit Abb. 3.13, denn dort liefern vor allem RK und 3/8R für die Beispiele mit Eigenwerten $\lambda_1 = \lambda_2 = 0$, 1a, 2a und 2b um einen Grad bessere Ordnungen, als durch die Struktur der speziellen Lösung bestimmt.

Somit ist es möglich, abhängig von λ_1, λ_2 ein bestimmtes Runge-Kutta Verfahren zu konstruieren, welches mit Hilfe der Koeffizienten b_1, \dots, b_4 und a_{21}, \dots, a_{43} für Beispiele mit den Eigenwerten λ_1 und λ_2 bessere Ordnungen erzielt. Denn wenn ein bestimmtes Verfahren bei einem Beispiel eine gewisse Ordnung zeigt, so hat es sie bei allen Beispielen mit den gleichen Eigenwerten, wobei natürlich zwischen linearen Anfangswertaufgaben mit konstanten und variablen Koeffizienten A_i unterschieden werden muß. Dies ist

jedoch nicht zielführend, da nur ein Spezialfall abgedeckt wird und für leicht geänderte Eigenwerte die Ordnung wieder auf zwei zurückfällt. Das heißt es muß in jedem Fall mit einer Ordnungsreduktion an der Singularität auf die Ordnung zwei gerechnet werden.

4.2 Ordnung des lokalen Fehlers IRK 4

Analog zu Abschnitt 4.1 sollen nun implizite Runge-Kutta Verfahren untersucht werden, deren Definition allgemein durchgeführt wird, wobei ein dreistufiges Verfahren,

```
> c1 := a11+a12+a13: c2 := a21+a22+a23: c3 := a31+a32+a33:
```

angenommen wird. Die Berechnung des lokalen Fehlers wird mit Hilfe des MAPLE Programms `calcilok` durchgeführt, siehe Abb. 4.2.

```
> calcilok := proc(sum)
> # Global F(x,z[.]) ... rechte Seite als Funktion in x und z[.].
> # Global x0 ... Startposition
> # Global v[.](x) ... exakte Loesung
> # Global v0[.] ... Startwert
> # sum die Anzahl der Folgenglieder fuer series
> local k1,k2,k3,k11,k12,k21,k22,k31,k32,k,
>     e1,e2,e3,e4,e5,e6,e,A,Ainv,b:
> k1 := vector([k11,k12]):
> k2 := vector([k21,k22]):
> k3 := vector([k31,k32]):
> e1 := k1[1]-F(x0+c1*h,evalm(v0+h*(a11*k1+a12*k2+a13*k3)))[1]:
> e2 := k1[2]-F(x0+c1*h,evalm(v0+h*(a11*k1+a12*k2+a13*k3)))[2]:
> e3 := k2[1]-F(x0+c2*h,evalm(v0+h*(a21*k1+a22*k2+a23*k3)))[1]:
> e4 := k2[2]-F(x0+c2*h,evalm(v0+h*(a21*k1+a22*k2+a23*k3)))[2]:
> e5 := k3[1]-F(x0+c3*h,evalm(v0+h*(a31*k1+a32*k2+a33*k3)))[1]:
> e6 := k3[2]-F(x0+c3*h,evalm(v0+h*(a31*k1+a32*k2+a33*k3)))[2]:
> e := vector([e1, e2, e3, e4, e5, e6]):
> A := transpose(matrix([map(coeff,map(collect,e,k11),k11),
>                             map(coeff,map(collect,e,k12),k12),
>                             map(coeff,map(collect,e,k21),k21),
>                             map(coeff,map(collect,e,k22),k22),
>                             map(coeff,map(collect,e,k31),k31),
>                             map(coeff,map(collect,e,k32),k32)])):
> b := -vector([subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[1]),
>               subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[2]),
>               subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[3]),
>               subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[4]),
>               subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[5]),
>               subs({k11=0,k12=0,k21=0,k22=0,k31=0,k32=0},e[6]) ]):
> Ainv := inverse(A):
> k := evalm(Ainv&b):
> k1 := vector([k[1],k[2]]):
> k2 := vector([k[3],k[4]]):
> k3 := vector([k[5],k[6]]):
> evalm(v0 + h*(b1*k1+b2*k2+b3*k3)):
> evalm(" - v(x0+h)):
> map(expand,map(series,"h,sum)):
> end:
```

Abbildung 4.2: MAPLE Programm zur Berechnung des lokalen Fehlers von IRK

Die höhere Komplexität impliziter Verfahren zeigt sich auch deutlich im erhöhten Programmieraufwand. Zuerst werden die Komponenten der 2-dimensionalen k_i für (2.10) definiert. Dann werden die 3 Gleichungen für k_1, k_2 und k_3 im Vektor e aufgestellt. Da nur lineare Probleme betrachtet werden, kommen die k_i nur linear vor und es kann ihre Koeffizientenmatrix A des Gleichungssystem $A \cdot k = b$, wobei $k = (k_1, k_2, k_3)^T$ ist, aufgestellt werden. Der Vektor b enthält die von k unabhängigen Beiträge.

Nach Lösung des linearen Gleichungssystems wird der Runge-Kutta Schritt durchgeführt, d.h. es wird (2.10) ausgewertet. Danach wird wie in `calclok` die exakte Lösung abgezogen und die Differenz nach Potenzen von h entwickelt, wobei die Länge der Entwicklung mittels `sum` angegeben werden kann.

Im Gegensatz zu den expliziten Verfahren ist im Fall $c_1 \neq 0$, also für Radau und Gauss Verfahren, keine Auswertung an der Stelle $x_0 = 0$ notwendig, wodurch die Berechnung von $F(x_0, y_0)$ entfällt.

4.2.1 Lineare Probleme mit konstanter Koeffizientenmatrix

Wir untersuchen wieder Anfangswertprobleme zweiter Ordnung mit konstanter Koeffizientenmatrix M , wobei je nach deren Eigenwerten unterschieden werden muß. In Fall 1, $\lambda_1 \leq \lambda_2 < 0$, liefert

```
> lok1 := calclok(5):
```

eine entsprechende Entwicklung des lokalen Fehlers mit

```
> subs(h=0,lok1[i]);
```

0

```
> simplify(collect(coeff(lok1[i],h),D(g)(0)));
      (b1 + b2 + b3 - 1) D(g)(0)
```

für $i = 1, 2$.

Im Vergleich mit den expliziten Verfahren fällt auf, daß diese Koeffizienten sowohl wegen $D(g)(0) = y'(0) = 0$ als auch wegen der Ordnungsbedingung $b_1 + b_2 + b_3 = 1$, welche für Ordnung 2 notwendig ist, verschwinden. Das heißt, daß Ordnung 2 in diesem Fall sowohl für singuläre Anfangswertaufgaben erster als auch zweiter Ordnung gesichert ist.

Weiters gilt, wie auch bei den expliziten Verfahren,

```
> coeff(lok1[i],h^k):
      u(b1, ..., b3, a21, ..., a33) D^(k)(g)(0)
```

etc., für $i = 1, 2$, und $k \geq 1$.

Analog verhält sich der lokale Fehler im zweiten und dritten Fall $\lambda_1 < \lambda_2 = 0$, bzw. $\lambda_1 = \lambda_2 = 0$,

```
> lok3 := calclok(5):
```

```
> subs(h=0,lok3[i]);
```

0

```
> simplify(collect(coeff(lok3[i],h),D(g)(0)));
      (b1 + b2 + b3 - 1) D(g)(0)
```

für $i = 1, 2$. Im Fall konstanter Koeffizienten A_0 und A_1 hat die Entwicklung des lokalen Fehlers nach Potenzen von h also ebenfalls die Form (4.1).

Die weiteren Koeffizienten zu h^k , $k = 2, \dots$, in `lok3` haben die gewohnte Form.

Zu beachten ist, daß der Koeffizient von `lok3[2]` zu h^2 wegen

```
> collect(coeff(lok3[2],h^2),D(D(g))(0));
```

$$\begin{aligned} & (-1 + 2 b_1 a_{11} + 2 b_1 a_{12} + 2 b_1 a_{13} + 2 b_2 a_{21} + 2 b_2 a_{22} + 2 b_2 a_{23} \\ & + 2 b_3 a_{31} + 2 b_3 a_{32} + 2 b_3 a_{33}) D^{(2)}(g)(0) \end{aligned}$$

ebenfalls verschwindet, da $b_1 c_1 + b_2 c_2 + b_3 c_3 = 1/2$ eine weitere notwendige Bedingung für Ordnung 2 darstellt, [18], Seite 202, Formel (7.9). Ein analoges Ergebnis konnte aber aufgrund der Komplexität der Koeffizienten und der Ordnungsbedingungen nicht für sämtliche Koeffizienten zu h^2 aller drei Fälle nachgewiesen werden.

4.2.2 Zusammenfassung

Wie auch für explizite Verfahren, vgl. Abschnitt 4.1.3, hat der Grad der Nullstelle $x = 0$ direkten Einfluß auf die Konvergenzordnung des lokalen Fehlers linearer Probleme, vgl. dazu auch Tabelle 3.14, Beispiel 1a. Prinzipiell ist mit einer Ordnungsreduktion auf 2 zu rechnen. Diese theoretische Aussage wird allerdings nicht durch die Überprüfung anderer Testbeispiele bestätigt. Die Ordnungen der Radau (RIIA3 und RIIA5) sowie Gauss Verfahren (G4, G6) bei $x^* = 0$, zeigen in den meisten Fällen mindestens Ordnung 3. Dies legt die Vermutung nahe, daß im allgemeinen nur mit einer Ordnungsreduktion auf 3 zu rechnen ist. Die impliziten Verfahren scheinen daher gegenüber der Ordnungsreduktion an der Singularität robuster zu sein als die expliziten Verfahren.

5 Schätzung des globalen Fehlers

Beim Lösen von Differentialgleichungen werden Genauigkeitsanforderungen meist an den globalen Fehler gestellt. Bei der Schrittweitensteuerung über den lokalen Fehler, vgl. Abschnitt 3, wurde vorausgesetzt, daß der lokale und der globale Fehler korreliert sind. Dies ist zwar im allgemeinen der Fall, muß aber nicht immer so sein. Eine sichere Methode ist, den globalen Fehler zu schätzen, wofür bereits eine Vielzahl von Methoden existiert, vgl. [39]. Das Ziel dabei ist es, eine möglichst gute Schätzung δ_j von ε_j zu erhalten. Aufgrund solcher Schätzungen kann die Qualität der Lösung bezüglich der vorgegebenen Toleranz verlässlich bestimmt werden. Die meisten Methoden basieren auf zweimaliger Integration der Anfangswertaufgabe, ein beträchtlich größerer Aufwand. Weiters gibt es keine einfache Möglichkeit die Schrittweite während der Integration an den globalen Fehler anzupassen, vgl. [38].

5.1 Der globale Fehler der Verfahren

Nachdem bei den singulären Anfangswertaufgaben der lokale Fehler unglatt ist, scheint es sinnvoll, vor der Anwendung verschiedener Methoden zur Schätzung des globalen Fehlers die Struktur des globalen Fehlers auszuloten. Teilweise wurde dies schon in den Abschnitten 3.1 und 3.2 durchgeführt, obwohl dort das Hauptaugenmerk auf dem lokalen Fehler lag. Dort lag der globale Fehler etwa immer zwischen $1/10$ und dem 10 fachen der lokalen Toleranz, je nach Verfahren und Testbeispiel.

5.1.1 Explizites Runge-Kutta 4

Wir schätzen die Ordnung des globalen Fehlers genau wie die des lokalen Fehlers in Abschnitt 3.3. Wir starten bei $x = 0$ und rechnen mit der Schrittweite h_1 bis 1 , wobei $h_1 = 1/n_1$, $n_1 \in \mathbb{N}$ gelten muß. Das Verfahren benötigt also genau n_1 Schritte. Der globale Fehler bei 1 ist $\varepsilon_{n_1} = y_{n_1} - y(1)$. Analog erhalten wir für h_2 den globalen Fehler $\varepsilon_{n_2} = y_{n_2} - y(1)$. Die Ordnung wird analog zu (3.5) mit

$$p \approx \frac{\ln\left(\frac{|\varepsilon_{n_1}|}{|\varepsilon_{n_2}|}\right)}{\ln\left(\frac{|h_1|}{|h_2|}\right)} \quad (5.1)$$

berechnet, bzw. die Fehlerkonstante wie in (3.6) mit

$$c \approx \frac{\varepsilon_{n_1}}{h_1^p}. \quad (5.2)$$

Dieser Zugang liefert für jede Komponente der Lösung eine Konvergenzordnung und Fehlerkonstante.

Anschließend sind die Konvergenzordnungen p der ersten Komponente einiger Beispiele aufgelistet. Es wird mit $h_1 = 1/5$ begonnen. Weitere Schrittweiten ergeben sich durch sukzessives Halbieren von h_1 . Dabei ist $\varepsilon = \varepsilon_{n_i}(h_i)$, $i = 1, \dots, 9$, der globale Fehler der ersten Komponente und c die Fehlerkonstante. Hat diese annähernd gleiche Werte für verschiedene h_i , so bestätigt dies die Korrektheit der Schätzung von p . Verwendet wird

das klassische Runge-Kutta Verfahren mit dem Butcher-Array aus Abschnitt 3.3.5. Die erwartete Ordnung ist daher 4. Die Tabellen zeigen nur die h -Bereiche, in denen sich Ordnungen einstellen.

h	ε	p	c
1/5	$9.091 \cdot 10^{-01}$	5.871129	$-1.154 \cdot 10^{+03}$
$1/5 \cdot 2^{-1}$	$1.553 \cdot 10^{-03}$	5.113945	$-2.019 \cdot 10^{+02}$
$1/5 \cdot 2^{-2}$	$4.485 \cdot 10^{-04}$	4.782977	$7.491 \cdot 10^{+02}$
$1/5 \cdot 2^{-3}$	$1.629 \cdot 10^{-06}$	0.640454	$-1.729 \cdot 10^{-05}$
$1/5 \cdot 2^{-4}$	$1.045 \cdot 10^{-06}$	2.749678	$-1.786 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$1.553 \cdot 10^{-07}$	3.213922	$-1.884 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	$1.674 \cdot 10^{-08}$	3.436347	$-6.799 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	$1.547 \cdot 10^{-09}$	3.569052	$-1.602 \cdot 10^{+01}$
$1/5 \cdot 2^{-8}$	$1.303 \cdot 10^{-10}$	3.670260	$-3.306 \cdot 10^{+02}$
$1/5 \cdot 2^{-9}$	$1.023 \cdot 10^{-11}$	3.117248	$-4.311 \cdot 10^{+00}$

Tabelle 5.1: Ordnung globaler Fehler ERK 4 für Beispiel 1a an $x = 1$

h	ε	p	c
1/5	$1.184 \cdot 10^{-04}$	4.015576	$7.590 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$7.323 \cdot 10^{-05}$	4.003873	$7.388 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$4.564 \cdot 10^{-06}$	4.000967	$7.324 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$2.851 \cdot 10^{-08}$	4.000239	$7.305 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$1.781 \cdot 10^{-09}$	3.999959	$7.296 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$1.113 \cdot 10^{-10}$	4.002002	$7.372 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$6.949 \cdot 10^{-11}$	4.105621	$1.340 \cdot 10^{-01}$
$1/5 \cdot 2^{-7}$	$4.037 \cdot 10^{-12}$	2.689650	$1.424 \cdot 10^{-05}$

Tabelle 5.2: Ordnung globaler Fehler ERK 4 für Beispiel 1b an $x = 1$

Die Tabelle 5.4 zeigt für das Testbeispiel 2b eine klare Ordnungsreduktionen von 4 auf $p = 3$. An der Singularität hatte dieses Beispiel Ordnungsreduktionen um 2 h -Potenzen, vgl. Tabelle 3.13. Diese zum Teil größtmöglichen Ordnungsreduktionen wirken sich allerdings nur auf eine h -Potenz im globalen Fehler aus. Die Fehlerkonstanten zwischen 0.1 und 100 bestätigen die Gültigkeit der gemessenen Ordnungen.

Für die Beispiele 1a, 2a und 3a zeigen die Tabellen 5.1, 5.3 und 5.5 Ordnungen zwischen $p = 3.5$ und $p = 3.8$. Die Fehlerkonstanten stabilisieren sich zwar nicht, sind aber auch nicht allzu groß. Diese Beispiele, vgl. Tabelle 3.13, zeigten für das klassische Runge-Kutta Verfahren (RK) nur mäßige bis gar keine Ordnungsreduktionen im lokalen Fehler.

In der Tabelle 5.2 beobachtet man, wie sich für 1b die klassischen Konvergenzordnungen einstellen. Die Fehlerkonstanten sind etwa in der Größenordnung von 10, was die Richtigkeit der Ordnungen bestätigt.

Außer im Testbeispiel 2b ist ε nach der 8. bis 9. Gitterverfeinerung durchwegs im Rundungsfehlerbereich, und es stellen sich keine Ordnungen mehr ein.

h	ε	p	c
$1/5$	$5.028 \cdot 10^{-03}$	3.478542	$-1.357 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$4.511 \cdot 10^{-04}$	3.161546	$-6.543 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$5.041 \cdot 10^{-05}$	3.270460	$-9.068 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$5.224 \cdot 10^{-06}$	3.421611	$-1.583 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$4.875 \cdot 10^{-07}$	3.540578	$-2.667 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$4.190 \cdot 10^{-08}$	3.628036	$-4.157 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	$3.389 \cdot 10^{-09}$	3.693043	$-6.049 \cdot 10^{+00}$
$1/5 \cdot 2^{-7}$	$2.620 \cdot 10^{-11}$	3.715743	$-7.005 \cdot 10^{+00}$
$1/5 \cdot 2^{-8}$	$1.994 \cdot 10^{-12}$	3.315877	$-4.008 \cdot 10^{-01}$

Tabelle 5.3: Ordnung globaler Fehler ERK 4 für Beispiel 2a an $x = 1$

h	ε	p	c
$1/5$	$3.466 \cdot 10^{-01}$	2.485072	$1.891 \cdot 10^{+00}$
$1/5 \cdot 2^{-1}$	$6.191 \cdot 10^{-02}$	2.650220	$2.766 \cdot 10^{+00}$
$1/5 \cdot 2^{-2}$	$9.862 \cdot 10^{-03}$	2.763838	$3.888 \cdot 10^{+01}$
$1/5 \cdot 2^{-3}$	$1.452 \cdot 10^{-04}$	2.842075	$5.189 \cdot 10^{+01}$
$1/5 \cdot 2^{-4}$	$2.025 \cdot 10^{-05}$	2.895807	$6.567 \cdot 10^{+01}$
$1/5 \cdot 2^{-5}$	$2.720 \cdot 10^{-06}$	2.932331	$7.905 \cdot 10^{+01}$
$1/5 \cdot 2^{-6}$	$3.564 \cdot 10^{-06}$	2.956777	$9.102 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	$4.591 \cdot 10^{-07}$	2.972909	$1.010 \cdot 10^{+01}$
$1/5 \cdot 2^{-8}$	$5.847 \cdot 10^{-08}$	2.983876	$1.092 \cdot 10^{+01}$
$1/5 \cdot 2^{-9}$	$7.391 \cdot 10^{-09}$	2.975906	$1.026 \cdot 10^{+01}$

Tabelle 5.4: Ordnung globaler Fehler ERK 4 für Beispiel 2b an $x = 1$

h	ε	p	c
$1/5$	$1.863 \cdot 10^{-04}$	3.686882	$-7.034 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$1.446 \cdot 10^{-05}$	3.799305	$-9.112 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$1.039 \cdot 10^{-06}$	3.835148	$-1.014 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$7.280 \cdot 10^{-07}$	3.852488	$-1.081 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$5.040 \cdot 10^{-08}$	3.864811	$-1.141 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$3.459 \cdot 10^{-09}$	3.875116	$-1.202 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$2.357 \cdot 10^{-11}$	3.882585	$-1.255 \cdot 10^{-01}$
$1/5 \cdot 2^{-7}$	$1.598 \cdot 10^{-12}$	3.972802	$-2.249 \cdot 10^{-01}$
$1/5 \cdot 2^{-8}$	$1.018 \cdot 10^{-13}$	6.840777	$-1.834 \cdot 10^{+08}$

Tabelle 5.5: Ordnung globaler Fehler ERK 4 für Beispiel 3a an $x = 1$

5.1.2 Implizites Runge-Kutta 4

Analog zum vorigen Abschnitt wird der globale Fehler ε , die Konvergenzordnung p und die Fehlerkonstante c der ersten Komponente für ein implizites Runge-Kutta Verfahren tabelliert. Verwendet wird die Gauss Formel 4. Ordnung aus [19], Seite 72, Abb. 5.1. Die erwartete Ordnung ist daher 4.

h	ε	p	c
1/5	$9.123 \cdot 10^{-01}$	3.212604	$1.605 \cdot 10^{+01}$
$1/5 \cdot 2^{-1}$	$9.841 \cdot 10^{-02}$	3.272877	$1.844 \cdot 10^{+01}$
$1/5 \cdot 2^{-2}$	$1.018 \cdot 10^{-03}$	3.452993	$3.164 \cdot 10^{+02}$
$1/5 \cdot 2^{-3}$	$9.298 \cdot 10^{-04}$	3.604613	$5.536 \cdot 10^{+02}$
$1/5 \cdot 2^{-4}$	$7.643 \cdot 10^{-05}$	3.699659	$8.396 \cdot 10^{+02}$
$1/5 \cdot 2^{-5}$	$5.882 \cdot 10^{-06}$	3.758463	$1.131 \cdot 10^{+02}$
$1/5 \cdot 2^{-6}$	$4.346 \cdot 10^{-07}$	3.796954	$1.412 \cdot 10^{+02}$
$1/5 \cdot 2^{-7}$	$3.127 \cdot 10^{-09}$	3.823271	$1.674 \cdot 10^{+02}$
$1/5 \cdot 2^{-8}$	$2.209 \cdot 10^{-10}$	3.832845	$1.793 \cdot 10^{+02}$

Tabelle 5.6: Ordnung globaler Fehler IRK 4 für Beispiel 1a an $x = 1$

h	ε	p	c
1/5	$7.903 \cdot 10^{-04}$	4.016608	$-5.073 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$4.883 \cdot 10^{-05}$	4.004131	$-4.929 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$3.043 \cdot 10^{-07}$	4.001031	$-4.884 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$1.900 \cdot 10^{-08}$	4.000260	$-4.870 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$1.187 \cdot 10^{-09}$	4.000217	$-4.869 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$7.422 \cdot 10^{-10}$	3.998208	$-4.820 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$4.644 \cdot 10^{-11}$	3.874164	$-2.356 \cdot 10^{-02}$
$1/5 \cdot 2^{-7}$	$3.167 \cdot 10^{-12}$	4.235097	$-2.427 \cdot 10^{-01}$
$1/5 \cdot 2^{-8}$	$1.681 \cdot 10^{-14}$	-2.848777	$2.366 \cdot 10^{-23}$

Tabelle 5.7: Ordnung globaler Fehler IRK 4 für Beispiel 1b an $x = 1$

Wie im Fall des expliziten Verfahrens zeigt die Tabelle 5.9 von Testbeispiel 2b eine klare Ordnungsreduktion von 4 auf $p = 3$.

In den Tabellen 5.6 und 5.8 der Beispiele 1a und 2a sind Ordnungen um $p = 3.8$ zu erkennen. Die Fehlerkonstanten liegen zwischen 1 und 100, stabilisieren sich jedoch nicht.

In den Tabellen 5.7 und 5.10 stellt sich für die Beispiele 1b und 3a die klassische Konvergenzordnung ein.

h	ε	p	c
1/5	$2.829 \cdot 10^{-03}$	3.515338	$8.105 \cdot 10^{+00}$
$1/5 \cdot 2^{-1}$	$2.474 \cdot 10^{-04}$	3.650708	$1.107 \cdot 10^{+00}$
$1/5 \cdot 2^{-2}$	$1.970 \cdot 10^{-05}$	3.730416	$1.405 \cdot 10^{+00}$
$1/5 \cdot 2^{-3}$	$1.484 \cdot 10^{-06}$	3.779881	$1.686 \cdot 10^{+00}$
$1/5 \cdot 2^{-4}$	$1.080 \cdot 10^{-07}$	3.812698	$1.947 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	$7.689 \cdot 10^{-08}$	3.836005	$2.192 \cdot 10^{+00}$
$1/5 \cdot 2^{-6}$	$5.384 \cdot 10^{-09}$	3.853102	$2.419 \cdot 10^{+00}$
$1/5 \cdot 2^{-7}$	$3.726 \cdot 10^{-10}$	3.886507	$3.002 \cdot 10^{+00}$
$1/5 \cdot 2^{-8}$	$2.519 \cdot 10^{-12}$	4.424200	$1.406 \cdot 10^{+02}$

Tabelle 5.8: Ordnung globaler Fehler IRK 4 für Beispiel 2a an $x = 1$

h	ε	p	c
1/5	$4.427 \cdot 10^{-01}$	3.054198	$6.038 \cdot 10^{+01}$
$1/5 \cdot 2^{-1}$	$5.330 \cdot 10^{-02}$	3.019657	$5.576 \cdot 10^{+01}$
$1/5 \cdot 2^{-2}$	$6.572 \cdot 10^{-03}$	3.004025	$5.321 \cdot 10^{+01}$
$1/5 \cdot 2^{-3}$	$8.192 \cdot 10^{-04}$	2.998745	$5.219 \cdot 10^{+01}$
$1/5 \cdot 2^{-4}$	$1.024 \cdot 10^{-05}$	2.997661	$5.194 \cdot 10^{+01}$
$1/5 \cdot 2^{-5}$	$1.283 \cdot 10^{-06}$	2.997955	$5.202 \cdot 10^{+01}$
$1/5 \cdot 2^{-6}$	$1.606 \cdot 10^{-07}$	2.998529	$5.219 \cdot 10^{+01}$
$1/5 \cdot 2^{-7}$	$2.010 \cdot 10^{-08}$	2.999176	$5.241 \cdot 10^{+01}$
$1/5 \cdot 2^{-8}$	$2.513 \cdot 10^{-09}$	3.000964	$5.308 \cdot 10^{+01}$

Tabelle 5.9: Ordnung globaler Fehler IRK 4 für Beispiel 2b an $x = 1$

h	ε	p	c
1/5	$4.929 \cdot 10^{-05}$	3.999027	$-3.076 \cdot 10^{-03}$
$1/5 \cdot 2^{-1}$	$3.083 \cdot 10^{-07}$	3.999760	$-3.081 \cdot 10^{-03}$
$1/5 \cdot 2^{-2}$	$1.927 \cdot 10^{-08}$	3.999950	$-3.083 \cdot 10^{-03}$
$1/5 \cdot 2^{-3}$	$1.204 \cdot 10^{-09}$	3.999993	$-3.083 \cdot 10^{-03}$
$1/5 \cdot 2^{-4}$	$7.528 \cdot 10^{-10}$	4.000000	$-3.083 \cdot 10^{-03}$
$1/5 \cdot 2^{-5}$	$4.705 \cdot 10^{-11}$	3.993479	$-2.983 \cdot 10^{-03}$
$1/5 \cdot 2^{-6}$	$2.954 \cdot 10^{-13}$	3.830858	$-1.167 \cdot 10^{-03}$
$1/5 \cdot 2^{-7}$	$2.076 \cdot 10^{-14}$	2.023332	$-9.887 \cdot 10^{-08}$
$1/5 \cdot 2^{-8}$	$5.107 \cdot 10^{-14}$	-0.309328	$5.584 \cdot 10^{-15}$

Tabelle 5.10: Ordnung globaler Fehler IRK 4 für Beispiel 3a an $x = 1$

5.1.3 Zusammenfassung

In Tabelle 5.11 sind die in weiteren Tests erhaltenen Konvergenzordnungen der Testbeispiele für weitere Verfahren tabelliert.

Verfahren	D3	RK	3/8R	DP5	DP8	RIIA3	G4	RIIA5
klass. Ord.	3	4	4	5	8	3	4	5
Beispiel 1a	3	3.5	3.5	4	5	3	3.8	4
Beispiel 1b	4	4	4	5	?	3	4	5
Beispiel 1c	3	4	4	4	4	3	4	5
Beispiel 2a	3	3.6	3.5	4	5	3	3.8	4
Beispiel 2b	2	3	3	3	4	2.8	3	4
Beispiel 3a	3	3.8	3.9	4	4	3	4	5

Tabelle 5.11: Ordnungen globaler Fehler RK aller Beispiele an $x = 1$

Der globale Fehler ist im Gegensatz zum lokalen Fehler glatt. Für kleinere Schrittweiten reduziert er sich dementsprechend, bis er im Rundungsfehlerbereich liegt, vgl. auch Beschreibung des globalen Fehlers in Abschnitt 3.1.6. Verfahren mit Ordnung $p \leq 4$ zeigen teilweise die erwartete Ordnung p , teilweise eine bis um Eins reduzierte Ordnung. Die Ordnungsreduktionen um 3 h -Potenzen an der Singularität wirken sich global auf maximal eine h -Potenz aus.

Für Verfahren mit hoher Genauigkeit, z.B. DP8, wirken sich die Ordnungsreduktionen im lokalen Fehler stärker auf den globalen Fehler aus. Für DP8 können theoretisch lokale Ordnungsreduktionen bis zu 7 h -Potenzen auftreten, vgl. Abschnitt 4.1, und es treten auch tatsächlich 5 h -Potenzen in Tabelle 3.13 auf, was eine dramatische Ordnungsreduktion darstellt. Die impliziten Verfahren scheinen jedoch etwas robuster zu sein, d.h. weniger Ordnungspotenzen zu verlieren, als die expliziten.

5.2 Zadunaisky-Schätzung

Zadunaisky beschreibt in [44] eine Möglichkeit den globalen Fehler ε zu schätzen. Er geht dabei folgendermaßen vor: Nachdem das Problem (2.7),

$$\begin{aligned} y'(x) &= F(x, y(x)), & a \leq x \leq b, \\ y(a) &= b_0, \end{aligned}$$

mit y_Δ numerisch gelöst wurde, konstruiert man ein weiteres Anfangswertproblem erster Ordnung, das sogenannte Nachbarproblem,

$$z'(x) = \bar{F}(x, z(x)), \quad a \leq x \leq b, \quad (5.3a)$$

$$z(a) = \bar{b}_0, \quad (5.3b)$$

derart, daß man die exakte Lösung $z(x)$ bereits kennt, und das Problem sich nur wenig vom ursprünglichen Problem (2.7) unterscheidet. Das Nachbarproblem wird genauso gelöst wie das eigentliche Problem, und man erhält die numerische Lösung z_Δ mit den

globalen Fehlern $\bar{\varepsilon}_j = z_j - z(x_j)$, $j = 1, \dots, n$, Falls die numerische Lösung z_j des Nachbarproblems nahe der numerischen Lösung y_j des eigentlichen Problems liegt, so kann man erwarten, daß sich die globalen Fehler gleich verhalten, also daß

$$\varepsilon_j \approx \delta_j := \bar{\varepsilon}_j \quad (5.4)$$

ist.

Eine Möglichkeit zur Konstruktion des Nachbarproblems besteht darin, die numerische Lösung y_Δ durch ein vektorwertiges Polynom $p(x)$ zu interpolieren, $z(x) = p(x)$. Dann hat das Nachbarproblem die Form

$$z'(x) = F(x, z(x)) + D(x), \quad a \leq x \leq b, \quad (5.5a)$$

$$z(a) = p(a) = b_0, \quad (5.5b)$$

mit $D(x) = p'(x) - F(x, p(x))$. Um hohe Polynomgrade zu vermeiden, setzt man $p(x)$ stückweise aus mehreren Polynomen vom Grad k zusammen, wobei k größer als die Ordnung p des verwendeten Verfahrens sein sollte, meist $k := p+1$. Somit ist $p \in \mathbf{C}[a, b]$, während $p'(x)$ an den Stützstellen x_{kj} , $j = 1, \dots$, Sprungstellen besitzt, was jedoch bei Systemen erster Ordnung keine Probleme bereitet. Die Schätzung (5.4) vereinfacht sich in diesem Fall zu

$$\varepsilon_j \approx \delta_j := \bar{\varepsilon}_j = z_j - y_j. \quad (5.6)$$

Für singuläre Anfangswertaufgaben stellt sich aufgrund des $1/x$ im Differentialoperator sofort die Frage, ob das Nachbarproblem (5.5) überhaupt sinnvoll definiert ist. Das Nachbarproblem der Anfangswertaufgabe mit konstanter Koeffizientenmatrix ist

$$z'(x) - \frac{M}{x} \cdot z(x) = p'(x) - \frac{M}{x} \cdot p(x), \quad 0 < x \leq 1,$$

$$B_0 \cdot z(0) = \beta,$$

$$M \cdot z(0) = 0.$$

Aufgrund von $z(0) = p(0) = y(0)$ gilt $M \cdot p(0) = 0$, woraus $M \cdot p(x) = xq(x)$, $q \in [x]$, und damit $1/xM \cdot p(x) = q(x)$ folgt. Das Nachbarproblem entspricht also dem homogenen ursprünglichen Problem mit einer polynomialen Inhomogenität.

Für lineare Anfangswertaufgaben mit variabler Koeffizientenmatrix hat die benachbarte Differentialgleichung die Form

$$z'(x) - \frac{M(x)}{x} \cdot z(x) = p'(x) - \frac{M(x)}{x} \cdot p(x), \quad 0 < x \leq 1.$$

Es gilt $M(x) \cdot p(x) = M(0) \cdot p(x) + xC(x) \cdot p(x)$. Wegen $M(0) \cdot z(0) = 0$ folgt daraus $(1/x)M(x) \cdot p(x) = q(x) + C(x) \cdot p(x)$. Das Nachbarproblem entspricht somit ebenfalls dem homogenen ursprünglichen Problem mit einer polynomialen Inhomogenität. Analoges gilt für das nichtlineare Problem (2.1), da die zusätzlichen nichtlinearen Beiträge $f(x, y(x))$ und $f(x, p(x))$ keine Probleme bereiten.

Diese Methode wird auch als Defektkorrektur Version A bezeichnet, vgl. [39], denn $D(x)$ aus (5.5a) ist der Defekt der Lösung y_Δ bezüglich eines Kollokationsverfahrens

zum Gitter Δ . Formel (5.6) liefert eine mit korrektem Vorzeichen versehene Schätzung des globalen Fehlers. Damit erhält man aus

$$y_j^{(1)} := y_j - \bar{\varepsilon}_j$$

eine verbesserte Näherung für $y(x_j)$ bezüglich des Kollokationsverfahrens. Mit der neuen Näherung $y_{\Delta}^{(1)}$ wird ein neuer Defekt $D^{(1)}(x)$ gebildet, und man erhält eine neue Fehlerschätzung. Iteriert man diesen Vorgang, so erhält man mit der Methode der Iterierten Defekt-Korrektur (IDeC), vgl. [9], immer bessere Näherungslösungen

$$y_j^{(i+1)} := y_j^{(i)} - \bar{\varepsilon}_j^{(i)} = y_j^{(i)} - (z_j^{(i)} - y_j). \quad (5.7)$$

Der Grenzwert $y_{\Delta}^{(*)}$ ist mit $D^{(*)}(x) \equiv 0$ die exakte Lösung des Kollokationsverfahrens, welche in diesem Fall das Zielverfahren darstellt. Das Verhalten der Iterierten Defekt-Korrektur für singuläre Probleme wurde bereits in [1] und [11] untersucht.

5.2.1 Basismethode explizites Runge-Kutta 4

In diesem Abschnitt überprüfen wir die Güte der Schätzung (5.6) des globalen Fehlers. Dazu werden die Testprobleme für verschiedene Schrittweiten h gelöst und der globale Fehler ε_j wird mit dem Wert des Zadunaisky-Schätzers δ_j verglichen. Die erste Abbildung zeigt jeweils die Maximumnorm des Fehlers des Zadunaisky-Schätzers für verschiedene Schrittweiten, also $|\delta_j - \varepsilon_j|$. Die weiteren Abbildungen zeigen die Maximumnorm des globalen Fehlers, $|\varepsilon_j|$, $0 \leq j \leq n$, (dicke Linie) und den Wert des Schätzers, $|\delta_j|$, $0 \leq j \leq n$, (punktierete Linie). Alle Graphen sind logarithmisch skaliert.

Verwendet wurde das Verfahren DP4, also die Runge-Kutta Methode von DOPRI5 mit der Ordnung $p = 4$. Daher wurde $k = 5$ gewählt.⁶

Zadunaisky-Schätzung für ERK 4 für Beispiel 1a: Der Fehler des Schätzers verläuft in Abb. 5.1 beinahe völlig geradlinig. Das heißt der globale Fehler wird entlang des Intervalls gleichmäßig gut approximiert. Mit kleiner werdender Schrittweite, $h_2 := h_1/2$, verringert sich der Fehler des Schätzers um je 1.5 Zehnerpotenzen, also grob um $1/2^p$.

Ein detaillierteres Bild geben die Abb. 5.2 bis 5.6. Für $h = 1/5 \cdot 2^{-3}$ liegt der Schätzer eine halbe Zehnerpotenz unter dem globalen Fehler, Abb. 5.2. Mit kleiner werdender Schrittweite gibt er das qualitative Verhalten und die Größenordnung des Fehlers immer besser wieder, Abb. 5.3 bis 5.5. Für eine relativ kleine Schrittweite, $h = 1/5 \cdot 2^{-7} \approx 0.0015$, folgt der Schätzer exakt dem Verlauf des globalen Fehlers.

⁶Experimente mit $p = 4$ und $k = 3$ haben zwar für reguläre Testbeispiele und die Anfangswertaufgaben 1b, 2b und 3a durchaus brauchbare Ergebnisse geliefert, nicht jedoch für 1a und 2a. In diesen Fällen liegt der Schätzer einige Zehnerpotenzen unter dem globalen Fehler und auch bei feiner werdendem Gitter nähert sich der Schätzer dem Fehler **nicht**.

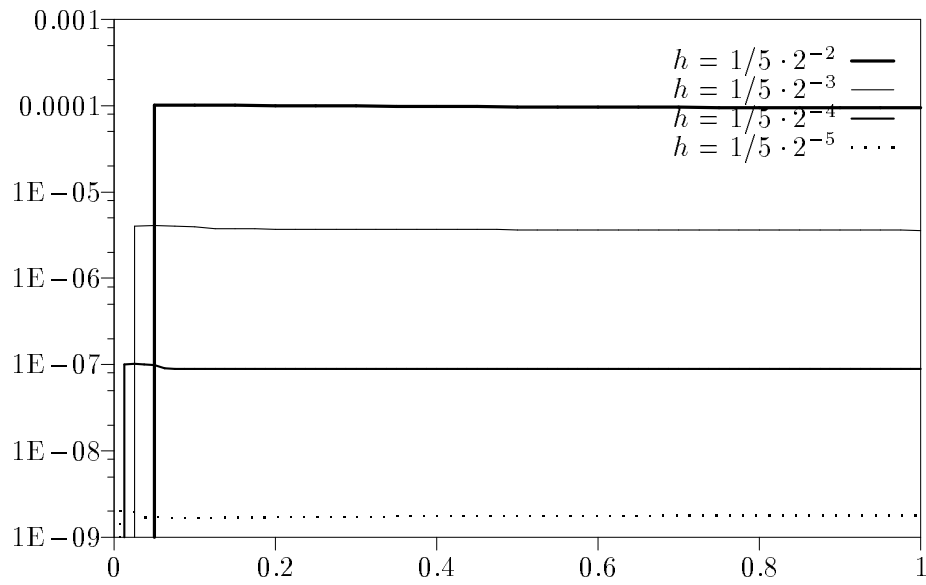
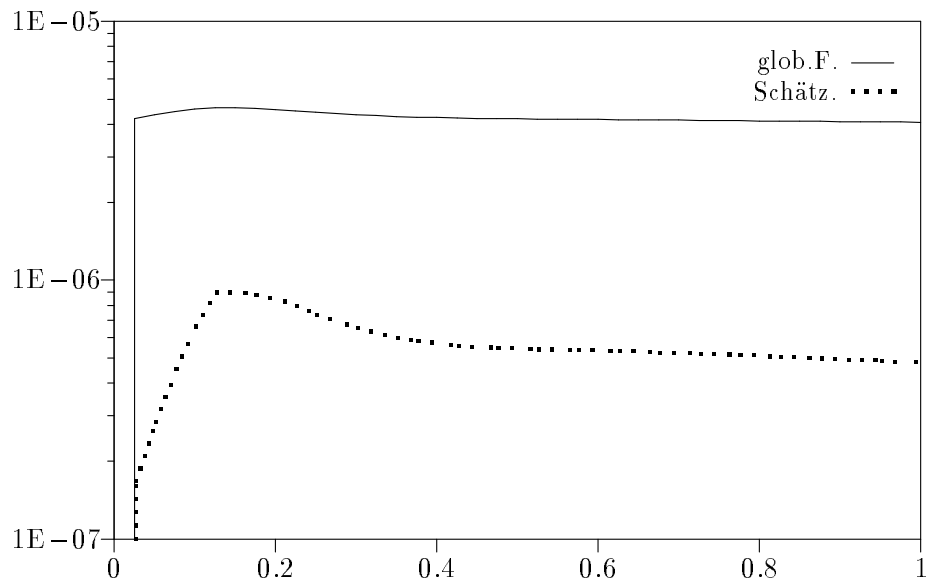
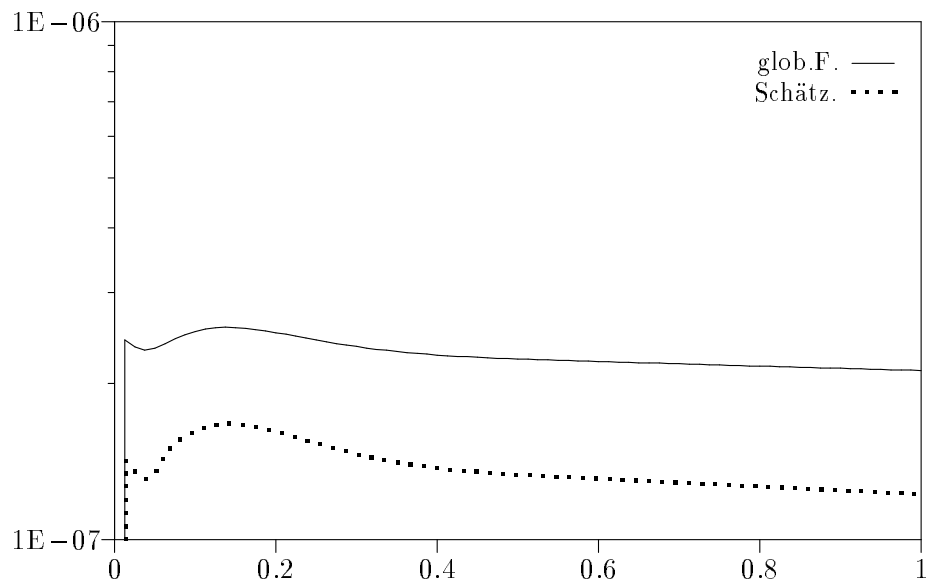
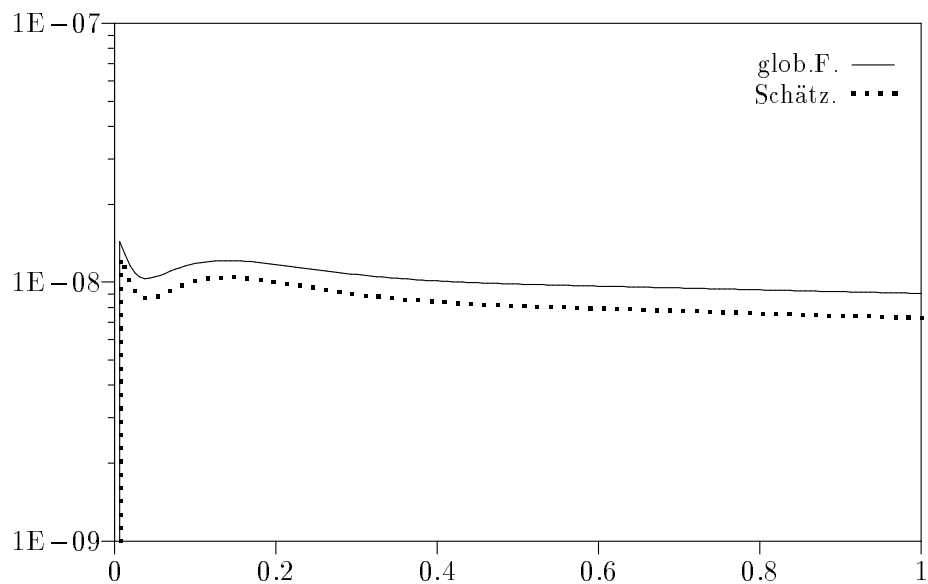
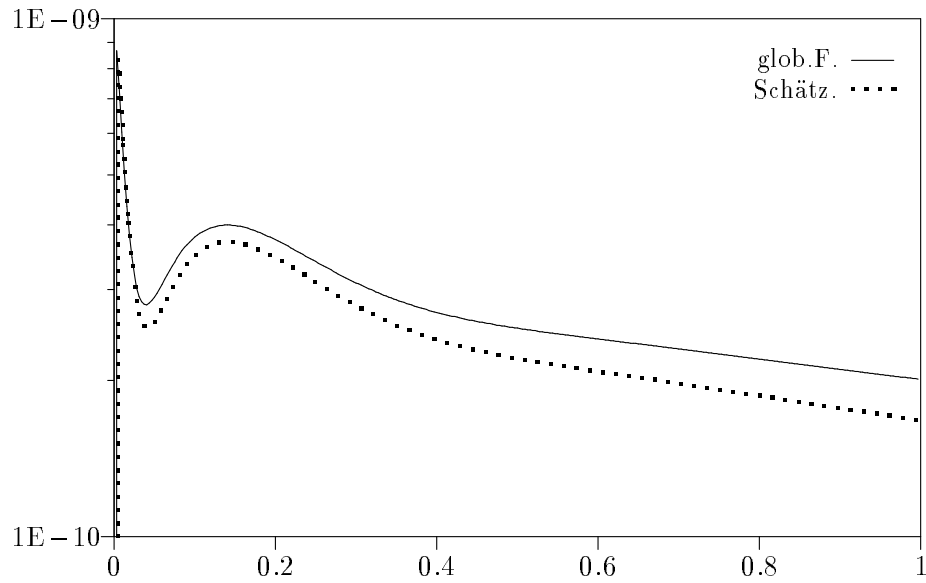
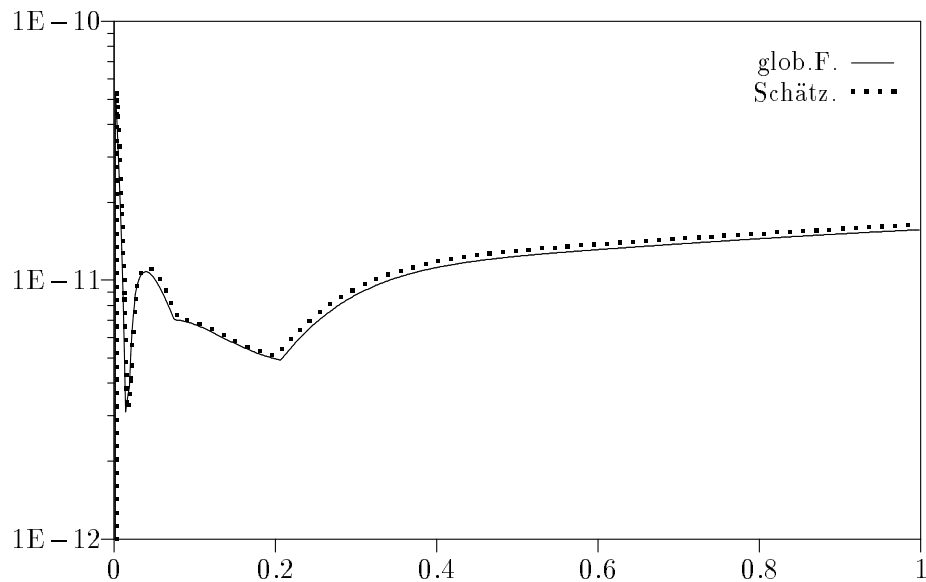


Abbildung 5.1: Fehler Zadunaisky für ERK 4 in Beispiel 1a

Abbildung 5.2: Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-3}$

Abbildung 5.3: Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-4}$ Abbildung 5.4: Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-5}$

Abbildung 5.5: Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-6}$ Abbildung 5.6: Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-7}$

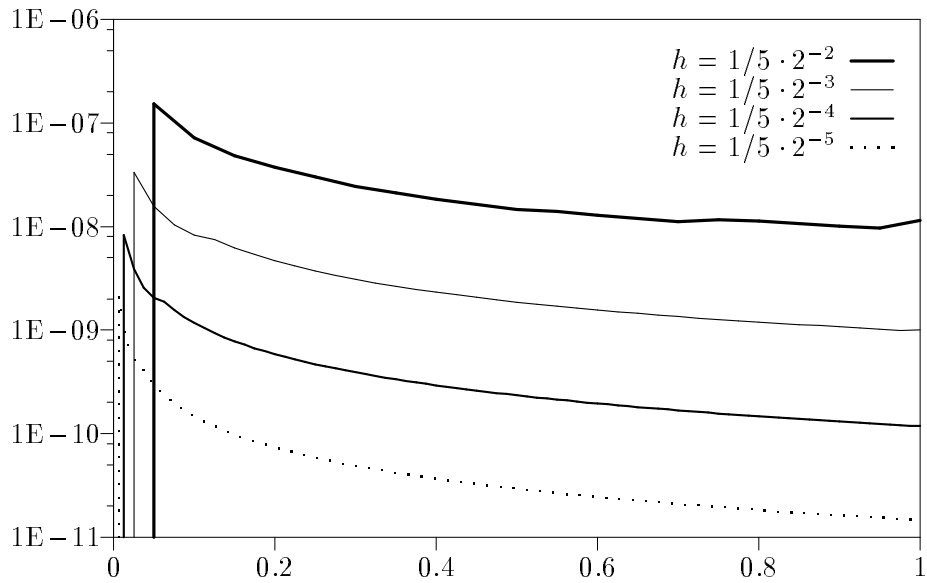
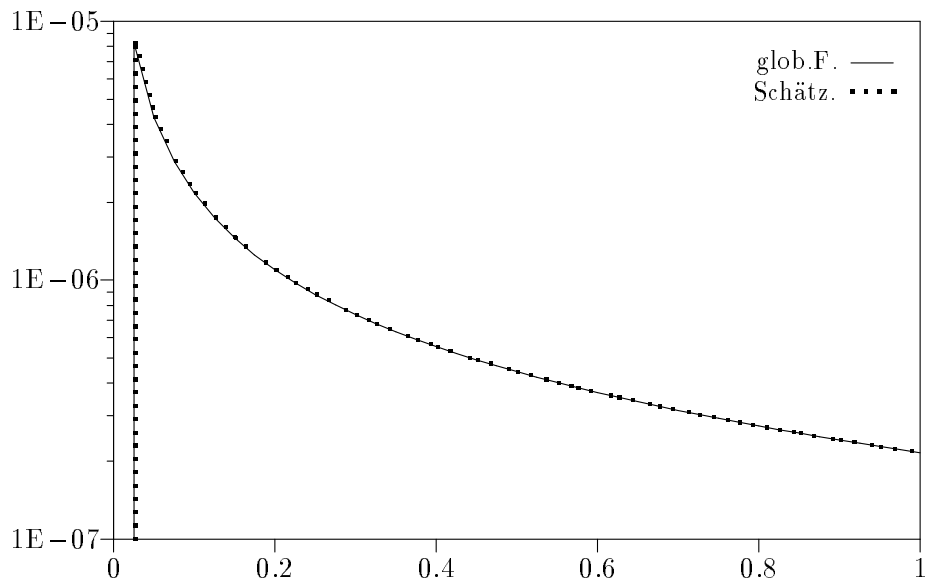


Abbildung 5.7: Fehler Zadunaisky für ERK 4 in Beispiel 1b

Abbildung 5.8: Qualität Zadunaisky für ERK 4 in Beispiel 1b, $h = 1/5 \cdot 2^{-3}$

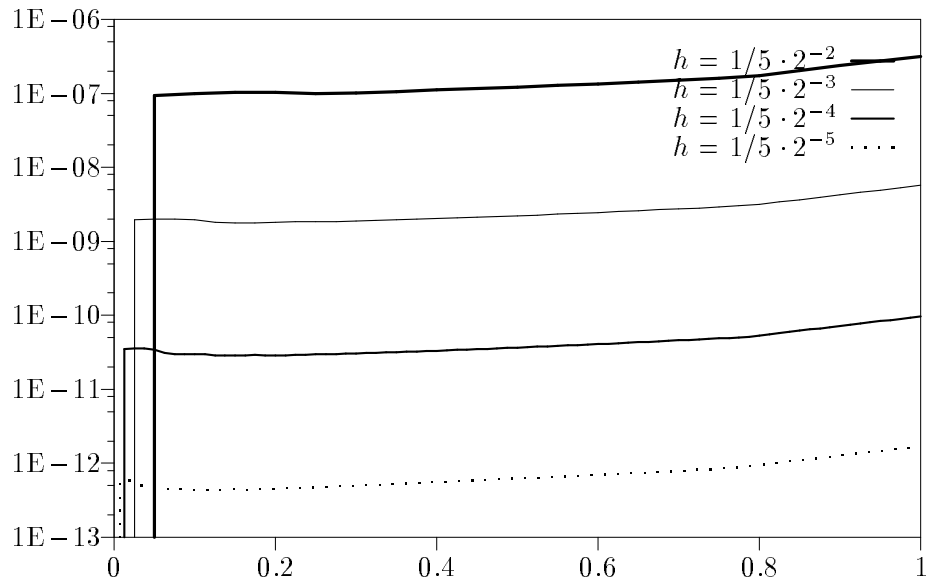
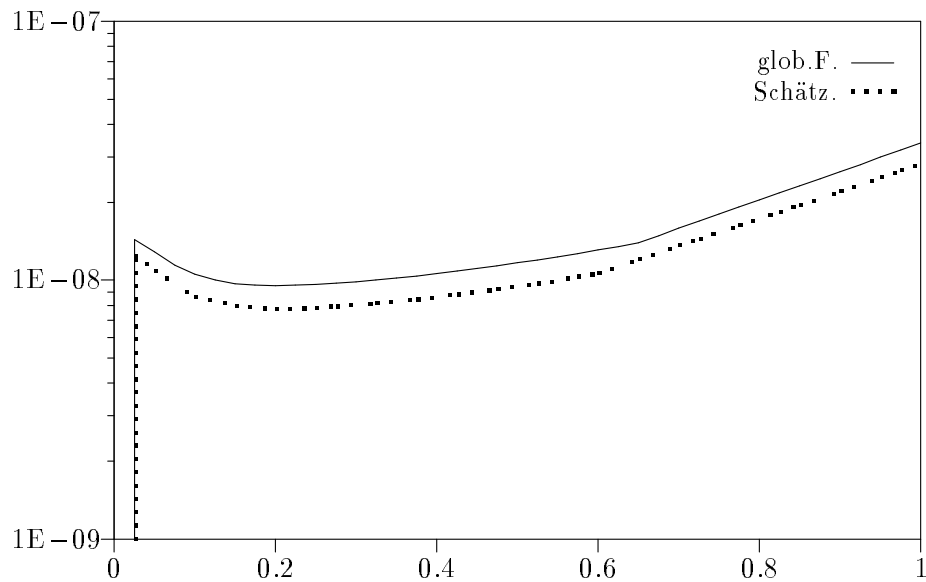


Abbildung 5.9: Fehler Zadunaisky für ERK 4 in Beispiel 2a

Abbildung 5.10: Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-3}$

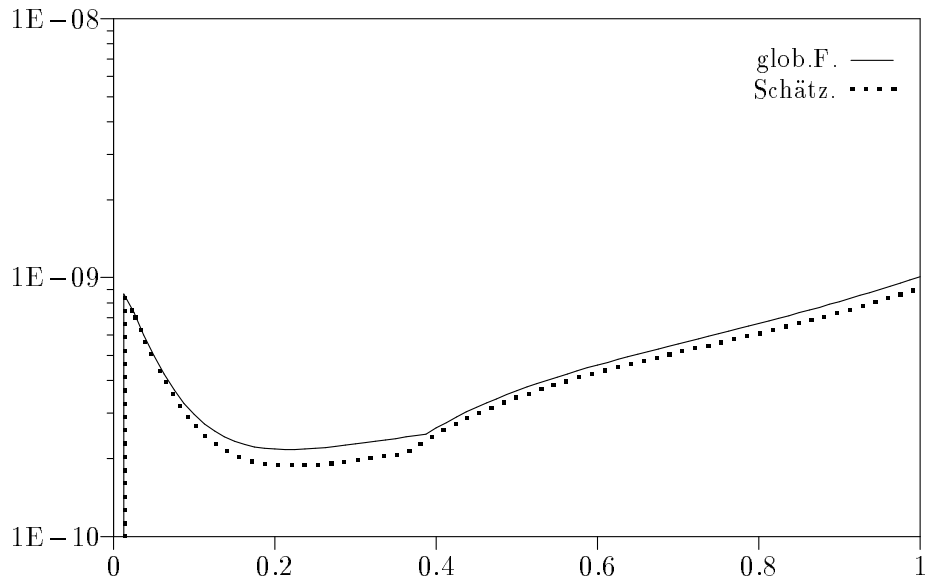


Abbildung 5.11: Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-4}$

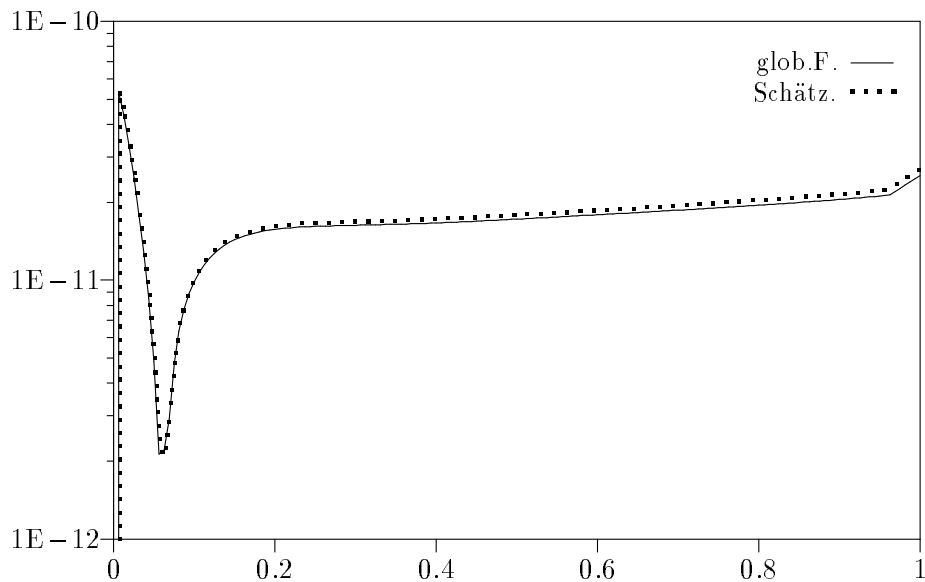


Abbildung 5.12: Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-5}$

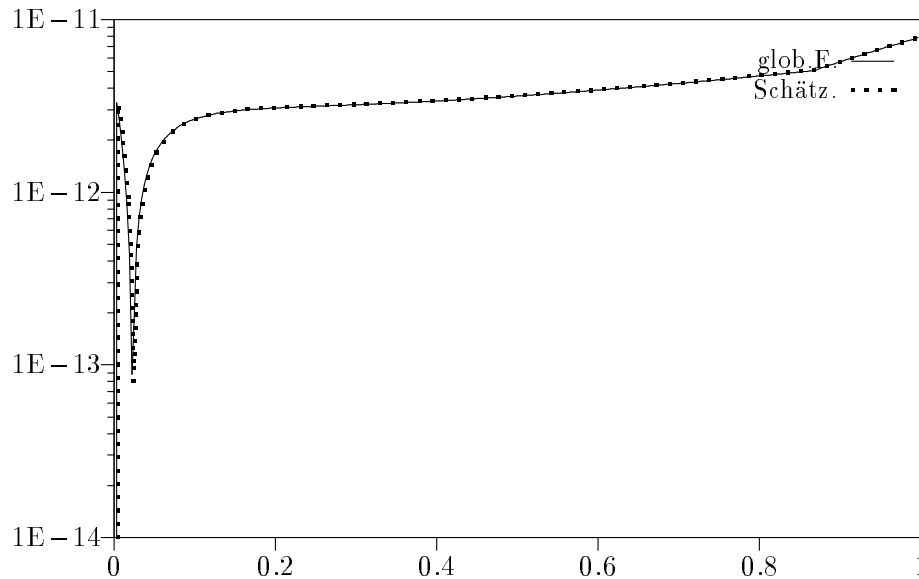


Abbildung 5.13: Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-6}$

Zadunaisky-Schätzung für ERK 4 für Beispiel 1b: In Abb. 5.7 hat der Fehler des Schätzers sein Maximum nach dem ersten Schritt und nimmt dann im Verlauf der Integration kontinuierlich ab. Der Zadunaisky-Schätzer gibt schon für moderates h den Verlauf und die Größenordnung des globalen Fehlers auf 3 Stellen genau an, vgl. in Abb. 5.8 liegen die Werte der Kurven bei $1E-5$ bis $1E-6$, der Fehler des Schätzers ist jedoch etwa $1E-8$. Da der globale Fehler sein Maximum ebenfalls am Beginn des Intervalls hat, ist der relative Fehler des Schätzers über das ganze Intervall annähernd konstant. Im Vergleich mit Beispiel 1a heißt das, daß eine vorgegebene Genauigkeit der Schätzung nicht unbedingt von der Schrittweite abhängt, d.h. bei gleichem h kann der Zadunaisky-Schätzer für unterschiedliche Beispiele unterschiedlich genau sein.

Zadunaisky-Schätzung für ERK 4 für Beispiel 2a: Ähnlich zu Beispiel 1a verläuft der Fehler des Zadunaisky-Schätzers fast geradlinig, wobei er am Ende des Intervalls leicht zunimmt, siehe Abb. 5.9. Die Abb. 5.10 bis 5.13 zeigen, wie sich die Schätzung dem globalen Fehler mit kleiner werdender Schrittweite immer besser annähert. Obwohl der globale Fehler am Beginn der Integration Schwankungen von mehr als einer Zehnerpotenz hat, hat die Schätzung im Gegensatz zu Beispiel 1b konstante absolute Genauigkeit.

Zadunaisky-Schätzung für ERK 4 für Beispiel 2b: Die Abb. 5.14 und 5.15 zeigen ein ähnlich gutes Verhalten des Zadunaisky-Schätzers, wie Abb. 5.7 und 5.8. Der Fehler des Schätzers nimmt im Laufe der Integration leicht zu. Mit kleiner werdendem h wird der Schätzer um 1.5 Zehnerpotenzen genauer. Der globale Fehler wird schon bei großen Schrittweiten gut aufgelöst. Die Zadunaisky-Methode zur Schätzung des globalen Fehlers funktioniert sehr gut.

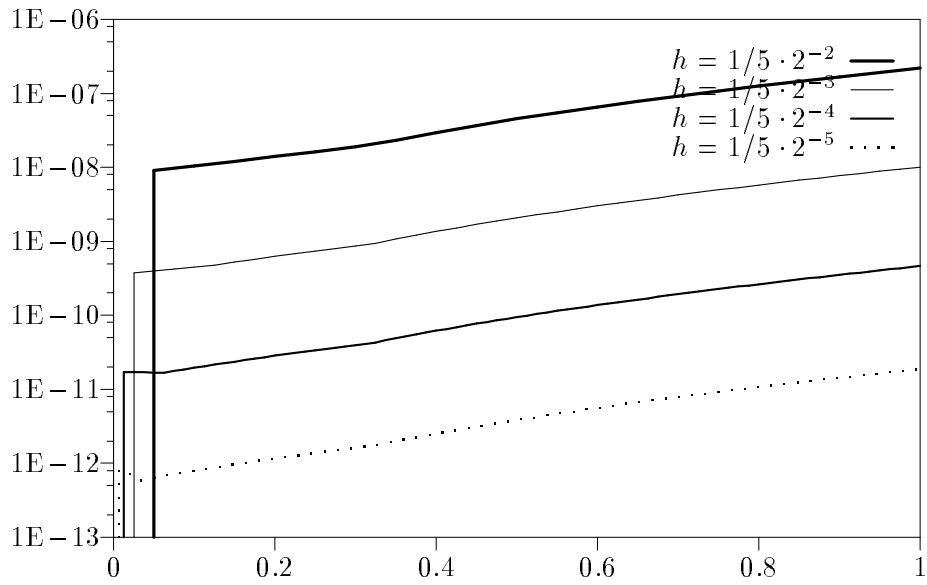
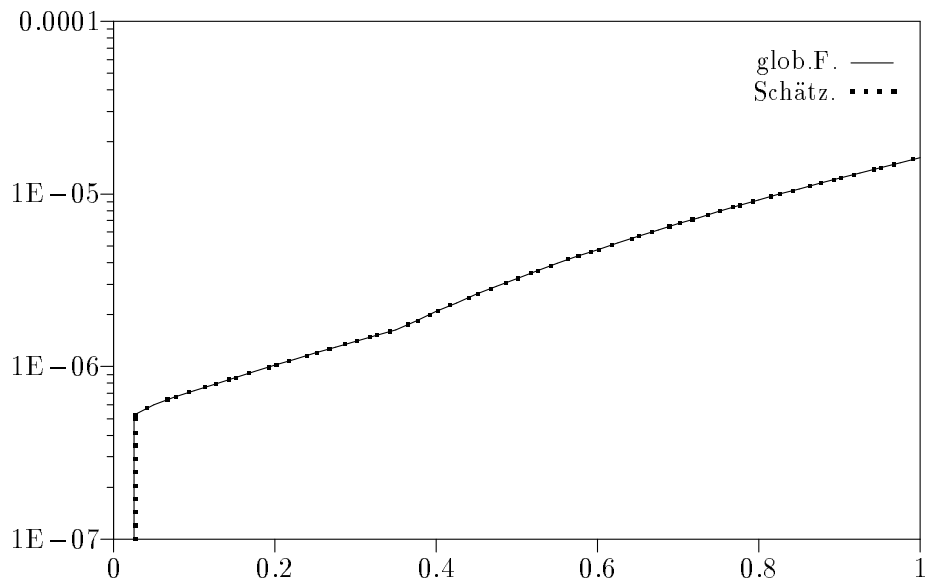


Abbildung 5.14: Fehler Zadunaisky für ERK 4 in Beispiel 2b

Abbildung 5.15: Qualität Zadunaisky für ERK 4 in Beispiel 2b, $h = 1/5 \cdot 2^{-3}$

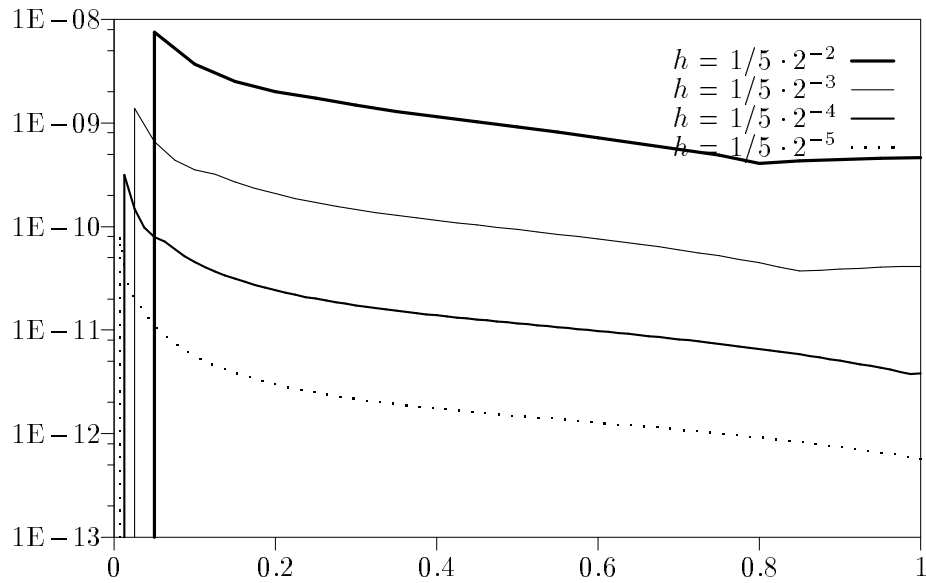
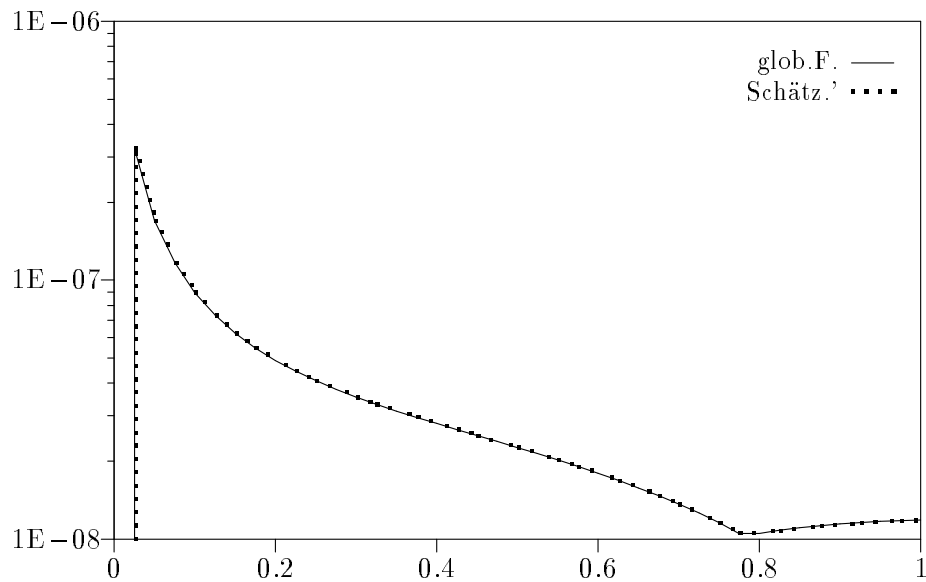


Abbildung 5.16: Fehler Zadunaisky für ERK 4 in Beispiel 3a

Abbildung 5.17: Qualität Zadunaisky für ERK 4 in Beispiel 3a, $h = 1/5 \cdot 2^{-3}$

Zadunaisky-Schätzung für ERK 4 für Beispiel 3a: Auch das nichtlineare Problem 3a wird zufriedenstellend gelöst. Der Fehler des Schätzers für $h = 1/5 \cdot 2^{-3}$, siehe Abb. 5.17, liegt 2.5 Zehnerpotenzen unter der Größe des Schätzers, vgl. Abb. 5.16. Die Schätzung des globalen Fehlers ist sehr gut.

5.2.2 Basismethode explizites Runge-Kutta 7

Zum Vergleich mit dem vorigen Abschnitt wurde das Verfahren DP7, also die Runge-Kutta Methode von DOPRI8 mit der Ordnung $p = 7$ getestet. Mit der höheren Ordnung mußte auch der Grad der Interpolationspolynome auf $k = 8$ erhöht werden. Die entsprechenden Experimente finden sich in [30].

Es zeigt sich, daß der globale Fehler, wie auch schon von DP4, entlang des Intervalls gleichmäßig gut approximiert wird. Mit kleiner werdender Schrittweite verringert sich der Fehler des Schätzers um etwa 2 Zehnerpotenzen, also grob um $1/2^2 = 1/4$. Die Qualität des Schätzers verbessert sich mit kleiner werdender Schrittweite immer mehr. Selbst unter Einfluß der Rechenfehler, für $h \leq 1/4 \cdot 2^{-3}$ hat in einigen Beispielen sowohl der globale Fehler als auch der Fehler des Schätzers die Größenordnung 100 EPS, und der dadurch bedingten Unsystematik folgt der Schätzer dem globalen Fehler gut. Die Schätzung löst den Verlauf des globalen Fehlers und die Größenordnung teilweise sogar auf 5 Dezimalstellen richtig auf.

Wird der Grad des Polynoms $p(x)$ von 8 auf $k = 10$ erhöht, machen sich auch die Rechenfehler stärker bemerkbar. Es zeigt sich, daß der Fehler des Schätzers bei kleiner werdender Schrittweite nicht verbessert, sondern wächst.

5.2.3 Ordnung des Fehlers des Zadunaisky-Schätzers für ERK 4

Im vorigen Abschnitt war zu erkennen, daß mit kleiner werdendem h der Fehler des Zadunaisky-Schätzers ebenfalls abnimmt. In diesem Abschnitt soll nun die Ordnung dieser Genauigkeitszunahme untersucht werden. Dazu lösen wir mit dem klassischen Runge-Kutta Verfahren, Butcher-Array auf Seite 64, mit $p = 4$ und $k = 5$ einige Testbeispiele zu verschiedenen Schrittweiten, $h_1 = 1/5$, $h_2 = h_1/2$, etc. Aus den Fehlern der entsprechenden Schätzungen $\delta_{n_i} - \varepsilon_{n_i}$ und $\delta_{n_{i+1}} - \varepsilon_{n_{i+1}}$ zu den Schrittweiten h_i und h_{i+1} kann mittels (5.1) und (5.2) die Ordnung und Fehlerkonstante berechnet werden. Bei regulären Testbeispielen hat sich als Konvergenzordnung $p = k + 1 = 6$ eingestellt.

Die folgenden Tabellen zeigen jeweils die erste Komponente des globalen Fehler, ε , die erste Komponente des Fehlers des Zadunaisky-Schätzers, $\delta - \varepsilon$, die Konvergenzordnung des Fehlers, p , und die Fehlerkonstante c .

Der Zadunaisky-Schätzer ist in den meisten Fällen 2 bis 3 Dezimalstellen genau, d.h. der Fehler des Schätzers liegt 2 bis 3 Zehnerpotenzen unter der Größenordnung des globalen Fehlers. In den Beispielen 1a und 2a, wie auch schon im vorigen Abschnitt, müssen die Schrittweiten schon relativ klein sein, $h_7 = 1/5 \cdot 2^{-6} \approx 0.003$, damit der Schätzer gute Werte liefert, vgl. Tabellen 5.12 und 5.14. In den anderen Fällen liefert die Zadunaisky-Methode auch schon für große Schrittweiten, teilweise sogar schon für die größtmögliche $h_1 = 1/k = 1/5$, gute Ergebnisse.

h	ε	$\delta - \varepsilon$	p	c
$1/5$	$9.0 \cdot 10^{-01}$	$2.0 \cdot 10^{-01}$	6.354	$5.6 \cdot 10^{+04}$
$1/5 \cdot 2^{-1}$	$1.5 \cdot 10^{-03}$	$2.4 \cdot 10^{-03}$	-0.160	$1.7 \cdot 10^{-03}$
$1/5 \cdot 2^{-2}$	$4.4 \cdot 10^{-04}$	$2.7 \cdot 10^{-03}$	4.240	$-9.1 \cdot 10^{+03}$
$1/5 \cdot 2^{-3}$	$1.6 \cdot 10^{-06}$	$1.4 \cdot 10^{-04}$	5.167	$-2.7 \cdot 10^{+04}$
$1/5 \cdot 2^{-4}$	$1.0 \cdot 10^{-06}$	$4.0 \cdot 10^{-05}$	5.551	$-1.4 \cdot 10^{+05}$
$1/5 \cdot 2^{-5}$	$1.5 \cdot 10^{-07}$	$8.7 \cdot 10^{-07}$	5.733	$-3.7 \cdot 10^{+06}$
$1/5 \cdot 2^{-6}$	$1.6 \cdot 10^{-08}$	$1.6 \cdot 10^{-09}$	5.837	$-6.8 \cdot 10^{+06}$
$1/5 \cdot 2^{-7}$	$1.5 \cdot 10^{-09}$	$2.8 \cdot 10^{-11}$	5.862	$-8.1 \cdot 10^{+06}$
$1/5 \cdot 2^{-8}$	$1.3 \cdot 10^{-10}$	$4.9 \cdot 10^{-12}$	3.264	$-6.8 \cdot 10^{-02}$

Tabelle 5.12: Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 1a an $x = 1$

h	ε	$\delta - \varepsilon$	p	c
$1/5$	$1.1 \cdot 10^{-04}$	$6.7 \cdot 10^{-04}$	6.451	$2.1 \cdot 10^{+00}$
$1/5 \cdot 2^{-1}$	$7.3 \cdot 10^{-05}$	$7.6 \cdot 10^{-06}$	6.096	$9.6 \cdot 10^{+00}$
$1/5 \cdot 2^{-2}$	$4.5 \cdot 10^{-06}$	$1.1 \cdot 10^{-08}$	6.023	$7.7 \cdot 10^{+00}$
$1/5 \cdot 2^{-3}$	$2.8 \cdot 10^{-08}$	$1.7 \cdot 10^{-10}$	6.006	$7.2 \cdot 10^{+00}$
$1/5 \cdot 2^{-4}$	$1.7 \cdot 10^{-09}$	$2.6 \cdot 10^{-12}$	6.067	$9.4 \cdot 10^{+00}$
$1/5 \cdot 2^{-5}$	$1.1 \cdot 10^{-10}$	$4.0 \cdot 10^{-13}$	2.263	$3.9 \cdot 10^{-08}$
$1/5 \cdot 2^{-6}$	$6.9 \cdot 10^{-11}$	$8.3 \cdot 10^{-14}$	0.129	$1.7 \cdot 10^{-14}$
$1/5 \cdot 2^{-7}$	$4.0 \cdot 10^{-12}$	$7.6 \cdot 10^{-14}$	0.019	$8.6 \cdot 10^{-14}$

Tabelle 5.13: Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 1b an $x = 1$

h	ε	$\delta - \varepsilon$	p	c
$1/5$	$1.1 \cdot 10^{-04}$	$4.8 \cdot 10^{-02}$	4.063	$-3.3 \cdot 10^{+01}$
$1/5 \cdot 2^{-1}$	$7.3 \cdot 10^{-05}$	$2.9 \cdot 10^{-04}$	5.089	$-3.5 \cdot 10^{+02}$
$1/5 \cdot 2^{-2}$	$4.5 \cdot 10^{-06}$	$8.6 \cdot 10^{-05}$	5.517	$-1.2 \cdot 10^{+02}$
$1/5 \cdot 2^{-3}$	$2.8 \cdot 10^{-08}$	$1.8 \cdot 10^{-07}$	5.718	$-2.7 \cdot 10^{+02}$
$1/5 \cdot 2^{-4}$	$1.7 \cdot 10^{-09}$	$3.5 \cdot 10^{-08}$	5.829	$-4.4 \cdot 10^{+03}$
$1/5 \cdot 2^{-5}$	$1.1 \cdot 10^{-10}$	$6.2 \cdot 10^{-10}$	5.935	$-7.5 \cdot 10^{+03}$
$1/5 \cdot 2^{-6}$	$6.9 \cdot 10^{-11}$	$1.0 \cdot 10^{-12}$	5.555	$-8.4 \cdot 10^{+02}$
$1/5 \cdot 2^{-7}$	$4.0 \cdot 10^{-12}$	$2.1 \cdot 10^{-14}$	0.011	$-2.3 \cdot 10^{-14}$

Tabelle 5.14: Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 2a an $x = 1$

Die Ordnung des Schätzers variiert je nach Beispiel. Während sich für 1b in der Tabelle 5.13 Ordnung 6 einstellt, sind in der Tabelle 5.15 Ordnungen um 3 bis 3.5 zu sehen. Für nichtlineare Beispiele, vgl. Tabelle 5.16, stellen sich keine Ordnungen ein.

h	ε	$\delta - \varepsilon$	p	c
1/5	$3.4 \cdot 10^{-01}$	$3.6 \cdot 10^{-02}$	8.528	$-3.3 \cdot 10^{+04}$
$1/5 \cdot 2^{-1}$	$6.1 \cdot 10^{-02}$	$9.8 \cdot 10^{-05}$	0.658	$4.4 \cdot 10^{-04}$
$1/5 \cdot 2^{-2}$	$9.8 \cdot 10^{-03}$	$6.2 \cdot 10^{-05}$	3.321	$1.3 \cdot 10^{-01}$
$1/5 \cdot 2^{-3}$	$1.4 \cdot 10^{-04}$	$6.2 \cdot 10^{-06}$	3.530	$2.8 \cdot 10^{-01}$
$1/5 \cdot 2^{-4}$	$2.0 \cdot 10^{-05}$	$5.4 \cdot 10^{-07}$	3.508	$2.5 \cdot 10^{-01}$
$1/5 \cdot 2^{-5}$	$2.7 \cdot 10^{-06}$	$4.7 \cdot 10^{-08}$	3.398	$1.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-6}$	$3.5 \cdot 10^{-06}$	$4.5 \cdot 10^{-09}$	3.270	$7.0 \cdot 10^{-01}$
$1/5 \cdot 2^{-7}$	$4.5 \cdot 10^{-07}$	$4.6 \cdot 10^{-10}$	3.107	$2.4 \cdot 10^{-02}$
$1/5 \cdot 2^{-8}$	$5.8 \cdot 10^{-08}$	$5.4 \cdot 10^{-11}$	2.506	$3.3 \cdot 10^{-03}$
$1/5 \cdot 2^{-9}$	$7.3 \cdot 10^{-09}$	$9.5 \cdot 10^{-12}$	0.440	$3.0 \cdot 10^{-11}$

Tabelle 5.15: Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 2b an $x = 1$

h	ε	$\delta - \varepsilon$	p	c
1/5	$1.8 \cdot 10^{-04}$	$8.1 \cdot 10^{-05}$	6.263	$-1.9 \cdot 10^{-01}$
$1/5 \cdot 2^{-1}$	$1.4 \cdot 10^{-05}$	$1.0 \cdot 10^{-07}$	5.624	$-4.4 \cdot 10^{-01}$
$1/5 \cdot 2^{-2}$	$1.0 \cdot 10^{-06}$	$2.1 \cdot 10^{-09}$	5.211	$-1.3 \cdot 10^{-02}$
$1/5 \cdot 2^{-3}$	$7.2 \cdot 10^{-07}$	$5.8 \cdot 10^{-10}$	4.602	$-1.3 \cdot 10^{-03}$
$1/5 \cdot 2^{-4}$	$5.0 \cdot 10^{-08}$	$2.3 \cdot 10^{-12}$	4.197	$-2.3 \cdot 10^{-04}$
$1/5 \cdot 2^{-5}$	$3.4 \cdot 10^{-09}$	$1.3 \cdot 10^{-13}$	4.707	$-3.1 \cdot 10^{-03}$
$1/5 \cdot 2^{-6}$	$2.3 \cdot 10^{-11}$	$4.9 \cdot 10^{-14}$	1.491	$-2.7 \cdot 10^{-11}$
$1/5 \cdot 2^{-7}$	$1.5 \cdot 10^{-12}$	$1.7 \cdot 10^{-15}$	-2.129	$1.8 \cdot 10^{-21}$
$1/5 \cdot 2^{-8}$	$1.0 \cdot 10^{-13}$	$7.7 \cdot 10^{-14}$	0.222	$-3.8 \cdot 10^{-13}$

Tabelle 5.16: Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 3a an $x = 1$

Analog wird die Zadunaisky-Methode der Schätzung des globalen Fehlers für implizite Runge-Kutta Methoden untersucht, die ebenfalls auf (5.6) beruht.

5.2.4 Basismethode implizites Runge-Kutta 4 (Gauss)

Wie in Abschnitt 5.2.1 wird für verschiedene Schrittweiten der globale Fehler ε_j mit dem Wert des Zadunaisky-Schätzers δ_j verglichen, die sich bei der Berechnung der Testbeispiele mit dem impliziten Gauss Verfahren 4. Ordnung und mit $k = 5$ ergeben. Die logarithmisch skalierten Abbildungen zeigen jeweils die Maximumnorm des Fehlers des Zadunaisky-Schätzers und den Verlauf sowohl des globalen Fehlers, (dicke Linie) als auch des Schätzers (punktierte Linie).

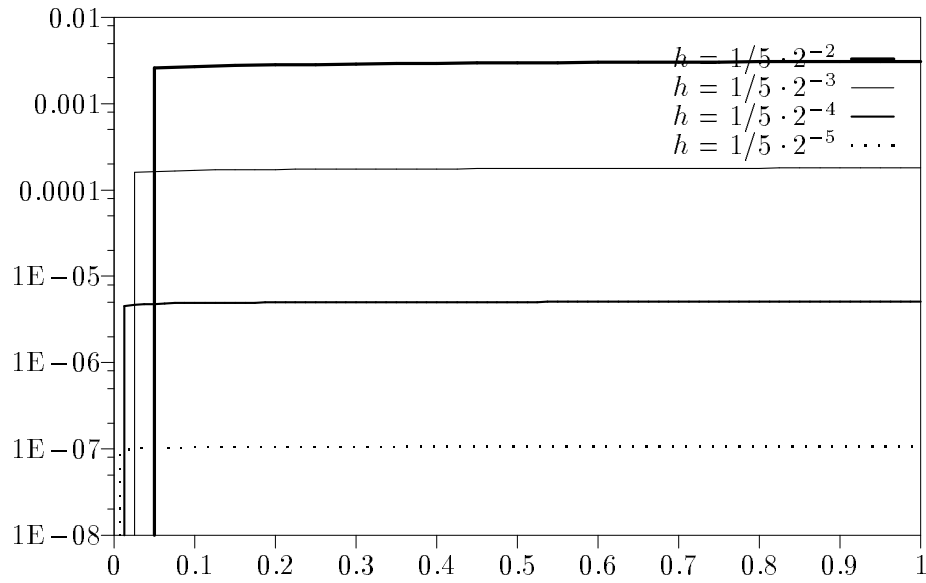
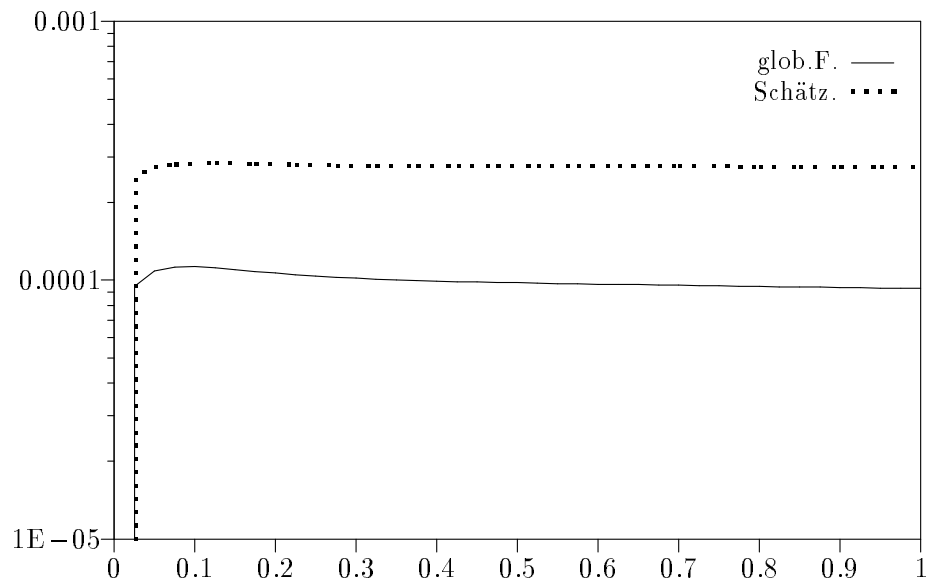


Abbildung 5.18: Fehler Zadunaisky für IRK 4 in Beispiel 1a

Abbildung 5.19: Qualität Zadunaisky für IRK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-3}$

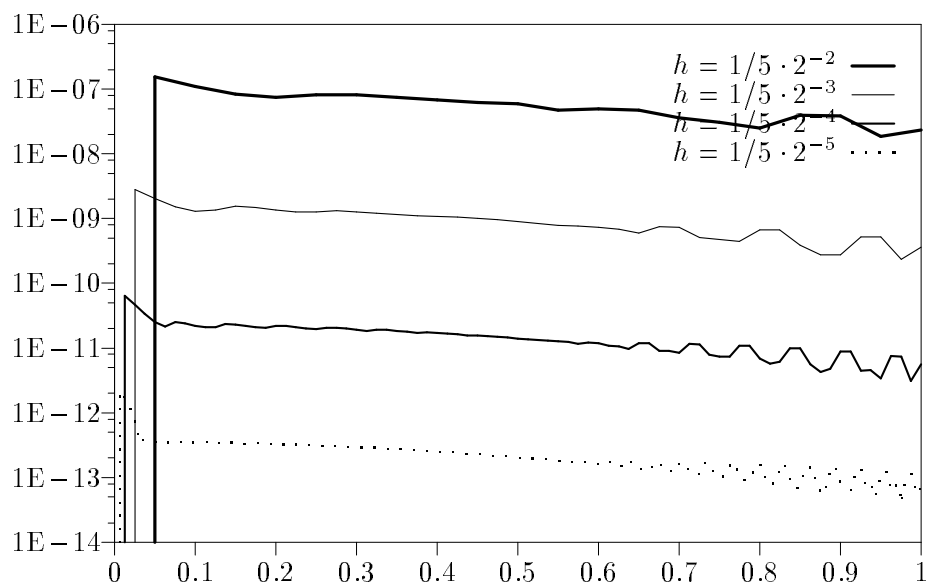
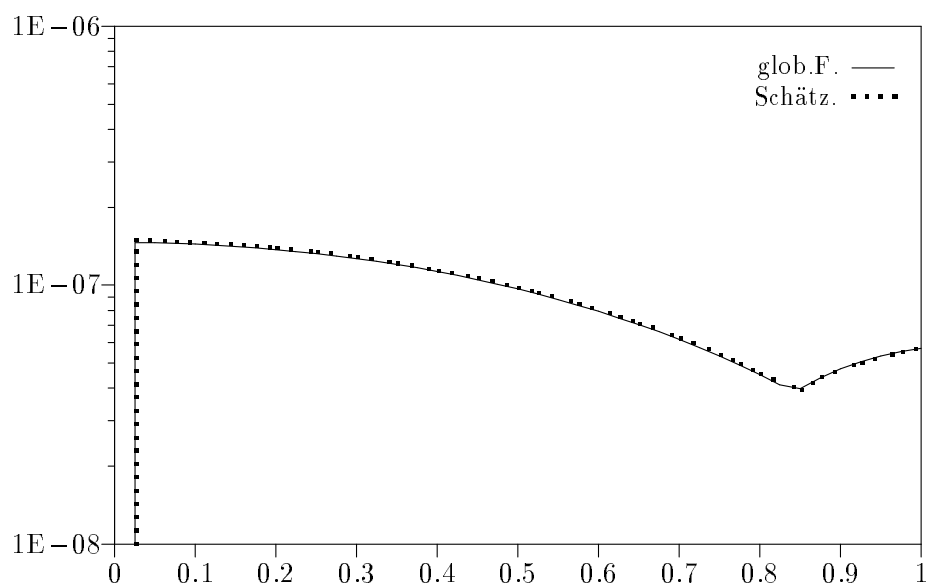


Abbildung 5.20: Fehler Zadunaisky für IRK 4 in Beispiel 1b

Abbildung 5.21: Qualität Zadunaisky für IRK 4 in Beispiel 1b, $h = 1/5 \cdot 2^{-3}$

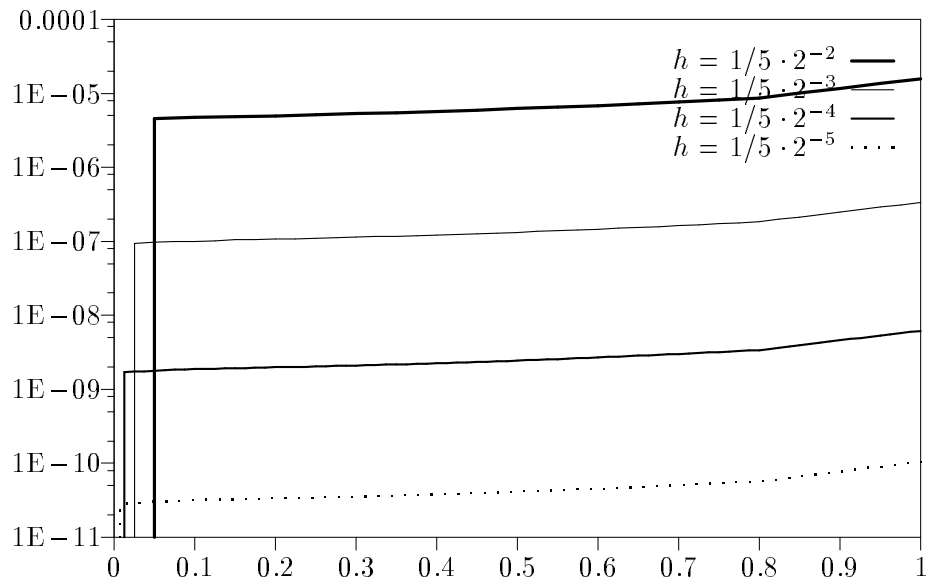
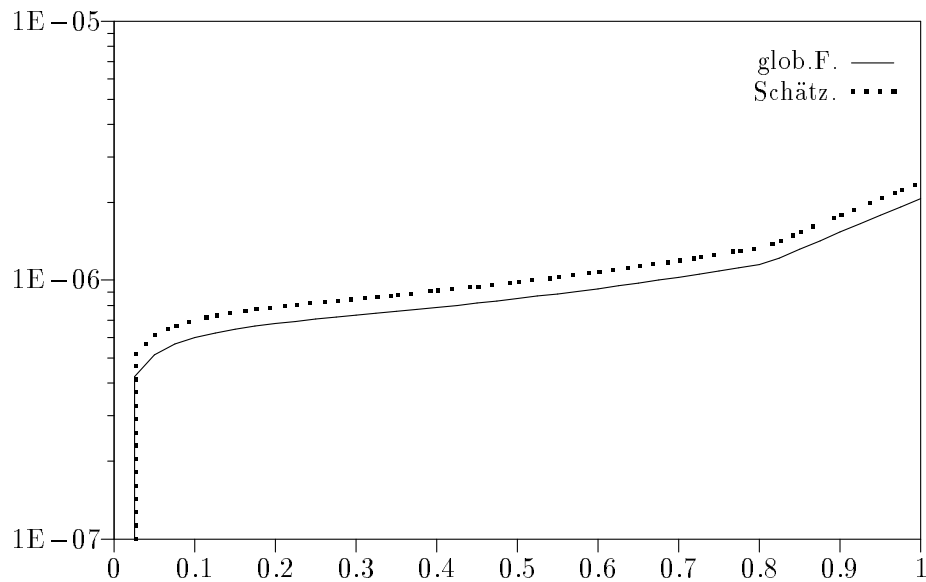


Abbildung 5.22: Fehler Zadunaisky für IRK 4 in Beispiel 2a

Abbildung 5.23: Qualität Zadunaisky für IRK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-3}$

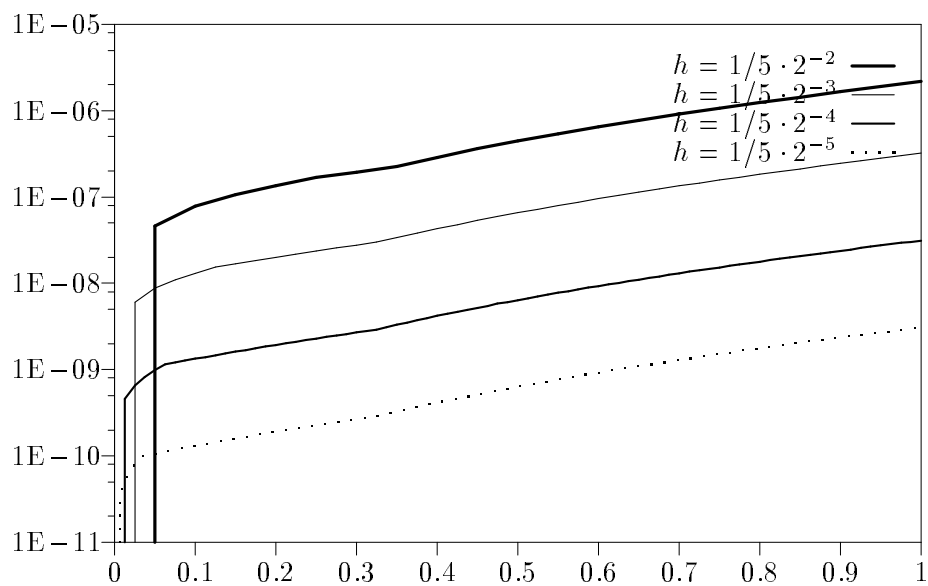
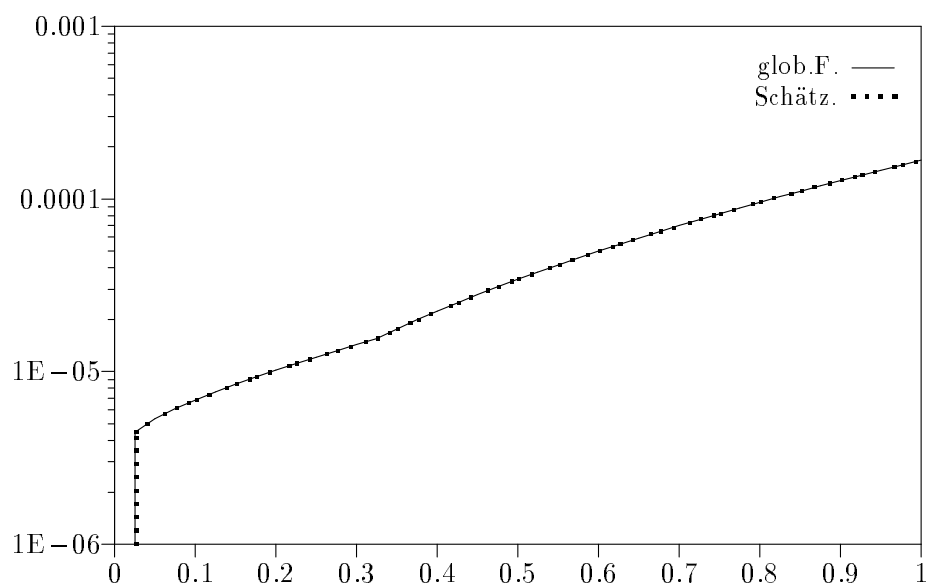


Abbildung 5.24: Fehler Zadunaisky für IRK 4 in Beispiel 2b

Abbildung 5.25: Qualität Zadunaisky für IRK 4 in Beispiel 2b, $h = 1/5 \cdot 2^{-3}$

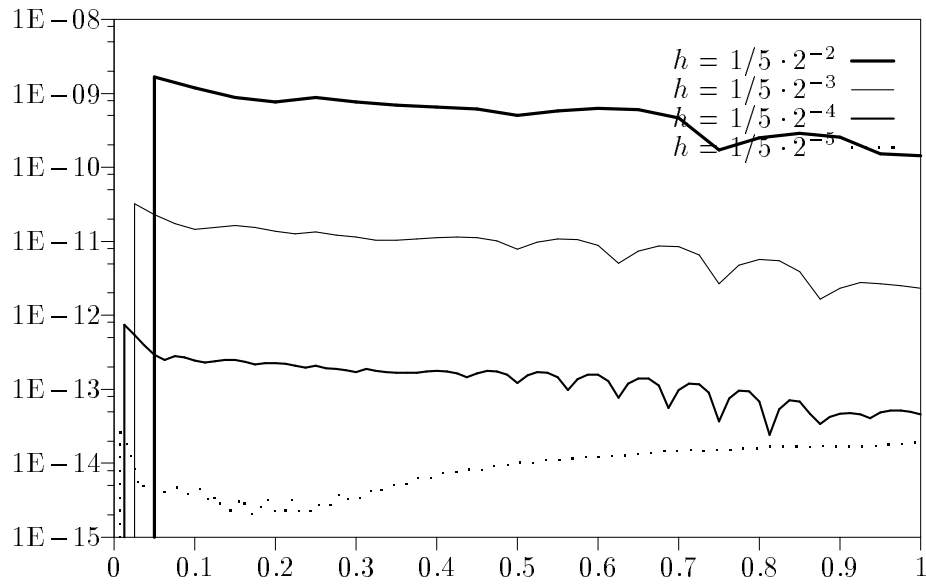
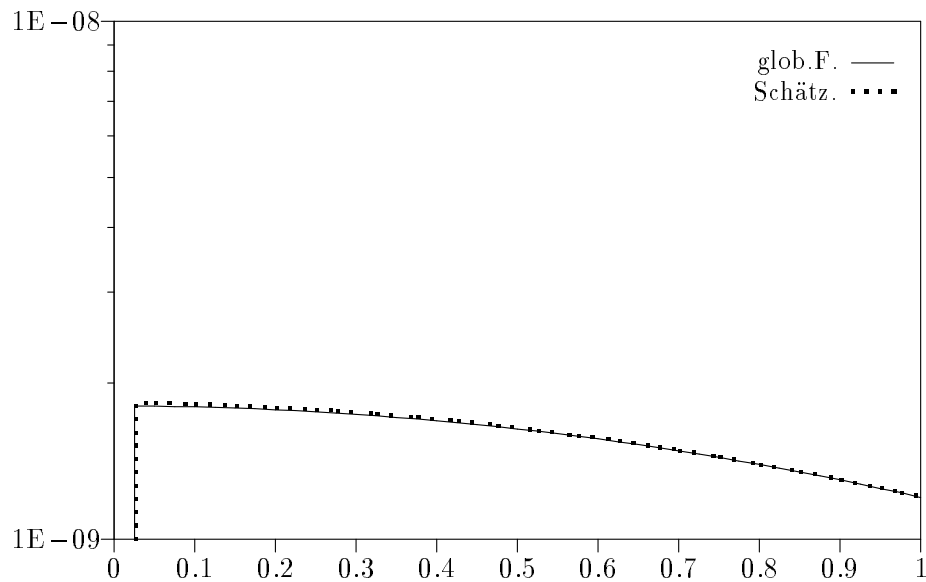


Abbildung 5.26: Fehler Zadunaisky für IRK 4 in Beispiel 3a

Abbildung 5.27: Qualität Zadunaisky für IRK 4 in Beispiel 3a, $h = 1/5 \cdot 2^{-3}$

Zadunaisky-Schätzung für IRK 4 für Beispiel 1a: Der Fehler des Schätzers verläuft geradlinig, Abb. 5.18 und wird mit kleiner werdender Schrittweite um je 1.5 Zehnerpotenzen reduziert. In Abb. 5.19 liegt der Schätzer weniger als eine Halbe Zehnerpotenz über dem globalen Fehler und ist damit ein wenig besser als das explizite Verfahren, vgl. Abb. 5.2, wobei dort der Fehler unterschätzt wurde.

Zadunaisky-Schätzung für IRK 4 für Beispiel 1b: Im Gegensatz zu Abb. 5.7 verläuft der Fehler des Schätzers eher geradlinig, vgl. 5.20, wobei am Ende des Intervalls Oszillationen auftreten. Die Fehler verschiedener Schrittweiten entsprechen in der Größenordnung den Fehlern des expliziten Verfahrens. Abb. 5.21 zeigt ein qualitativ ähnliches Ergebnis wie Abb. 5.8, der Zadunaisky-Schätzer gibt den Verlauf und die Größenordnung des globalen Fehlers auf 1 bis 2 Stellen genau an, vgl. Abb. 5.21.

Zadunaisky-Schätzung für IRK 4 für Beispiel 2a: Wie in Beispiel 1a, Abb. 5.18, verläuft der Fehler des Zadunaisky-Schätzers fast geradlinig und verbessert sich mit kleiner werdender Schrittweite ebenfalls um 1.5 Zehnerpotenzen, vgl. Abb. 5.22. In Abb. 5.23 wird im Gegensatz zu Abb. 5.10 der globale Fehler etwas überschätzt.

Zadunaisky-Schätzung für IRK 4 für Beispiel 2b: Die Abb. 5.24 und 5.25 unterscheiden sich kaum von den Abb. 5.14 und 5.15. Der Zadunaisky-Schätzer liefert gute Näherungen für den globalen Fehler von G4, welcher im Gegensatz zum expliziten DP4 eine Zehnerpotenz größer ist.

Zadunaisky-Schätzung für IRK 4 für Beispiel 3a: Problem 3a wird ebenfalls genau gelöst. Der Schätzer, Abb. 5.27, liegt 2.5 Zehnerpotenzen über dem globalen Fehler, vgl. Abb. 5.26. Die Schätzung nach Zadunaisky funktioniert.

5.2.5 Zusammenfassung

Die Schätzung des globalen Fehlers mittels der Zadunaisky-Methode funktioniert überraschend gut. Der globale Fehler wird entlang des Intervalls gut genähert. Teilweise kann schon mit sehr groben Gittern eine auf drei Stellen genaue Schätzung erhalten werden. Für kleine Schrittweiten löst δ den globalen Fehler ε sehr genau auf, und die Näherung verbessert sich mit kleiner werdender Schrittweite mit Ordnungen zwischen 3 und 6, vgl. Abschnitt 5.2.3.

Für Verfahren höherer Ordnung wird das Verfahren empfindlicher auf Rechenfehler. Mit dem durch die Ordnung bedingten höheren Polynomgrad $k = p + 1$, ist man teilweise schon nach wenigen Gitterverfeinerungen im Rundungsfehlerbereich. Für gröbere Gitter funktioniert die Zadunaisky-Methode aber auch mit Verfahren hoher Ordnung, wobei sich mit wachsender Ordnung des Verfahrens die Situation immer mehr verschlechtert, so daß die Fehler bei kleiner werdender Schrittweite aufgrund der Rechenfehler wachsen. Prinzipiell funktioniert die Fehlerschätzung nach Zadunaisky so gut, daß ihre Verwendung in ernsthafter Software sinnvoll erscheint.

Der Fehler der Schätzung für implizite Runge-Kutta Verfahren wird im allgemeinen überschätzt, während er im Fall der expliziten Methode unterschätzt wurde. Sonst sind keine gravierenden Unterschiede zwischen expliziten und impliziten Verfahren zu erkennen, obwohl es scheint, daß die impliziten Verfahren etwas bessere Lösungen liefern.

In der Zadunaisky-Methode stellt die zusätzliche numerische Lösung von (5.5) einen beträchtlichen Mehraufwand dar. Stetter hat in [40], Seite 440, eine mögliche Vereinfachung der Berechnung von δ vorgeschlagen, die auf der Methode der Iterierten Defekt-Korrektur basiert. Man löst das Anfangswertproblem (2.7) numerisch und erhält y_Δ . Diese Lösung wird als Zwischenergebnis $\tilde{y}_\Delta^{(i)}$ einer Defekt-Korrektur Iteration basierend auf einem einfacheren Basisverfahren, z.B. Euler-Verfahren, interpretiert. Man führt einen weiteren Schritt dieser hypothetischen Defekt-Korrektur Iteration durch, und erhält eine Schätzung des globalen Fehlers. Praktisch heißt das, man löst zusätzlich (2.7) auch mit dem einfacheren Verfahren für \tilde{y}_Δ , und das Nachbarproblem (5.5), um $\tilde{z}_\Delta^{(i)}$ zu erhalten. Der Aufwand wird somit von zweimaliger Anwendung des höheren Verfahrens auf dessen einmalige Anwendung und die zweimalige Anwendung des einfacheren Verfahrens reduziert.

Diese Schätzung des globalen Fehlers funktioniert aber nur sehr eingeschränkt. Selbst auf relativ feinen Gittern stellen sich keine brauchbaren Schätzungen der Größenordnung ein. Im Vergleich mit der Zadunaisky-Schätzung ist der Fehler des Schätzers im Durchschnitt um 3 Zehnerpotenzen größer, unabhängig von der Schrittweite, und mit kleiner werdender Schrittweite verbessert sich die Qualität des Schätzers nicht unbedingt.

6 Zusammenfassung

Aus den Ergebnissen der Abschnitte 3 und 5 ergeben sich einige Konsequenzen für die Programmentwicklung eines experimentellen Anfangswertaufgaben-Integrators. Aufgrund der Ordnungsreduktion im lokalen Fehler, vgl. Zusammenfassung 3.3.5, und der daraus resultierenden Unterschätzung des lokalen Fehlers an der Singularität durch die klassischen Verfahren, siehe Abschnitte 3.1.6 und 3.2.3, steht die Verwendung lokaler Strategien in Frage.

Das ausgezeichnete Verhalten der Zadunaisky-Schätzung des globalen Fehlers, vgl. Abschnitt 5.2, bestärkt noch mehr die Idee, globale Strategien zumindestens nahe der Singularität zu verwenden. Weg von der Singularität, wo sich der lokale Fehler durchwegs klassisch verhält, vgl. Abschnitt 3.3, können aus Effizienzgründen weiterhin lokale Gitterstrategien verwendet werden. Ein Algorithmus zur numerischen Lösung singulärer Anfangswertaufgaben⁷ könnte wie folgt aussehen:

1. Integration der Anfangswertaufgabe bei konstanter Schrittweite h mittels Runge-Kutta Methoden bis zu einer festen Stelle $x^* \in (0, 1]$, die möglichst nahe bei Null liegen soll, um den zusätzlichen Aufwand zu minimieren (Umschaltpunkt).
2. Schätzung des globalen Fehlers bei x^* mittels der Zadunaisky-Methode.
3. Bei ungenügender Genauigkeit Anpassung der Schrittweite h und neuerliche Integration (Punkt 1).
4. Integration der Anfangswertaufgabe ab x^* mittels einer auf Schätzung des lokalen Fehlers basierenden Methode, z.B. Runge-Kutta Paare.

Ein entsprechender Code in Englischer Sprache wurde in [30] verfaßt, wobei versucht wurde, die speziellen Strukturen von FORTRAN 90, vor allem die vektorwertigen Operationen voll auszunützen, und für maximale Portabilität zu anderen Computersystemen zu sorgen. Im dem dort bereitgestellten Programm wurde ein Algorithmus zur Integration singulärer Anfangswertprobleme implementiert, der den Schwierigkeiten der Singularität zum Teil gerecht wurde. Das Kernstück des Programms stellt die Zadunaisky Methode zur Schätzung des globalen Fehlers dar. Obwohl auch der globale Fehler Ordnungsreduktionen aufweist, funktioniert sie sowohl für explizite, als auch für implizite Verfahren sehr gut. Der globale Fehler wird gut wiedergegeben.

Genügend weit weg von der Singularität können Schrittweitensteuerungen basierend auf der Schätzung des lokalen Fehlers verwendet werden, da Ordnungsreduktionen im lokalen Fehler mit der wachsenden Entfernung von der Singularität schnell abnehmen, wodurch eine Unterschätzung des lokalen Fehlers nicht mehr gegeben ist.

⁷Und mittels Schießverfahren somit für Randwertaufgaben.

Tabellenverzeichnis

3.1	Ordnung lokaler Fehler ERK 4 für Beispiel 1a an $x^* = 0, \dots, 0.5$	51
3.2	Ordnung lokaler Fehler ERK 4 für Beispiel 1b an $x^* = 0, \dots, 0.5$	52
3.3	Ordnung lokaler Fehler ERK 4 für Beispiel 2a an $x^* = 0, \dots, 0.5$	53
3.4	Ordnung lokaler Fehler ERK 4 für Beispiel 2b an $x^* = 0, \dots, 0.5$	54
3.5	Ordnung lokaler Fehler ERK 4 für Beispiel 3a an $x^* = 0, \dots, 0.5$	55
3.6	Ordnung lokaler Fehler IRK 5 für Beispiel 1a an $x^* = 0, 0.5$	56
3.7	Ordnung lokaler Fehler IRK 5 für Beispiel 1b an $x^* = 0, 0.5$	56
3.8	Ordnung lokaler Fehler IRK 5 für Beispiel 2a an $x^* = 0, 0.5$	57
3.9	Ordnung lokaler Fehler IRK 5 für Beispiel 2b an $x^* = 0, 0.5$	57
3.10	Ordnung lokaler Fehler IRK 5 für Beispiel 3a an $x^* = 0, 0.5$	58
3.11	Anzahl Schritte bis zum Erreichen der Ordnung für alle Beispiele	60
3.12	Anzahl Schritte bis zum Erreichen der Ordnung in Beispiel 1c	61
3.13	Ordnungen lokaler Fehler ERK aller Beispiele an $x^* = 0$	64
3.14	Ordnungen lokaler Fehler IRK aller Beispiele an $x^* = 0$	64
5.1	Ordnung globaler Fehler ERK 4 für Beispiel 1a an $x = 1$	84
5.2	Ordnung globaler Fehler ERK 4 für Beispiel 1b an $x = 1$	84
5.3	Ordnung globaler Fehler ERK 4 für Beispiel 2a an $x = 1$	85
5.4	Ordnung globaler Fehler ERK 4 für Beispiel 2b an $x = 1$	85
5.5	Ordnung globaler Fehler ERK 4 für Beispiel 3a an $x = 1$	85
5.6	Ordnung globaler Fehler IRK 4 für Beispiel 1a an $x = 1$	86
5.7	Ordnung globaler Fehler IRK 4 für Beispiel 1b an $x = 1$	86
5.8	Ordnung globaler Fehler IRK 4 für Beispiel 2a an $x = 1$	87
5.9	Ordnung globaler Fehler IRK 4 für Beispiel 2b an $x = 1$	87
5.10	Ordnung globaler Fehler IRK 4 für Beispiel 3a an $x = 1$	87
5.11	Ordnungen globaler Fehler RK aller Beispiele an $x = 1$	88
5.12	Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 1a an $x = 1$	101
5.13	Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 1b an $x = 1$	101
5.14	Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 2a an $x = 1$	101
5.15	Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 2b an $x = 1$	102
5.16	Ordnung Fehler Zad.-Schätzer ERK 4 für Beispiel 3a an $x = 1$	102

Abbildungsverzeichnis

2.1	Die Lösungskomponenten von Beispiel 1a	9
2.2	Die Lösungskomponenten von Beispiel 1b	9
2.3	Die Lösungskomponenten von Beispiel 1c	10
2.4	Die Lösungskomponenten von Beispiel 2a	11
2.5	Die Lösungskomponenten von Beispiel 2b	12
2.6	Die Lösungskomponenten von Beispiel 3a	13
2.7	FORTRAN 90 Code für ein explizites Runge-Kutta Verfahren	16
3.1	Fehler 5(4) Paar in Beispiel 1a, Toleranz $1E-6$	19
3.2	Fehler 5(4) Paar in Beispiel 1a, Toleranz $1E-9$	20
3.3	Fehler 5(4) Paar in Beispiel 1a, Toleranz $1E-12$	20
3.4	xjh_j -Plot 5(4) Paar für Beispiel 1a	21

3.5	Fehler 5(4) Paar in Beispiel 1b, Toleranz $1E-6$	22
3.6	Fehler 5(4) Paar in Beispiel 1b, Toleranz $1E-9$	22
3.7	$x_j h_j$ -Plot 5(4) Paar für Beispiel 1b	23
3.8	Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-6$	23
3.9	Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-9$	24
3.10	Fehler 5(4) Paar in Beispiel 2a, Toleranz $1E-12$	24
3.11	$x_j h_j$ -Plot 5(4) Paar für Beispiel 2a	25
3.12	Fehler 5(4) Paar in Beispiel 2b, Toleranz $1E-6$	25
3.13	Fehler 5(4) Paar in Beispiel 2b, Toleranz $1E-9$	26
3.14	$x_j h_j$ -Plot 5(4) Paar für Beispiel 2b	26
3.15	Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-6$	27
3.16	Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-9$	27
3.17	Fehler 5(4) Paar in Beispiel 3a, Toleranz $1E-12$	28
3.18	$x_j h_j$ -Plot 5(4) Paar für Beispiel 3a	28
3.19	Fehler 4(5) Paar in Beispiel 1a, Toleranz $1E-6$	30
3.20	Fehler 4(5) Paar in Beispiel 1a, Toleranz $1E-9$	31
3.21	Fehler 4(5) Paar in Beispiel 1b, Toleranz $1E-6$	31
3.22	Fehler 4(5) Paar in Beispiel 1b, Toleranz $1E-9$	32
3.23	Fehler 4(5) Paar in Beispiel 2a, Toleranz $1E-6$	32
3.24	Fehler 4(5) Paar in Beispiel 2a, Toleranz $1E-9$	33
3.25	Fehler 5(4) Paar exakt in Beispiel 1a, Toleranz $1E-6$	34
3.26	Fehler 5(4) Paar exakt in Beispiel 1a, Toleranz $1E-9$	34
3.27	$x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 1a	35
3.28	Fehler 5(4) Paar exakt in Beispiel 1b, Toleranz $1E-6$	35
3.29	Fehler 5(4) Paar exakt in Beispiel 1b, Toleranz $1E-9$	36
3.30	$x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 1b	36
3.31	Fehler 5(4) Paar exakt in Beispiel 2a, Toleranz $1E-6$	37
3.32	Fehler 5(4) Paar exakt in Beispiel 2a, Toleranz $1E-9$	37
3.33	$x_j h_j$ -Plot 5(4) Paar exakt für Beispiel 2a	38
3.34	Fehler Richardson für ERK 4 in Beispiel 1a, Toleranz $1E-6$	40
3.35	Fehler Richardson für ERK 4 in Beispiel 1a, Toleranz $1E-9$	41
3.36	$x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 1a	41
3.37	Fehler Richardson für ERK 4 in Beispiel 2a, Toleranz $1E-6$	42
3.38	Fehler Richardson für ERK 4 in Beispiel 2a, Toleranz $1E-9$	42
3.39	$x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 2a	43
3.40	Fehler Richardson für ERK 4 in Beispiel 3a, Toleranz $1E-6$	43
3.41	Fehler Richardson für ERK 4 in Beispiel 3a, Toleranz $1E-9$	44
3.42	$x_j h_j$ -Plot Richardson für ERK 4 für Beispiel 3a	44
3.43	Fehler Richardson für IRK 5 in Beispiel 1a, Toleranz $1E-6$	46
3.44	Fehler Richardson für IRK 5 in Beispiel 1a, Toleranz $1E-9$	46
3.45	$x_j h_j$ -Plot Richardson für IRK 5 für Beispiel 1a	47
3.46	Fehler Richardson für IRK 5 in Beispiel 2a, Toleranz $1E-6$	47
3.47	Fehler Richardson für IRK 5 in Beispiel 2a, Toleranz $1E-9$	48
3.48	$x_j h_j$ -Plot Richardson für IRK 5 für Beispiel 2a	48
3.49	Ordnungskurven lokaler Fehler ERK 4 für Beispiel 2b	59
3.50	Ordnungskurven lokaler Fehler IRK 4 für Beispiel 2b	59
3.51	Fehler ERK 4 exakt in Beispiel 1a, Toleranz $1E-6$	66

3.52 Fehler ERK 4 exakt in Beispiel 1a, Toleranz $1E-9$	66
3.53 $x_j h_j$ -Plot ERK 4 exakt für Beispiel 1a	67
3.54 Fehler ERK 4 exakt in Beispiel 2a, Toleranz $1E-6$	67
3.55 Fehler ERK 4 exakt in Beispiel 2a, Toleranz $1E-9$	68
3.56 $x_j h_j$ -Plot ERK 4 exakt für Beispiel 2a	68
4.1 MAPLE Programm zur Berechnung des lokalen Fehlers von ERK	71
4.2 MAPLE Programm zur Berechnung des lokalen Fehlers von IRK	80
5.1 Fehler Zadunaisky für ERK 4 in Beispiel 1a	91
5.2 Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-3}$	91
5.3 Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-4}$	92
5.4 Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-5}$	92
5.5 Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-6}$	93
5.6 Qualität Zadunaisky für ERK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-7}$	93
5.7 Fehler Zadunaisky für ERK 4 in Beispiel 1b	94
5.8 Qualität Zadunaisky für ERK 4 in Beispiel 1b, $h = 1/5 \cdot 2^{-3}$	94
5.9 Fehler Zadunaisky für ERK 4 in Beispiel 2a	95
5.10 Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-3}$	95
5.11 Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-4}$	96
5.12 Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-5}$	96
5.13 Qualität Zadunaisky für ERK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-6}$	97
5.14 Fehler Zadunaisky für ERK 4 in Beispiel 2b	98
5.15 Qualität Zadunaisky für ERK 4 in Beispiel 2b, $h = 1/5 \cdot 2^{-3}$	98
5.16 Fehler Zadunaisky für ERK 4 in Beispiel 3a	99
5.17 Qualität Zadunaisky für ERK 4 in Beispiel 3a, $h = 1/5 \cdot 2^{-3}$	99
5.18 Fehler Zadunaisky für IRK 4 in Beispiel 1a	103
5.19 Qualität Zadunaisky für IRK 4 in Beispiel 1a, $h = 1/5 \cdot 2^{-3}$	103
5.20 Fehler Zadunaisky für IRK 4 in Beispiel 1b	104
5.21 Qualität Zadunaisky für IRK 4 in Beispiel 1b, $h = 1/5 \cdot 2^{-3}$	104
5.22 Fehler Zadunaisky für IRK 4 in Beispiel 2a	105
5.23 Qualität Zadunaisky für IRK 4 in Beispiel 2a, $h = 1/5 \cdot 2^{-3}$	105
5.24 Fehler Zadunaisky für IRK 4 in Beispiel 2b	106
5.25 Qualität Zadunaisky für IRK 4 in Beispiel 2b, $h = 1/5 \cdot 2^{-3}$	106
5.26 Fehler Zadunaisky für IRK 4 in Beispiel 3a	107
5.27 Qualität Zadunaisky für IRK 4 in Beispiel 3a, $h = 1/5 \cdot 2^{-3}$	107

Literatur

- [1] W. AUZINGER, P. TRIEB, E. WEINMÜLLER, *Beschleunigte Algorithmen zur Lösung von singulären Randwertaufgaben*, Bericht Nr. 113(1994) des Institutes für Angewandte und Numerische Mathematik, Technische Universität Wien, Wien.
- [2] C. Y. CHAN, S. W. DU, *A Constructive Method for the Thomas-Fermi Equation*, Quart. Appl. Math. 44(1986), pp. 303–307.
- [3] C. Y. CHAN, Y. C. HON, *A Constructive Solution for a Generalized Thomas-Fermi Theory of Ionized Atoms*, Quart. Appl. Math. 45(1987), pp. 591–599.
- [4] M. M. CHAWLA, C. P. KATTI, *A Uniform Mesh Finite Difference Method for a Class of Singular Two-Point Boundary Value Problems*, SIAM J. Numer. Anal. 22(1985), pp. 561–565.
- [5] M. M. CHAWLA, C. P. KATTI, *Finite Difference Methods and Their Convergence for a Class of Singular Two-Point Boundary Value Problems*, Numer. Math. 39(1982), pp. 341–350.
- [6] E. J. DOEDEL, G. W. REDDIEN, *Finite Difference Methods for Singular Two-Point Boundary Value Problems*, SIAM J. Numer. Anal. 21(1984), pp. 300–313.
- [7] J. R. DORMAND, P. J. PRINCE, *A family of embedded Runge-Kutta formulae*, J. Comp. a. Appl. Math. 6(1980), pp. 19–26.
- [8] J. R. DORMAND, *Numerical Methods for Differential Equations - A Computational Approach*, CRC Press (1996).
- [9] R. FRANK, *The method of Iterated Defect-Correction and its application to two-point boundary value problems, Part I*, Numer. Math. 25(1976), pp. 409–419.
- [10] R. FRANK, *The method of Iterated Defect-Correction and its application to two-point boundary value problems, Part II*, Numer. Math. 27(1977), pp. 407–420.
- [11] F. FROMMLET, E. WEINMÜLLER *Asymptotic Error Expansions for Singular BVP's*, submitted to M³AS (Richard Weiß Memorial Issue).
- [12] M. GRÄFF, E. WEINMÜLLER, *Schätzungen des lokalen Diskretisierungsfehlers bei singulären Randwertproblemen*. Bericht Nr. 66(1986) des Institutes für Angewandte und Numerische Mathematik, Technische Universität Wien, Wien.
- [13] M. GRÄFF, E. WEINMÜLLER, *Testing of Error Estimation Procedures for Boundary Value Problems with a Singularity of the First Kind*, Proceedings of the 1986 Annual Seminar “Oscillation, Bifurcation, Chaos”, 13.–25. July 1986, Toronto, Canada.
- [14] R. D. GRIGORIEFF, *Numerik gewöhnlicher Differentialgleichungen, Band 1*, Teubner Verlag (1972).
- [15] C. GROSSMANN, *Enclosures and Semi-Analytic Discretization of Boundary Value Problems*, Num. Anal. a. Math. Mod. 29(1994), pp. 283–304.

- [16] C. GROSSMANN, *Enclosures of the solution of the Thomas-Fermi equation by monotone discretization*, J. Comp. Phys. 98(1992), pp. 26–38.
- [17] B. GUSTAFSSON, *A Numerical Method for Solving Singular Boundary Value Problems*, Numer. Math. 21(1973), pp. 328–344.
- [18] E. HAIRER, S. P. NORSETT, G. WANNER, *Solving Ordinary Differential Equations I*, Springer series in Computational Mathematics 8, Springer Verlag Berlin Heidelberg (1987).
- [19] E. HAIRER, G. WANNER, *Solving Ordinary Differential Equations II*, Springer series in Computational Mathematics 14, Springer Verlag Berlin Heidelberg (1991).
- [20] F. R. DE HOOG, R. WEISS, *Collocation Methods for Singular Boundary Value Problems*, SIAM J. Numer. Anal. 15(1978), pp. 198–217.
- [21] F. R. DE HOOG, R. WEISS, *Difference Methods for Boundary Value Problems with a Singularity of the First Kind*, SIAM J. Numer. Anal. 13(1976), pp. 775–813.
- [22] F. R. DE HOOG, R. WEISS, *On the Boundary Value Problem for Systems of Ordinary Differential Equations with a Singularity of the Second Kind*, SIAM J. Math. Anal. 11(1980), pp. 41–60.
- [23] F. R. DE HOOG, R. WEISS, *The Application of Linear Multistep Methods to Singular Initial Value Problems*, Math. Comp. 31(1977), pp. 676–690.
- [24] F. R. DE HOOG, R. WEISS, *The Application of Runge-Kutta Schemes to Singular Initial Value Problems*, Math. Comp. 44(1985), pp. 93–103.
- [25] F. R. DE HOOG, R. WEISS, *The numerical Solution of Boundary Value Problems with an Essential Singularity*, SIAM J. Numer. Anal. 16(1979), pp. 637–669.
- [26] P. JAMET, *On the Convergence of Finite Difference Approximations to One-Dimensional Singular Boundary Value Problems*, Numer. Math. 14(1970), pp. 335–378.
- [27] H. B. KELLER, *Numerical Methods for Two-Point Boundary Value Problems*, Blaisdell Publishing Company (1968).
- [28] O. KOCH, P. KOFLER, E. WEINMÜLLER, *On the Initial Value Problems for Systems of Ordinary First and Second Order Differential Equations with a Singularity of the First Kind*, submitted to SIAM J. Numer. Anal.
- [29] P. KOFLER, E. WEINMÜLLER, *Numerical Treatment of Singular Boundary and Initial Value Problems*, submitted for Proceedings of RAAM 1996, Kuwait, Vorabdruck: Bericht Nr. 121(1996) des Institutes für Angewandte und Numerische Mathematik, Technische Universität Wien, Wien.
- [30] P. KOFLER, *Theorie und numerische Lösung singulärer Anfangswertaufgaben gewöhnlicher Differentialgleichungen mit einer Singularität erster Art*, Dissertation, Institutes für Angewandte und Numerische Mathematik, Technische Universität Wien, Wien 1998.

- [31] M. METCALF, J. REID, *Fortran 90 Explained*, Oxford University Press (1990).
- [32] F. NATTERER, *Das Differenzenverfahren für singuläre Rand-Eigenwertaufgaben gewöhnlicher Differentialgleichungen*, Numer. Math. 23(1975), pp. 387–409.
- [33] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, B. P. FLANNERY, *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing*, Cambridge University Press (1996).
- [34] P. RENTROP, *Eine Taylorreihenmethode zur numerischen Lösung von Zwei-Punkt Randwertproblemen mit Anwendung auf singuläre Probleme der nichtlinearen Schalentheorie*, TUM Rep. 7733, Institut für Mathematik, München (1977).
- [35] P. RENTROP, *Numerical solution of the singular Ginzburg-Landau equations by multiple shooting*, Computing 16(1976), pp. 61–67.
- [36] R. D. RUSSEL, L. F. SHAMPINE, *A Collocation Method for Boundary Value Problems*, Numer. Math. 19(1972), pp. 1–28.
- [37] R. D. RUSSEL, L. F. SHAMPINE, *Numerical Methods for Singular Boundary Value Problems*, SIAM J. Numer. Anal. 12(1975), pp. 13–36.
- [38] L. F. SHAMPINE, H. A. WATTS, *Global Error Estimation for Ordinary Differential Equations*, ACM Trans. Math. Soft. 2(1976), pp. 172–186.
- [39] R. D. SKEEL, *Thirteen Ways to Estimate Global Error*, Numer. Math. 48(1986), pp. 1–20.
- [40] H. J. STETTER, *The Defect Correction Principle and Discretization Methods*, Numer. Math. 29(1978), pp. 425–443.
- [41] E. WEINMÜLER, *A Difference Method for a Singular Boundary Value Problems of Second Order*, Math. Comp. 42(1984), pp. 441–464.
- [42] E. WEINMÜLER, *On the Numerical Solution of Singular Boundary Value Problems of Second Order by a Difference Method*, Math. Comp. 46(1986), pp. 93–117.
- [43] E. WEINMÜLER, *Stability of Singular Boundary Value Problems and Their Discretization by Finite Differences*, SIAM J. Numer. Anal. 26(1989), pp. 180–213.
- [44] P. E. ZADUNAISKY, *On the Estimation of Errors Propagated in the Numerical Integration of Ordinary Differential Equations*, Numer. Math. 27(1976), pp. 21–39.

Aktuelle Publikationen aus Gebieten der angewandten Wissenschaften

- [45] P. BAUMAN, D. PHILLIPS, Q. TANG, *On the onset of superconductivity for the Ginzburg-Landau equations*, Z. Angewandte Mathematik und Mechanik 76(1996), pp. 277–279.

- [46] T.W. CARR, T. ERNEUX, *Understanding the bifurcation to traveling waves in a class-B laser using a degenerate Ginzburg-Landau equation*, Physical Review A 50(1994), pp. 4219–4227.
- [47] J. FLECKINGER-PELLE, H.G. KAPER, *Gauges for the Ginzburg-Landau equations of superconductivity*, Z. Angewandte Mathematik und Mechanik 76(1996), pp. 345–348.
- [48] G.W. PARKER, *Solution of the Thomas-Fermi equation for molecules by an efficient relaxation method*, J. Math. Phys. 31(1990), pp. 2535–2537.
- [49] Y. REN, J.-H. XU, C.S. TING, *Ginzburg-Landau equations for mixed $s+d$ symmetry superconductors*, Physical Review B 53(1996), pp. 2249–2252.
- [50] N. REZLESCU, M. AGOP, C. BUZEA, C.G. BUZEA, *Perturbative solutions of the Ginzburg-Landau equation and the superconducting parameters*, Physical Review B 53(1996), pp. 2229–2232.
- [51] R. SCHREIBER, *Finite Element Methods of High-Order Accuracy for Singular Two-Point Boundary Value Problems with Nonsmooth Solutions*, SIAM J. Numer. Anal. 17(1980), pp. 547–566.
- [52] R. SCHREIBER, *Regularity of Singular Two-Point Boundary Value Problems*, SIAM J. Math. Anal. 12(1981), pp. 104–109.
- [53] F. SIRINGO, G. PICCITTO, R. PUCCI, *Thomas-Fermi model for the C_{60} molecule*, Physical Review E 48(1993), pp. 263–272.
- [54] K. STALIUNAS, *Laser Ginzburg-Landau equation and laser hydrodynamics*, Physical Review A 48(1993), pp. 1573–1581.
- [55] W. XU, Y. REN, C.S. TING, *Ginzburg-Landau equations for a d -wave superconductor with nonmagnetic impurities*, Physical Review B 53(1996), pp. 12481–12495.