D I S S E R T A T I O N

Defektkorrektur zur numerischen Lösung steifer Anfangswertprobleme

ausgeführt zum Zweck der Erlangung des
akademischen Grades eines Doktors der
technischen Wissenschaften

unter der Leitung von

Ao. Univ. Prof. Dr. Ewa Weinmüller
Institut für Angewandte und Numerische Mathematik

eingereicht an der

Technischen Universität Wien
Technisch-Naturwissenschaftliche Fakultät

von

Harald Hofstätter
Hirschfeldspitz 50, 7100 Neusiedl am See

Neusiedl am See, im Mai 2000

Inhaltsverzeichnis

Kapitel 1

Einleitung

Zur numerischen Lösung steifer Differentialgleichungen werden in der Praxis hauptsächlich implizite Mehrschrittverfahren vom BDF[1]-Typ verwendet. Gegenüber den impliziten Runge-Kutta-Verfahren (IRK-Verfahren) haben die BDF-Verfahren den Vorteil, daß bei ihrer Anwendung auf ein n-dimensionales Anfangswertproblem in jedem Integrationsschritt lediglich ein algebraisches Gleichungssystem der Dimension n zu lösen ist. Bei voll-impliziten Runge-Kutta-Verfahren hingegen, z.B. bei den Gauß- oder RadauIIA-Verfahren, beträgt diese Dimension $n \times s$, wobei hier mit s die Anzahl der Stufen des IRK-Verfahrens bezeichnet wird.

Dennoch sind gewisse IRK-Verfahren den Mehrschrittverfahren in wesentlichen Belangen überlegen:

- Es gibt IRK-Verfahren mit optimalen Stabilitätseigenschaften von beliebig hoher Ordnung. So ist für jede Stufenanzahl s das entsprechende Gauß- bzw. RadauIIA-Verfahren sowohl A- als auch B-stabil mit einer (Super-) Konvergenzordnung $p = 2s$ bzw. $p = 2s - 1$.[2] Hingegen sind die BDF-Verfahren nur bis zur Ordnung 2 A-stabil und nur bis zur Ordnung 6 $A(\alpha)$-stabil.

- Wegen ihrer Natur als Einschritt-Verfahren ist eine Änderung der Schrittweite bei den IRK-Verfahren unproblematisch, was sowohl Rechenaufwand als auch Stabilität betrifft.

[1] BDF ist eine Abkürzung für *backward differentiation formula*. Für Details zu diesen Verfahren siehe z.B. [9].

[2] Für gewisse steife Modellprobleme beobachtet man eine Reduktion dieser hohen Konvergenzordnung, vgl. [12]. Neuere Untersuchungen auf die wir später (vgl. Abschnitt 1.2.4) noch näher eingehen werden, ergeben aber, daß in vielen Fällen dennoch die volle klassische Ordnung beobachtet werden kann.

Diese Eigenschaften lassen insbesondere die Gauß- und RadauIIA-Verfahren als attraktive Kandidaten zum Lösen von steifen Differentialgleichungen erscheinen, vorausgesetzt, es gelingt, die oben erwähnten algebraischen Gleichungen der Dimension $n \times s$ auf effiziente Weise zu lösen.

Gauß- und RadauIIA-Verfahren lassen sich als Kollokationsverfahren auffassen. Mit Kollokationsverfahren gewonnene Lösungen treten in bestimmten Fällen als Fixpunkte von Iterationen von Defektkorrekturverfahren auf, die so realisiert werden können, daß in jedem Integrationsschritt nur algebraische Gleichungen der Dimension n gelöst werden müssen (bei Verwendung eines einstufigen Basisverfahrens wie z.B. des impliziten Eulerverfahrens). Somit stellt sich die Frage nach Möglichkeiten, auch Gauß- und RadauIIA-Verfahren durch Defektkorrekturalgorithmen zu realisieren.

1.1 Iterierte Defektkorrektur (IDeC)

Der IDeC-Algorithmus basiert auf einer Idee von Zadunaisky [15], seine Anwendung auf steife Anfangswertprobleme wurde zuerst in [8] untersucht.

1.1.1 Beschreibung des IDeC-Algorithmus[3]

Gegeben sei das Anfangswertproblem

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

wobei $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. Wir wählen eine Intervallänge H, eine ganze Zahl $m \geq 2$ und setzen $h := H/m$. Auf dem äquidistanten Gitter $t_\nu = t_0 + \nu \cdot h$ berechnen wir mit Hilfe eines geeigneten Basisverfahrens (z.B. implizites Eulerverfahren) eine „Basisapproximation“ $\eta_\nu \approx y(t_\nu)$ für die exakte Lösung $y(t)$ von (1.1). Diese η_ν werden zusammen mit y_0 durch ein Polynom $P^{[0]}(t)$ vom Grad m interpoliert, sodaß $P^{[0]}(t_\nu) = \eta_\nu$ und $P^{[0]}(t_0) = y_0$ gilt. Mit Hilfe des Defekts

$$d^{[0]}(t) := \frac{d}{dt} P^{[0]}(t) - f(t, P^{[0]}(t)) \tag{1.2}$$

[3] In dieser Einleitung geben wir nur vereinfachte Beschreibungen der jeweiligen Defektkorrekturalgorithmen, wobei insbesondere jeweils nur das erste Integrationsintervall der Länge H betrachtet wird. Für Details wie z.B. die Fortsetzung auf die weiteren Intervalle, siehe die entsprechenden Abschnitte in Kapitel 2.

Um den Text nicht mit Phrasen wie „für alle $\nu = 1, \ldots, m$“ zu überladen, vereinbaren wir, daß in dieser Einleitung jede Gleichung, in der der Index ν vorkommt, so zu verstehen ist, daß diese Gleichung für alle $\nu = 1, \ldots, m$ gilt.

bilden wir das sogenannte Nachbarproblem

$$y' = f(t,y) + d^{[0]}(t), \quad y(t_0) = y_0, \tag{1.3}$$

dessen exakte Lösung $P^{[0]}(t)$ ist. Dieses Nachbarproblem wird nun mit Hilfe des Basisverfahrens gelöst, was eine Approximation $\pi_\nu^{[0]} \approx P^{[0]}(t_\nu)$ ergibt. Der explizit berechenbare globale Fehler $\pi_\nu^{[0]} - P^{[0]}(t_\nu)$ des Basisverfahrens für das Nachbarproblem wird als Schätzung für den unbekannten ursprünglichen globalen Fehler $\eta_\nu - y(t_\nu)$ in der Identität

$$y(t_\nu) = \eta_\nu - (\eta_\nu - y(t_\nu)) \tag{1.4}$$

verwendet, was eine verbesserte Approximation

$$\eta_\nu^{[1]} := \eta_\nu - (\pi_\nu^{[0]} - P^{[0]}(t_\nu)) \tag{1.5}$$

für $y(t_\nu)$ ergibt. Dieser Vorgang kann nun iterativ fortgesetzt werden: $P^{[1]}(t)$ interpoliert die $\eta_\nu^{[1]}$; damit wird neuer Defekt $d^{[1]}(t)$ und damit neues Nachbarproblem definiert; darauf angewendetes Basisverfahren liefert neue Werte $\pi_\nu^{[1]}$ und damit eine nochmals verbesserte Approximation $\eta_\nu^{[2]} := \eta_\nu - (\pi_\nu^{[1]} - P^{[1]}(t_\nu))$ für $y(t_\nu)$, usw.

1.1.2 Fixpunkteigenschaft der Kollokationslösung

Das Kollokationspolynom bezüglich der t_ν als Kollokationsknoten ist eindeutig als jenes Polynom $P^*(t)$ vom Grad m definiert, welches die Bedingungen

$$P^*(t_0) \;=\; y_0, \tag{1.6}$$

$$\frac{d}{dt}P^*(t_\nu) - f(t_\nu, P^*(t_\nu)) \;=\; 0, \quad \nu = 1,\ldots,m \tag{1.7}$$

erfüllt. Wir zeigen nun, daß das Kollokationspolynom $P^*(t)$ die folgende Fixpunkteigenschaft bezüglich der IDeC-Iteration hat: Wählt man zur Berechnung der $\eta_\nu^{[k+1]}$ die Werte $\eta_\nu^{[k]}$ als

$$\eta_\nu^{[k]} := P^*(t_\nu), \tag{1.8}$$

so gilt $\eta_\nu^{[k+1]} = \eta_\nu^{[k]}$.

Denn für das die $\eta_\nu^{[k]}$ interpolierende Polynom $P^{[k]}(t)$ gilt natürlich $P^{[k]}(t) \equiv P^*(t)$, und damit für den Defekt

$$d^{[k]}(t) \;:=\; \frac{d}{dt}P^{[k]}(t) - f(t, P^{[k]}(t)) \;=\; \frac{d}{dt}P^*(t) - f(t, P^*(t)). \tag{1.9}$$

Aus (1.7) folgt daher

$$d^{[k]}(t_\nu) \;=\; 0, \quad \nu = 1,\ldots,m. \tag{1.10}$$

Für das implizite Eulerverfahren ist die rechte Seite $f(t,y)$ des Originalproblems bzw. $f(t,y)+d^{[k]}(t)$ des Nachbarproblems nur für die Werte $t \in \{t_1,\ldots,t_m\}$ von Relevanz. Für diese t-Werte gilt aber $f(t,y) = f(t,y)+d^{[k]}(t)$. Daher ergibt die Anwendung des impliziten Eulerverfahrens auf das Nachbarproblem $y' = f(t,y)+d^{[k]}(t)$, $y(t_0)=y_0$ die bei der Anwendung des impliziten Eulerverfahrens auf das Originalproblem (1.1) gewonnene Lösung: $\pi_\nu^{[k]} = \eta_\nu$. Daraus folgt

$$\eta_\nu^{[k+1]} \;:=\; \eta_\nu - (\pi_\nu^{[k]} - P^{[k]}(t_\nu)) \;=\; P^{[k]}(t_\nu) \;=\; \eta_\nu^{[k]}, \tag{1.11}$$

d.h. die Fixpunkteigenschaft $\eta_\nu^{[k+1]} = \eta_\nu^{[k]}$ der durch (1.8) definierten Kollokationslösung $(\eta_1^{[k]},\ldots,\eta_m^{[k]})$.

1.1.3 Verwendung der IDeC zur Realisierung des Kollokationsverfahrens

Für einfache steife Modelle wurde in [8] die schnelle Konvergenz der oben beschriebenen Defektkorrekturiteration zu ihrem Fixpunkt d.h. zur Kollokationslösung gezeigt. Damit ergibt sich eine attraktive Möglichkeit, diese Kollokationslösung zu gewinnen, da sich der IDeC-Algorithmus sehr effizient implementieren läßt: Die Berechnung des Defekts gelingt durch Verwendung von Gewichtsmatrizen (für Details siehe Abschnitt 2.4.1) mit wenig Aufwand, und wenn zur Lösung der bei den jeweiligen impliziten Euler-Schritten auftretenden n-dimensionalen algebraischen Gleichungssysteme ein vereinfachtes Newton-Verfahren verwendet wird, genügt in vielen Fällen die LU-Zerlegung einer einzigen $(n \times n)$-Matrix. Im Gegensatz dazu wäre zur direkten Auflösung der Kollokationsgleichungen die LU-Zerlegung einer $(mn \times mn)$-Matrix notwendig.

1.1.4 Grenzen der IDeC

Die als Fixpunkte der IDeC auftretenden Kollokationsverfahren basieren auf äquidistanten Kollokationsknoten. Diese Verfahren sind nicht superkonvergent (auch nicht im nichtsteifen Fall), ihre Konvergenzordnungen sind durch $p = s$ begrenzt. Hier bezeichnet s die Anzahl der Stufen des als IRK-Verfahren aufgefaßten Kollokationsverfahrens, sie ist gleich dem Grad m des oben beschriebenen Kollokationspolynoms $P^*(t)$. Außerdem sind diese Verfahren für größere s auch nicht mehr A-stabil, dafür aber $A(\alpha)$-stabil mit relativ großen Winkeln α, vgl. [8].

Die superkonvergenten Kollokationsverfahren vom Typ Gauß bzw. RadauIIA basieren hingegen auf nichtäquidistanten Kollokationsknoten. Eine naheliegende Möglichkeit, diese Verfahren als Fixpunkte von Defektkorrekturiterationen zu gewinnen, ist, das Basisverfahren auf den entsprechenden nichtäquidistanten

Gittern arbeiten zu lassen. Hier wird praktisch in jedem Integrationsschritt des Basisverfahrens die Schrittweite h geändert, was zu großen Schwierigkeiten führt: Es genügt jetzt nicht mehr die LU-Zerlegung einer einzigen $(n \times n)$-Matrix, sondern man benötigt für die symmetrischen Gauß-Knoten bzw. für die nichtsymmetrischen RadauIIA-Knoten die LU-Zerlegung von $\lfloor m/2 \rfloor + 1$ bzw. m $(n \times n)$-Matrizen. Schwerer wiegt aber, daß die Defektkorrekturiteration, wie die numerische Erfahrung zeigt, auf nichtäquidistanten Gittern selbst im nichtsteifen Fall meist nur schlecht, oft aber auch gar nicht gegen ihren Fixpunkt konvergiert.[4]

1.2 Iterierte Interpolierte Defektkorrektur (IIDeC)

Um die Schwierigkeiten der IDeC im Fall von nichtäquidistanten Gittern zu überwinden, wurde in [13] eine modifizierte Version der Defektkorrektur vorgeschlagen. Wie sich herausstellte, ist diese Modifikation sehr eng mit der interpolierten Defektkorrektur (IIDeC) verwandt, die auf einer Idee von W. Auzinger, R. Frank und E. Weinmüller basiert, und die im Zusammenhang mit skalaren Randwertproblemen in [10] beschrieben wurde, vgl. auch [1].

1.2.1 Beschreibung des IIDeC-Algorithmus[5]

Zusätzlich zu den Größen H und m, mit deren Hilfe in Abschnitt 1.1 das t_ν-Gitter definiert wurde, seien jetzt die auf das Intervall $[0, 1]$ normierten (i.a. nichtäquidistanten) Kollokationsknoten $c_1, \ldots, c_m$ gegeben, mit deren Hilfe wir durch $\tau_\nu = t_0 + c_\nu H$ ein zusätzliches Gitter definieren.

Der einzige kleine aber wesentliche Unterschied der interpolierten Defektkorrektur (IIDeC) gegenüber der klassischen IDeC ist nun, daß wir das Nachbarproblem nicht wie in (1.3) mit Hilfe des Defekts $d^{[0]}(t)$ aufstellen, sondern mit Hilfe des Polynoms $D^{[0]}(t)$ vom Grad $m-1$, das den Defekt $d^{[0]}(t)$ an den Knoten des τ_ν-Gitters interpoliert, d.h.

$$D^{[0]}(\tau_\nu) = d^{[0]}(\tau_\nu) = \frac{d}{dt} P^{[0]}(\tau_\nu) - f(\tau_\nu, P^{[0]}(\tau_\nu)). \tag{1.12}$$

Im ersten Defektkorrekturschritt wird also statt (1.3) jetzt

$$y' = f(t, y) + D^{[0]}(t), \quad y(t_0) = y_0 \tag{1.13}$$

[4] Das Verhältnis ist eher umgekehrt: oft schlecht, meistens gar nicht.

[5] vgl. Fußnote 3.

als Nachbarproblem aufgestellt und mit Hilfe des (auf dem äquidistanten t_ν-Gitter arbeitenden!) Basisverfahrens gelöst. Diese Lösung wird wieder mit $\pi_\nu^{[0]}$ bezeichnet. Alle anderen Details bleiben gegenüber der klassischen IDeC unverändert, insbesondere die Definition (1.5) der $\eta_\nu^{[1]}$. Die weiteren Defektkorrekturschritte werden analog wie bei der IDeC iterativ gewonnen, die Vorgangsweise dazu ist offensichtlich.

1.2.2 Fixpunkteigenschaft der Kollokationslösung

Das zu den Kollokationsknoten τ_ν gehörige Kollokationspolynom $P^*(t)$ vom Grad m ist nun durch

$$P^*(t_0) \;=\; y_0, \tag{1.14}$$

$$\frac{d}{dt}P^*(\tau_\nu) - f(\tau_\nu, P^*(\tau_\nu)) \;=\; 0, \quad \nu = 1, \ldots, m \tag{1.15}$$

definiert. Wenn wir

$$\eta_\nu^{[k]} := P^*(t_\nu) \tag{1.16}$$

setzen, so gilt $P^{[k]}(t) \equiv P^*(t)$ für das die $\eta_\nu^{[k]}$ interpolierende Polynom $P^{[k]}(t)$. Daher gilt für den Defekt

$$d^{[k]}(t) \;:=\; \frac{d}{dt}P^{[k]}(t) - f(t, P^{[k]}(t)) \;=\; \frac{d}{dt}P^*(t) - f(t, P^*(t)), \tag{1.17}$$

woraus wegen (1.15)

$$d^{[k]}(\tau_\nu) \;=\; 0, \quad \nu = 1, \ldots, m \tag{1.18}$$

folgt. Wir erhalten somit das Nullpolynom, wenn wir $d^{[k]}(t)$ an den Stellen τ_ν interpolieren, d.h. $D^{[k]}(t) \equiv 0$. Also sind jetzt Originalproblem (1.1) und Nachbarproblem $y' = f(t,y) + D^{[k]}(t),\ y(t_0) = y_0$ identisch gleich, das Basisverfahren liefert für beide dieselbe Lösung, d.h. $\pi_\nu^{[k]} = \eta_\nu$, woraus

$$\eta_\nu^{[k+1]} \;:=\; \eta_\nu - (\pi_\nu^{[k]} - P^{[k]}(t_\nu)) \;=\; P^{[k]}(t_\nu) \;=\; \eta_\nu^{[k]}, \tag{1.19}$$

d.h. die Fixpunkteigenschaft $\eta_\nu^{[k+1]} = \eta_\nu^{[k]}$ der durch (1.16) definierten Kollokationslösung $(\eta_1^{[k]}, \ldots, \eta_m^{[k]})$ folgt.

1.2.3 Verwendung der IIDeC zur Realisierung des Kollokationsverfahrens

Unter der Voraussetzung, daß die IIDeC-Iteration zu ihrem Fixpunkt d.h. zur Kollokationslösung konvergiert, kann der IIDeC-Algorithmus ähnlich effizient wie

der IDeC-Algorithmus implementiert werden, d.h. es genügt bei Verwendung des impliziten Eulerverfahrens als Basisverfahren, wobei die algebraischen Gleichungen mit einem vereinfachten Newton-Verfahren gelöst werden, die LU-Zerlegung einer einzigen $(n \times n)$-Matrix, wieder im Gegensatz zu der LU-Zerlegung einer $(mn \times mn)$-Matrix, die zur direkten Auflösung der Kollokationsgleichungen notwendig wäre.

Was nun die Konvergenz betrifft, so wurden im Rahmen dieser Dissertation umfangreiche numerische Experimente durchgeführt, von denen einige in den Abschnitten 3.4 und 3.5 beschrieben werden. Zusammenfassend kann gesagt werden, daß i.a. die IIDeC für jene steife Probleme gut funktioniert, für die die steifen Eigenrichtungen der Jacobi-Matrix hinreichend langsam variieren. Für eindimensionale Probleme (d.h. $n = 1$) ist das natürlich immer der Fall.

Für $n = 2$ entspricht das im linearen Fall Anfangswertproblemen der Gestalt

$$y' = A(t)y + g(t), \quad y(t_0) = y_0 \tag{1.20}$$

mit

$$A(t) = X(t)\Lambda(t)X^{-1}(t) = \begin{bmatrix} | & | \\ x_1(t) & x_2(t) \\ | & | \end{bmatrix} \begin{bmatrix} -\frac{c_1(t)}{\varepsilon} & \\ & c_2(t) \end{bmatrix} \begin{bmatrix} | & | \\ x_1(t) & x_2(t) \\ | & | \end{bmatrix}^{-1}, \tag{1.21}$$

wobei $c_1(t) \geq C_1$ mit einer Konstanten $C_1 > 0$ und

$$\frac{d^k}{dt^k} x_1(t) = O(\varepsilon), \quad k \geq 1. \tag{1.22}$$

Die $c_i(t)$ und $x_2(t)$ sind hinreichend glatte Funktionen mit moderaten, vom steifen Parameter $0 < \varepsilon \ll 1$ unabhängigen Ableitungen.

Im nichtlinearen Fall haben z.B. Differentialgleichungen vom singulär gestörten Typ, also

$$\begin{aligned} y_1'(t) &= f(t, y_1(t), y_2(t)), \\ \varepsilon y_2'(t) &= g(t, y_1(t), y_2(t)), \end{aligned} \tag{1.23}$$

wobei $\frac{\partial g}{\partial y_2} \leq G_2$ mit einer Konstanten $G_2 < 0$, die Eigenschaft, daß die steife Eigenrichtung der Jakobimatrix nur wie $O(\varepsilon)$ variiert. Eine Analyse der Konvergenzeigenschaften von IRK-Verfahren angewendet auf singulär gestörte Differentialgleichungen findet man in [9] und [11].

1.2.4 Grenzen der IIDeC

Wenn wir die Forderung nach der langsamen Variation der steifen Eigenrichtungen fallen lassen, wenn wir also z.B. für das Anfangswertproblem (1.20), (1.21)

statt (1.22) nur noch $\frac{d^k}{dt^k}x_1(t) = O(1)$ fordern, dann treten bei der Anwendung der IIDeC in vielen Fällen große Probleme auf, die IIDeC zeigt ein divergentes Verhalten, vgl. Abschnitt 3.5. Somit ist für solche steife Differentialgleichungen die IIDeC keine geeignete Lösungsmethode. Das ist nun gerade deshalb sehr bedauerlich, weil die Forderung, daß die steifen Eigenrichtungen der Jacobi-Matrix wie $O(1)$ und nicht nur wie $O(\varepsilon)$ variieren dürfen, mit der Tatsache vereinbar ist, daß die Differentialgleichung in jene Klasse steifer Systeme fällt, die durch die neue nichtlineare Konvergenztheorie für IRK-Verfahren abgedeckt wird, welche am Institut für Angewandte und Numerische Mathematik der Technischen Universität Wien entwickelt wurde. In dieser Theorie, zu deren genauen Inhalt auf [3], [4] und [6] verwiesen sei, wird der globale Diskretisierungsfehler in „glatte“ bzw. „steife“ Komponenten zerlegt, wobei für das stark stabile RadauIIA-Verfahren für die glatte Fehlerkomponente Schranken der Gestalt $\max(O(\varepsilon h^s), O(h^{2s-1}))$ und für die steife Komponente Schranken der Gestalt $O(\varepsilon h^s)$ hergeleitet werden. (Hier bezeichnet h die Schrittweite des RadauIIA-Verfahrens, s die Stufenanzahl, und $\varepsilon > 0$ ist ein kleiner Parameter, durch den die Steifheit des Problems charakterisiert wird.) Im stark steifen Fall, wo ε hinreichend klein im Vergleich zu h ist, bedeutet das, daß die klassische Ordnung $p = 2s - 1$ des RadauIIA-Verfahrens wirklich beobachtet werden kann.[6]

1.3 Mit QR-Verfahren kombinierte Iterierte Interpolierte Defektkorrektur (QR-IIDeC)

Im Laufe dieser Dissertation wurde eine vielversprechende Möglichkeit gefunden, die Schwierigkeiten der IIDeC mit „schnell“ (d.h. wie $O(1)$) variierenden steifen Eigenrichtungen zu überwinden. Die grundlegende Idee dazu ist, den Defekt $d^{[0]}(t)$ nicht bezüglich des kartesischen Standardkoordinatensystems des $\mathbb{R}^n$ zu interpolieren, sondern bezüglich eines solchen, wo die steifen Eigenvektoren der Jacobi-Matrix Basisvektoren sind. Da sich die steifen Eigenrichtungen nun von Punkt zu Punkt (bei autonomen nichtlinearen Systemen) bzw. von t-Wert zu t-Wert (bei linearen Systemen mit nichtkonstanten Koeffizienten) ändern, tun das auch die lokalen Basisvektoren dieses Koordinatensystems.

Anhand des Anfangswertproblems (1.20), für dessen Jacobi-Matrix die Eigenvektoren aus (1.21) bekannt sind, wollen wir diese Idee näher erläutern: Als neues Koordinatensystem nehmen wir das $\{x_1(t), x_2(t)\}$-Koordinatensystem, der steife Eigenvektor $x_1(t)$ ist Basisvektor. Ausgehend von der ersten Approximation η_ν für die exakte Lösung von (1.20) bilden wir wie bei der IIDeC das Polynom $P^{[0]}(t)$ und damit den Defekt $d^{[0]}(t)$. Der Defekt $d^{[0]}(t)$ wird nun aber nicht an

[6] Hierauf bezieht sich die Fußnote 2.

den τ_ν-Stellen interpoliert, sondern zuerst im $\{x_1(t), x_2(t)\}$-Koordinatensystem dargestellt:

$$\tilde{d}^{[0]}(t) := X^{-1}(t) \cdot d^{[0]}(t). \tag{1.24}$$

Diese Darstellung $\tilde{d}^{[0]}(t)$ wird nun an den τ_ν-Stellen durch ein Polynom $\tilde{D}^{[0]}(t)$ vom Grad $m-1$ interpoliert, d.h. $\tilde{D}^{[0]}(\tau_\nu) = \tilde{d}^{[0]}(\tau_\nu)$. Zur Aufstellung des Nachbarproblems

$$y' = A(t)y + g(t) + D^{[0]}(t), \quad y(t_0) = y_0. \tag{1.25}$$

brauchen wir für den im $\{x_1(t), x_2(t)\}$-Koordinatensystem dargestellten interpolierten Defekt $\tilde{D}^{[0]}(t)$ die Darstellung

$$D^{[0]}(t) := X(t) \cdot \tilde{D}^{[0]}(t). \tag{1.26}$$

bezüglich des Standardkoordinatensystems des $\mathbb{R}^2$. Mit Ausnahme der Definition der Funktionen $D^{[k]}(t)$ bleiben alle weiteren algorithmischen Details gegenüber der IIDeC unverändert.

Numerische Experimente zeigen für die so modifizierte interpolierte Defektkorrektur ein sehr befriedigendes Verhalten, wenn sie auf Probleme der Gestalt (1.20), (1.21) angewendet wird, bei denen $x_1(t)$ wie $O(1)$ variiert, vgl. Kapitel 4.

Allerdings sind in der Praxis die Jacobi-Matrizen selten in faktorisierter Form wie in (1.21) gegeben. Es stellt sich daher die Frage nach einer effizienten algorithmischen Umsetzung der oben beschriebenen Idee. Eine Möglichkeit bietet, wie wir sehen werden, das QR-Verfahren zur Berechnung von Eigenwerten und Eigenvektoren einer Matrix, an das wir nun im speziellen Fall einer (2×2)-Matrix erinnern:[7]

1.3.1 QR-Verfahren

Gegeben sei eine Matrix $A \in \mathbb{R}^{2\times 2}$. Damit definieren wir eine Folge $\{A_k\}$ von Matrizen durch

$$A_0 := A, \tag{1.27}$$
$$A_{k+1} := R_k \cdot Q_k, \quad k = 0, 1, 2, \ldots, \tag{1.28}$$

wobei die orthogonale Matrix Q_k und die obere Dreiecksmatrix R_k jeweils durch die QR-Zerlegung von A_k,

$$A_k = Q_k \cdot R_k, \tag{1.29}$$

konstruiert werden. Weiters definieren wir die Matrixfolge $\{\tilde{Q}_k\}$ durch

$$\tilde{Q}_k := Q_0 \cdot Q_1 \cdots Q_k, \quad k = 0, 1, 2, \ldots. \tag{1.30}$$

[7] Für Details zum QR-Verfahren, siehe z.B. [5].

Unter bestimmten Voraussetzungen konvergiert nun die Folge $\{A_k\}$ gegen eine obere Dreiecksmatrix R und die Folge $\{\tilde{Q}_k\}$ gegen eine orthogonale Matrix Q, sodaß

$$A = QRQ^T = \begin{bmatrix} | & | \\ q_1 & q_2 \\ | & | \end{bmatrix} \begin{bmatrix} \lambda_1 & r_{12} \\ & \lambda_2 \end{bmatrix} \begin{bmatrix} | & | \\ q_1 & q_2 \\ | & | \end{bmatrix}^T \tag{1.31}$$

mit $|\lambda_1| > |\lambda_2|$ gilt, wobei λ_1 und λ_2 die dem Betrag nach geordneten Eigenwerte von A sind.[8] Wichtig ist nun, daß dabei die $(2,1)$-Einträge der Matrizen A_k wie $\left|\frac{\lambda_2}{\lambda_1}\right|^k$ für $k \to \infty$ gegen 0 konvergieren. In dem uns interessierenden Fall, wo für den steifen Eigenwert $|\lambda_1| = O(1/\varepsilon)$ und für den nichtsteifen Eigenwert $|\lambda_2| = O(1)$ gilt, konvergieren die $(2,1)$-Einträge der Matrizen A_k also wie ε^k gegen 0, was für $\varepsilon \ll 1$ eine sehr rasche Konvergenz des QR-Verfahrens bedeutet.

1.3.2 Kombination der IIDeC mit dem QR-Verfahren

Die QR-IIDeC basiert nun auf der Annahme, daß man im steifen Fall schon nach einem Schritt des QR-Verfahrens,

d.h. mit Hilfe einer einzigen QR-Zerlegung von A, mit hinreichender Genauigkeit einen Eigenvektor der Matrix A zum steifen Eigenwert λ_1 erhält. Sei also durch

$$A(t) = Q(t) \cdot R(t) = \begin{bmatrix} | & | \\ q_1(t) & q_2(t) \\ | & | \end{bmatrix} \cdot \begin{bmatrix} r_{11}(t) & r_{12}(t) \\ & r_{22}(t) \end{bmatrix} \tag{1.32}$$

eine stetige QR-Zerlegung für die Matrix $A(t)$ aus (1.20) gegeben, wobei die $q_i(t)$ und die $r_{ij}(t)$ stetige Funktionen von t sind.[9] Nach unserer Annahme ist hier die erste Spalte $q_1(t)$ von $Q(t)$ mit hinreichender Genauigkeit ein Eigenvektor von $A(t)$ zum steifen Eigenwert $\lambda_1(t)$. Im Fall der QR-IIDeC wird nun der (genauso wie bei der IIDeC mit Hilfe der ersten Approximation η_ν definierte) Defekt $d^{[0]}(t)$

[8] Eine Zerlegung der Form (1.31) heißt „Schur-Zerlegung“ von A.

[9] Eine solche stetige QR-Zerlegung ist nicht eindeutig bestimmt, was schon deswegen nicht der Fall sein kann, weil auch für eine feste (d.h. nicht von t abhängige) Matrix A die QR-Zerlegung nicht eindeutig bestimmt ist: Wenn $A = Q_0 \cdot R_0$ eine QR-Zerlegung von A ist, dann sind alle anderen QR-Zerlegungen von A durch

$$A = \underbrace{(Q_0 D)}_{Q} \cdot \underbrace{(D R_0)}_{R}, \quad D \in \left\{ \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \right\} \tag{1.33}$$

gegeben. Man kann aber durch eine geeignete Strategie für die Wahl der QR-Zerlegung (1.32) an einer festen Stelle t sicherstellen, daß damit insgesamt eine geeignete eindeutig bestimmte stetige QR-Zerlegung von $A(t)$ gegeben ist, für Details siehe Abschnitt 4.2.4.

im durch (1.32) gegebenen $\{q_1(t), q_2(t)\}$-Koordinatensystem dargestellt:

$$\tilde{d}^{[0]}(t) := Q^T(t) \cdot d^{[0]}(t). \tag{1.34}$$

Diese Darstellung $\tilde{d}^{[0]}(t)$ wird nun an den τ_ν-Stellen durch ein Polynom $\tilde{D}^{[0]}(t)$ vom Grad $m-1$ interpoliert, d.h. $\tilde{D}^{[0]}(\tau_\nu) = \tilde{d}^{[0]}(\tau_\nu)$. Zur Aufstellung des Nachbarproblems

$$y' = A(t)y + g(t) + D^{[0]}(t), \quad y(t_0) = y_0. \tag{1.35}$$

brauchen wir für den im $\{q_1(t), q_2(t)\}$-Koordinatensystem dargestellten interpolierten Defekt $\tilde{D}^{[0]}(t)$ die Darstellung

$$D^{[0]}(t) := Q(t) \cdot \tilde{D}^{[0]}(t). \tag{1.36}$$

bezüglich des Standardkoordinatensystems des $\mathbb{R}^2$. Mit Ausnahme der Definition der Funktionen $D^{[k]}(t)$ bleiben alle weiteren algorithmischen Details gegenüber der IIDeC unverändert.

1.3.3 Zu hoher Rechenaufwand bei der QR-IIDeC (?)

Numerische Experimente mit Problemen der Gestalt (1.20) zeigen ein vielversprechendes Verhalten der QR-IIDeC.Dennoch scheint der hohe Rechenaufwand wegen der vielen notwendigen QR-Zerlegungen ein großes Problem der QR-IIDeC zu sein. Zur Reduktion dieses Aufwands bietet es sich an, die Matrizen $Q(t)$ nur noch für die Stellen t_ν mit Hilfe von QR-Zerlegungen zu berechnen, die ebenfalls benötigten Matrizen $Q(\tau_\nu)$ hingegen durch eine geeignete Interpolation der $Q(t_\nu)$ zu gewinnen.

Zur Lösung der algebraischen Gleichungen des impliziten Euler-Verfahrens benötigt man eine Zerlegung der Matrix $I - hA(t_\nu)$. Damit stellt sich die Frage, ob es nicht auch möglich ist, mit Hilfe der einer QR-Zerlegung von $A(t_\nu)$ diese algebraischen Gleichungen zu lösen. Die Antwort auf diese Frage ist offen, scheint aber negativ zu sein. Da die Matrizen $I - hA(t)$ und $A(t)$ dieselben Eigenvektoren haben, bietet es sich nun an, die QR-IIDeC so zu modifizieren, daß zur Definition der Matrizen $Q(t_\nu)$ nicht QR-Zerlegungen der Matrizen $A(t_\nu)$ berechnet werden, sondern stattdessen QR-Zerlegungen der Matrizen $I - hA(t_\nu)$. Diese Zerlegungen können dann auch für die Lösung der jeweiligen algebraischen Gleichungen des impliziten Euler-Verfahrens verwendet werden. Bei Anwendung auf eine Matrix $I - hA(t)$ reduziert sich zwar die schnelle $O(\varepsilon^k)$-Konvergenz des QR-Verfahrens auf $O(\varepsilon^k/h^k)$, aber für ε hinreichend klein im Vergleich zu h bedeutet das noch immer eine sehr rasche Konvergenz des QR-Verfahrens, sodaß auch für diese Variante der QR-IIDeC bei Anwendung auf Probleme der Gestalt (1.20) ein vielversprechendes Verhalten festzustellen ist.

1.4 Ausblick auf die weiteren Kapitel

Im nun folgenden 2. Kapitel werden die in dieser Einleitung nur vereinfacht beschriebenen Defektkorrekturalgorithmen IDeC und IIDeC exakt spezifiziert, wobei diese Algorithmen insofern verallgemeinert werden, daß nicht nur das Implizite Eulerverfahren sondern auch andere IRK-Verfahren als Basisverfahren betrachtet werden. Im 3. Kapitel werden Varianten der Interpolierten Defektkorrektur auf lineare Anfangswertprobleme angewendet, wobei insbesondere der skalare Fall studiert wird. Mit Hilfe dieses speziellen Falls gewinnen wir einen Eindruck davon, welche IRK-Verfahren für die Verwendung als Basisverfahren der Interpolierten Defektkorrektur geeignet sind. Wie wir sehen werden, erzielt man mit Hilfe eines speziellen 2-stufigen diagonalimpliziten Verfahrens als Basisverfahren die besten Ergebnisse. Dieses Basisverfahren betrachten wir daher in unseren weiteren Untersuchungen. Im letzten Abschnitt des 3. Kapitels führen wir numerische Experimente durch, die das in Abschnitt 1.2.4 erwähnte instabile Verhalten der Interpolierten Defektkorrektur im Fall einer variierenden steifen Eigenrichtung demonstrieren. Dieses instabile Verhalten wird im 4. Kapitel näher untersucht, weiters wird dort die QR-IIDeC exakt spezifiziert, mit der die Schwierigkeiten der Interpolierten Defektkorrektur mit einer variierenden steifen Eigenrichtung überwunden werden können, was wir anhand von numerischen Experimenten demonstrieren, die den Abschluß unserer Untersuchungen bilden.

1.5 Resümee

In dieser Dissertation werden erste Untersuchungen von verschiedenen Varianten der Interpolierten Defektkorrektur im Zusammenhang mit deren Anwendung auf steife Anfangswertprobleme durchgeführt. Dabei gewinnen wir Erkenntnisse darüber, welche Varianten zur Lösung steifer Anfangswertproblem besser geeignet sind und welche weniger gut geeignet sind. Insbesondere wird in numerischen Experimenten festgestellt, daß Schwierigkeiten auftreten, wenn die Interpolierte Defektkorrektur auf Anfangswertprobleme mit einer variierenden steifen Eigenrichtung angewendet wird. Zur Überwindung dieser Schwierigkeiten schlagen wir den QR-IIDeC-Algorithmus, eine Kombination der Interpolierten Defektkorrektur mit dem QR-Verfahren zur Bestimmung von Eigenvektoren vor. Erste numerische Experimente anhand von einfachen Testproblemen deuten auf ein vielversprechendes Verhalten dieses Algorithmus auch bei Anwendung auf Probleme mit einer variierenden steifen Eigenrichtung hin. Inwieweit dieser Algorithmus auch in allgemeineren Fällen anwendbar ist, muß durch weitere Untersuchungen noch geklärt werden.

Kapitel 2

Defektkorrekturalgorithmen

In diesem Kapitel spezifizieren wir die im 1. Kapitel nur vereinfacht beschriebenen Defektkorrekturalgorithmen, insbesondere IDeC und IIDeC. Eine genaue Behandlung der QR-IIDeC erfolgt erst in Kapitel 4. Zunächst legen wir einige Bezeichnungen fest, die im folgenden verwendet werden.

2.1 Definitionen und Bezeichnungen

2.1.1 Das numerisch zu lösende Problem

Gegeben sei das Anfangswertproblem

$$y' = f(t, y), \quad y(t_0) = y_0 \tag{2.1}$$

mit $f : [t_0, t_{end}] \times \mathcal{G} \to \mathbb{R}^n$ und $\mathcal{G} \subseteq \mathbb{R}^n$. Wir setzen voraus, daß $f(t, y)$ auf $[t_0, t_{end}] \times \mathcal{G}$ stetig und dort bezüglich y Lipschitz-stetig ist. Damit ist die Existenz einer eindeutig bestimmten exakten Lösung $y(t)$ von (2.1) für $t \in [t_0, t_{end}]$ sichergestellt.

Gesucht ist eine numerische Approximation für diese exakte Lösung $y(t)$. Die von uns betrachteten Defektkorrekturalgorithmen liefern numerische Approximationen für $y(t)$ auf einem auf dem Intervall $[t_0, t_{end}]$ definierten äquidistanten Gitter Γ_h, das wie folgt definiert ist:

2.1.2 Äquidistantes Gitter

Wir unterteilen das Integrationsintervall $[t_0, t_{end}]$ in N Teilintervalle der Länge $H := (t_{end} - t_0)/N$:

$$[t_0, t_{end}] = [T_0, T_1] \cup [T_1, T_2] \cup \ldots [T_{N-1}, T_N], \tag{2.2}$$

wobei

$$T_\ell := t_0 + \ell H, \quad \ell = 0, \ldots, N. \tag{2.3}$$

In jedem dieser Teilintervalle $[T_{\ell-1}, T_\ell]$ fügen wir $m+1$ äquidistante Punkte

$$t_{\ell,\nu} := T_\ell + \nu h, \quad \ell = 0, \ldots, N-1, \quad \nu = 0, \ldots, m, \quad h := \frac{H}{m} = \frac{t_{end} - t_0}{N \cdot m} \tag{2.4}$$

ein, und definieren mit Hilfe dieser Punkte das äquidistante Gitter

$$\begin{aligned} \Gamma_h &:= \{t_{\ell,\nu} : \ \ell = 0, \ldots N-1, \ \nu = 0, \ldots, m\} \\ &= \{t_0 + \nu h : \ \nu = 0, \ldots, N \cdot m\} \end{aligned} \tag{2.5}$$

auf $[t_0, t_{end}]$.

Weiters sei

$$\mathcal{E}_h := \{\eta_h : \ \Gamma_h \to \mathbb{R}^n\} \tag{2.6}$$

der Raum der auf diesem Gitter definierten Gitterfunktionen. Wir schreiben $\eta_{\ell,\nu}$ statt $\eta_h(t_{\ell,\nu})$ für $\eta_h \in \mathcal{E}_h$. Wegen $t_{\ell-1,m} = T_\ell = t_{\ell,0}$ setzen wir dabei immer $\eta_{\ell-1,m} = \eta_{\ell,0}, \ \ell = 1, \ldots, N-1$ voraus. Die Einführung des Raumes $\mathcal{E}_h$ dient hauptsächlich der bequemeren Schreibweise, im wesentlichen handelt es sich dabei einfach um den Raum $(\mathbb{R}^n)^{m \cdot N+1} \cong \mathbb{R}^{n \cdot (m \cdot N+1)}$.

Unter einer Approximation für die exakte Lösung $y(t)$ von (2.1) auf dem Gitter Γ_h verstehen wir ein Element $\eta_h \in \mathcal{E}_h$ mit $\eta_h(t_0) = y_0$, für das $\eta_{\ell,\nu} = \eta_h(t_{\ell,\nu}) \approx y(t_{\ell,\nu}), \ \ell = 0, \ldots, N-1, \ \nu = 0, \ldots, m$ gelten soll.

2.1.3 Raum stetiger stückweiser Polynomfunktionen

Es gibt einen bijektiven Zusammenhang zwischen dem Raum $\mathcal{E}_h$ der Gitterfunktionen auf Γ_h und dem Raum $\mathcal{P}_h$ aller stetigen stückweisen Polynomfunktionen $p : [t_0, t_{end}] \to \mathbb{R}^n$ der Gestalt

$$p(t) \ = \ \begin{cases} p_0(t) & \text{für } t \in [T_0, T_1], \\ p_1(t) & \text{für } t \in (T_1, T_2], \\ & \vdots \\ p_{N-1} & \text{für } t \in (T_{N-1}, T_N], \end{cases} \tag{2.7}$$

wobei die $p_\ell(t), \ \ell = 0, \ldots, N-1$ Polynome vom Grad m mit

$$p_\ell(T_\ell) \ = \ p_{\ell-1}(T_\ell), \quad \ell = 1, \ldots, N-1 \qquad \text{(d.h. } p(t) \text{ ist stetig)} \tag{2.8}$$

sind. Dieser ist durch den Interpolationsoperator

$$P_h : \mathcal{E}_h \to \mathcal{P}_h, \quad \eta_h \mapsto P_h \eta_h, \tag{2.9}$$

wobei für $p(t) \equiv (P_h \eta_h)(t)$ das entsprechende $p_\ell(t)$ in (2.7) jeweils als das Interpolationspolynom von $(t_{\ell,0}, \eta_{\ell,0}), \ldots, (t_{\ell,m}, \eta_{\ell,m})$ definiert ist, und den dazu inversen Einschränkungsoperator

$$P_h^{-1} : \mathcal{P}_h \to \mathcal{E}_h, \quad p \mapsto p|_{\Gamma_h} \tag{2.10}$$

gegeben.

An den Stellen $t = T_\ell,\ \ell = 1, \ldots, N-1$ sind die $p(t) \in \mathcal{P}_h$ i.a. nicht differenzierbar. Wir machen daher die

Vereinbarung 2.1.1. *Für $p(t) \in \mathcal{P}_h$ der Form (2.7) ist $p'(T_\ell),\ \ell = 1, \ldots, N-1$ als die linksseitige Ableitung $p'_{\ell-1}(T_\ell)$ zu interpretieren. Ebenso interpretieren wir an der rechten Endstelle $T_N = t_{end}$, $p'(T_N)$ als die linksseitige Ableitung $p'_{N-1}(T_N)$.*

2.1.4 Kollokationsgitter

Für die interpolierte Defektkorrektur (IIDeC und QR-IIDeC) benötigen wir zusätzlich zum äquidistanten Gitter Γ_h ein weiteres i.a. nichtäquidistantes Gitter

$$\widehat{\Gamma}_h \ = \ \{\tau_{\ell,\nu} : \ \ell = 0, \ldots N-1,\ \nu = 1, \ldots, m\}, \tag{2.11}$$

bestehend aus den Kollokationspunkten $\tau_{\ell,\nu}$ jenes Kollokationsverfahrens, das der jeweilige Defektkorrekturalgorithmus als Fixpunkt haben soll. Dazu sei eine feste Menge $\{\gamma_1, \ldots, \gamma_m\}$ von m Zahlen mit[1]

$$0 < \gamma_1 < \gamma_2 < \ldots < \gamma_m \leq 1 \tag{2.12}$$

gegeben, mit deren Hilfe die Kollokationspunkte durch

$$\tau_{\ell,\nu} := T_\ell + \gamma_\nu H, \quad \ell = 0, \ldots N-1,\ \nu = 1, \ldots, m \tag{2.13}$$

definiert werden.

Von besonderem Interesse sind die „klassischen“ Kollokationsverfahren Gauß und RadauIIA. Dazu wählt man die Abszissen γ_j in (2.12) als die auf das Intervall $[0,1]$ transformierten Stützstellen des Gauß- bzw. Radau-Quadraturverfahrens, d.h. man definiert im Fall des Gauß-Verfahrens die $\gamma_j,\ j = 1, \ldots, m$, als die m paarweise verschiedenen, der Größe nach geordneten Nullstellen des auf das Intervall $[0,1]$ transformierten m-ten Legendre-Polynoms

$$\widehat{P}_m(x) \ := \ \frac{1}{m!} \frac{d^m}{dx^m} (x^m (x-1)^m). \tag{2.14}$$

[1] Die Bedingung $\gamma_1 > 0$ statt $\gamma_1 \geq 0$ ist nicht wesentlich. Sie dient hier lediglich dazu, die Vereinbarung 2.1.1 zu ermöglichen und so die Schreibweise zu vereinfachen. Für alle von uns betrachteten Mengen $\{\gamma_1, \ldots, \gamma_m\}$ ist diese Bedingung erfüllt.

Im Fall des RadauIIA-Verfahrens wählt man die entsprechenden Nullstellen des Polynoms

$$\widehat{P}_m(x) - \widehat{P}_{m-1}(x). \tag{2.15}$$

2.1.5 Kollokationslösung

Das zum Gitter $\widehat{\Gamma}_h$ gehörige Kollokationspolynom ist nun als jene stetige stückweise Polynomfunktion $P^*(t) \in \mathcal{P}_h$ definiert, welche die Anfangswertbedingung

$$P^*(T_0) \;=\; y_0, \tag{2.16}$$

und für jedes $\tau_{\ell,\nu} \in \widehat{\Gamma}_h$ die Kollokationsbedingung[2]

$$\frac{d}{dt}P^*(\tau_{\ell,\nu}) - f(\tau_{\ell,\nu}, P^*(\tau_{\ell,\nu})) \;=\; 0 \tag{2.17}$$

erfüllt.[3]

Als (zum Gitter $\widehat{\Gamma}_h$ gehörige) Kollokationslösung bezeichnen wir jene Gitterfunktion $\eta_h^* \in \mathcal{E}_h$, die durch

$$\eta_h^*(t_{\ell,\nu}) \;=\; \eta^*_{\ell,\nu} \;=\; P^*(t_{\ell,\nu}), \quad t_{\ell,\nu} \in \Gamma_h \tag{2.18}$$

d.h. $\eta_h^* = P^*|_{\Gamma_h}$, gegeben ist.

2.1.6 Basisverfahren

Bei den verschiedenen Varianten der iterativen Defektkorrektur ist, wie in Abschnitt 2.2 genauer ausgeführt wird, sowohl im Basisschritt als auch in den einzelnen Defektkorrekturschritten jeweils die numerische Lösung von (2.1) bzw. eines zu (2.1) benachbarten Problems mit Hilfe eines geeigneten Basisverfahrens erforderlich. Dieses Basisverfahren, für das sich theoretisch jedes (implizite) Runge-Kutta-Verfahren (IRK-Verfahren) einsetzen läßt,[4] legen wir durch Angabe seines Butcher-Arrays

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \quad = \quad \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} \quad . \tag{2.19}$$

[2]Wenn in (2.12) $\gamma_m = 1$ ist, dann gilt hier für $\tau_{\ell,m} = T_{\ell+1},\ \ell = 0, \ldots, N-1$ die Vereinbarung 2.1.1.

[3]Wir setzen voraus, daß ein solches $P^*(t) \in \mathcal{P}_h$ existiert und eindeutig bestimmt ist.

[4]In der Praxis wird man sich natürlich auf IRK-Verfahren mit wenigen Stufen (d.h. $s = 1$ oder $s = 2$) und hinreichend guten Stabilitäts- und Konvergenzeigenschaften beschränken. Für Details zu IRK-Verfahren siehe [9].

fest. In einem Integrationsschritt $(t_\nu, \eta_\nu) \mapsto (t_\nu + h, \eta_{\nu+1})$ mit Schrittweite h wird dabei aus einer gegebenen Approximation η_ν für die exakte Lösung $y(t_\nu)$ von (2.1) an der Stelle $t = t_\nu$ gemäß der Formel

$$\eta_{\nu+1} \;:=\; \eta_\nu + h\big(b_1 f(t_\nu + c_1 h, Y_1) + \ldots + b_s f(t_\nu + c_s h, Y_s)\big) \tag{2.20}$$

eine Approximation $\eta_{\nu+1}$ für die exakte Lösung $y(t_\nu + h)$ an der Stelle $t = t_\nu + h$ berechnet, wobei die „Stufen“ Y_i implizit durch das i.a. nichtlineare[5] $(n \times s)$-dimensionale Gleichungssystem

$$\begin{aligned} Y_1 &= \eta_\nu + h\big(a_{11} f(t_\nu + c_1 h, Y_1) + \ldots + a_{1s} f(t_\nu + c_s h, Y_s)\big), \\ Y_2 &= \eta_\nu + h\big(a_{21} f(t_\nu + c_1 h, Y_1) + \ldots + a_{2s} f(t_\nu + c_s h, Y_s)\big), \\ &\vdots \\ Y_s &= \eta_\nu + h\big(a_{s1} f(t_\nu + c_1 h, Y_1) + \ldots + a_{ss} f(t_\nu + c_s h, Y_s)\big) \end{aligned} \tag{2.21}$$

definiert sind.

Vereinbarung 2.1.2. *Falls für das Basisverfahren (2.19) $c_1 = 0$ gilt und $f(t, y)$ die Gestalt $f(t, y) = \tilde{f}(t, y) + \delta(t)$ hat, wobei $\delta(t)$ an der Stelle $t = t_\nu$ nicht rechtsseitig stetig ist, dann ist in (2.20) und (2.21) $f(t_\nu + c_1 h, Y_1) = \tilde{f}(t_\nu, Y_1) + \delta(t_\nu)$ durch den rechtsseitigen Grenzwert $\tilde{f}(t_\nu, Y_1) + \lim_{t \to t_\nu +} \delta(t)$ zu ersetzen.*

Diese Vereinbarung ist notwendig, da die rechten Seiten $f(t, y)$ der in den einzelnen Defektkorrekturschritten zu lösenden Nachbarprobleme für gewisse $t = t_{\ell,\nu} \in \Gamma_h$ die genannte Unstetigkeit aufweisen.

Wenn wir das Basisverfahren auf dem äquidistanten Gitter Γ_h arbeiten lassen, werden die rechten Seiten der jeweiligen Differentialgleichungen nur für t-Werte der Form $t = t_{\ell,\nu} + c_i h$ ausgewertet. Alle diese t-Werte fassen wir zum Gitter $\widetilde{\Gamma}_h$ zusammen, d.h.

$$\widetilde{\Gamma}_h \;:=\; \{t_{\ell,\nu} + c_i h \;:\; \ell = 0, \ldots, N-1,\; \nu = 0, \ldots, m-1,\; i = 1, \ldots, s\}. \tag{2.22}$$

2.1.7 Beispiele von IRK-Verfahren, die zur Verwendung als Basisverfahren in Betracht kommen

Zur leichteren Referenzierung listen wir jene Basisverfahren auf, die wir in unseren Untersuchungen hauptsächlich betrachten werden.

Implizites Euler-Verfahren:

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}\;; \tag{2.23}$$

[5] Das Gleichungssystem ist genau dann linear wenn die Funktion $f(t, y)$ linear in y ist.

Implizite Mittelpunktregel (IMR):

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} ; \tag{2.24}$$

IMR2:

$$\begin{array}{c|cc} \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \tag{2.25}$$

(einem IMR2-Integrationsschritt mit Schrittweite h entsprechen zwei IMR-Integrationsschritte mit Schrittweite $h/2$);

Implizite Trapezregel (ITR):

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} ; \tag{2.26}$$

ITR2:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ 1 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \hline & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \tag{2.27}$$

(einem ITR2-Integrationsschritt mit Schrittweite h entsprechen zwei ITR-Integrationsschritte mit Schrittweite $h/2$);

2-stufiges diagonalimplizites Verfahren (SDIRK(2)):

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array} \quad \text{mit} \quad \gamma = 1 - \frac{\sqrt{2}}{2} \tag{2.28}$$

(dieses Verfahren ist stark A-stabil und hat Ordnung 2);

2-stufiges RadauIIA-Verfahren (RadauIIA(2)):

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} . \tag{2.29}$$

2.2 Bauart der hier betrachteten Defektkorrekturalgorithmen

Die iterative Defektkorrektur ist ein iteratives Verfahren, bei dem ausgehend von einer *Basisapproximation* $\eta_h^{[0]} \in \mathcal{E}_h$ schrittweise verbesserte Approximationen $\eta_h^{[1]}, \eta_h^{[2]}, \ldots \in \mathcal{E}_h$ für die exakte Lösung $y(t)$ von (2.1) berechnet werden. Die Anzahl $K \geq 1$ der Iterationen sei dabei im Vorhinein festgelegt.[6] Man kann dabei so vorgehen, daß die Werte sämtlicher $\eta_h^{[k]},\ k = 0, \ldots, K$ zuerst auf dem Intervall $[T_0, T_1]$ berechnet werden, dann die Werte aller $\eta_h^{[k]}$ auf dem Intervall $[T_1, T_2]$, und so weiter schrittweise für alle Intervalle $[T_\ell, T_{\ell+1}],\ \ell = 0, \ldots, N-1$.

2.2.1 Berechnung der $\eta_h^{[k]}$ im Intervall $[T_0, T_1]$

Die Rechenvorschriften für die einzelnen Defektkorrekturvarianten (IDeC, IIDeC und QR-IIDeC) im ersten Teilintervall $[T_0, T_1]$ wurden schon in der Einleitung beschrieben. Ihre allgemeine Bauart läßt sich wie folgt angeben:

Algorithmus 2.2.1 (Berechnung der $\eta_h^{[k]}$ im Intervall $[T_0, T_1]$)

Führe m Integrationsschritte

$$(t_0, y_0) \mapsto (t_{0,1}, \eta_{0,2}^{[0]}) \mapsto (t_{0,2}, \eta_{0,2}^{[0]}) \mapsto \ldots \mapsto (t_{0,m}, \eta_{0,m}^{[0]}) \tag{2.30}$$

des Basisverfahrens angewendet auf das Originalproblem (2.1) durch.

for $k := 0, 1, \ldots, K-1$ **do**

Bestimme Interpolationspolynom $P^{[k]}(t)$ vom Grad m mit

$$P^{[k]}(t_0) = y_0 \quad \text{und} \quad P^{[k]}(t_{0,\nu}) = \eta_{0,\nu}^{[k]}, \quad \nu = 1, \ldots, m. \tag{2.31}$$

Bilde Defekt:

$$d^{[k]}(t) := \frac{d}{dt} P^{[k]}(t) - f(t, P^{[k]}(t)). \tag{2.32}$$

Mit Hilfe des Defekts $d^{[k]}(t)$ definiere eine Störung $\delta^{[k]}(t)$ der Differentialgleichung (2.1) und stelle damit das k-te Nachbarproblem

$$y' = f(t, y) + \delta^{[k]}(t), \quad y(t_0) = y_0 \tag{2.33}$$

auf.

[6] In der Praxis wird sich diese Anzahl K im Lauf der Integration von (2.1) ändern dürfen, sie ist ein wichtiger Parameter zur Steuerung der Konvergenzordnung des Defektkorrekturalgorithmus.

Führe m Integrationsschritte

$$(t_0, y_0) \mapsto (t_{0,1}, \pi_{0,1}^{[k]}) \mapsto (t_{0,2}, \pi_{0,2}^{[k]}) \mapsto \ldots \mapsto (t_{0,m}, \pi_{0,m}^{[k]}) \tag{2.34}$$

des Basisverfahrens angewendet auf (2.33) durch.

Setze

$$\eta_{0,\nu}^{[k+1]} := \eta_{0,\nu}^{[0]} - (\pi_{0,\nu}^{[k]} - \eta_{0,\nu}^{[k]}), \quad \nu = 1, \ldots, m. \tag{2.35}$$

end for

Im Fall der klassischen IDeC ist hier die Störung $\delta^{[k]}(t)$ gleich dem Defekt, d.h. $\delta^{[k]}(t) \equiv d^{[k]}(t)$, im Fall der IIDeC ist $\delta^{[k]}(t)$ gleich dem interpolierten Defekt, d.h. $\delta^{[k]}(t) \equiv D^{[k]}(t)$, wobei $D^{[k]}(t)$ jenes Polynom vom Grad $m-1$ ist, für das $D^{[k]}(\tau_{0,\nu}) = d^{[k]}(\tau_{0,\nu}),\ \nu = 1, \ldots, m$ gilt.

2.2.2 Fortsetzung auf die weiteren Intervalle

Zur Fortsetzung der Berechnung der Werte von $\eta_h^{[k]}$ auf die weiteren Intervalle $[T_\ell, T_{\ell+1}],\ \ell = 1, \ldots, N-1$ bieten sich zwei Alternativen an:

1. Lokale Verbindungsstrategie: Hier interpretiert man die durch den Algorithmus 2.2 beschriebene Vorgangsweise als einen Integrationsschritt $(t_0, y_0) \mapsto (T_1, \eta_{0,m}^{[K]})$ eines Einschrittverfahrens mit Schrittweite H. Die Berechnung der $\eta_{\ell,\nu}^{[k]}$ im Intervall $[T_\ell, T_{\ell+1}]$ für $\ell \geq 1$ erfolgt dabei völlig analog zur Berechnung der $\eta_{0,\nu}^{[k]}$ im Intervall $[T_0, T_1]$, wobei die Rolle des Anfangswerts y_0 für $t = t_0$ nun die im vorangegangenen Intervall $[T_{\ell-1}, T_\ell]$ gewonnene letzte Approximation $\eta_{\ell-1,m}^{[K]}$ für die Stelle $t = T_\ell$ übernimmt. Da wir hauptsächlich die globale Verbindungsstrategie anwenden werden, soll diese kurze Beschreibung der lokalen Verbindungsstrategie genügen, Details entnehme man dem Algorithmus 2.4.1 in Abschnitt 2.4.

2. Globale Verbindungsstrategie: Diese Variante ist zur folgenden Vorgangsweise äquivalent:

- Im Basisschritt berechnen wir $\eta_h^{[0]}$ auf ganz Γ_h durch $N \times m$ Integrationsschritte des Basisverfahrens angewendet auf (2.1).

für $k = 0, 1, \ldots$**:**

- Die im Basisschritt ($k = 0$) bzw. im vorangegangenen Defektkorrekturschritt ($k \geq 1$) berechneten Werte von $\eta_h^{[k]}$ interpolieren wir durch eine stückweise Polynomfunktion $P^{[k]}(t) \in \mathcal{P}_h$, d.h. $P^{[k]}(t) := (P_h \eta_h^{[k]})(t)$.

- Wir bilden den Defekt[7]

$$d^{[k]}(t) := \frac{d}{dt}P^{[k]}(t) - f(t, P^{[k]}(t)), \tag{2.36}$$

mit dessen Hilfe wir eine Störung $\delta^{[k]}(t)$ der Differentialgleichung (2.1) definieren: Im Fall der klassischen Defektkorrektur (IDeC) setzen wir $\delta^{[k]}(t) \equiv d^{[k]}(t)$ und im Fall der interpolierten Defektkorrektur (IIDeC) definieren wir $\delta^{[k]}(t)$ stückweise durch

$$\delta^{[k]}(t) \;:=\; D^{[k]}(t) \;=\; \begin{cases} D_0^{[k]}(t) & \text{für } t \in [T_0, T_1], \\ D_1^{[k]}(t) & \text{für } t \in (T_1, T_2], \\ & \vdots \\ D_{N-1}^{[k]}(t) & \text{für } t \in (T_{N-1}, T_N], \end{cases} \tag{2.37}$$

wobei $D_\ell^{[k]}(t),\ \ell = 0, \ldots, N-1$ jeweils das durch $D_\ell^{[k]}(\tau_{\ell,\nu}) = d^{[k]}(\tau_{\ell,\nu}),\ \nu = 1, \ldots, m$ definierte Interpolationspolynom vom Grad $m-1$ ist.

- Mit Hilfe von $\delta^{[k]}(t)$ stellen wir das Nachbarproblem

$$y' = f(t, y) + \delta^{[k]}(t), \quad y(t_0) = y_0 \tag{2.38}$$

auf. Die rechte Seite dieses Anfangswertproblems ist an den Stellen $t = T_\ell,\ \ell = 1, \ldots, N-1$ i.a. unstetig (wegen der Unstetigkeit von $\delta^{[k]}(t)$ an diesen Stellen), daher ist i.a. auch die Ableitung der exakten Lösung von (2.38) an diesen Stellen unstetig. Dennoch läßt sich, da sämtliche Unstetigkeitsstellen Gitterpunkte von Γ_h sind, durch $N \times m$ Integrationsschritte unseres Basisverfahrens eine Approximation $\pi_h^{[k]} \in \mathcal{E}_h$ für diese exakte Lösung berechnen.[8]

- Die verbesserte Approximation $\eta_h^{[k+1]} \in \mathcal{E}_h$ für die exakte Lösung von (2.1) ist nun durch

$$\eta_h^{[k+1]} \;:= \eta_h^{[0]} - (\pi_h^{[k]} - \eta_h^{[k]}) \tag{2.39}$$

definiert.

2.2.3 Fixpunkteigenschaft der Kollokationslösung

Die Frage nach dem Fixpunkt der oben beschriebenen Defektkorrekturiteration $\eta_h^{[0]} \mapsto \eta_h^{[1]} \mapsto \eta_h^{[2]} \mapsto \ldots$ bei globaler Verbindungsstrategie ist für die klassische

[7] Hier gilt für $t = T_\ell,\ \ell = 1, \ldots, N$ die Vereinbarung 2.1.1.

[8] Für Basisverfahren (2.19), für die $c_1 = 0$ gilt (z.B. Implizite Trapezregel), kommt hier im Integrationsschritt $(T_\ell, \pi_{\ell,0}^{[k]}) \mapsto (t_{\ell,1}, \pi_{\ell,1}^{[k]})$ die Vereinbarung 2.1.2 zur Anwendung.

Defektkorrektur (IDeC) nur für spezielle Basisverfahren (z.B. Implizites Eulerverfahren, vgl. Abschnitt 1.1.2) einfach zu beantworten.

Im Fall der interpolierten Defektkorrektur (IIDeC) ist aber leicht zu sehen, daß durch die in Abschnitt 2.1.5 beschriebene Kollokationslösung η_h^* unabhängig vom verwendeten Basisverfahren ein Fixpunkt ist. Denn für $\eta_h^{[k]} := \eta_h^*$ gilt natürlich

$$P^{[k]}(t) \;:=\; (P_h \eta_h^{[k]})(t) \;=\; P^*(t).$$

und wegen der definierenden Eigenschaft (2.17) von $P^*(t)$ gilt

$$d^{[k]}(\tau_{\ell,\nu}) = \frac{d}{dt} P^{[k]}(\tau_{\ell,\nu}) - f(t, P^{[k]}(\tau_{\ell,\nu})) = \frac{d}{dt} P^*(\tau_{\ell,\nu}) - f(t, P^*(\tau_{\ell,\nu})) = 0$$

für $\ell = 0, \ldots, N-1$ und $\nu = 1, \ldots, m$. Die Polynome $D_\ell^{[k]}(t),\ \ell = 0, \ldots, N-1$ aus (2.37), die den Defekt $d^{[k]}(t)$ jeweils an den Stellen $\tau_{\ell,1}, \ldots, \tau_{\ell,m}$ interpolieren, sind daher alle gleich dem Nullpolynom, $D_\ell^{[k]}(t) \equiv 0$, und es gilt $\delta^{[k]}(t) = D^{[k]}(t) \equiv 0$. Nachbarproblem (2.38) und Originalproblem (2.1) sind somit identisch gleich und so auch deren jeweils mit dem Basisverfahren gewonnenen Lösungen $\pi_h^{[k]}$ bzw. $\eta_h^{[0]}$. Einsetzen von $\pi_h^{[k]} = \eta_h^{[0]}$ in (2.39) ergibt schließlich die Fixpunkteigenschaft $\eta_h^{[k+1]} = \eta_h^{[k]}$ der Kollokationslösung η_h^*.

2.3 Defektkorrektur nach Schild

In [13] wurde von K.H. Schild ein Algorithmus angegeben, der eng mit unserer Interpolierten Defektkorrektur (IIDeC) verwandt ist. In diesem Abschnitt wollen wir zunächst diesen Algorithmus in knappen Worten beschreiben (für Details siehe [13]) und anschließend seinen Zusammenhang mit der IIDeC aufzeigen.

2.3.1 Beschreibung des Schild-Verfahrens

Sei $\eta_h \in \mathcal{E}_h$. Für das interpolierende stückweise Polynom $p(t) \equiv (P_h \eta_h)(t)$ der Gestalt (2.7) gilt

$$\frac{1}{h} \cdot (\eta_{\ell,i} - \eta_{\ell,i-1}) \;=\; \frac{1}{h} \cdot \int_{t_{\ell,i-1}}^{t_{\ell,i}} p'(t)\, dt \;=\; \sum_{j=1}^{m} w_{i,j} \cdot p'(\tau_{\ell,j}), \tag{2.40}$$

mit

$$w_{i,j} \;:=\; m \cdot \int_{(i-1)/m}^{i/m} \prod_{\substack{k=1 \\ k \neq j}}^{m} \frac{t - \gamma_k}{\gamma_j - \gamma_k}\, dt. \tag{2.41}$$

Hier wurde in (2.40) die Tatsache verwendet, daß die Quadraturformel

$$\int_{t_{\ell,i-1}}^{t_{\ell,i}} f(t)\,dt \;\approx\; h \cdot \sum_{j=1}^{m} w_{i,j} \cdot f(\tau_{\ell,j}), \tag{2.42}$$

für Polynome $f(t)$ vom Grad $m-1$ exakt ist. Für η_h^* und $P^*(t)$ wie in Abschnitt 2.1.5 gilt nun

$$\begin{aligned}
\eta_h = \eta_h^* \;&\Leftrightarrow\; p(t) \equiv P^*(t) \\
&\Leftrightarrow\; p(t_0) = y_0 \;\text{ und }\; p'(\tau_{\ell,i}) - f(\tau_{\ell,i}, p(\tau_{\ell,i})) \;=\; 0, \\
&\qquad\qquad \ell = 0,\ldots,N-1,\; i = 1,\ldots,m \\
&\Leftrightarrow\; p(t_0) = y_0 \;\text{ und }\; \sum_{j=0}^{m} w_{i,j} \cdot \big(p'(\tau_{\ell,j}) - f(\tau_{\ell,j}, p(\tau_{\ell,j}))\big) \;=\; 0, \\
&\qquad\qquad \ell = 0,\ldots,N-1,\; i = 1,\ldots,m \\
&\Leftrightarrow\; \eta_{0,0} = y_0 \;\text{ und }\; \frac{1}{h} \cdot (\eta_{\ell,i} - \eta_{\ell,i-1}) - \sum_{j=0}^{m} w_{i,j} \cdot f(\tau_{\ell,j}, p(\tau_{\ell,j})) \;=\; 0, \\
&\qquad\qquad \ell = 0,\ldots,N-1,\; i = 1,\ldots,m.
\end{aligned}$$

Hier wurde für die vorletzte Äquivalenz die Regularität der Matrix $(w_{i,j})$, und für die letzte Äquivalenz (2.40) verwendet. Die Kollokationslösung η_h^* läßt sich daher als Lösung der Gleichung

$$F_h^* \eta_h^* = 0 \tag{2.43}$$

identifizieren, wobei

$$\begin{aligned}
&F_h^*\colon\; \mathcal{E}_h \to \mathcal{E}_h, \\
&(F_h^* \eta_h)_{0,0} := \eta_{0,0} - y_0, \\
&(F_h^* \eta_h)_{\ell,i} := \frac{1}{h} \cdot (\eta_{\ell,i} - \eta_{\ell,i-1}) - \sum_{j=0}^{m} w_{i,j} \cdot f(\tau_{\ell,j}, (P_h \eta_h)(\tau_{\ell,j})), \\
&\qquad\qquad \ell = 0,\ldots,N-1,\; i = 1,\ldots,m.
\end{aligned} \tag{2.44}$$

Dieser Operator F_h^* hat eine ähnliche Gestalt wie der Operator F_h der Impliziten Trapezregel:

$$\begin{aligned}
&F_h\colon\; \mathcal{E}_h \to \mathcal{E}_h, \\
&(F_h \eta_h)_{0,0} := \eta_{0,0} - y_0, \\
&(F_h \eta_h)_{\ell,i} := \frac{1}{h} \cdot (\eta_{\ell,i} - \eta_{\ell,i-1}) - \frac{1}{2} \cdot \big(f(t_{\ell,i}, \eta_{\ell,i}) + f(t_{\ell,i-1}, \eta_{\ell,i-1})\big), \\
&\qquad\qquad \ell = 0,\ldots,N-1,\; i = 1,\ldots,m.
\end{aligned} \tag{2.45}$$

Dies legt nahe, die Gleichung (2.43) iterativ, mit Hilfe von „Version B“ der iterierten Defektkorrektur nach Stetter [14] zu lösen,[9]

$$\begin{aligned} \eta_h^{[0]} &:= \text{Lösung von } F_h\eta_h^{[0]} = 0, \\ \eta_h^{[k]} &:= \text{Lösung von } F_h\eta_h^{[k]} = F_h\eta_h^{[k-1]} - F_h^*\eta_h^{[k-1]}, \quad k = 1, 2, 3, \ldots, \end{aligned} \tag{2.46}$$

was im wesentlichen dem in [13] beschriebenen Verfahren entspricht, wenn es auf Anfangswertprobleme der Gestalt (2.1) angewendet wird.

2.3.2 Zusammenhang mit der Interpolierten Defektkorrektur

Wenn das numerisch zu lösende Anfangswertproblem (2.1) linear ist, d.h. wenn die rechte Seite $f(t, y)$ die Gestalt $f(t, y) = A(t)y+g(t)$ hat, dann ist offensichtlich auch der Operator F_h linear, und $F_h\eta_h$ läßt sich als

$$F_h\eta_h \;=\; L_h\eta_h - b_h \tag{2.47}$$

mit $b_h \in \mathcal{E}_h$ und einem homogen-linearen Operator $L_h\colon\ \mathcal{E}_h \to \mathcal{E}_h$ darstellen. Für lineares F_h ist die Version B (2.46) mit der Version A der Defektkorrektur nach Stetter [14] identisch, die durch

$$\left.\begin{aligned} \eta_h^{[0]} &:= \text{Lösung von } F_h\eta_h^{[0]} = 0, \\ \pi_h^{[k-1]} &:= \text{Lösung von } F_h\pi_h^{[k-1]} = F_h^*\eta_h^{[k-1]}, \\ \eta_h^{[k]} &:= \eta_h^{[0]} - (\pi_h^{[k-1]} - \eta_h^{[k-1]}), \end{aligned}\right\} \quad k = 1, 2, 3, \ldots, \tag{2.48}$$

gegeben ist. Das folgt aus

$$\begin{aligned} F_h\pi_h^{[k-1]} &= F_h(-\eta_h^{[k]} + \eta_h^{[k-1]} + \eta_h^{[0]}) \\ &= -(L_h\eta_h^{[k]} - b_h) + (L_h\eta_h^{[k-1]} - b_h) + \underbrace{(L_h\eta_h^{[0]} - b_h)}_{=0} \\ &= -F_h\eta_h^{[k]} + F_h\eta_h^{[k-1]}, \end{aligned}$$

[9] Folgende Heuristik liegt dieser Version B zugrunde: Gesucht ist eine Lösung η_h^* von $F_h^*\eta_h^* = 0$, wobei die direkte Auflösung dieser Gleichung aufwendig ist. Wir haben aber ein einfacheres $F_h \approx F_h^*$ zur Verfügung. Damit können wir durch Lösen von $F_h\eta_h^{[0]} = 0$ eine Approximation $\eta_h^{[0]} \approx \eta_h^*$ berechnen. Es ist nun die Annahme plausibel, daß

$$F_h\eta_h^* - \underbrace{F_h^*\eta_h^*}_{=0} \approx F_h\eta_h^{[0]} - F_h^*\eta_h^{[0]}$$

gilt, wodurch wir erwarten können, daß die Lösung $\eta_h^{[1]}$ von

$$F_h\eta_h^{[1]} = F_h\eta_h^{[0]} - F_h^*\eta_h^{[0]}$$

eine verbesserte Approximation für η_h^* ist. Die iterative Fortsetzung dieser Idee führt schließlich zu (2.46).

also

$$\left.\begin{array}{l} F_h\pi_h^{[k-1]} = F_h^*\eta^{[k-1]} \quad \text{und} \\ \pi_h^{[k-1]} = -\eta_h^{[k]} + \eta_h^{[k-1]} + \eta_h^{[0]} \end{array}\right\} \quad \Leftrightarrow \quad -F_h\eta_h^{[k]} + F_h\eta_h^{[k-1]} = F_h^*\eta_h^{[k-1]},$$

d.h.

$$\text{die } \eta_h^{[k]} \text{ erfüllen (2.48)} \quad \Leftrightarrow \quad \text{die } \eta_h^{[k]} \text{ erfüllen (2.46).}$$

Offenbar hat das durch (2.48) gegebene Verfahren die in Abschnitt 2.2.2 beschriebene Bauart, wenn wir dort

1. die globale Verbindungsstrategie anwenden,
2. als Basisverfahren (2.19) die Implizite Trapezregel (2.26) verwenden, und
3. die Störung $\delta^{[k]}(t)$ in (2.38) so definieren, daß

$$(F_h^*\eta_h^{[k]})_{\ell,i} \;=\; \frac{1}{2}\cdot\big(\delta^{[k]}(t_{\ell,i}) + \delta^{[k]}(t_{\ell,i-1})\big),$$

bzw. (vgl. Vereinbarung 2.1.2)

$$(F_h^*\eta_h^{[k]})_{\ell,i} \;=\; \frac{1}{2}\cdot\big(\delta^{[k]}(t_{\ell,i}) + \lim_{t\to t_{\ell,i-1}+}\delta^{[k]}(t)\big) \tag{2.49}$$

gilt.

Am einfachsten ist (2.49) zu erreichen, wenn wir $\delta^{[k]}(t)$ stückweise konstant durch

$$\delta^{[k]}(t) \;:=\; (F_h^*\eta_h^{[k]})_{\ell,i} \text{ für } t\in(t_{\ell,i-1},t_{\ell,i}], \quad \ell=0,\ldots,N-1,\ i=1,\ldots,m \tag{2.50}$$

definieren. Sei nun $D^{[k]}(t)$ wie in (2.37) die stückweise Polynomfunktion vom Grad $m-1$, die den Defekt $d^{[k]}(t) := \frac{d}{dt}(P_h\eta_h^{[k]})(t) - f(t,(P_h\eta_h^{[k]})(t))$ an den Stellen $\tau_{\ell,\nu}\in\widehat{\Gamma}_h$ interpoliert. Unter Verwendung von (2.40) und der Tatsache, daß die Quadraturformel (2.42) exakt ist, wenn sie auf den interpolierten Defekt $D^{[k]}(t)$ angewendet wird, ergibt sich

$$\begin{aligned}(F_h^*\eta_h)_{\ell,i} &= \frac{1}{h}\cdot(\eta_{\ell,i}-\eta_{\ell,i-1}) - \sum_{j=0}^{m} w_{i,j}\cdot f(\tau_{\ell,j},(P_h\eta_h)(\tau_{\ell,j})) \\ &= \sum_{j=0}^{m} w_{i,j}\cdot\big(\frac{d}{dt}(P_h\eta_h^{[k]})(\tau_{\ell,j}) - f(t,(P_h\eta_h^{[k]})(\tau_{\ell,j}))\big) \\ &= \sum_{j=0}^{m} w_{i,j}\cdot D^{[k]}(\tau_{\ell,j}) \;=\; \frac{1}{h}\cdot\int_{t_{\ell,i-1}}^{t_{\ell,i}} D^{[k]}(t)\,dt,\end{aligned}$$

d.h. man kann das Verfahren von Schild als Variante der interpolierten Defektkorrektur auffassen, bei der im Nachbarproblem (2.38) der interpolierte Defekt $\delta^{[k]}(t) = D^{[k]}(t)$ auf jedem Intervall $(t_{\ell,i-1},t_{\ell,i}],\ \ell=0,\ldots,N-1,\ i=1,\ldots,m$ jeweils durch seinen dortigen Mittelwert $\frac{1}{h}\cdot\int_{t_{\ell,i-1}}^{t_{\ell,i}} D^{[k]}(t)\,dt$ ersetzt wird.

2.4 Algorithmische Details

Wir geben zunächst eine Pseudocode-Darstellung des in Abschnitt 2.2.2 beschriebenen Defektkorrekturalgorithmus, die als Grundlage für eine Implementierung dienen kann. Eine Implementierung des Defektkorrekturalgorithmus als Matlab-Funktion, die sich eng an diese Pseudocode-Darstellung anlehnt, findet man im Anhang, Abschnitt A.1.7.

Algorithmus 2.4.1

Setze $s_0 := y_0$, $s_0^{[k]} := y_0,\ k = 0, \ldots, K-1$ und $\eta_{0,0}^{[k]} := y_0,\ k = 0, \ldots, K-1$.

for $\ell = 0, 1, \ldots, N-1$ **do**

if $\ell \geq 1$ **then**

if Verbindungsstrategie = lokal **then**[10]

$$\begin{array}{l}\text{Setze } s_\ell := \eta_{\ell-1,m}^{[K]},\ s_\ell^{[k]} := \eta_{\ell-1,m}^{[K]},\ k = 0, \ldots K-1 \text{ und} \\ \eta_{\ell,0}^{[k]} := \eta_{\ell-1,m}^{[K]},\ k = 0, \ldots K-1.\end{array} \tag{2.51}$$

else *(globale Verbindungsstrategie:)*

$$\begin{array}{l}\text{Setze } s_\ell := \eta_{\ell-1,m}^{[0]},\ s_\ell^{[k]} := \pi_{\ell-1,m}^{[k]},\ k = 0, \ldots K-1 \text{ und} \\ \eta_{\ell,0}^{[k]} := \eta_{\ell-1,m}^{[k]},\ k = 0, \ldots K-1.\end{array} \tag{2.52}$$

end if

end if

Führe m Integrationsschritte

$$(T_\ell, s_\ell) \mapsto (t_{\ell,1}, \eta_{\ell,1}^{[0]}) \mapsto (t_{\ell,2}, \eta_{\ell,2}^{[0]}) \mapsto \ldots \mapsto (t_{\ell,m}, \eta_{\ell,m}^{[0]}) \tag{2.53}$$

des Basisverfahrens angewendet auf

$$y' = f(t, y), \quad y(T_\ell) = s_\ell \tag{2.54}$$

durch.

for $k := 0, 1, \ldots, K-1$ **do**

[10]Die leichte Inkonsistenz zu unserer Vereinbarung $\eta_{\ell,0} = \eta_{\ell-1,m}$ für $\eta_h \in \mathcal{E}_h$ aus Abschnitt 2.1.2, daß hier $\eta_{\ell,0}^{[k]} := \eta_{\ell-1,m}^{[K]}$ gesetzt wird, obwohl für $k < K$ i.a. $\eta_{\ell-1,m}^{[k]} \neq \eta_{\ell-1,m}^{[K]}$ gilt, ist notwendig, damit in (2.55) die richtigen Werte interpoliert werden.

Bestimme Polynom $P_\ell^{[k]}(t)$ vom Grad m mit

$$P_\ell^{[k]}(t_{\ell,\nu}) = \eta_{\ell,\nu}^{[k]}, \quad \nu = 0, \ldots, m. \tag{2.55}$$

Bilde Defekt:

$$d_\ell^{[k]}(t) := \frac{d}{dt} P_\ell^{[k]}(t) - f(t, P_\ell^{[k]}(t)). \tag{2.56}$$

Definiere mit Hilfe des Defekts $d_\ell^{[k]}(t)$ die entsprechende Störung $\delta_\ell^{[k]}(t)$ der Differentialgleichung (2.54) und stelle damit das k-te Nachbarproblem

$$y' = f(t, y) + \delta_\ell^{[k]}(t), \quad y(T_\ell) = s_\ell^{[k]} \tag{2.57}$$

für $t \in [T_\ell, T_{\ell+1}]$ auf.
Führe m Integrationsschritte

$$(T_\ell, s_\ell^{[k]}) \mapsto (t_{\ell,1}, \pi_{\ell,1}^{[k]}) \mapsto (t_{\ell,2}, \pi_{\ell,2}^{[k]}) \mapsto \ldots \mapsto (t_{\ell,m}, \pi_{\ell,m}^{[k]}) \tag{2.58}$$

des Basisverfahrens angewendet auf (2.57) durch.
Setze

$$\eta_{\ell,\nu}^{[k+1]} := \eta_{\ell,\nu}^{[0]} - (\pi_{\ell,\nu}^{[k]} - \eta_{\ell,\nu}^{[k]}), \quad \nu = 1, \ldots, m. \tag{2.59}$$

end for

end for

Man beachte, daß hier die Schleife über ℓ die äußere und die Schleife über k die innere ist, d.h. alle Rechenschritte, die sich auf das Intervall $[T_\ell, T_{\ell+1}]$ beziehen, werden abgeschlossen bevor zum nächsten Intervall $[T_{\ell+1}, T_{\ell+2}]$ fortgeschritten wird.

2.4.1 Darstellung der Interpolationspolynome

Die explizite Berechnung der Koeffizienten des Polynoms $P_\ell^{[k]}(t)$ aus (2.55) ist nicht notwendig. Im Fall der klassischen Defektkorrektur (IDeC) benötigen wir lediglich die Werte von $P_\ell^{[k]}(t)$ bzw. die Werte der Ableitung $\frac{d}{dt} P_\ell^{[k]}(t)$ an den Stellen $t = t_{\ell,\nu} + c_r h,\ \nu = 0, \ldots, m-1,\ r = 1, \ldots, s$ des durch (2.22) gegebenen Gitters $\widetilde{\Gamma}_h$. Mit diesen Werten lassen sich dann die Werte des Defekts $d_\ell^{[k]}(t) = \frac{d}{dt} P_\ell^{[k]}(t) - f(t, P_\ell^{[k]}(t))$ an diesen Stellen berechnen, die zur Lösung des Nachbarproblems (2.57) mit Hilfe des Basisverfahrens erforderlich sind. Im Fall

der interpolierten Defektkorrektur (IIDeC) benötigen wir zur Berechnung des Defekts $d_\ell^{[k]}(t)$ an jenen Stellen $t = \tau_{\ell,\nu},\ \nu = 0,\ldots,m$ des Gitters $\widehat{\Gamma}_h$ aus (2.11), an denen $d_\ell^{[k]}(t)$ interpoliert werden soll, die Werte von $P_\ell^{[k]}(t)$ bzw. die Werte der Ableitung $\frac{d}{dt}P_\ell^{[k]}(t)$ an diesen Stellen. Alle diese Werte berechnen wir wie folgt durch Anwendung der Lagrangeschen Interpolationsformel.

Das Polynom $P_\ell^{[k]}(t)$ läßt sich als

$$P_\ell^{[k]}(t) \ = \ \sum_{j=0}^{m} \eta_{\ell,i}^{[k]} L_{\ell,j}(t) \tag{2.60}$$

darstellen, wobei die $L_{\ell,j}(t),\ j = 0,\ldots,m$ die durch

$$L_{\ell,j}(t) \ := \ \prod_{\substack{k=0\\k\neq j}}^{m} \frac{t - t_{\ell,k}}{t_{\ell,j} - t_{\ell,k}} \tag{2.61}$$

definierten, zur Knotenmenge $\{t_{\ell,0},\ldots,t_{\ell,m}\}$ gehörigen Lagrange-Basispolynome sind.

Für die Ableitung $L'_{\ell,j}(t)$ der Lagrange-Basispolynome ergibt sich aus der Ableitungsregel für Produkte

$$L'_{\ell,j}(t) \ = \ \frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m} \prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m} (t - t_{\ell,k})}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m} (t_{\ell,j} - t_{\ell,k})}. \tag{2.62}$$

Berechnung der Werte $P_\ell^{[k]}(t_{\ell,i-1} + c_r h)$ bzw. $\frac{d}{dt}P_\ell^{[k]}(t_{\ell,i-1} + c_r h)$ bei der IDeC: Dazu benötigen wir die Werte

$$\begin{aligned} L_{\ell,j}(t_{\ell,i-1} + c_r h) \ &= \ \prod_{\substack{k=0\\k\neq j}}^{m} \frac{t_{\ell,i-1} + c_r h - t_{\ell,k}}{t_{\ell,j} - t_{\ell,k}} \\ &= \ \prod_{\substack{k=0\\k\neq j}}^{m} \frac{(T_\ell + (i - 1 + c_r)h) - (T_\ell + kh)}{(T_\ell + jh) - (T_\ell + kh)} \\ &= \ \prod_{\substack{k=0\\k\neq j}}^{m} \frac{i - 1 + c_r - k}{j - k}, \end{aligned}$$

bzw.

$$\begin{aligned} L'_{\ell,j}(t_{\ell,i-1}+c_r h) &= \frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m}\ \prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m}(t_{\ell,i-1}+c_r h-t_{\ell,k})}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m}(t_{\ell,j}-t_{\ell,k})} \\ &= \frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m}\ \prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m}\big((T_\ell+(i-1+c_r)h)-(T_\ell+kh)\big)}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m}\big((T_\ell+jh)-(T_\ell+kh)\big)} \\ &= \frac{1}{h}\cdot\frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m}\ \prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m}(i-1+c_r-k)}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m}(j-k)}. \end{aligned}$$

Offensichtlich sind diese Werte von ℓ unabhängig, wir definieren daher jeweils für $i=1,\ldots,m,\ r=1,\ldots,s$ und $j=0,\ldots,m$:[11]

$$\widetilde{w}_{i,r,j} := \prod_{\substack{k=0\\k\neq j}}^{m}\frac{i-1+c_r-k}{j-k} \quad \Big(=\ L_{\ell,j}(t_{\ell,i-1}+c_r h)\ \Big) \tag{2.63}$$

bzw.[12]

$$\widetilde{w}'_{i,r,j} := \frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m}\ \prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m}(i-1+c_r-k)}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m}(j-k)} \quad \Big(=\ h\cdot L'_{\ell,j}(t_{\ell,i-1}+c_r h)\ \Big). \tag{2.64}$$

Die Werte von $P_\ell^{[k]}(t_{\ell,i-1}+cr_h)$ bzw. $\frac{d}{dt}P_\ell^{[k]}(t_{\ell,i-1}+c_r h)$, lassen sich daher einfach durch Bildung der gewichteten Summen

$$P_\ell^{[k]}(t_{\ell,i-1}+c_r h) \ =\ \sum_{j=0}^{m}\widetilde{w}_{i,r,j}\eta_{\ell,j}^{[k]}, \quad i=1,\ldots,m,\ r=1,\ldots,s \tag{2.65}$$

[11] Die Akzente „˜“ in den Gewichten $\widetilde{w}_{i,r,j}$ bzw. $\widetilde{w}'_{i,r,j}$ sollen anzeigen, daß sie zur Berechnung von Werten von Polynomen an Punkten des Gitters $\widetilde{\Gamma}_h$ verwendet werden.

[12] Der Grund dafür, daß wir in (2.64) den Faktor $\frac{1}{h}$ nicht in die Definition der $\widetilde{w}'_{i,r,j}$ inkludiert haben, ist, daß dadurch die $\widetilde{w}'_{i,r,j}$ von h unabhängig sind, was bei Änderung der Schrittweite h eine Neuberechnung der $\widetilde{w}'_{i,r,j}$ überflüssig macht.

bzw.

$$\frac{d}{dt}P_\ell^{[k]}(t_{\ell,i-1}+c_r h) \;=\; \frac{1}{h}\cdot\sum_{j=0}^{m}\widetilde{w}'_{i,r,j}\eta_{\ell,j}^{[k]},\quad i=1,\ldots,m,\ r=1,\ldots,s \tag{2.66}$$

bestimmen.

Berechnung der Werte $P_\ell^{[k]}(\tau_{\ell,i})$ bzw. $\frac{d}{dt}P_\ell^{[k]}(\tau_{\ell,i})$ bei der IIDeC: Diese Werte lassen sich analog wie oben durch Bildung der gewichteten Summen

$$P_\ell^{[k]}(\tau_{\ell,i}) \;=\; \sum_{j=0}^{m}\widehat{w}_{i,j}\eta_{\ell,j}^{[k]},\quad i=1,\ldots,m \tag{2.67}$$

bzw.

$$\frac{d}{dt}P_\ell^{[k]}(\tau_{\ell,i}) \;=\; \frac{1}{h}\cdot\sum_{j=0}^{m}\widehat{w}'_{i,j}\eta_{\ell,j}^{[k]},\quad i=1,\ldots,m \tag{2.68}$$

berechnen, wobei die Gewichte $\widehat{w}_{i,j}$ bzw. $\widehat{w}'_{i,j}$ jetzt jeweils für $i=1,\ldots,m,\ j=0,\ldots,m$ durch[13]

$$\widehat{w}_{i,j} \;:=\; \prod_{\substack{k=0\\k\neq j}}^{m}\frac{\gamma_i m-k}{j-k}\quad\Big(=\;L_{\ell,j}(\tau_{\ell,i})\;\Big) \tag{2.69}$$

bzw.

$$\widehat{w}'_{i,j} \;:=\; \frac{\displaystyle\sum_{\substack{k_1=0\\k_1\neq j}}^{m}\;\prod_{\substack{k=0\\k\neq j,k\neq k_1}}^{m}(\gamma_i m-k)}{\displaystyle\prod_{\substack{k=0\\k\neq j}}^{m}(j-k)}\quad\Big(=\;h\cdot L'_{\ell,j}(\tau_{\ell,i})\;\Big) \tag{2.70}$$

definiert sind.

Berechnung der Werte des interpolierten Defekts an den Stellen $t=t_{\ell,i}+c_r h$ bei der IIDeC: Das Polynom $D_\ell^{[k]}(t)$ vom Grad $m-1$, das die Werte

$$d_{\ell,\nu}^{[k]} \;:=\; \frac{d}{dt}P_\ell^{[k]}(\tau_{\ell,\nu})-f(t,P_\ell^{[k]}(\tau_{\ell,\nu})),\quad \nu=1,\ldots,m.$$

des Defekts an den Stellen $\tau_{\ell,\nu},\ \nu=1,\ldots,m$ interpoliert, läßt sich darstellen als

$$D_\ell^{[k]}(t) \;=\; \sum_{j=1}^{m}d_{\ell,j}^{[k]}\widehat{L}_{\ell,j}(t), \tag{2.71}$$

[13]Die Akzente „$\widehat{\ }$“ in den Gewichten $\widehat{w}_{i,j}$ bzw. $\widehat{w}'_{i,j}$ zeigen jetzt an, daß diese Gewichte zur Berechnung von Werten von Polynomen an Punkten des Gitters $\widehat{\Gamma}_h$ verwendet werden.

wobei die $\widehat{L}_{\ell,j}(t),\ j = 1, \ldots, m$ die durch

$$\widehat{L}_{\ell,j}(t) \ := \ \prod_{\substack{k=1 \\ k \neq j}}^{m} \frac{t - \tau_{\ell,k}}{\tau_{\ell,j} - \tau_{\ell,k}} \tag{2.72}$$

definierten, zur Knotenmenge $\{\tau_{\ell,1}, \ldots, \tau_{\ell,m}\}$ gehörigen Lagrange-Basispolynome sind.

Im Fall der interpolierten Defektkorrektur sind für die Anwendung des Basisverfahrens auf (2.57) die Werte des interpolierten Defekts $D_\ell^{[k]}(t)$ an den Stellen $t = t_{\ell,i-1} + c_r h$ erforderlich. Diese lassen sich durch Bildung der gewichteten Summe

$$D_\ell^{[k]}(t_{\ell,i-1} + c_r h) \ = \ \sum_{j=1}^{m} \widetilde{v}_{i,r,j} d_{\ell,j}^{[k]}, \quad i = 1, \ldots, m,\ r = 1, \ldots, s \tag{2.73}$$

berechnen, wobei die (von ℓ unabhängigen) Gewichte $\widetilde{v}_{i,r,j}$ durch[14]

$$\begin{aligned} \widehat{L}_{\ell,j}(t_{\ell,i-1} + c_r h) \ &= \ \prod_{\substack{k=1 \\ k \neq j}}^{m} \frac{t_{\ell,i-1} + c_r h - \tau_{\ell,k}}{\tau_{\ell,j} - \tau_{\ell,k}} \\ &= \ \prod_{\substack{k=1 \\ k \neq j}}^{m} \frac{(T_\ell + (i - 1 + c_r)h) - (T_\ell + \gamma_k m h)}{(T_\ell + \gamma_j m h) - (T_\ell + \gamma_k m h)} \\ &= \ \prod_{\substack{k=1 \\ k \neq j}}^{m} \frac{i - 1 + c_r - \gamma_k m}{\gamma_j m - \gamma_k m} \ =: \ \widetilde{v}_{i,r,j} \end{aligned} \tag{2.74}$$

für $i = 1, \ldots, m,\ r = 1, \ldots, s$ und $j = 1, \ldots, m$ gegeben sind.

2.4.2 Implementierung des Basisverfahrens

Das algebraische Gleichungssystem (2.21) wird im allgemeinen mit Hilfe eines (vereinfachten) Newton-Verfahren gelöst. Wie dieses effizient implementiert werden kann, dazu sei auf die entsprechende Literatur, insbesondere auf [9, Abschnitt IV.8] verwiesen. Viele der dort behandelten Aspekte, speziell die Wahl von geeigneten Startwerten für die Newton-Iteration und die Verwendung derselben Jacobi-Matrix für alle Iterationsschritte, lassen sich direkt bei der numerischen Integration (2.53) des Originalproblems (2.54) anwenden.

[14] Die Akzente „˜“ in den Gewichten $\widetilde{v}_{i,r,j}$ zeigen wieder an, daß diese Gewichte zur Berechnung von Werten von Polynomen an Punkten des Gitters $\widetilde{\Gamma}_h$ verwendet werden.

Bei der numerischen Integration (2.58) der Nachbarprobleme (2.57) bieten es sich nun an, bei der Lösung der algebraischen Gleichungen (2.21) mit Hilfe eines vereinfachten Newton-Verfahrens als Approximation für die Jacobi-Matrix, jeweils die gleiche Matrix zu verwenden, die schon bei der Lösung der entsprechenden algebraischen Gleichungen im Basisschritt verwendet wurde. Dadurch läßt sich die im Basisschritt berechnete LU-Zerlegung dieser Matrix wiederverwenden, und es sind in den einzelnen Defektkorrekturschritten keine zusätzlichen LU-Zerlegungen erforderlich. Weiters ist durch die Lösung der Gleichungen (2.21) aus dem Basisschritt aufgrund der engen Nachbarschaft von (2.57) und (2.54) ein sehr guter Startwert für die Newton-Iteration zur Lösung der entsprechenden Gleichungen (2.21) in den einzelnen Defektkorrekturschritten gegeben.

Kapitel 3

Anwendung der Defektkorrekturalgorithmen

3.1 Anwendung der Defektkorrekturalgorithmen auf lineare Anfangswertprobleme

Für lineare Anfangswertprobleme (2.1), d.h. für Anfangswertprobleme der Gestalt

$$y' = A(t) \cdot y + g(t), \qquad y(t_0) = y_0 \tag{3.1}$$

mit

$$A\colon\ [t_0, t_{end}] \to \mathbb{R}^{n\times n} \quad \text{und} \quad g\colon\ [t_0, t_{end}] \to \mathbb{R}^n, \tag{3.2}$$

lassen sich die in Kapitel 2 beschriebenen Defektkorrekturiterationen bei Verwendung der globalen Verbindungsstrategie als iterative Anwendung

$$\eta_h^{[k+1]} \ :=\ S_h \eta_h^{[k]} + v_h, \quad k = 0, 1, 2, \ldots \tag{3.3}$$

eines affinen Operators $\eta_h \mapsto S_h\eta_h + v_h$ auffassen, wobei $S_h\colon\ \mathcal{E}_h \to \mathcal{E}_h$ homogenlinear und $v_h \in \mathcal{E}_h$ ist.

Beim Raum $\mathcal{E}_h$ der Gitterfunktionen auf Γ_h handelt es sich im wesentlichen um den Vektorraum $\mathbb{R}^{n(Nm+1)}$, und der Unterraum

$$\mathcal{E}_h^0 := \{\eta_h \in \mathcal{E}_h\colon\ \eta_h(t_0) = y_0\} \tag{3.4}$$

ist zum Untervektorraum $\mathbb{R}^{Nmn}$ isomorph, wobei jedem Element $\eta_h \in \mathcal{E}_h^0$ der

Vektor

$$\boldsymbol{\eta} \;=\; \begin{pmatrix} \eta_{0,1} \\ \vdots \\ \eta_{0,m} \\ \vdots \\ \eta_{N-1,1} \\ \vdots \\ \eta_{N-1,m} \end{pmatrix} \;\in\; \mathbb{R}^{Nmn} \tag{3.5}$$

zugeordnet wird. Unter dieser Isomorphie entspricht dem Operator S_h bzw. der Gitterfunktion v_h aus (3.3) eine Matrix $\boldsymbol{S} \in \mathbb{R}^{(Nmn)\times(Nmn)}$ bzw. ein Vektor $\boldsymbol{v} \in \mathbb{R}^{Nmn}$, und damit entspricht der Iteration (3.3) die äquivalente Iteration

$$\boldsymbol{\eta}^{[k+1]} \;:=\; \boldsymbol{S}\cdot\boldsymbol{\eta}^{[k]} + \boldsymbol{v}, \quad k = 0, 1, 2, \ldots. \tag{3.6}$$

Ziel dieses Abschnitts ist, die Matrix $\boldsymbol{S}$ und den Vektor $\boldsymbol{v}$ in einer relativ expliziten Form anzugeben, die für weitere Untersuchungen nützlich ist, und mit der es nur mehr eine Routineangelegenheit ist, $\boldsymbol{S}$ bzw. $\boldsymbol{v}$ für konkrete Probleme (3.1) mit Hilfe eines Matrix-orientierten Programms wie MATLAB zu berechnen.

In diesem Abschnitt verwenden wir folgende Bezeichnungen:

- Kronecker-Produkt zweier Matrizen $A \in \mathbb{R}^{n_1\times m_1}$ und $B \in \mathbb{R}^{n_2\times m_2}$:

$$A \otimes B \;:=\; \begin{pmatrix} a_{11}B & \cdots & a_{1m_1}B \\ \vdots & & \vdots \\ a_{n_11}B & \cdots & a_{n_1m_1}B \end{pmatrix} \in \mathbb{R}^{(n_1n_2)\times(m_1m_2)};$$

- $e_n := (1, \ldots, 1)^T$ (n Komponenten);
- $I_n :=$ $(n \times n)$-Einheitsmatrix.

Die Gitterpunkte $t_{\ell,\nu} \in \Gamma_h$ bezeichnen wir, wo es zweckmäßig ist, auch durch $t_\nu := t_0 + \nu \cdot h,\ \nu = 0, \ldots, Nm$.

3.1.1 Vektordarstellung $\boldsymbol{\eta}^{[0]}$ der Basisapproximation $\eta_h^{[0]}$

Ein Schritt $(t_\nu, \eta_\nu) \mapsto (t_{\nu+1}, \eta_{\nu+1})$ des Basisverfahrens (2.19) angewendet auf (3.1) läßt sich darstellen als

$$\eta_{\nu+1} \;=\; J_\nu \cdot \eta_\nu + K_\nu \cdot g_\nu, \tag{3.7}$$

wobei[1]

$$g_\nu := \begin{pmatrix} g(t_\nu + c_1 h) \\ \vdots \\ g(t_\nu + c_s h) \end{pmatrix}, \tag{3.8}$$

$$A_\nu := \text{blockdiag}(A(t_\nu + c_1 h), \ldots, A(t_\nu + c_s h)), \tag{3.9}$$

$$J_\nu := I_n + h(b^T \otimes I_n) A_\nu \big(I_{sn} - h(A \otimes I_n) A_\nu\big)^{-1} (e_s \otimes I_n), \tag{3.10}$$

$$K_\nu := h(b^T \otimes I_n) + h(b^T \otimes I_n) A_\nu \big(I_{sn} - h(A \otimes I_n) A_\nu\big)^{-1} h(A \otimes I_n). \tag{3.11}$$

Wir setzen[2]

$$\widetilde{\boldsymbol{g}} := \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{Nm-1} \end{pmatrix}, \tag{3.12}$$

$$\boldsymbol{J} := \begin{pmatrix} J_0 \\ J_1 J_0 \\ \vdots \\ J_{Nm-1} \cdots J_1 J_0 \end{pmatrix}, \tag{3.13}$$

$$\boldsymbol{K} := \begin{pmatrix} K_0 & 0 & 0 & \ldots & 0 \\ J_1 K_0 & K_1 & 0 & \ldots & 0 \\ J_2 J_1 K_0 & J_2 K_1 & K_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ J_{Nm-1} \cdots J_1 K_0 & J_{Nm-1} \cdots J_2 K_1 & J_{Nm-1} \cdots J_3 K_2 & \ldots & K_{Nm-1} \end{pmatrix}. \tag{3.14}$$

Damit läßt sich die durch Nm Integrationsschritte des auf (3.1) angewendeten Basisverfahrens definierte Basisapproximation $\eta_h^{[0]} \in \mathcal{E}_h^0$, bzw. deren gemäß (3.5) zugeordneter Vektor $\boldsymbol{\eta}^{[0]} \in \mathbb{R}^{Nmn}$, als

$$\boldsymbol{\eta}^{[0]} = \boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot \widetilde{\boldsymbol{g}} \tag{3.15}$$

darstellen.

[1] Wir hoffen, daß hier die Verwendung des Buchstabens „A“ sowohl für die Koeffizientenmatrix A des Basisverfahrens (2.19) als auch für die t-abhängige Matrix $A(t)$ aus (3.1) zu keiner Verwirrung führt.

[2] Der Akzent „˜“ bei Vektoren wie hier beim Vektor $\widetilde{\boldsymbol{g}}$ soll im folgenden andeuten, daß der jeweilige Vektor einer auf dem Gitter $\widetilde{\Gamma}_h$ definierten Gitterfunktion entspricht.

3.1.2 Vektordarstellungen des auf $\widetilde{\Gamma}_h$ bzw. $\widehat{\Gamma}_h$ eingeschränkten stückweisen Interpolationspolynoms $P^{[k]}(t)$ und dessen Ableitung $\frac{d}{dt}P^{[k]}(t)$

Wir definieren mit Hilfe der in Abschnitt 2.4.1 definierten Gewichte $\widetilde{w}_{i,r,j}$, $\widetilde{w}'_{i,r,j}$, $\widehat{w}_{i,j}$ bzw. $\widehat{w}'_{i,j}$ die Matrizen

$$\widetilde{W}_0 := \begin{pmatrix} \widetilde{w}_{1,1,0} \\ \vdots \\ \widetilde{w}_{1,s,0} \\ \vdots \\ \widetilde{w}_{m,1,0} \\ \vdots \\ \widetilde{w}_{m,s,0} \end{pmatrix} \otimes I_n, \qquad \widetilde{W}_1 := \begin{pmatrix} \widetilde{w}_{1,1,1} & \cdots & \widetilde{w}_{1,1,m} \\ \vdots & & \vdots \\ \widetilde{w}_{1,s,1} & \cdots & \widetilde{w}_{1,s,m} \\ \vdots & & \vdots \\ \widetilde{w}_{m,1,1} & \cdots & \widetilde{w}_{m,1,m} \\ \vdots & & \vdots \\ \widetilde{w}_{m,s,1} & \cdots & \widetilde{w}_{m,s,m} \end{pmatrix} \otimes I_n, \quad (3.16)$$

$$\widetilde{W}'_0 := \begin{pmatrix} \widetilde{w}'_{1,1,0} \\ \vdots \\ \widetilde{w}'_{1,s,0} \\ \vdots \\ \widetilde{w}'_{m,1,0} \\ \vdots \\ \widetilde{w}'_{m,s,0} \end{pmatrix} \otimes I_n, \qquad \widetilde{W}'_1 := \begin{pmatrix} \widetilde{w}'_{1,1,1} & \cdots & \widetilde{w}'_{1,1,m} \\ \vdots & & \vdots \\ \widetilde{w}'_{1,s,1} & \cdots & \widetilde{w}'_{1,s,m} \\ \vdots & & \vdots \\ \widetilde{w}'_{m,1,1} & \cdots & \widetilde{w}'_{m,1,m} \\ \vdots & & \vdots \\ \widetilde{w}'_{m,s,1} & \cdots & \widetilde{w}'_{m,s,m} \end{pmatrix} \otimes I_n, \quad (3.17)$$

$$\widehat{W}_0 := \begin{pmatrix} \widehat{w}_{1,0} \\ \vdots \\ \widehat{w}_{m,0} \end{pmatrix} \otimes I_n, \qquad \widehat{W}_1 := \begin{pmatrix} \widehat{w}_{1,1} & \cdots & \widehat{w}_{1,m} \\ \vdots & & \vdots \\ \widehat{w}_{m,1} & \cdots & \widehat{w}_{m,m} \end{pmatrix} \otimes I_n \quad (3.18)$$

bzw.

$$\widehat{W}'_0 := \begin{pmatrix} \widehat{w}'_{1,0} \\ \vdots \\ \widehat{w}'_{m,0} \end{pmatrix} \otimes I_n, \qquad \widehat{W}'_1 := \begin{pmatrix} \widehat{w}'_{1,1} & \cdots & \widehat{w}'_{1,m} \\ \vdots & & \vdots \\ \widehat{w}'_{m,1} & \cdots & \widehat{w}'_{m,m} \end{pmatrix} \otimes I_n. \quad (3.19)$$

Sei $p(t)$ ein Polynom (bzw. ein n-dimensionaler Vektor von Polynomen) vom Grad $\leq m$. Dann gilt

$$[\widetilde{W}_0|\widetilde{W}_1] \cdot p = \widetilde{p}, \quad [\widetilde{W}'_0|\widetilde{W}'_1] \cdot p = \widetilde{p}', \quad [\widehat{W}_0|\widehat{W}_1] \cdot p = \widehat{p}, \quad [\widehat{W}'_0|\widehat{W}'_1] \cdot p = \widehat{p}' \quad (3.20)$$

mit

$$p = \begin{pmatrix} p(T) \\ p(T + \frac{1}{m} \cdot H) \\ \vdots \\ p(T + \frac{m}{m} \cdot H) \end{pmatrix},$$

$$\widetilde{p} = \begin{pmatrix} p(T + c_1 \cdot \frac{H}{m}) \\ \vdots \\ p(T + c_s \cdot \frac{H}{m}) \\ \vdots \\ p(T + (m-1+c_1) \cdot \frac{H}{m}) \\ \vdots \\ p(T + (m-1+c_s) \cdot \frac{H}{m}) \end{pmatrix}, \quad \widetilde{p}' = \begin{pmatrix} p'(T + c_1 \cdot \frac{H}{m}) \\ \vdots \\ p'(T + c_s \cdot \frac{H}{m}) \\ \vdots \\ p'(T + (m-1+c_1) \cdot \frac{H}{m}) \\ \vdots \\ p'(T + (m-1+c_s) \cdot \frac{H}{m}) \end{pmatrix}$$

und

$$\widehat{p} = \begin{pmatrix} p(T + \gamma_1 H) \\ \vdots \\ p(T + \gamma_m H) \end{pmatrix}, \quad \widehat{p}' = \begin{pmatrix} p'(T + \gamma_1 H) \\ \vdots \\ p'(T + \gamma_1 H) \end{pmatrix}$$

für beliebiges T und $H > 0$. Mit Hilfe der Matrizen $\widetilde{W}_0$, $\widetilde{W}_1$, $\widetilde{W}_0'$, $\widetilde{W}_1'$, $\widehat{W}_0$, $\widehat{W}_1$, $\widehat{W}_0'$ und $\widehat{W}_1'$ definieren wir die Matrizen $\widetilde{\boldsymbol{W}}_0$, $\widetilde{\boldsymbol{W}}_1$, $\widetilde{\boldsymbol{W}}_0'$, $\widetilde{\boldsymbol{W}}_1'$, $\widehat{\boldsymbol{W}}_0$, $\widehat{\boldsymbol{W}}_1$, $\widehat{\boldsymbol{W}}_0'$ und $\widehat{\boldsymbol{W}}_1'$ nach dem Schema

$$\left(\boldsymbol{W}_0 \| \;\; \boldsymbol{W}_1 \;\; \right) := \left(\begin{array}{c||cc|cc|cc|cc|c} W_0 & W_1 & & & & & & & & \\ \hline 0 & 0 & W_0 & W_1 & & & & & & \\ \hline & & & 0 & W_0 & W_1 & & & & \\ \hline \vdots & & & & \ddots & \ddots & \ddots & & & \\ \hline & & & & & 0 & W_0 & W_1 & & \\ \hline 0 & & & & & & 0 & W_0 & W_1 \end{array}\right), \tag{3.21}$$

wobei hier der Block $[W_0 | \;\; W_1 \;\;]$ N-mal vorkommt. Sei $P^{[k]}(t) := (P_h \eta_h^{[k]})(t)$ das stückweise Polynom aus $\mathcal{P}_h$, das $\eta_h^{[k]} \in \mathcal{E}_h^0$ interpoliert. Mit[3]

$$\widetilde{\boldsymbol{p}}^{[k]} := \begin{pmatrix} P^{[k]}(t_0 + c_1 h) \\ \vdots \\ P^{[k]}(t_0 + c_s h) \\ \vdots \\ P^{[k]}(t_{Nm-1} + c_1 h) \\ \vdots \\ P^{[k]}(t_{Nm-1} + c_s h) \end{pmatrix}, \qquad \widetilde{\boldsymbol{p}}'^{[k]} := \begin{pmatrix} \frac{d}{dt} P^{[k]}(t_0 + c_1 h) \\ \vdots \\ \frac{d}{dt} P^{[k]}(t_0 + c_s h) \\ \vdots \\ \frac{d}{dt} P^{[k]}(t_{Nm-1} + c_1 h) \\ \vdots \\ \frac{d}{dt} P^{[k]}(t_{Nm-1} + c_s h) \end{pmatrix} \tag{3.22}$$

gilt

$$\widetilde{\boldsymbol{p}}^{[k]} = \widetilde{\boldsymbol{W}}_0 \cdot y_0 + \widetilde{\boldsymbol{W}}_1 \cdot \boldsymbol{\eta}^{[k]}, \tag{3.23}$$

$$\widetilde{\boldsymbol{p}}'^{[k]} = \frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_0' \cdot y_0 + \frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_1' \cdot \boldsymbol{\eta}^{[k]}, \tag{3.24}$$

[3]Für Basisverfahren (2.19) mit $c_1 = 0$ ist hier in der Definition von $\widetilde{\boldsymbol{p}}'^{[k]}$, $\frac{d}{dt} P^{[k]}(t_\nu + c_1 h)$ für $\nu = 0, m, 2m, \ldots, (N-1)m$ jeweils durch $\lim_{t \to t_\nu +} \left(\frac{d}{dt} P^{[k]}(t)\right)$ zu ersetzen, vgl. Vereinbarung 2.1.2.

wobei hier $\boldsymbol{\eta}^{[k]}$ der Vektor ist, der $\eta_h^{[k]}$ gemäß der Zuordnung (3.5) entspricht. Analog gilt[4]

$$\widehat{\boldsymbol{p}}^{[k]} = \widehat{\boldsymbol{W}}_0 \cdot y_0 + \widehat{\boldsymbol{W}}_1 \cdot \boldsymbol{\eta}^{[k]}, \tag{3.25}$$

$$\widehat{\boldsymbol{p}}'^{[k]} = \frac{1}{h} \cdot \widehat{\boldsymbol{W}}_0' \cdot y_0 + \frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' \cdot \boldsymbol{\eta}^{[k]}, \tag{3.26}$$

wobei

$$\widehat{\boldsymbol{p}}^{[k]} := \begin{pmatrix} P^{[k]}(\tau_{0,1}) \\ \vdots \\ P^{[k]}(\tau_{0,m}) \\ \vdots \\ P^{[k]}(\tau_{N-1,1}) \\ \vdots \\ P^{[k]}(\tau_{N-1,m}) \end{pmatrix}, \qquad \widehat{\boldsymbol{p}}'^{[k]} := \begin{pmatrix} \frac{d}{dt}P^{[k]}(\tau_{0,1}) \\ \vdots \\ \frac{d}{dt}P^{[k]}(\tau_{0,m}) \\ \vdots \\ \frac{d}{dt}P^{[k]}(\tau_{N-1,m}) \\ \vdots \\ \frac{d}{dt}P^{[k]}(\tau_{N-1,m}) \end{pmatrix}. \tag{3.27}$$

3.1.3 Matrix S und Vektor v aus (3.6) im Fall der klassischen Defektkorrektur (IDeC)

Der Defekt ist durch

$$d^{[k]} = \frac{d}{dt}P^{[k]}(t) - A(t) \cdot P^{[k]}(t) - g(t) \tag{3.28}$$

gegeben. Wenn wir

$$\widetilde{\boldsymbol{A}} := \text{blockdiag}(A_0, A_1, \ldots, A_{Nm-1}) \tag{3.29}$$

setzen, wobei die A_ν, $\nu = 0, \ldots, Nm-1$ durch (3.9) gegeben sind, dann folgt mit $\widetilde{\boldsymbol{g}}$ aus (3.12) und $\widetilde{\boldsymbol{p}}^{[k]}$, $\widetilde{\boldsymbol{p}}'^{[k]}$ aus (3.22)[5]

$$\begin{aligned} \widetilde{\boldsymbol{d}}^{[k]} &:= \begin{pmatrix} d^{[k]}(t_0 + c_1 h) \\ \vdots \\ d^{[k]}(t_0 + c_s h) \\ \vdots \\ d^{[k]}(t_{Nm-1} + c_1 h) \\ \vdots \\ d^{[k]}(t_{Nm-1} + c_s h) \end{pmatrix} = \widetilde{\boldsymbol{p}}'^{[k]} - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{p}}^{[k]} - \widetilde{\boldsymbol{g}} \\ &= \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_1' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_1 \right) \cdot \boldsymbol{\eta}^{[k]} + \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_0' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_0 \right) \cdot y_0 - \widetilde{\boldsymbol{g}}. \end{aligned}$$

[4]In Analogie zu Fußnote 2 kennzeichnen wir im folgenden Vektoren wie $\widehat{\boldsymbol{p}}^{[k]}$ und $\widehat{\boldsymbol{p}}'^{[k]}$, die einer auf dem Gitter $\widehat{\Gamma}_h$ definierten Gitterfunktion entsprechen, mit dem Akzent „$\widehat{\ }$“.

[5]Für Basisverfahren mit $c_1 = 0$ ist hier $d^{[k]}(t_\nu + c_1 h)$ für $\nu = 0, m, 2m, \ldots, (N-1)m$ jeweils durch $\lim_{t \to t_\nu +} d^{[k]}(t)$ zu ersetzen.

Das Nachbarproblem

$$y' = A(t) \cdot y + g(t) + d^{[k]}(t), \qquad y(t_0) = y_0 \tag{3.30}$$

hat dieselbe Gestalt wie das Originalproblem (3.1). Die Anwendung des Basisverfahrens ergibt daher (vgl. (3.15))

$$\boldsymbol{\pi}^{[k]} \;=\; \boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot (\widetilde{\boldsymbol{g}} + \widetilde{\boldsymbol{d}}^{[k]}). \tag{3.31}$$

Es folgt

$$\begin{aligned}
\boldsymbol{\eta}^{[k+1]} &= \boldsymbol{\eta}^{[0]} - (\boldsymbol{\pi}^{[k]} - \boldsymbol{\eta}^{[k]}) \\
&= \big(\boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot \widetilde{\boldsymbol{g}}\big) - \big(\boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot (\widetilde{\boldsymbol{g}} + \widetilde{\boldsymbol{d}}^{[k]})\big) + \boldsymbol{\eta}^{[k]} \\
&= \boldsymbol{\eta}^{[k]} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{d}}^{[k]} \\
&= \boldsymbol{\eta}^{[k]} - \boldsymbol{K} \cdot \left(\left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_1' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_1 \right) \cdot \boldsymbol{\eta}^{[k]} + \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_0' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_0 \right) \cdot y_0 - \widetilde{\boldsymbol{g}} \right),
\end{aligned}$$

also

$$\boldsymbol{\eta}^{[k+1]} \;=\; \boldsymbol{S} \cdot \boldsymbol{\eta}^{[k]} + \boldsymbol{v}$$

mit

$$\boldsymbol{S} \;=\; \boldsymbol{S}_{\mathrm{IDeC}} \;:=\; I_{Nmn} - \boldsymbol{K} \cdot \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_1' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_1 \right), \tag{3.32}$$

$$\boldsymbol{v} \;=\; \boldsymbol{v}_{\mathrm{IDeC}} \;:=\; \boldsymbol{K} \cdot \left(\widetilde{\boldsymbol{g}} - \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_0' - \widetilde{\boldsymbol{A}} \cdot \widetilde{\boldsymbol{W}}_0 \right) \cdot y_0 \right). \tag{3.33}$$

3.1.4 Matrix $\boldsymbol{S}$ und Vektor $\boldsymbol{v}$ aus (3.6) im Fall der interpolierten Defektkorrektur (IIDeC)

Wir setzen

$$\widehat{\boldsymbol{A}} \;:=\; \mathrm{blockdiag}\big(A(\tau_{0,1}), \ldots, A(\tau_{0,m}), \ldots, A(\tau_{N-1,1}), \ldots, A(\tau_{N-1,m})\big) \tag{3.34}$$

und

$$\widehat{\boldsymbol{g}} \;:=\; \begin{pmatrix} g(\tau_{0,1}) \\ \vdots \\ g(\tau_{0,m}) \\ \vdots \\ g(\tau_{N-1,1}) \\ \vdots \\ g(\tau_{N-1,m}) \end{pmatrix}. \tag{3.35}$$

Damit und mit $d^{[k]}(t)$ aus (3.28) und $\widehat{\boldsymbol{p}}^{[k]}$, $\widehat{\boldsymbol{p}}'^{[k]}$ aus (3.27) folgt

$$\begin{aligned}\widehat{\boldsymbol{d}}^{[k]} &:= \begin{pmatrix} d^{[k]}(\tau_{0,1}) \\ \vdots \\ d^{[k]}(\tau_{0,m}) \\ \vdots \\ d^{[k]}(\tau_{N-1,1}) \\ \vdots \\ d^{[k]}(\tau_{N-1,m}) \end{pmatrix} = \widehat{\boldsymbol{p}}'^{[k]} - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{p}}^{[k]} - \widehat{\boldsymbol{g}} \\ &= \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_1\right) \cdot \boldsymbol{\eta}^{[k]} + \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_0' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_0\right) \cdot y_0 - \widehat{\boldsymbol{g}}.\end{aligned}$$

Mit Hilfe der in (2.74) definierten Gewichte $\widetilde{v}_{i,r,j}$ definieren wir die Matrix

$$\widetilde{V} := \begin{pmatrix} \widetilde{v}_{1,1,1} & \cdots & \widetilde{v}_{1,1,m} \\ \vdots & & \vdots \\ \widetilde{v}_{1,s,1} & \cdots & \widetilde{v}_{1,s,m} \\ \vdots & & \vdots \\ \widetilde{v}_{m,1,1} & \cdots & \widetilde{v}_{m,1,m} \\ \vdots & & \vdots \\ \widetilde{v}_{m,s,1} & \cdots & \widetilde{v}_{m,s,m} \end{pmatrix} \otimes I_n. \tag{3.36}$$

Für ein Polynom (bzw. einen n-dimensionaler Vektor von Polynomen) $p(t)$ vom Grad $\leq m-1$ gilt damit

$$\widetilde{V} \cdot \begin{pmatrix} p(T+\gamma_1 H) \\ \vdots \\ p(T+\gamma_m H) \end{pmatrix} = \begin{pmatrix} p(T+c_1 \cdot \frac{H}{m}) \\ \vdots \\ p(T+c_s \cdot \frac{H}{m}) \\ \vdots \\ p(T+(m-1+c_1) \cdot \frac{H}{m}) \\ \vdots \\ p(T+(m-1+c_s) \cdot \frac{H}{m}) \end{pmatrix} \tag{3.37}$$

für beliebiges T und $H > 0$. Wir setzen[6]

$$\widetilde{\boldsymbol{V}} := \text{blockdiag}(\widetilde{V}, \ldots, \widetilde{V}), \tag{3.38}$$

wobei hier der Block $\widetilde{V}$ N-mal vorkommt. Sei $D^{[k]}(t)$ die wie in (2.37) definierte stückweise Polynomfunktion, die den Defekt $d^{[k]}(t)$ an den Stellen $\tau_{\ell,\nu} \in \widehat{\Gamma}_h$

[6]Wir verwenden die Schreibweise blockdiag$(\widetilde{V}, \ldots, \widetilde{V})$ in offensichtlicher Weise, auch wenn die Blöcke $\widetilde{V}$ für $s \geq 2$ nicht quadratisch sind.

interpoliert. Damit gilt[7]

$$\widetilde{\boldsymbol{D}}^{[k]} := \begin{pmatrix} D^{[k]}(t_0 + c_1 h) \\ \vdots \\ D^{[k]}(t_0 + c_s h) \\ \vdots \\ D^{[k]}(t_{Nm-1} + c_1 h) \\ \vdots \\ D^{[k]}(t_{Nm-1} + c_s h) \end{pmatrix} = \widetilde{\boldsymbol{V}} \cdot \widehat{\boldsymbol{d}}^{[k]}, \tag{3.39}$$

Die Anwendung des Basisverfahrens auf das Nachbarproblem

$$y' = A(t) \cdot y + g(t) + D^{[k]}(t), \qquad y(t_0) = y_0 \tag{3.40}$$

ergibt nun

$$\boldsymbol{\pi}^{[k]} = \boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot (\widetilde{\boldsymbol{g}} + \widetilde{\boldsymbol{D}}^{[k]}) \tag{3.41}$$

und daher

$$\begin{aligned} \boldsymbol{\eta}^{[k+1]} &= \boldsymbol{\eta}^{[0]} - (\boldsymbol{\pi}^{[k]} - \boldsymbol{\eta}^{[k]}) \\ &= \left(\boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot \widetilde{\boldsymbol{g}}\right) - \left(\boldsymbol{J} \cdot y_0 + \boldsymbol{K} \cdot (\widetilde{\boldsymbol{g}} + \widetilde{\boldsymbol{D}}^{[k]})\right) + \boldsymbol{\eta}^{[k]} \\ &= \boldsymbol{\eta}^{[k]} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{D}}^{[k]} \\ &= \boldsymbol{\eta}^{[k]} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \widehat{\boldsymbol{d}}^{[k]} \\ &= \boldsymbol{\eta}^{[k]} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_1 \right) \cdot \boldsymbol{\eta}^{[k]} + \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_0' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_0 \right) \cdot y_0 - \widehat{\boldsymbol{g}} \right), \end{aligned}$$

also

$$\boldsymbol{\eta}^{[k+1]} = \boldsymbol{S} \cdot \boldsymbol{\eta}^{[k]} + \boldsymbol{v}$$

mit

$$\boldsymbol{S} = \boldsymbol{S}_{\mathrm{IIDeC}} := I_{Nmn} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_1 \right), \tag{3.42}$$

$$\boldsymbol{v} = \boldsymbol{v}_{\mathrm{IIDeC}} := \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\widehat{\boldsymbol{g}} - \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_0' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_0 \right) \cdot y_0 \right). \tag{3.43}$$

[7] Für Basisverfahren mit $c_1 = 0$ ist hier $D^{[k]}(t_\nu + c_1 h)$ für $\nu = 0, m, 2m, \ldots, (N-1)m$ jeweils durch $\lim_{t \to t_\nu +} D^{[k]}(t)$ zu ersetzen.

3.2 Anwendung der Defektkorrekturalgorithmen auf Anfangswertprobleme der Gestalt $y' = \lambda y + g(t),\ y(0) = y_0$

Dieser Abschnitt ist stark durch die Untersuchungen in [8] motiviert. Wir wenden die Ergebnisse aus Abschnitt 3.1 auf ein skalares Anfangswertproblem

$$y' = \lambda y + g(t), \qquad y(t_0) = y_0 \tag{3.44}$$

mit $\lambda \in \mathbb{C}$ an.

Ein Schritt $(t_\nu, \eta_\nu) \mapsto (t_{\nu+1} + h, \eta_{\nu+1})$ des Basisverfahrens (2.19) angewendet auf (3.44) entspricht jetzt dem linearen Gleichungssystem

$$\left(\begin{array}{c|c} & 0 \\ I_s - h\lambda A & \vdots \\ & 0 \\ \hline -h\lambda b^T & 1 \end{array}\right) \cdot \left(\begin{array}{c} Y_1 \\ \vdots \\ Y_s \\ \hline \eta_{\nu+1} \end{array}\right) = \left(\begin{array}{c} \eta_\nu + ha_{11}g(t_\nu + hc_1) + \ldots + ha_{1s}g(t_\nu + hc_s) \\ \vdots \\ \eta_\nu + ha_{s1}g(t_\nu + hc_1) + \ldots + ha_{ss}g(t_\nu + hc_s) \\ \hline \eta_\nu + hb_1g(t_\nu + hc_1) + \ldots + hb_{1b}g(t_\nu + hc_s) \end{array}\right).$$

Durch Anwendung der Cramerschen Regel erhält man daraus

$$\eta_{\nu+1} = \frac{1}{\det[I_s - h\lambda A]} \cdot \det \left(\begin{array}{c|c} & \eta_\nu + ha_{11}g(t_\nu + hc_1) + \ldots + ha_{1s}g(t_\nu + hc_s) \\ I_s - h\lambda A & \vdots \\ & \eta_\nu + ha_{s1}g(t_\nu + hc_1) + \ldots + ha_{ss}g(t_\nu + hc_s) \\ \hline -h\lambda b^T & \eta_\nu + hb_1g(t_\nu + hc_1) + \ldots + hb_sg(t_\nu + hc_s) \end{array}\right).$$

Wir definieren die rationalen Funktionen[8]

$$R(z) := \frac{1}{\det[I_s - zA]} \cdot \det \left(\begin{array}{c|c} & 1 \\ I_s - zA & \vdots \\ & 1 \\ \hline -zb^T & 1 \end{array}\right), \tag{3.45}$$

$$R_j(z) := \frac{1}{\det[I_s - zA]} \cdot \det \left(\begin{array}{c|c} & a_{1j} \\ I_s - zA & \vdots \\ & a_{sj} \\ \hline -zb^T & b_j \end{array}\right), \quad j = 1, \ldots, s. \tag{3.46}$$

Damit gilt

$$\eta_{\nu+1} = R(h\lambda)\eta_\nu + h\sum_{j=1}^{s} R_j(h\lambda)g(t_\nu + hc_j).$$

[8] $R(z)$ ist die Stabilitätsfunktion für die A-Stabilität.

Basisverfahren	$R(z)$	$R_1(z)$	$R_2(z)$	$R_3(z)$
Imp. Euler	$\frac{1}{1-z}$	$\frac{1}{1-z}$		
IMR	$\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$	$\frac{1}{1-\frac{1}{2}z}$		
IMR2	$\left(\frac{1+\frac{1}{4}z}{1-\frac{1}{4}z}\right)^2$	$\frac{1}{2}\cdot\frac{1+\frac{1}{4}z}{(1+\frac{1}{4}z)^2}$	$\frac{1}{2}\cdot\frac{1}{1-\frac{1}{4}z}$	
ITR	$\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$	$\frac{1}{2}\cdot\frac{1}{1-\frac{1}{2}z}$	$\frac{1}{2}\cdot\frac{1}{1-\frac{1}{2}z}$	
ITR2	$\left(\frac{1+\frac{1}{4}z}{1-\frac{1}{4}z}\right)^2$	$\frac{1}{4}\cdot\frac{1+\frac{1}{4}z}{(1-\frac{1}{4}z)^2}$	$\frac{1}{2}\cdot\frac{1}{(1-\frac{1}{4}z)^2}$	$\frac{1}{4}\cdot\frac{1}{1-\frac{1}{4}z}$
SDIRK(2)	$\frac{1+(1-2\gamma)z}{(1-\gamma z)^2}$	$\frac{1-\gamma}{(1-\gamma z)^2}$	$\frac{\gamma}{1-\gamma z}$	
RadauIIA(2)	$\frac{1+\frac{1}{3}z}{1-\frac{2}{3}z+\frac{1}{6}z^2}$	$\frac{3}{4}\cdot\frac{1}{1-\frac{2}{3}z+\frac{1}{6}z^2}$	$\frac{1}{4}\cdot\frac{1-\frac{2}{3}z}{1-\frac{2}{3}z+\frac{1}{6}z^2}$	

Tabelle 3.1: $R(z)$ und $R_j(z)$, $j = 1,\ldots,s$ für die IRK-Verfahren aus Abschnitt 2.1.7.

In der Gleichung (3.7) ist J_ν jetzt also ein Skalar,

$$J_\nu \;=\; R(h\lambda),$$

und K_ν ein s-dimensionaler Zeilenvektor,

$$K_\nu \;=\; h\cdot\Big(R_1(h\lambda),\ldots,R_s(h\lambda)\Big)\;.$$

Für die Matrizen $\boldsymbol{J}$ und $\boldsymbol{K}$ aus (3.15) gilt somit

$$\boldsymbol{J} \;=\; \Big(\; R(h\lambda), R(h\lambda)^2,\ldots,R(h\lambda)^{Nm}\Big)^T, \tag{3.47}$$

$$\boldsymbol{K} \;=\; h\cdot\begin{pmatrix} 1 & 0 & \ldots & 0\\ R(h\lambda) & 1 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ R(h\lambda)^{Nm-1} & R(h\lambda)^{Nm-2} & \ldots & 1\end{pmatrix}\otimes\Big(R_1(h\lambda),\ldots,R_s(h\lambda)\Big) \tag{3.48}$$

Die Matrizen $\boldsymbol{S}$ aus (3.32) bzw. (3.42) und die Vektoren $\boldsymbol{v}$ aus (3.33) bzw. (3.43) vereinfachen sich jetzt zu (man beachte, daß der Multiplikation mit der Matrix

$\widetilde{\boldsymbol{A}}$ bzw. $\widehat{\boldsymbol{A}}$ nun die Multiplikation mit dem Skalar λ entspricht)

$$\boldsymbol{S}_{\mathrm{IDeC}} := I_{Nm} - \boldsymbol{K} \cdot \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_1' - \lambda \cdot \widetilde{\boldsymbol{W}}_1 \right), \tag{3.49}$$

$$\boldsymbol{v}_{\mathrm{IDeC}} := \boldsymbol{K} \cdot \left(\widetilde{\boldsymbol{g}} - \left(\frac{1}{h} \cdot \widetilde{\boldsymbol{W}}_0' - \lambda \cdot \widetilde{\boldsymbol{W}}_0 \right) \cdot y_0 \right), \tag{3.50}$$

$$\boldsymbol{S}_{\mathrm{IIDeC}} := I_{Nm} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \lambda \cdot \widehat{\boldsymbol{W}}_1 \right), \tag{3.51}$$

$$\boldsymbol{v}_{\mathrm{IIDeC}} := \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\widehat{\boldsymbol{g}} - \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_0' - \lambda \cdot \widehat{\boldsymbol{W}}_0 \right) \cdot y_0 \right). \tag{3.52}$$

Die Matrizen $\boldsymbol{S}_{\mathrm{IDeC}}$ und $\boldsymbol{S}_{\mathrm{IIDeC}}$ sind bei gegebenen Basisverfahren (2.19), festem N und m und (im Fall von $\boldsymbol{S}_{\mathrm{IIDeC}}$) festen γ_ν aus (2.12) nur vom Produkt $h\lambda$ abhängig, da auch $\frac{1}{h} \cdot \boldsymbol{K}$ nur von $h\lambda$ abhängig ist, und die $\boldsymbol{W}$-Matrizen und die Matrix $\widetilde{\boldsymbol{V}}$ von h und λ unabhängig sind. Wir verwenden daher die Schreibweisen $\boldsymbol{S}_{\mathrm{IDeC}} = \boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)$ und $\boldsymbol{S}_{\mathrm{IIDeC}} = \boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)$.

3.2.1 Spektralradius von $\boldsymbol{S}_{\mathbf{IDeC}}(h\lambda)$ bzw. $\boldsymbol{S}_{\mathbf{IIDeC}}(h\lambda)$

Bekanntlich ist für die Konvergenz der Iteration (3.6) hinreichend, daß für den Spektralradius

$$\rho(\boldsymbol{S}) := \max\{|\lambda| \colon \lambda \text{ Eigenwert von } \boldsymbol{S}\} \tag{3.53}$$

der Iterationsmatrix $\boldsymbol{S}$,

$$\rho(\boldsymbol{S}) < 1 \tag{3.54}$$

gilt. Umgekehrt folgt aus $\rho(\boldsymbol{S}) > 1$ zwar nicht notwendigerweise die Divergenz dieser Iteration (wenn der Startvektor $\boldsymbol{\eta}^{[0]}$ und der Vektor $\boldsymbol{v}$ keinen Anteil in Richtung von Eigenvektoren zu Eigenwerten λ von $\boldsymbol{S}$ mit $|\lambda| \geq 1$ haben, dann ist die Iteration konvergent), aber bei der Durchführung der Iteration auf einem Computer mit begrenzter Rechengenauigkeit werden in den $\boldsymbol{\eta}^{[k]}$ aufgrund von Rundungsfehlern mit großer Wahrscheinlichkeit bald Anteile in Richtung von Eigenvektoren zu Eigenwerten λ mit $|\lambda| > 1$ auftreten, die dann im Lauf der weiteren Iteration verstärkt werden, d.h. die Iteration (3.6) ist im Fall $\rho(\boldsymbol{S}) > 1$ numerisch instabil.

Bei der Untersuchung von $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda))$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda))$ darf man sich auf den Fall $N = 1$ beschränken, da $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda))$ und $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda))$ von N unabhängig sind. Das folgt aus der speziellen Block-Struktur der $\boldsymbol{W}$-Matrizen $\widetilde{\boldsymbol{W}}_1$, $\widetilde{\boldsymbol{W}}_1'$, $\widehat{\boldsymbol{W}}_1$, $\widehat{\boldsymbol{W}}_1'$ (vgl. (3.21)), der Matrix $\boldsymbol{V}$ (vgl. (3.38)) und der Matrix $\boldsymbol{K}$, die offensichtlich

die „Block-Dreiecks-Gestalt“

$$\boldsymbol{K} \;=\; \begin{pmatrix} K & & \\ & \ddots & \\ * & & K \end{pmatrix} \qquad (N\ K\text{-Blöcke})$$

mit

$$K \;:=\; h \cdot \begin{pmatrix} 1 & 0 & \dots & 0 \\ R(h\lambda) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ R(h\lambda)^{m-1} & R(h\lambda)^{m-2} & \dots & 1 \end{pmatrix} \otimes \Big(R_1(h\lambda), \dots, R_s(h\lambda) \Big)$$

hat (vgl. (3.48)). Die Matrizen $\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)$ bzw. $\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)$ haben damit die Gestalt

$$\boldsymbol{S} \;=\; \begin{pmatrix} S & & \\ & \ddots & \\ * & & S \end{pmatrix} \qquad (N\ S\text{-Blöcke}) \tag{3.55}$$

mit

$$S \;=\; S_{\mathrm{IIDeC}}(h\lambda) \;:=\; I_m - K \cdot \left(\frac{1}{h} \cdot \widetilde{W}_1' - \lambda \cdot \widetilde{W}_1 \right) \tag{3.56}$$

bzw.

$$S \;=\; S_{\mathrm{IIDeC}}(h\lambda) \;:=\; I_m - K \cdot \widetilde{V} \cdot \left(\frac{1}{h} \cdot \widehat{W}_1' - \lambda \cdot \widehat{W}_1 \right). \tag{3.57}$$

Für quadratische Matrizen der Gestalt (3.55) gilt

$$\det \boldsymbol{S} \;=\; (\det S)^N,$$

also gilt für das charakteristische Polynom

$$\chi_{\boldsymbol{S}}(x) \;=\; \det(\boldsymbol{S} - xI_{Nm}) \;=\; (\det(S - xI_m))^N \;=\; (\chi_S(x))^N,$$

da $\boldsymbol{S} - xI_{Nm}$ die zu (3.55) analoge Gestalt

$$\boldsymbol{S} - xI_{Nm} \;=\; \begin{pmatrix} S - xI_m & & \\ & \ddots & \\ * & & S - xI_m \end{pmatrix} \qquad (N\ (S - xI_m)\text{-Blöcke})$$

hat. Die Nullstellen von $\chi_{\boldsymbol{S}}(x)$ und $\chi_S(x)$, d.h. die Eigenwerte von $\boldsymbol{S}$ und S sind daher gleich, und somit ist der Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)) = \rho(S_{\mathrm{IDeC}}(h\lambda))$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) = \rho(S_{\mathrm{IIDeC}}(h\lambda))$ wie behauptet von N unabhängig.

Auf den folgenden Seiten finden sich nun mit Matlab[9] erzeugte Plots der Funktionen $z \mapsto \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ bzw. $z \mapsto \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $z \in [-1 \cdot 10^{10}, -1 \cdot 10^{-5}]$, wobei

[9] Die entsprechenden Matlab-Dateien findet man im Anhang A.2.

die horizontale z-Achse logarithmisch skaliert ist. Dabei ist jede Seite jeweils einem Basisverfahren aus Abschnitt 2.1.7 gewidmet. Die obere Abbildung zeigt jeweils die Spektralradien der zur klassischen Defektkorrektur mit $m = 3, 4, 5, 6$ gehörigen Matrizen $\boldsymbol{S}_{\mathrm{IDeC}}(z)$. In der mittleren bzw. unteren Abbildung findet man die Spektralradien der zur interpolierten Defektkorrektur gehörigen Matrizen $\boldsymbol{S}_{\mathrm{IIDeC}}(z)$, wobei in der mittleren Abbildung für die γ_j in (2.12) die Abszissen der m-stufigen Gauß-verfahren mit $m = 3, 4, 5, 6$, und in der unteren Abbildung die Abszissen der entsprechenden m-stufigen RadauIIA-Verfahren verwendet wurden.

Die Abbildungen legen die Vermutung nahe, daß immer $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) \to 0$ und $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) \to 0$ für $z \to 0$ gilt. Das ist tatsächlich der Fall, die Begründung dafür geben wir in Abschnitt 3.2.2. Es folgt die rasche Konvergenz der entsprechenden Defektkorrekturalgorithmen zu ihrem jeweiligen Fixpunkt, wenn sie auf nichtsteife Anfangswertprobleme (3.44) mit $\lambda = O(1)$ angewendet werden, und eine hinreichend kleine Schrittweite h verwendet wird.

Im Fall der klassischen Defektkorrektur (vgl. jeweils die obere Abbildung) beobachtet man, wenn nicht die Implizite Mittelpunktregel als Basisverfahren verwendet wird, daß $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) \to 0$ für $z \to -\infty$ gilt, wir begründen das in Abschnitt 3.2.3. Es folgt die rasche Konvergenz der entsprechenden Defektkorrekturalgorithmen bei Anwendung auf stark steife Anfangswertprobleme (3.44). Bei Verwendung der Impliziten Mittelpunktregel als Basisverfahren läßt die Abbildung 3.4 ein divergentes oder zumindest instabiles Verhalten der klassischen Defektkorrektur erwarten, wenn sie auf steife Anfangswertprobleme (3.44) angewendet wird. Dies wird durch numerische Experimente bestätigt, vgl. Abschnitt 3.4.2.

Im Fall der interpolierten Defektkorrektur ist bei Verwendung von IMR bzw. ITR als Basisverfahren jeweils $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) > 2$ für stark negative z zu beobachten. Die dadurch erwartete Instabilität der interpolierten Defektkorrektur angewendet auf steife Anfangswertprobleme (3.44) wird durch numerische Experimente bestätigt, vgl. Abschnitt 3.4.2. Bei Verwendung von IMR2 bzw. ITR2 als Basisverfahren beobachtet man $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) \approx 1$ für stark negative z. Tatsächlich gilt in diesem Fall, wie wir in Abschnitt 3.2.3 bestätigen werden, $\lim_{z\to-\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) = 1$. Numerische Experimente zeigen, daß in diesem Fall die auf steife Anfangswertprobleme (3.44) angewendete interpolierte Defektkorrektur nach wenigen Defektkorrekturschritten sich kaum mehr verändernde Approximationen $\eta_h^{[k]}$ liefert, die aber alle ungleich der Kollokationslösung sind, d.h. auch in diesem Fall tritt keine Konvergenz zu dieser Kollokationslösung ein.

Schließlich beobachtet man bei Verwendung von stark A-stabilen Basisverfahren wie dem Impliziten Eulerverfahren, SDIRK(2) oder RadauIIA(2), daß zumindest für die hier betrachteten $m = 3, 4, 5, 6$, $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) < 1$ für negative z gilt, woraus die Fixpunkt-Konvergenz der entsprechenden, auf steife Anfangswertprobleme (3.44) angewendeten interpolierten Defektkorrekturalgorithmen folgt.

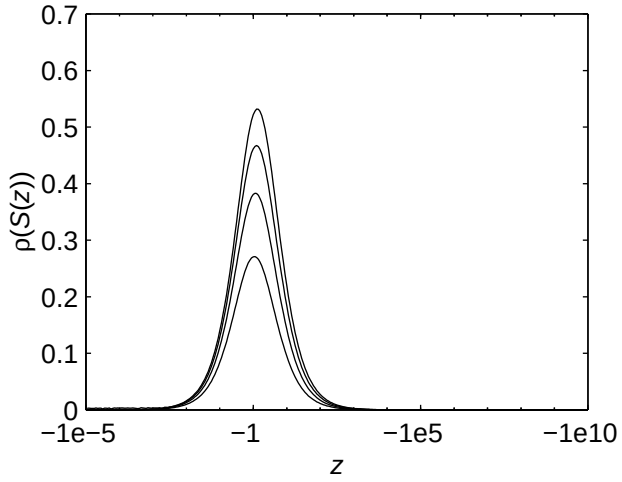

Abbildung 3.1: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung des Impliziten Eulerverfahren (2.23) als Basisverfahren.

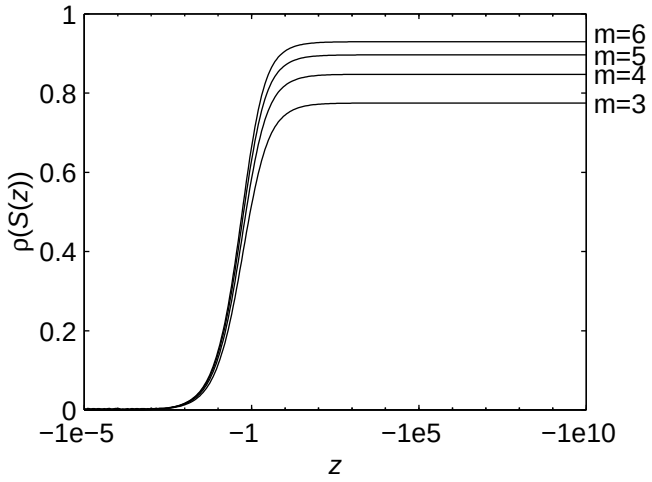

Abbildung 3.2: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und Implizitem Eulerverfahren (2.23) als Basisverfahren.

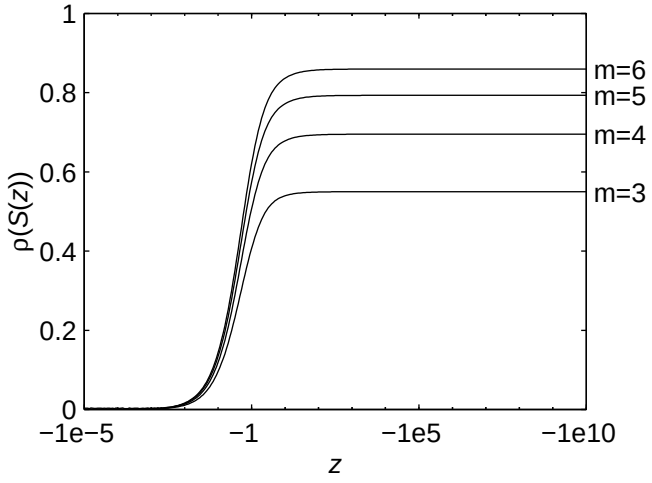

Abbildung 3.3: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und Implizitem Eulerverfahren (2.23) als

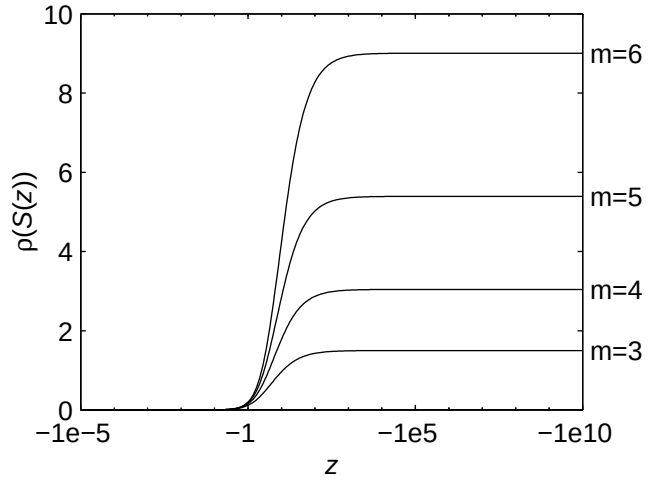

Abbildung 3.4: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von IMR (2.24) als Basisverfahren.

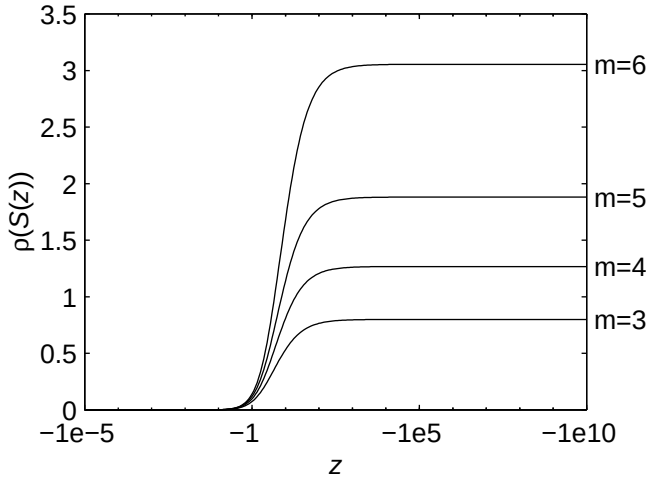

Abbildung 3.5: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und IMR (2.24) als Basisverfahren.

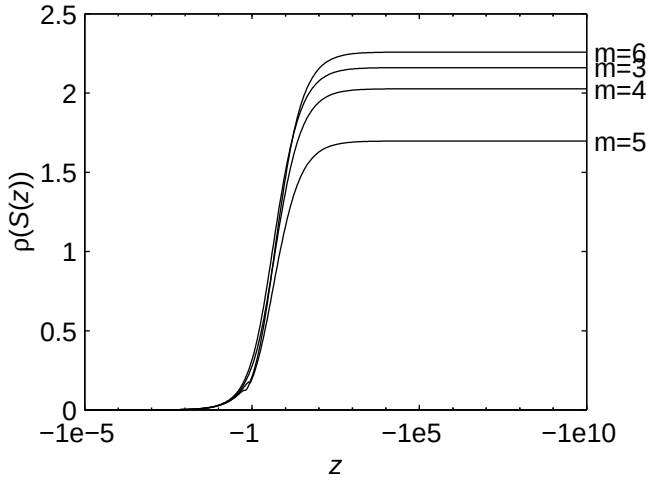

Abbildung 3.6: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und IMR (2.24) als Basisverfahren.

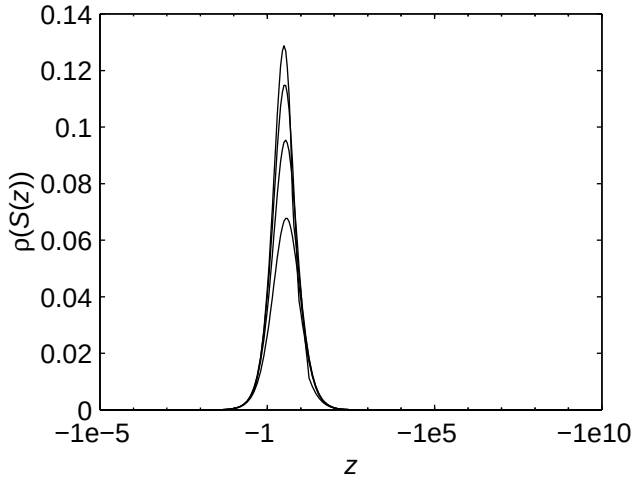

Abbildung 3.7: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von IMR2 (2.25) als Basisverfahren.

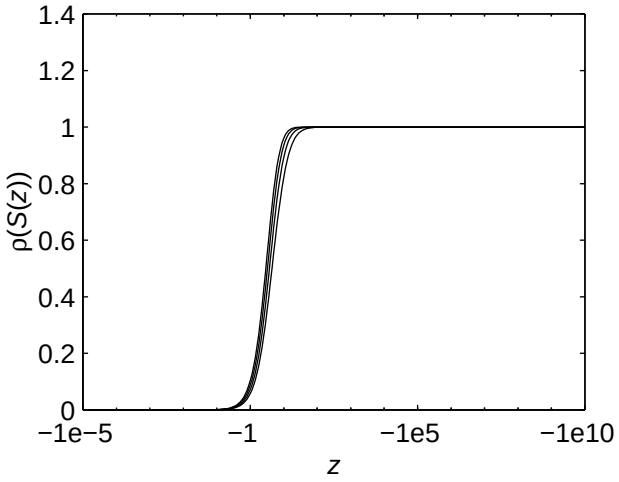

Abbildung 3.8: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und IMR2 (2.25) als Basisverfahren.

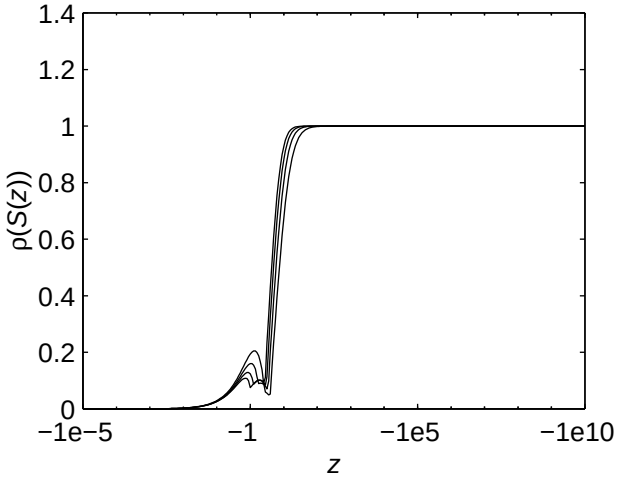

Abbildung 3.9: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und IMR2 (2.25) als Basisverfahren.

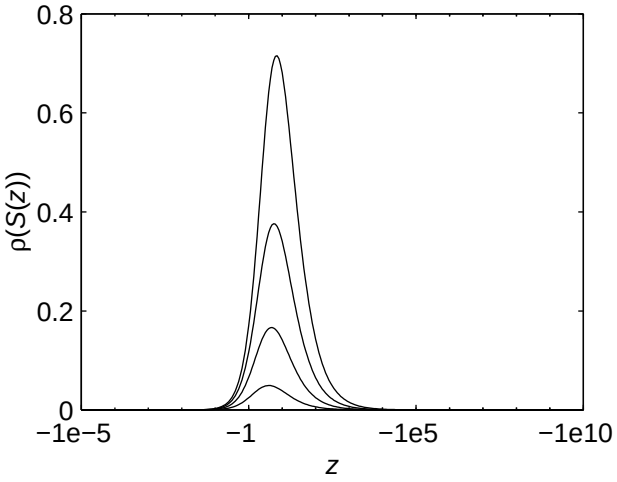

Abbildung 3.10: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von ITR (2.26) als Basisverfahren.

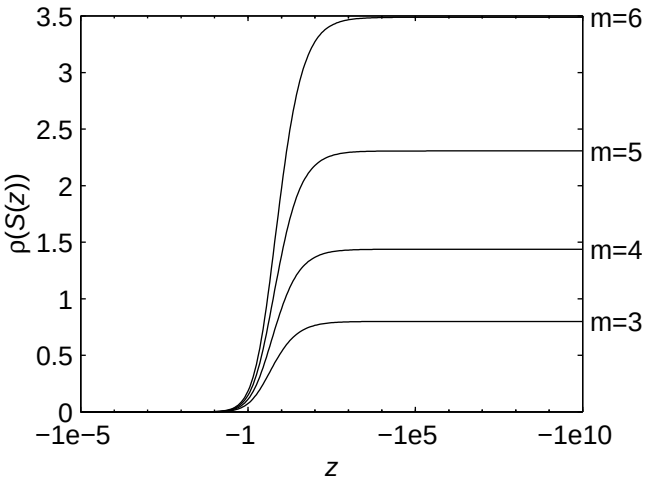

Abbildung 3.11: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und ITR (2.26) als Basisverfahren.

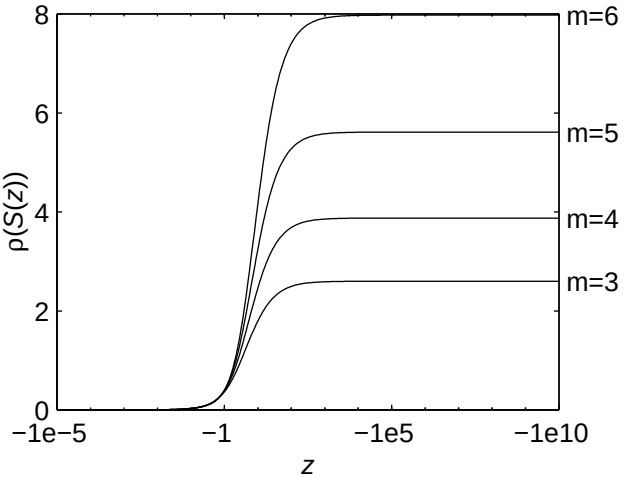

Abbildung 3.12: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und ITR (2.26) als Basisverfahren.

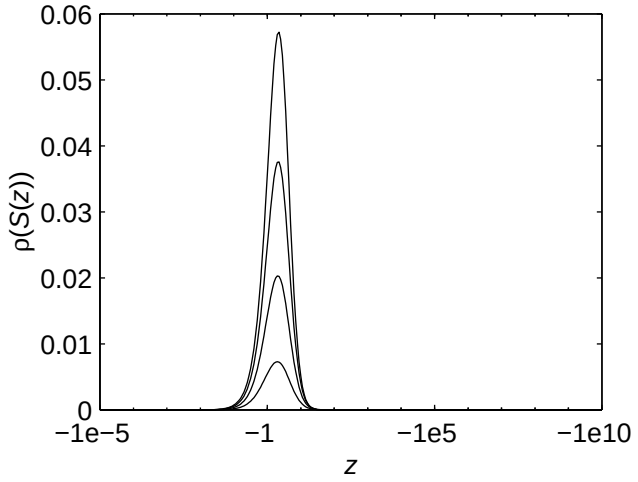

Abbildung 3.13: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von ITR2 (2.27) als Basisverfahren.

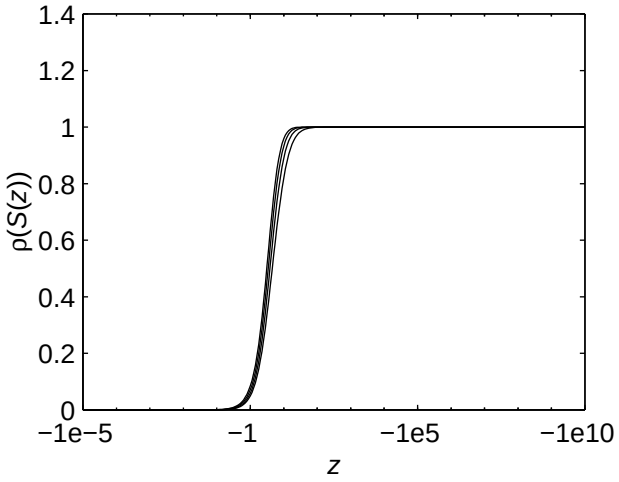

Abbildung 3.14: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und ITR2 (2.27) als Basisverfahren.

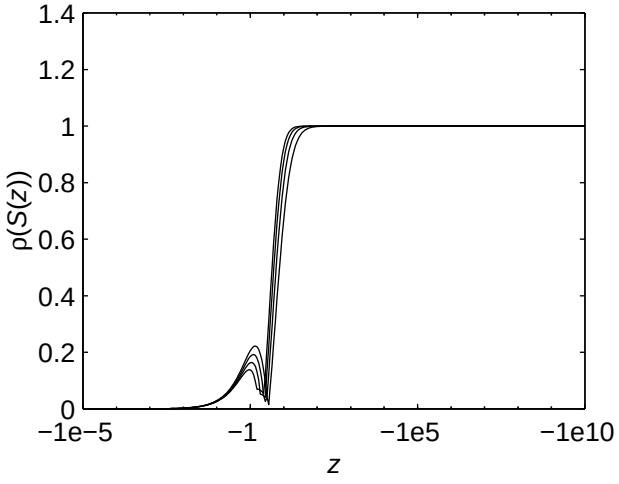

Abbildung 3.15: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und ITR2 (2.27) als Basisverfahren.

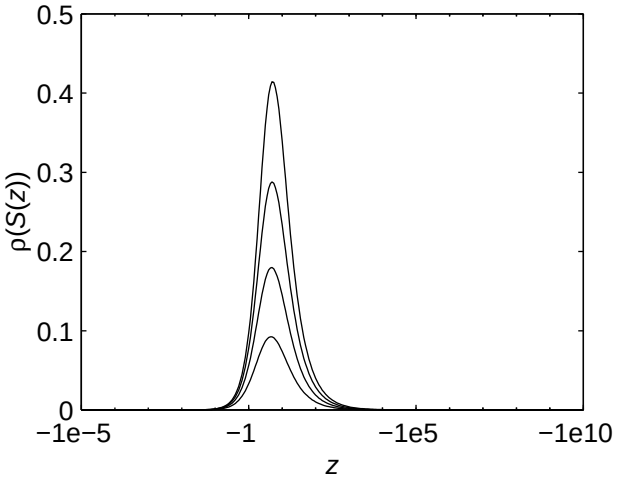

Abbildung 3.16: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von SDIRK(2) (2.28) als Basisverfahren.

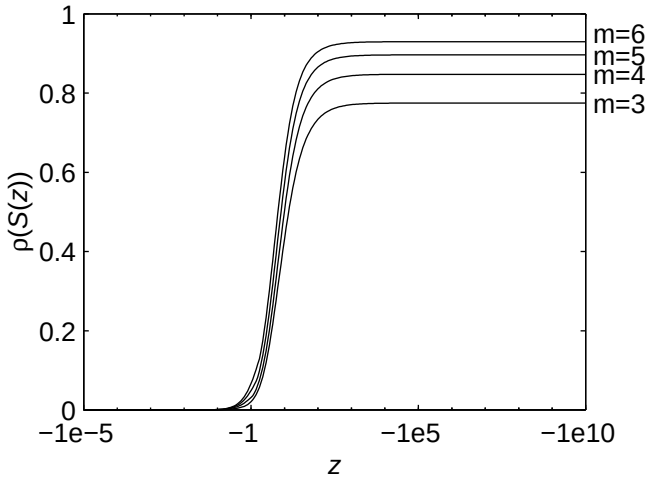

Abbildung 3.17: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und SDIRK(2) (2.28) als Basisverfahren.

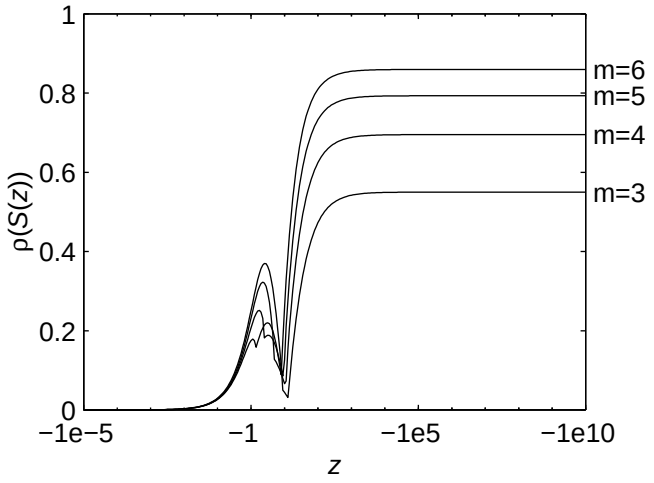

Abbildung 3.18: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und SDIRK(2) (2.28) als Basisverfahren.

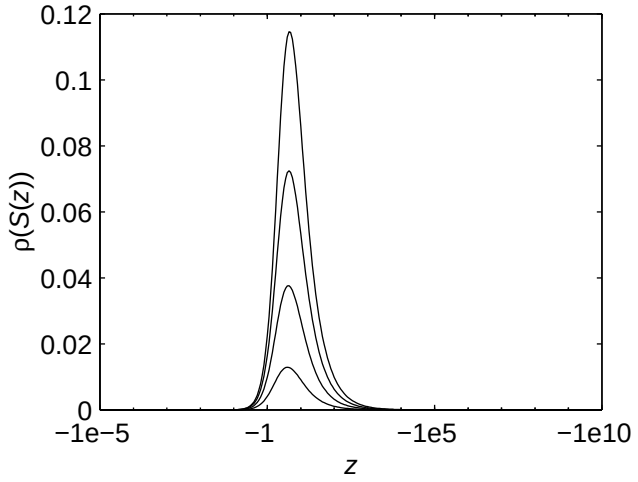

Abbildung 3.19: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ für $m = 3, 4, 5, 6$ bei Verwendung von RadauIIA(2) (2.29) als Basisverfahren.

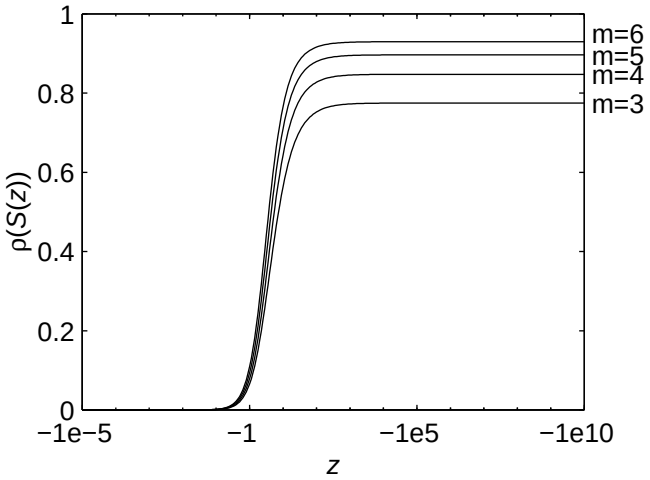

Abbildung 3.20: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von Gauß-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und RadauIIA(2) (2.29) als Basisverfahren.

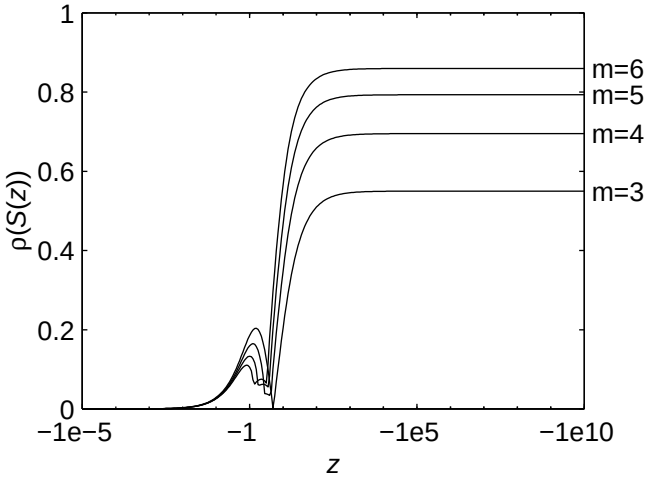

Abbildung 3.21: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ bei Verwendung von RadauIIA-Abszissen ($m = 3, 4, 5, 6$) für (2.12) und RadauIIA(2) (2.29) als Basisverfahren.

3.2.2 Analyse für $h\lambda \to 0$ (nichtsteifer Fall)

Satz 3.2.1. *Das Basisverfahren (2.19) erfülle die Butcher-Bedingung*[10]

$$B(p): \quad \sum_{j=1}^{s} b_j c_j^{i-1} \;=\; \frac{1}{i}, \quad i = 1, \ldots, p. \tag{3.58}$$

Dann ist $\boldsymbol{S}_{\mathrm{IIDeC}}(0) = \boldsymbol{S}_{\mathrm{IDeC}}(0) =: \boldsymbol{S}_0$ *nilpotent der Ordnung*[11] $\lceil \frac{m}{p} \rceil$*, d.h.* $\boldsymbol{S}_0^{\lceil \frac{m}{p} \rceil} = 0$.

Beweis. Im Fall $\lambda = 0$ reduziert sich (3.44) auf das Quadraturproblem

$$y' \;=\; g(t), \quad y(t_0) = y_0.$$

Da $\boldsymbol{S}_0$ von $g(t)$ und y_0 unabhängig ist, genügt es, das triviale Problem

$$y' \;=\; 0, \quad y(t_0) = 0 \tag{3.59}$$

zu betrachten. Für dieses Problem reduziert sich wegen $\boldsymbol{v} = 0$ die Iteration (3.6) auf

$$\boldsymbol{\eta}^{[k+1]} = \boldsymbol{S}_0 \cdot \boldsymbol{\eta}^{[k]}.$$

Wir haben zu zeigen, daß für beliebiges $\boldsymbol{\eta}^{[k]} \in \mathbb{R}^{Nm}$, $\boldsymbol{\eta}^{[k+\lceil \frac{m}{p} \rceil]} = \boldsymbol{S}_0^{\lceil \frac{m}{p} \rceil} \cdot \boldsymbol{\eta}^{[k]} = 0$ gilt, oder äquivalenterweise, daß für beliebiges $\eta_h^{[k]} \in \mathcal{E}_h^0$, $\lceil \frac{m}{p} \rceil$ IDeC-Schritte angewendet auf (3.59), $\eta_h^{[k+\lceil \frac{m}{p} \rceil]} = 0$ ergeben. Zum Beweis zeigen wir, daß wenn

$$\mathrm{grad} P_\ell^{[k]}(t) \le q \le m, \quad \ell = 0, \ldots, N-1$$

für $P^{[k]}(t) := (P_h \eta^{[k]})(t)$ gilt (die $P_\ell^{[k]}(t)$ seien gemäß (2.7) die Polynom-Stücke, aus denen $P^{[k]}(t)$ besteht), dann für $P^{[k+1]}(t) := (P_h \eta^{[k+1]})(t)$,

$$\mathrm{grad} P_\ell^{[k+1]}(t) \le \max(0, q-p), \quad \ell = 0, \ldots, N-1 \tag{3.60}$$

gilt. Damit reduziert sich dann der Grad von $P^{[k]}(t)$ pro IDeC-Schritt um p, nach maximal $\lceil \frac{m}{p} \rceil$ IDeC-Schritten ist der Grad 0 erreicht, d.h. $P^{[k+\lceil \frac{m}{p} \rceil]}(t)$ ist konstant und $\equiv 0$ wegen $P^{[k+\lceil \frac{m}{p} \rceil]}(t_0) = y_0 = 0$.

Der (interpolierte) Defekt bezüglich $\eta_h^{[k]}$ ist jetzt $d^{[k]}(t) = \frac{d}{dt} P^{[k]}(t)$, das Nachbarproblem hat daher die Gestalt

$$y' \;=\; \frac{d}{dt} P^{[k]}(t), \quad y(t_0) = 0.$$

[10] Für Näheres zu den vereinfachenden Bedingungen von Butcher vgl. [9, Abschnitt IV.5]. Von den IRK-Verfahren aus Abschnitt 2.1.7 erfüllt RadauIIA(3) $B(p)$ mit $p = 3$; $B(p)$ mit $p = 2$ erfüllen IMR, IMR2, ITR, ITR2 und SDIRK(2); und das Implizite Eulerverfahren erfüllt $B(p)$ mit $p = 1$.

[11] $\lceil x \rceil$ ist die kleinste ganze Zahl $\ge x$.

Wenn wir darauf das Basisverfahren (2.19) anwenden, ergibt sich

$$\begin{aligned}
&\pi_{0,0}^{[k]} = 0, \\
&\pi_{\ell,\nu+1}^{[k]} \;=\; \pi_{\ell,\nu}^{[k]} + h\sum_{j=1}^{s} b_j \frac{d}{dt} P_\ell^{[k]}(t_{\ell,\nu} + c_j h), \qquad (3.61)\\
&\qquad\qquad \ell = 0,\ldots,N-1, \quad \nu = 0,\ldots,m-1,
\end{aligned}$$

und es gilt unter Berücksichtigung der Tatsache, daß das auf das Originalproblem (3.59) angewendete Basisverfahren $\eta_h^{[0]} = 0$ liefert,

$$\eta_h^{[k+1]} \;=\; \eta_h^{[0]} - (\pi_h^{[k]} - \eta_h^{[k]}) \;=\; \eta_h^{[k]} - \pi_h^{[k]},$$

also

$$P_\ell^{[k+1]}(t) \;=\; P_\ell^{[k]}(t) - Q_\ell^{[k]}(t), \quad \ell = 0,\ldots,N-1,$$

wobei $Q_\ell^{[k]}(t),\ \ell = 0,\ldots,N-1$ jeweils das Polynom vom Grad m ist, das die Punkte $(t_{\ell,0}, \pi_{\ell,0}^{[k]}),\ldots,(t_{\ell,m}, \pi_{\ell,m}^{[k]})$ interpoliert.

Sei ∇_h der auf der Menge der Polynome durch $\nabla_h P(t) := P(t+h) - P(t)$ definierte Rückwärtsdifferenzen-Operator. Dieser Operator hat die einfach zu verifizierende Eigenschaft

$$\mathrm{grad} P(t) \le q \;\Leftrightarrow\; \mathrm{grad}\nabla_h P(t) \le q-1,$$

woraus folgt, daß für (3.60) im Fall $q \ge p+1$, $\mathrm{grad}\nabla_h P_\ell^{[k+1]}(t) \le q-p-1$ und andernfalls $\nabla_h P_\ell^{[k+1]}(t) \equiv 0$ zu zeigen ist. Aus der Definition von $Q_\ell^{[k]}(t)$ und (3.61) folgt

$$\nabla_h Q_\ell^{[k]}(t_{\ell,\nu}) \;=\; \pi_{\ell,\nu+1}^{[k]} - \pi_{\ell,\nu}^{[k]} \;=\; h\sum_{j=1}^{s} b_j \frac{d}{dt} P_\ell^{[k]}(t_{\ell,\nu} + c_j h), \quad \nu = 0,\ldots,m-1,$$

d.h. das Polynom $\nabla_h Q_\ell^{[k]}(t)$ stimmt mit dem Polynom $h\sum_{j=1}^{s} b_j \frac{d}{dt} P_\ell^{[k]}(t + c_j h)$ an den m Stellen $t_{\ell,0},\ldots,t_{\ell,m-1}$ überein, und daher gilt, da beide Polynome einen Grad $\le m-1$ haben,

$$\nabla_h Q_\ell^{[k]}(t) \;\equiv\; h\sum_{j=1}^{s} b_j \frac{d}{dt} P_\ell^{[k]}(t + c_j h).$$

Durch Entwickeln nach Taylor ergibt sich für $j = 1,\ldots,s$

$$\frac{d}{dt} P_\ell^{[k]}(t + c_j h) \;=\; \frac{d}{dt} P_\ell^{[k]}(t) + c_j h \frac{d^2}{dt^2} P_\ell^{[k]}(t) + \ldots + \frac{c_j^{p-1} h^{p-1}}{(p-1)!} \frac{d^p}{dt^p} P_\ell^{[k]}(t) + R_{\ell,j}^{[k]}(t)$$

mit $\mathrm{grad} R_{\ell,j}^{[k]}(t) \le q-p-1$ für $q \ge p+1$ und $R_{\ell,j}^{[k]}(t) \equiv 0$ sonst, also unter Verwendung von (3.58)

$$\begin{aligned}
\nabla_h Q_\ell^{[k]}(t) &= h\sum_{j=1}^{s} b_j \frac{d}{dt} P_\ell^{[k]}(t+c_j h) \\
&= h\underbrace{\left(\sum_{j=1}^{s} b_j\right)}_{=1} \frac{d}{dt} P_\ell^{[k]}(t) + h^2 \underbrace{\left(\sum_{j=1}^{s} b_j c_j\right)}_{=\frac{1}{2}} \frac{d^2}{dt^2} P_\ell^{[k]}(t) + \ldots \\
&\qquad \ldots + \frac{h^p}{(p-1)!} \underbrace{\left(\sum_{j=1}^{s} b_j c_j^{p-1}\right)}_{=\frac{1}{p}} \frac{d^p}{dt^p} P_\ell^{[k]}(t) + h\sum_{j=1}^{s} b_j R_{\ell,j}^{[k]}(t) \\
&= h\frac{d}{dt} P_\ell^{[k]}(t) + \frac{h^2}{2}\frac{d^2}{dt^2} P_\ell^{[k]}(t) + \ldots + \frac{h^p}{p!}\frac{d^p}{dt^p} P_\ell^{[k]}(t) + h\sum_{j=1}^{s} b_j R_{\ell,j}^{[k]}(t).
\end{aligned}$$

Andererseits gilt

$$\begin{aligned}
\nabla_h P_\ell^{[k]}(t) &= P_\ell^{[k]}(t+h) - P_\ell^{[k]}(t) \\
&= h\frac{d}{dt} P_\ell^{[k]}(t) + \frac{h^2}{2}\frac{d^2}{dt^2} P_\ell^{[k]}(t) + \ldots + \frac{h^p}{p!}\frac{d^p}{dt^p} P_\ell^{[k]}(t) + R_\ell^{[k]}(t)
\end{aligned}$$

mit $\mathrm{grad} R_\ell^{[k]}(t) \le q-p-1$ für $q \ge p+1$ und $R_\ell^{[k]}(t) \equiv 0$ sonst. Die Subtraktion der Ausdrücke für $\nabla_h P_\ell^{[k]}(t)$ bzw. $\nabla_h Q_\ell^{[k]}(t)$ ergibt

$$\nabla_h P_\ell^{[k+1]}(t) = \nabla_h P_\ell^{[k]}(t) - \nabla_h Q_\ell^{[k]}(t) = R_\ell^{[k]}(t) - h\sum_{j=1}^{s} b_j R_{\ell,j}^{[k]}(t),$$

d.h. der Grad von $\nabla_h P_\ell^{[k+1]}(t)$ ist $\le q-p-1$ für $q \ge p+1$ und $\nabla_h P_\ell^{[k+1]}(t) \equiv 0$ sonst, und es folgt $\mathrm{grad} P_\ell^{[k+1]}(t) \le \max(0, q-p)$, d.h. die zu zeigende Behauptung (3.60). □

Folgerung. *Für hinreichend kleine $|h\lambda|$ konvergiert die auf (3.44) angewendete IDeC- bzw. IIDeC-Iteration.*

Beweis. Für jedes „vernünftige“ Basisverfahren gilt zumindest $B(p)$ mit $p = 1$,[12] daher ist $\boldsymbol{S}_0$ nilpotent der Ordnung $\le m$, und für den Spektralradius $\rho(\boldsymbol{S}_0)$ gilt $\rho(\boldsymbol{S}_0) = 0$. Wegen der stetigen Abhängigkeit von $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ von z gilt daher $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) < 1$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) < 1$ für hinreichend kleine $|z|$, woraus für diese $z = h\lambda$ die Konvergenz der Iteration (3.6) und damit die Behauptung folgt. □

[12] $B(p)$ mit $p = 1$ ist zur Forderung äquivalent, daß das Anfangswertproblem $y' = 1,\ y(t_0) = y_0$ durch das Basisverfahren exakt integriert wird.

3.2.3 Analyse für $|h\lambda| \to \infty$ (stark steifer Fall)

In diesem Unterabschnitt setzen wir voraus, daß für das betrachtete Basisverfahren (2.19) die zugehörigen Funktionen $R_j(z)$ aus (3.46)

$$\lim_{z\to\infty} R_j(z) = 0, \quad j = 1, \ldots, s \tag{3.62}$$

erfüllen. Diese Bedingung ist, wie wir gleich sehen werden, für alle Runge-Kutta-Verfahren (2.19) mit regulärer Koeffizientenmatrix A erfüllt. Sie gilt aber auch z.B. für ITR (2.26) und ITR2 (2.27) und damit für alle in Abschnitt 2.1.7 angegebenen Verfahren, allgemein gilt sie jedoch nicht.[13] Wir setzen

$$r_j := \lim_{z\to\infty} zR_j(z), \quad j = 1, \ldots, s \tag{3.63}$$

$$r := \lim_{z\to\infty} R(z) = 1 + \sum_{j=1}^{s} r_j. \tag{3.64}$$

Hier folgt das letzte Gleichheitszeichen aus der Beziehung

$$R(z) = 1 + z\sum_{j=1}^{s} R_j(z),$$

die man durch einfache Umformungen von Determinanten unter Verwendung der üblicherweise erfüllten Konsistenzbedingung[11] $B(1): \sum_{j=1}^{s} b_j = 1$ erhält. Wegen der Voraussetzung (3.62) sind die Definitionen (3.63) und (3.64) sinnvoll. Wenn nun A regulär ist, dann gilt, wie man durch Umformen von Determinanten bestätigt,

$$\begin{aligned} r_j &= \lim_{z\to\infty} zR_j(z) \\ &= \lim_{z\to\infty} \frac{1}{\det\left[\frac{1}{z}I_s - A\right]} \cdot \det \left(\begin{array}{c|c} \frac{1}{z}I_s - A & \begin{smallmatrix}0\\ \vdots \\ 1 \\ \vdots \\ 0\end{smallmatrix} \\ \hline -b^T & 0 \end{array}\right) \\ &= \frac{1}{\det A} \cdot \det \left(\begin{array}{c|c} A & \begin{smallmatrix}0\\ \vdots \\ 1 \\ \vdots \\ 0\end{smallmatrix} \\ \hline b^T & 0 \end{array}\right), \qquad j = 1, \ldots, s. \end{aligned}$$

[13] Ein Gegenbeispiel ist LobattoIIIB(2),

$$\begin{array}{c|cc} 0 & \frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

mit $\lim_{z\to\infty} R_1(z) = -\frac{1}{2}$ und $\lim_{z\to\infty} R_2(z) = \frac{1}{2}$.

Basisverfahren	r	r_1	r_2	r_3
Imp. Euler	0	-1		
IMR	-1	-2		
IMR2	1	2	-2	
ITR	-1	-1	-1	
ITR2	1	1	0	-1
SDIRK(2)	0	0	-1	
RadauIIA(2)	0	0	-1	

Tabelle 3.2: r und $r_j,\ j = 1, \ldots, s$ für die IRK-Verfahren aus Abschnitt 2.1.7.

(Hier steht der Einser in der letzten Spalte jeweils in der j-ten Zeile.) Daraus folgt (3.62) für RK-Verfahren mit regulärer Matrix A. Außerdem erhält man durch Anwendung der Cramerschen Regel die Beziehung

$$(r_1, \ldots, r_s) \ = \ -b^T \cdot A^{-1}. \tag{3.65}$$

Sei $\boldsymbol{K}$ die Matrix aus (3.48). Für $h\lambda \to \infty$ konvergiert die Matrix $\frac{1}{h} \cdot \boldsymbol{K}$ wegen (3.62) gegen eine Nullmatrix und die Matrix $\lambda \cdot \boldsymbol{K}$ gegen die Matrix

$$\boldsymbol{K}_\infty \ := \ \begin{pmatrix} 1 & 0 & \cdots & 0 \\ r & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ r^{Nm-1} & r^{Nm-2} & \cdots & 1 \end{pmatrix} \otimes (r_1, \ldots, r_s), \tag{3.66}$$

daher gilt

$$\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IDeC}}(z) \ = \ I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{W}}_1, \tag{3.67}$$

$$\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z) \ = \ I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{V}} \cdot \widehat{\boldsymbol{W}}_1. \tag{3.68}$$

Im Anhang A.4 findet sich ein Maple-Arbeitsblatt, in dem die Werte des Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ für $z \to \infty$, also die Werte $\rho(I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{W}}_1)$ bzw. $\rho(I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{V}} \cdot \widehat{\boldsymbol{W}}_1)$ für die Basisverfahren aus Abschnitt 2.1.7 bei Verwendung von Gauß- bzw. RadauIIA-Abszissen für die γ_j in (2.12) berechnet werden. Ohne Beschränkung der Allgemeinheit (vgl. Abschnitt 3.2.1) wird dort nur der Fall $N = 1$ betrachtet. In Tabelle 3.3 sind die erhaltenen Ergebnisse zusammengefaßt.

Wir untersuchen nun genauer die Matrizen (3.67) und (3.68) für die einzelnen Basisverfahren in Abschnitt 2.1.7, wobei wir für einige der Werte aus Tabelle 3.3 eine Begründung bzw. eine explizite Formel erhalten werden.

Basisverfahren	m	$\lim_{z\to\infty} \rho(\boldsymbol{S}_{\text{(I)IDeC}}(z))$		
		klassisch	Gauß	RadauIIA
Impl. Euler, SDIRK(2) und RadauIIA(2)	3	0.000000	0.775000	0.550000
	4	0.000000	0.847619	0.695238
	5	0.000000	0.896660	0.793320
	6	0.000000	0.929870	0.859740
IMR	3	1.500000	0.800000	2.161187
	4	3.042840	1.267732	2.027420
	5	5.393228	1.881461	1.697756
	6	9.008910	3.054852	2.259314
IMR2	3	0.000000	1.000000	1.000000
	4	0.000000	1.000000	1.000000
	5	0.000000	1.000000	1.000000
	6	0.000000	1.000000	1.000000
ITR	3	0.000000	0.800000	2.600000
	4	0.000000	1.438095	3.876190
	5	0.000000	2.306878	5.613757
	6	0.000000	3.488312	7.976623
ITR2	3	0.000000	1.000000	1.000000
	4	0.000000	1.000000	1.000000
	5	0.000000	1.000000	1.000000
	6	0.000000	1.000000	1.000000

Tabelle 3.3: $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\text{IDeC}}(z))$ und $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\text{IIDeC}}(z))$ für die Basisverfahren aus Abschnitt 2.1.7 bei Verwendung von Gauß- bzw. RadauIIA-Abszissen für (2.12).

Implizites Eulerverfahren, SDIRK(2) und RadauIIA(2): Wir betrachten allgemeiner Basisverfahren (2.19), die die Bedingungen

$$A \text{ ist regulär,} \tag{3.69}$$
$$b^T = (s\text{-te Zeile von } A), \quad \text{und} \tag{3.70}$$
$$c_s = 1 \tag{3.71}$$

erfüllen. Diese Bedingungen werden z.B. von allen RadauIIA-Verfahren, also insbesondere vom Impliziten Eulerverfahren (2.23) und von RadauIIA(2) (2.29) erfüllt. Weiters genügen SDIRK(2) (2.28) und alle LobattoIIIC-Verfahren diesen Bedingungen.

Aus (3.65), (3.70) und (3.64) folgt jetzt $(r_1, \ldots, r_{s-1}, r_s) = (0, \ldots, 0, -1)$ und $r = 0$, und daher

$$-\boldsymbol{K}_\infty = I_{Nm} \otimes (0, \ldots, 0, 1). \tag{3.72}$$

Seien $\widetilde{W}_0$ und $\widetilde{W}_1$ die Matrizen aus (3.16). Wegen $c_s = 1$ gilt (vgl. (2.63))

$$\widetilde{w}_{i,s,j} \;=\; \begin{cases} 1 & \text{für } i = j, \\ 0 & \text{sonst,} \end{cases} \qquad i = 1, \ldots, m,\ j = 0, \ldots, m,$$

und daher

$$\big(I_m \otimes (0, \ldots, 0, 1)\big) \cdot \widetilde{W}_0 \;=\; \begin{pmatrix} \widetilde{w}_{1,s,0} \\ \vdots \\ \widetilde{w}_{m,s,0} \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

und

$$\big(I_m \otimes (0, \ldots, 0, 1)\big) \cdot \widetilde{W}_1 \;=\; \begin{pmatrix} \widetilde{w}_{1,s,1} & \cdots & \widetilde{w}_{1,s,m} \\ \vdots & & \vdots \\ \widetilde{w}_{m,s,1} & \cdots & \widetilde{w}_{m,s,m} \end{pmatrix} \;=\; I_m.$$

Unter Beachtung der speziellen Blockstruktur (3.21) von $\widetilde{\boldsymbol{W}}_1$ gilt daher

$$\begin{aligned} -\boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{W}}_1 &= \operatorname{blockdiag}\big(I_m \otimes (0, \ldots, 0, 1), \ldots, I_m \otimes (0, \ldots, 0, 1)\big) \cdot \widetilde{\boldsymbol{W}}_1 \\ &= \operatorname{blockdiag}(I_m, \ldots, I_m) \\ &= I_{Nm}, \end{aligned}$$

und es folgt

$$\lim_{z \to \infty} \boldsymbol{S}_{\mathrm{IDeC}}(z) \;=\; I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{W}}_1 \;=\; (0).$$

Wir haben somit den folgenden Satz gezeigt:

Satz 3.2.2. *Wenn das Basisverfahren (2.19) die Bedingungen (3.69), (3.70) und (3.71) erfüllt, dann konvergiert die Iterationsmatrix $\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)$ der auf (3.44) angewendeten klassischen Defektkorrektur für $h\lambda \to \infty$ gegen eine Nullmatrix.*

Für stark steife ($\lambda \ll 0$) Differentialgleichungen (3.44) hat dieser Satz die sehr rasche Konvergenz der klassischen Defektkorrekturiteration gegen ihren Fixpunkt zur Folge, was durch numerische Experimente (vgl. Abschnitt 3.4.2) bestätigt wird. Wir kommen nun zum Fall der interpolierten Defektkorrektur:

Satz 3.2.3. *Das Basisverfahren (2.19) erfülle die Bedingungen (3.69), (3.70) und (3.71). Dann gilt*

$$\lim_{h\lambda \to \infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) \;=\; \left| 1 - \frac{m^m}{m!} \prod_{j=1}^{m} \gamma_j \right| \tag{3.73}$$

für den Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda))$ der Iterationsmatrix der auf (3.44) angewendeten interpolierten Defektkorrektur.

Beweis. Da $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ von N unabhängig ist (vgl. Abschnitt 3.2.1), genügt es, den Fall $N = 1$ zu betrachten. In diesem Fall gilt (vgl. 3.72)

$$-\boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{V}} \;=\; \big(I_m \otimes (0,\ldots,0,1)\big) \cdot \widetilde{V} \;=\; \begin{pmatrix} \widetilde{v}_{1,s,1} & \cdots & \widetilde{v}_{1,s,m} \\ \vdots & & \vdots \\ \widetilde{v}_{m,s,1} & \cdots & \widetilde{v}_{m,s,m} \end{pmatrix} \;=\; \widetilde{V}_{\mathrm{E}}.$$

Hier ist $\widetilde{V}_{\mathrm{E}}$ die Matrix $\widetilde{V}$ (3.36) für das Implizite Eulerverfahren, d.h. für $s = 1$ und $c_1 = 1$, und das letzte Gleichheitszeichen folgt wegen $c_s = 1$ aus der Definition (2.74) der $\widetilde{v}_{i,s,j}$. Aus (3.20) und (3.37) folgt mit $p_k(t) := t^k$ für $k = 1,\ldots,m-1$

$$\begin{aligned} \widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1 \cdot \begin{pmatrix} \left(\frac{1}{m}\right)^k \\ \vdots \\ \left(\frac{m}{m}\right)^k \end{pmatrix} \;&=\; \widetilde{V}_{\mathrm{E}} \cdot [\widehat{W}_0|\ \widehat{W}_1\ ] \cdot \begin{pmatrix} 0 \\ \hline \left(\frac{1}{m}\right)^k \\ \vdots \\ \left(\frac{m}{m}\right)^k \end{pmatrix} \;=\; \widetilde{V}_{\mathrm{E}} \cdot [\widehat{W}_0|\ \widehat{W}_1\ ] \cdot \begin{pmatrix} p_k(0) \\ p_k(\frac{1}{m}) \\ \vdots \\ p_k(\frac{m}{m}) \end{pmatrix} \\ &=\; \widetilde{V}_{\mathrm{E}} \cdot \begin{pmatrix} p_k(\gamma_1) \\ \vdots \\ p_k(\gamma_m) \end{pmatrix} \;=\; \begin{pmatrix} p_k(\frac{1}{m}) \\ \vdots \\ p_k(\frac{m}{m}) \end{pmatrix} \;=\; \begin{pmatrix} \left(\frac{1}{m}\right)^k \\ \vdots \\ \left(\frac{m}{m}\right)^k \end{pmatrix}, \end{aligned}$$

d.h. die Vektoren $\big((\frac{1}{m})^k,\ldots,(\frac{m}{m})^k\big)^T$, $k = 1,\ldots,m-1$ sind $m-1$ linear unabhängige Eigenvektoren von $\widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert 1, und damit sind diese Vektoren $m-1$ linear unabhängige Eigenvektoren von $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z) = I_m - \widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert 0.

Wir zeigen nun, daß für $\gamma_j \neq \frac{j}{m},\ j = 1,\ldots,m$,[14]

$$\boldsymbol{x} \;:=\; \left(\prod_{j=1}^{m}\left(\frac{1}{m}-\gamma_j\right),\ldots,\prod_{j=1}^{m}\left(\frac{m}{m}-\gamma_j\right)\right)^T \tag{3.74}$$

ein von der Menge $\big\{\big((\frac{1}{m})^k,\ldots,(\frac{m}{m})^k\big)^T\colon\ k = 1,\ldots,m-1\big\}$ linear unabhängiger Eigenvektor von $\widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert $\frac{m^m}{m!}\prod_{j=1}^m \gamma_j$, und folglich ein Eigenvektor von $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z) = I_m - \widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert $1 - \frac{m^m}{m!}\prod_{j=1}^m \gamma_j$ ist. Damit ist $\Big\{1 - \frac{m^m}{m!}\prod_{j=1}^m \gamma_j,\ 0\Big\}$ die Menge *aller* Eigenwerte von $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z)$, woraus unmittelbar die Behauptung (3.73) folgt.

Wir definieren das Polynom $p(t) := \prod_{j=1}^m (t - \gamma_j)$ vom Grad m, womit $\boldsymbol{x} =$

[14] Im Fall $\gamma_j = \frac{j}{m},\ j = 1,\ldots,m$, wo der Vektor (3.74) der Nullvektor ist, gilt offenbar $\widehat{W}_1 = \widetilde{V}_E = I_m$, daher $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z) = I_m - \widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1 = (0)$, und die Formel (3.73) liefert den richtigen Wert $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) = 0$.

$\big(p(\frac{1}{m}),\ldots,p(\frac{m}{m})\big)^T$ gilt. Wegen (3.20) gilt

$$[\widehat{W}_0|\ \widehat{W}_1\ ]\cdot\begin{pmatrix}p(0)\\ \hline p(\frac{1}{m})\\ \vdots\\ p(\frac{m}{m})\end{pmatrix}\ =\ \begin{pmatrix}p(\gamma_1)\\ \vdots\\ p(\gamma_m)\end{pmatrix}\ =\ \begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix},$$

also

$$\widehat{W}_1\cdot\boldsymbol{x}\ =\ \widehat{W}_1\cdot\begin{pmatrix}p(\frac{1}{m})\\ \vdots\\ p(\frac{m}{m})\end{pmatrix}\ =\ -p(0)\cdot\widehat{W}_0\ =\ -(-1)^m\prod_{j=1}^{m}\gamma_j\cdot\widehat{W}_0.$$

Aus der Definition (2.69) der $\widehat{w}_{i,0}$ folgt

$$\widehat{W}_0\ =\ \begin{pmatrix}\widehat{w}_{1,0}\\ \vdots\\ \widehat{w}_{m,0}\end{pmatrix}\ =\ (-1)^m\frac{m^m}{m!}\cdot\begin{pmatrix}\prod_{j=1}^{m}(\gamma_1-\frac{j}{m})\\ \vdots\\ \prod_{j=1}^{m}(\gamma_m-\frac{j}{m})\end{pmatrix},$$

und daher

$$\widehat{W}_1\cdot\boldsymbol{x}\ =\ -\frac{m^m}{m!}\prod_{j=1}^{m}\gamma_j\cdot\begin{pmatrix}q(\gamma_1)\\ \vdots\\ q(\gamma_m)\end{pmatrix}\tag{3.75}$$

mit $q(t):=\prod_{j=1}^{m}(t-\frac{j}{m})$. Sei $Q(t)$ das Polynom vom Grad $m-1$, das die Funktion $q(t)$ an den Stellen $\gamma_1,\ldots,\gamma_m$ interpoliert. Für die wohlbekannte Darstellung des Interpolationsrestgliedes

$$q(t)-Q(t)\ =\ \frac{q^{(m)}(\xi(t))}{m!}\cdot(t-\gamma_1)\cdots(t-\gamma_m)$$

mit einer von t abhängigen Zwischenstelle $\xi(t)\in[\gamma_1,\gamma_m]$ gilt jetzt

$$q(t)-Q(t)\ =\ \prod_{j=1}^{m}(t-\gamma_j)\ =\ p(t),$$

da die m-te Ableitung $q^{(m)}(t)$ des Polynoms $q(t)$ vom Grad m mit 1 als Koeffizienten von t^m konstant gleich $m!$ ist. Aus (3.37) folgt

$$\begin{aligned}\widetilde{V}_{\mathrm{E}}\cdot\begin{pmatrix}q(\gamma_1)\\ \vdots\\ q(\gamma_m)\end{pmatrix}\ &=\ \widetilde{V}_{\mathrm{E}}\cdot\begin{pmatrix}Q(\gamma_1)\\ \vdots\\ Q(\gamma_m)\end{pmatrix}\ =\ \begin{pmatrix}Q(\frac{1}{m})\\ \vdots\\ Q(\frac{m}{m})\end{pmatrix}\\ &=\ \begin{pmatrix}q(\frac{1}{m})-p(\frac{1}{m})\\ \vdots\\ q(\frac{m}{m})-p(\frac{m}{m})\end{pmatrix}\ =\ -\begin{pmatrix}p(\frac{1}{m})\\ \vdots\\ p(\frac{m}{m})\end{pmatrix}\ =\ -\boldsymbol{x},\end{aligned}$$

also zusammen mit (3.75)

$$\widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1 \cdot \boldsymbol{x} \;=\; \frac{m^m}{m!} \prod_{j=1}^{m} \gamma_j \cdot \boldsymbol{x},$$

d.h. der Vektor $\boldsymbol{x}$ aus (3.74) ist wie behauptet ein Eigenvektor von $\widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert $\frac{m^m}{m!} \prod_{j=1}^{m} \gamma_j$ und daher ein Eigenvektor von $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z) = I_m - \widetilde{V}_{\mathrm{E}} \cdot \widehat{W}_1$ zum Eigenwert $1 - \frac{m^m}{m!} \prod_{j=1}^{m} \gamma_j$. □

Sei

$$\widehat{P}_m(x) \;=\; \widehat{a}_{m,m} x^m + \ldots + \widehat{a}_{m,0} \;:=\; \frac{1}{m!} \frac{d^m}{dx^m} (x^m (x-1)^m)$$

das transformierte m-te Legendre-Polynom aus (2.15). Dann gilt

$$\begin{aligned} \widehat{a}_{m,0} &= \text{Koeffizient von } x^0 \text{ in } \frac{1}{m!} \frac{d^m}{dx^m} (x^m (x-1)^m) \\ &= \text{Koeffizient von } x^m \text{ in } x^m (x-1)^m \\ &= (-1)^m \end{aligned}$$

und

$$\begin{aligned} \widehat{a}_{m,m} &= \text{Koeffizient von } x^m \text{ in } \frac{1}{m!} \frac{d^m}{dx^m} (x^m (x-1)^m) \\ &= \underbrace{(2m) \cdot (2m-1) \cdots (m+1)}_{= (2m)!/m!} \cdot \left[\text{Koeffizient von } x^{2m} \text{ in } \frac{1}{m!} (x^m (x-1)^m) \right] \\ &= \frac{(2m)!}{(m!)^2} \; . \end{aligned}$$

Wenn für die $\gamma_j,\ j = 1, \ldots, m$ die Abszissen des Gauß(m)-Verfahrens gewählt werden, dann sind diese γ_j die Nullstellen des Polynoms $\widehat{P}_m(x)$, und es gilt nach einem Satz der Algebra

$$\prod_{j=1}^{m} \gamma_j \;=\; (-1)^m \cdot \frac{\widehat{a}_{m,0}}{\widehat{a}_{m,m}} \;=\; \frac{(m!)^2}{(2m)!} \; . \tag{3.76}$$

Analog gilt, wenn für die $\gamma_j,\ j = 1, \ldots, m$ die Abszissen des RadauIIA(m)-Verfahrens, d.h. die Nullstellen des Polynoms $\widehat{P}_m(x) - \widehat{P}_{m-1}(x)$ gewählt werden,

$$\prod_{j=1}^{m} \gamma_j \;=\; (-1)^m \cdot \frac{\widehat{a}_{m,0} - \widehat{a}_{m-1,0}}{\widehat{a}_{m,m}} \;=\; 2 \cdot \frac{(m!)^2}{(2m)!} \; . \tag{3.77}$$

Durch Einsetzen von (3.76) bzw. (3.77) in (3.73) erhält man den folgenden

Satz 3.2.4. *Das Basisverfahren (2.19) erfülle die Bedingungen (3.69), (3.70) und (3.71). Dann gilt für den Spekralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda))$ der Iterationsmatrix der auf (3.44) angewendeten interpolierten Defektkorrektur*[15]

$$\lim_{h\lambda\to\infty}\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) \;=\; \begin{cases} 1-\dfrac{m!m^m}{(2m)!} & \textit{für Gauß(m)-Abszissen (2.12),} \\ 1-2\cdot\dfrac{m!m^m}{(2m)!} & \textit{für RadauIIA(m)-Abszissen (2.12).} \end{cases} \tag{3.78}$$

Offenbar sind die beiden Ausdrücke in (3.78) für alle $m \geq 1$ kleiner als 1.[16] Daraus folgt die Fixpunkt-Konvergenz und numerische Stabilität der entsprechenden Defektkorrekturalgorithmen, wenn sie auf stark steife Anfangswertprobleme (3.44) mit $\lambda \ll 0$ angewendet werden.

IMR und IMR2: Wie man aus Tabelle 3.3 entnimmt, gilt bei Verwendung der Impliziten Mittelpunktregel (IMR) als Basisverfahren *nicht* $\lim_{z\to\infty}\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) = 0$ wie bei allen übrigen Basisverfahren, vielmehr ist dieser Grenzwert gößer als 1, wodurch ein instabiles Verhalten des entsprechenden, auf stark steife ($\lambda \ll 0$) Anfangswertprobleme (3.44) angewendeten klassischen Defektkorrekturalgorithmus zu erwarten ist. Diese Instabilität wird in numerischen Experimenten tatsächlich beobachtet, vgl. Abschnitt 3.4.2.

Wegen der aufgrund von oszillierenden Termen im globalen Fehler der auf steife Differentialgleichungen angewendeten IMR (für Details siehe [2]) zu erwartenden Schwierigkeiten der klassischen Defektkorrektur mit der IMR als Basisverfahren, wurde in [2] vorgeschlagen, den klassischen Defektkorrekturalgorithmus so zu modifizieren, daß die stückweise Polynomfunktion $P^{[0]}(t)$ die Basislösung $\eta_h^{[0]}$ nur an Gitterpunkten $t_\nu \in \Gamma_h$ mit geradem Index ν interpoliert (entsprechendes gilt für $P^{[k]}(t),\ k = 1, 2, \ldots$). Diese Modifikation ist äquivalent dazu, IMR2 statt IMR als Basisverfahren zu verwenden, wobei gleichzeitig die Schrittweite h verdoppelt wird. Durch diese Modifikation kann nun tatsächlich die oben erwähnte Instabilität der klassischen Defektkorrektur mit IMR als Basisverfahren beseitigt werden. Dies ist eine Folgerung aus dem folgenden Satz:

Satz 3.2.5. *Bei Verwendung von IMR2 (2.25) als Basisverfahren (2.19) ist die Matrix $\lim_{h\lambda\to\infty}\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda) = I_{Nm} + \boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{W}}_1$ nilpotent der Ordnung $\lceil\frac{m}{2}\rceil$, und daher gilt $\lim_{h\lambda\to\infty}\rho(\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)) = \rho(I_{Nm} + \boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{W}}_1) = 0$.*

[15] Die Betragsstriche darf man hier weglassen, da für alle $m \geq 1$ die leicht zu verifizierende Ungleichung $0 \leq \frac{m!m^m}{(2m)!} \leq \frac{1}{2}$ gilt.

[16] Allerdings nähern sich die beiden Ausdrücke für große m immer mehr dem Wert 1 an. Das folgt aus der durch Anwendung der Stirlingschen Formel zu gewinnenden Grenzwertbeziehung $\lim_{m\to\infty}\frac{m!m^m}{(2m)!} = 0$.

Beweis. Wir beschränken uns auf den Fall $N = 1$ (dieser Fall genügt für die Aussage $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) = 0$). Wir zeigen mit einer ähnlichen Methode wie im Beweis von Satz 3.2.1: Ist $P(t)$ ein Polynom vom Grad $q \le m$ mit $P(0) = 0$, dann ist $P(t) + Q(t)$ ein Polynom vom Grad $\le q - 2$ mit $P(0) + Q(0) = 0$, wenn $q \ge 2$ ist, und $P(t) + Q(t) \equiv 0$ sonst, wobei $Q(t)$ das eindeutig bestimmte Polynom vom Grad $\le m$ mit $Q(0) = 0$ und (K_∞ ist die Matrix $\boldsymbol{K}_\infty$ aus (3.66) für $N = 1$)

$$\begin{aligned}\begin{pmatrix} Q(1) \\ \vdots \\ Q(m) \end{pmatrix} &= K_\infty \cdot \widetilde{W}_1 \cdot \begin{pmatrix} P(1) \\ \vdots \\ P(m) \end{pmatrix} \\ &= 2 \cdot \begin{pmatrix} 1 & -1 & & & & \\ 1 & -1 & 1 & -1 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ 1 & -1 & 1 & \cdots & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} P(\frac{1}{4}) \\ P(\frac{3}{4}) \\ \vdots \\ P(m-1+\frac{1}{4}) \\ P(m-1+\frac{3}{4}) \end{pmatrix}\end{aligned} \tag{3.79}$$

ist. Durch wiederholte Anwendung dieser Behauptung folgt daraus

$$(I_m + K_\infty \cdot \widetilde{W}_1)^{\lceil \frac{m}{2} \rceil} \cdot \begin{pmatrix} P(1) \\ \vdots \\ P(m) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.80}$$

Indem wir hier speziell $P(t) = t^k,\ k = 1, \ldots, m$ setzen, folgt, daß (3.80) für die m linear unabhängigen Vektoren $(P(1), \ldots, P(m))^T = (1^k, \ldots, m^k))^T,\ k = 1, \ldots, m$ gilt, woraus sich die Nilpotenz

$$(I_m + K_\infty \cdot \widetilde{W}_1)^{\lceil \frac{m}{2} \rceil} = (0)$$

von $I_m + K_\infty \cdot \widetilde{W}_1$ ergibt.

Wir betrachten das Polynom $\nabla Q(t) = Q(t+1) - Q(t)$. Aus (3.79) folgt offenbar

$$\nabla Q(k) = Q(k+1) - Q(k) = 2 \cdot (P(k+\tfrac{1}{4}) - P(k+\tfrac{3}{4})), \quad k = 0, \ldots, m-1, \tag{3.81}$$

d.h. die Polynome $\nabla Q(t)$ und $2 \cdot (P(t-\frac{1}{4}) - P(t+\frac{3}{4}))$ stimmen an den m Stellen $0, 1, \ldots, m-1$ überein. Da beide Polynome einen Grad $\le m-1$ haben gilt daher $\nabla Q(t) \equiv 2 \cdot (P(t+\frac{1}{4}) - P(t+\frac{3}{4}))$. Entwickeln von $P(t)$ nach Taylor ergibt

$$\begin{aligned} P(t+\tfrac{1}{4}) &= P(t) + \tfrac{1}{4}P'(t) + \tfrac{1}{32}P''(t) + R_1(t), \\ P(t+\tfrac{3}{4}) &= P(t) + \tfrac{3}{4}P'(t) + \tfrac{9}{32}P''(t) + R_2(t), \\ P(t+1) &= P(t) + P'(t) + \tfrac{1}{2}P''(t) + R(t), \end{aligned}$$

wobei die Polynome $R_1(t),\ R_2(t)$ und $R(t)$ für $q \geq 3$ den Grad $q-3$ haben und sonst $\equiv 0$ sind, und es folgt

$$\begin{aligned}\nabla(P+Q)(t) &= (P(t+1)+Q(t+1)) - (P(t)-Q(t)) \\ &= P(t+1) - P(t) + 2\cdot(P(t-\tfrac{1}{4}) - P(t-\tfrac{3}{4})) \\ &= R(t) + 2\cdot(R_1(t) - R_2(t)),\end{aligned}$$

d.h. das Polynom $\nabla(P+Q)(t)$ ist vom Grad $\leq q-3$ für $q \geq 3$ und gleich dem Nullpolynom sonst. Damit ist, was zu zeigen war, das Polynom $P(t)+Q(t)$ vom Grad $\leq q-2$ für $q \geq 2$ und $\equiv 0$ sonst. $\square$

Sei das Polynom $P(t)$ mit $P(0)=0$ in (3.79) jetzt vom Grad $q \leq m-1$ (statt $q \leq m$). Dann gilt wegen (3.20) und (3.37)

$$\widetilde{V}\cdot\widehat{W}_1\cdot\begin{pmatrix}P(1)\\ \vdots \\ P(m)\end{pmatrix} \;=\; \widetilde{W}_1\cdot\begin{pmatrix}P(1)\\ \vdots \\ P(m)\end{pmatrix},$$

und somit ergibt sich analog wie im obigen Beweis

$$(I_m + K_\infty\cdot\widetilde{V}\cdot\widehat{W}_1)^{\lceil\frac{m-1}{2}\rceil}\cdot\begin{pmatrix}P(1)\\ \vdots \\ P(m)\end{pmatrix} \;=\; \begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix}.$$

Indem wir speziell $P(t)=t^k,\ k=1,\ldots,m-1$ setzen, erhalten wir die Gültigkeit dieser Gleichung für die $m-1$ linear unabhängigen Vektoren $(P(1),\ldots,P(m))^T = (1^k,\ldots,m^k)^T,\ k=1,\ldots,m-1$, woraus folgt, daß 0 ein Eigenwert der Matrix $I_m+K_\infty\cdot\widetilde{V}\cdot\widehat{W}_1$ mit algebraischer Vielfachheit $m-1$ ist. Sei jetzt $P(t)$ das eindeutig definierte Polynom vom Grad $\leq m$ mit $P(0)=0$ und $P(\gamma_j)=1,\ \gamma=1,\ldots,m$. Dann gilt wegen (3.20), (3.37) und der Tatsache, daß das Polynom vom Grad $\leq m-1$, welches $P(t)$ an den Stellen $t=\gamma_1,\ldots,\gamma_m$ interpoliert, konstant gleich 1 ist,

$$K_\infty\cdot\widetilde{V}\cdot\widehat{W}_1\cdot\begin{pmatrix}P(\frac{1}{m})\\ \vdots \\ P(\frac{m}{m})\end{pmatrix} \;=\; K_\infty\cdot\widetilde{V}\cdot\begin{pmatrix}P(\gamma_1)\\ \vdots \\ P(\gamma_m)\end{pmatrix} \;=\; K_\infty\cdot\widetilde{V}\cdot\begin{pmatrix}1\\ \vdots \\ 1\end{pmatrix}$$

$$= \;K_\infty\cdot\begin{pmatrix}1\\1\\ \vdots \\1\\1\end{pmatrix} \;=\; 2\cdot\begin{pmatrix}1 & -1 & & & & \\ 1 & -1 & 1 & -1 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ 1 & -1 & 1 & \cdots & 1 & -1\end{pmatrix}\cdot\begin{pmatrix}1\\1\\ \vdots \\1\\1\end{pmatrix} \;=\; \begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix},$$

d.h. der Vektor $(P(\frac{1}{m}),\ldots,P(\frac{m}{m}))^T$ ist ein Eigenvektor der Matrix $I_m+K_\infty\cdot\widetilde{V}\cdot\widehat{W}_1$ zum Eigenwert 1. Damit ist $\{0,1\}$ die Menge *aller* Eigenwerte von $I_m+K_\infty\cdot\widetilde{V}\cdot\widehat{W}_1$ und damit von $\lim_{z\to\infty}\boldsymbol{S}_{\mathrm{IIDeC}}(z)=I_{Nm}+\boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{V}}\cdot\widehat{\boldsymbol{W}}_1$, und der folgende Satz ist bewiesen:

Satz 3.2.6. *Bei Verwendung von IMR2 (2.25) als Basisverfahren (2.19) gilt*

$$\lim_{h\lambda\to\infty}\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) \;=\; 1.$$

für beliebige Wahl der γ_j in (2.12).

ITR und ITR2: Bei Verwendung der Impliziten Trapezregel (ITR) als Basisverfahren gilt

$$\boldsymbol{K}_\infty \;=\; -\begin{pmatrix} 1 & 1 & & & & \\ -1 & -1 & 1 & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ (-1)^{Nm-1} & (-1)^{Nm-1} & (-1)^{Nm-2} & \cdots & 1 & 1 \end{pmatrix}.$$

Für beliebiges $\boldsymbol{\eta}=(\eta_1,\ldots,\eta_{Nm})^T\in\mathbb{R}^{Nm}$ definieren wir $\eta_h\in\mathcal{E}_h$ durch $\eta_h(t_0):=0$ und $\eta_h(t_\nu):=\eta_\nu,\ \nu=1,\ldots,Nm$ und damit die stückweise Polynomfunktion $P(t)\in\mathcal{P}_h$ durch $P(t):=(P_h\eta_h)(t)$. Damit gilt aufgrund der Definition von $\widetilde{\boldsymbol{W}}_1$ im Fall der ITR

$$\boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{W}}_1\cdot\boldsymbol{\eta} \;=\; \boldsymbol{K}_\infty\cdot\begin{pmatrix} P(t_0)\\ P(t_1)\\ P(t_1)\\ P(t_2)\\ \vdots\\ P(t_{Nm-1})\\ P(t_{Nm})\end{pmatrix} \;=\; -\begin{pmatrix} P(t_0)+P(t_1)\\ -P(t_0)+P(t_2)\\ \vdots\\ (-1)^{Nm-1}P(t_0)+P(t_{Nm})\end{pmatrix} \;=\; -\boldsymbol{\eta},$$

also $(I_{Nm}+\boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{W}}_1)\cdot\boldsymbol{\eta}=0$ für alle $\boldsymbol{\eta}\in\mathbb{R}^{Nm}$, und damit

$$\lim_{z\to\infty}\boldsymbol{S}_{\mathrm{IDeC}}(z) \;=\; I_{Nm}+\boldsymbol{K}_\infty\cdot\widetilde{\boldsymbol{W}}_1 \;=\; (0). \tag{3.82}$$

Auf ähnliche Weise zeigt man, daß (3.82) auch im Fall von ITR2 als Basisverfahren gültig ist, und wir erhalten den folgenden

Satz 3.2.7. *Bei Verwendung von ITR (2.26) oder ITR2 (2.27) als Basisverfahren konvergiert die Iterationsmatrix $\boldsymbol{S}_{\mathrm{IDeC}}(h\lambda)$ der auf (3.44) angewendeten klassischen Defektkorrektur für $h\lambda\to\infty$ gegen eine Nullmatrix.*

Wiederum folgt aus diesem Satz die rasche Fixpunktkonvergenz der entsprechenden klassischen Defektkorrekturalgorithmen, wenn sie auf stark steife Differentialgleichungen (3.44) angewendet werden.

Wir kommen nun zum Fall der interpolierten Defektkorrektur:

Satz 3.2.8. *Bei Verwendung von ITR2 (2.27) als Basisverfahren (2.19) gilt*

$$\lim_{h\lambda\to\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) \;=\; 1.$$

für beliebige Wahl der γ_j in (2.12).

Beweis. Es genügt den Fall $N = 1$ zu betrachten (vgl. Abschnitt 3.2.1). Die Matrix $\boldsymbol{K}_\infty$ aus (3.66) hat jetzt die Gestalt

$$K_\infty \;=\; \begin{pmatrix} 1 & 0 & -1 & & & & & & \\ 1 & 0 & -1 & 1 & 0 & -1 & & & \\ \vdots & \vdots & \vdots & & & \ddots & \ddots & \ddots & \\ 1 & 0 & -1 & & \dots & & 1 & 0 & -1 \end{pmatrix}.$$

Mit $p_k(t) := t^k,\ k = 1, \dots, m-1$ folgt aus (3.20) und (3.20)

$$K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1 \cdot \begin{pmatrix} p_k(\frac{1}{m}) \\ \vdots \\ p_k(\frac{m}{m}) \end{pmatrix} \;=\; K_\infty \cdot \widetilde{V} \cdot \begin{pmatrix} p_k(\gamma_1) \\ \vdots \\ p_k(\gamma_m) \end{pmatrix} \;=\; K_\infty \cdot \begin{pmatrix} p_k(0) \\ * \\ p_k(\frac{1}{m}) \\ \vdots \\ p_k(\frac{m-1}{m}) \\ * \\ p_k(\frac{m}{m}) \end{pmatrix} \;=\; - \begin{pmatrix} p_k(\frac{1}{m}) \\ \vdots \\ p_k(\frac{m}{m}) \end{pmatrix}.$$

Hier symbolisieren die Sterne „$*$“ Einträge im Vektor, die nicht von Belang sind. Die Vektoren $((\frac{1}{m})^k, \dots, (\frac{m}{m})^k)^T,\ k = 1, \dots, m-1$ sind also $m-1$ linear unabhängige Eigenvektoren der Matrix $K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1$ zum Eigenwert -1, und daher $m-1$ linear unabhängige Eigenvektoren der Matrix $I_m + K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1$ zum Eigenwert 0. Sei nun $P(t)$ das eindeutig bestimmte Polynom vom Grad $\le m$ mit $P(0) = 0$ und $P(\gamma_j) = 1,\ j = 1, \dots, m$. Damit gilt wegen (3.20), (3.37) und der Tatsache, daß das Polynom vom Grad $\le m-1$, welches $P(t)$ an den Stellen $t = \gamma_1, \dots, \gamma_m$ interpoliert, konstant gleich 1 ist,

$$K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1 \cdot \begin{pmatrix} P(\frac{1}{m}) \\ \vdots \\ P(\frac{m}{m}) \end{pmatrix} \;=\; K_\infty \cdot \widetilde{V} \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \;=\; K_\infty \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix},$$

also ist $P(\frac{1}{m}), \ldots, P(\frac{m}{m}))^T$ ein Eigenvektor von $I_m + K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1$ zum Eigenwert 1. 0 und 1 sind somit die einzigen Eigenwerte von $I_m + K_\infty \cdot \widetilde{V} \cdot \widehat{W}_1$, daher sind 0 und 1 die einzigen Eigenwerte von $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) = I_{Nm} + \boldsymbol{K}_\infty \cdot \widetilde{\boldsymbol{W}}_1$, woraus die Behauptung des Satzes folgt. □

3.3 Vorbemerkungen zu den numerischen Experimenten

Bevor wir im nächsten Abschnitt beginnen, Ergebnisse von numerischen Experimenten mit den Defektkorrekturalgorithmen zu dokumentieren, machen wir hier einige Bemerkungen zur Durchführung dieser Experimente.

Wir konzentrieren uns auf folgende zwei Aspekte:

1. Die Untersuchung der Konvergenz der durch Defektkorrektur gewonnenen Approximationen $\eta_h^{[k]}$ für $k \to \infty$ und fester Schrittweite h gegen den Fixpunkt, d.h. im Fall der interpolierten Defektkorrektur gegen die Kollokationslösung η_h^*.

2. Die Untersuchung der Konvergenz der k-ten Approximation $\eta_h^{[k]}$ bei festem k für $h \to 0$ gegen die exakte Lösung $y(t)$ des jeweils betrachteten Anfangswertproblems.

Der wesentliche Punkt ist hier der zweite, denn wir sind im Grunde nicht so sehr daran interessiert, Kollokationslösungen zu bestimmen, sondern vielmehr daran, bei vorgegebener Anforderung an die Genauigkeit, auf möglichst effiziente Weise eine Approximation für die exakte Lösung des betrachteten Anfangswertproblems zu berechnen. Der erste Punkt ist jedoch vielfach für die Beurteilung der Approximationsgüte der $\eta_h^{[k]}$ in Abhängigkeit von der Schrittweite h von Bedeutung, und zwar insbesondere dann, wenn die $\eta_h^{[k]}$ schnell gegen den Fixpunkt η_h^* konvergieren. Hier zeigen oft schon nach wenigen Defektkorrekturschritten die Approximationen $\eta_h^{[k]}$ für $h \to 0$ dasselbe Konvergenzverhalten wie der Fixpunkt η_h^*. Andererseits ist der erste Punkt aber auch dann von Interesse, wenn die $\eta_h^{[k]}$ für $k \to \infty$ überhaupt nicht gegen den Fixpunkt η_h^* konvergieren, sondern im Gegenteil ein stark divergentes Verhalten zeigen. In diesem Fall ist nämlich nicht zu erwarten, daß die $\eta_h^{[k]}$ für größere k vernünftige Approximationen der exakten Lösung $y(t)$ sind.

3.3.1 Konvergenzordnung für $h \to 0$

Im Fall der klassischen Defektkorrektur gibt es zur Frage der Konvergenz von $\eta_h^{[k]}$ für $h \to 0$ gegen die exakte Lösung $y(t)$ theoretische Resultate, die ohne Einschränkungen allerdings nur für nichtsteife Probleme gültig sind: Und zwar gilt, vgl. [7], wenn die Ordnung[17] des Basisverfahrens p beträgt,

$$\begin{aligned} \eta_h^{[0]}(t_\nu) - y(t_\nu) &= O(h^p), \\ \eta_h^{[1]}(t_\nu) - y(t_\nu) &= O(h^{2p}), \\ \eta_h^{[2]}(t_\nu) - y(t_\nu) &= O(h^{3p}), \\ &\vdots \end{aligned} \tag{3.83}$$

für $t_\nu \in \Gamma_h$. In dieser Folge ist die erzielbare Ordnung durch den Grad m der Interpolationspolynome $P^{[k]}(t)$ begrenzt, d.h. es gilt

$$\eta_h^{[k]}(t_\nu) - y(t_\nu) = \begin{cases} O(h^{(k+1)p}) & \text{für } (k+1)p \le m, \\ O(h^m) & \text{für } (k+1)p \ge m. \end{cases} \tag{3.84}$$

Bei Verwendung von äquidistanten Kollokationsabszissen (2.12), d.h. $\gamma_i = i/m$, $i = 1, \ldots, m$, ist m auch die Konvergenzordnung der entsprechenden Kollokationslösung η_h^*:

$$\eta_h^*(t_\nu) - y(t_\nu) \;=\; O(h^m) \quad (t_\nu \in \Gamma_h). \tag{3.85}$$

Speziell hat also bei Verwendung des Impliziten Eulerverfahrens als Basisverfahren, der Fixpunkt η_h^* der klassischen Defektkorrektur dieselbe Konvergenzordnung wie die $(m-1)$-te Approximation $\eta_h^{[m-1]}$, und durch Fortführen der Defektkorrekturiteration kann diese Konvergenzordnung im allgemeinen nicht mehr verbessert werden.

Im Fall von Gauß- bzw. RadauIIA-Abszissen als Kollokationsabszissen (2.12) tritt für die entsprechenden Kollokationslösungen η_h^* an den Stellen $T_\ell = t_0 + \ell \cdot H,\ \ell = 1, \ldots, N,\ H = m \cdot h$ das Phänomen der *Superkonvergenz* auf:[18]

$$\eta_h^*(T_\ell) - y(T_\ell) \;=\; \begin{cases} O(h^{2m}) & \text{für Gauß}(m)\text{-Abszissen (2.12)}, \\ O(h^{2m-1}) & \text{für RadauIIA}(m)\text{-Abszissen (2.12)}. \end{cases} \tag{3.86}$$

[17] Nach der Konvention bedeutet die Schreibweise $f(h) = O(h^p)$, daß es Konstanten C und h_0 gibt, sodaß $\|f(h)\| \le C \cdot h^p$ für alle $h \in (0, h_0]$ gilt. In diesem Sinn sind Ordnungsrelationen wie (3.83) auch für steife Anfangswertprobleme gültig. Der wesentliche Punkt ist aber, daß die in der „Groß-O"-Schreibweise nicht explizit vorkommenden Konstanten C und h_0 in (3.83) von den Parametern des betrachteten Anfangswertproblems in solcher Weise abhängen, daß im allgemeinen für steife Anfangswertprobleme C sehr groß und/oder h_0 sehr klein wird, und somit die entsprechenden Fehlerschranken praktisch wertlos werden. Eine Diskussion dieser Problematik findet man in [3, p. 642].

[18] Diese hohe Konvergenzordnung ist nur im Fall von nichtsteifen Anfangswertproblemen zu erwarten. Tatsächlich beobachtet man für viele steife Anfangswertprobleme eine Reduktion dieser Ordnung, vgl. Abschnitt 3.4.2

Damit gilt es in unseren numerischen Experimenten zu untersuchen, inwieweit sich im Fall der interpolierten Defektkorrektur dieses superkonvergente Verhalten des Fixpunktes η_h^* auf die Approximationen $\eta_h^{[k]}$ für endliches k überträgt. Eine sorgfältige Analyse der Argumentationsweise in [7] führt zu dem Schluß, daß diese für den Fall der interpolierten Defektkorrektur adaptiert werden kann, sodaß (3.84) auch im Fall der interpolierten Defektkorrektur (bei Anwendung auf nichtsteife Anfangswertprobleme) gültig ist. Insbesondere sichert also die Theorie für die Stellen $t = T_\ell$ die Ordnungsrelationen $\eta_h^{[k]}(T_\ell) - y(T_\ell) = O(h^{(k+1)p})$ für jene k, für die $(k+1)p \leq m$ ist, was durch numerische Experimente auch bestätigt wird. Für größere k sind mit Hilfe der Argumente aus [7] keine Aussagen zu gewinnen. In unseren numerischen Experimenten werden wir beobachten, daß sich für diese k, die Ordnung von $\eta_h^{[k]}(T_\ell)$ pro Defektkorrekturschritt i.a. nicht mehr auf ganz so systematische Weise erhöht, wie das für die k mit $(k+1)p \leq m$ der Fall ist, dennoch kann aber für nichtsteife Anfangswertprobleme bei Verwendung von Gauß- bzw. RadauIIA-Abszissen nach einer voraussagbaren Anzahl an Defektkorrekturschritten die Ordnung $2m$ bzw. $2m-1$ des Fixpunktes η_h^* erreicht werden.

3.3.2 Empirische Bestimmung von Konvergenzordnungen

Zur experimentellen Bestimmung der Konvergenzordnung für $h \to 0$ eines numerischen Integrationsverfahrens (in unserem Fall ist das ein Defektkorrekturverfahren mit einer festen Anzahl an Defektkorrekturschritten) gehen wir folgendermaßen vor: Wir integrieren mit dem Verfahren ein gegebenes Anfangswertproblem von $t = t_0$ bis $t = t_{end}$ einmal mit einer Schrittweite h und einmal mit der halben Schrittweite $h/2$ (und daher der doppelten Anzahl an Integrationsschritten) und erhalten dadurch die Approximationen $\eta_h(t_{end})$ bzw. $\eta_{\frac{h}{2}}(t_{end})$ für die exakte Lösung $y(t_{end})$ an der Stelle $t = t_{end}$. Unter der Annahme, daß der globale Fehler nicht nur[19]

$$\|\eta_h(t_\nu) - y(t_\nu)\| \;=\; O(h^p), \tag{3.87}$$

sondern näherungsweise sogar die Proportionalität

$$\|\eta_h(t_\nu) - y(t_\nu)\| \;\approx\; C(t_\nu) \cdot h^p \tag{3.88}$$

erfüllt, was insbesondere dann der Fall ist, wenn der globale Fehler eine asymptotische Entwicklung der Form

$$\eta_h(t_\nu) - y(t_\nu) \;=\; e_p(t_\nu) \cdot h^p + O(h^{p+1}) \tag{3.89}$$

[19]Mit $\|.\|$ meinen wir im folgenden immer die Euklidische Norm des $\mathbb{R}^n$. Die Wahl der Norm ist aber unwesentlich, da alle Normen des $\mathbb{R}^n$ äquivalent sind.

besitzt, läßt sich die Konvergenzordnung p des betrachteten Verfahrens durch

$$p \;\approx\; \log\left(\frac{\|\eta_h(t_{end}) - y(t_{end})\|}{\|\eta_{\frac{h}{2}}(t_{end}) - y(t_{end})\|}\right) \Big/ \log 2 \tag{3.90}$$

empirisch bestimmen. (Dazu muß natürlich die exakte Lösung $y(t_{end})$ an der Stelle $t = t_{end}$ explizit bekannt sein.)

3.3.3 Gleitpunktarithmetik mit erweiterter Genauigkeit

Beim Versuch, empirische Aussagen über das asymptotische Verhalten von Diskretisierungsverfahren zur numerischen Lösung von Differentialgleichungen zu gewinnen, stößt man bald an die Grenzen der Rechengenauigkeit der in den heute gebräuchlichen Computern hardwaremäßig eingebauten Gleitpunktarithmetik. Eine Alternative bieten Programme wie Maple, die softwaremäßig eine beliebig genaue Arithmetik zur Verfügung stellen. Allerdings ist bei Durchführung umfangreicher Testserien die damit erzielbare Rechengeschwindigkeit völlig inakzeptabel. Aus diesem Grund wurden die in den Abschnitten 3.4 und 3.5 beschriebenen numerischen Experimente mit einem Computerprogramm durchgeführt, das auf dem Softwarepaket „doubledouble“ von Keith Briggs basiert, welches unter der Internet-Adresse

http://www-epidem.plantsci.cam.ac.uk/~kbriggs/doubledouble.html

verfügbar ist. In „doubledouble“ ist softwaremäßig eine vierfach-genaue Gleitpunktarithmetik implementiert, was ca. 30 Dezimalstellen entspricht. Die erzielbare Rechengeschwindigkeit ist aufgrund der Programmierung in C++ hoch: Unser in C++ geschriebenes Testprogramm für die verschiedenen Defektkorrekturalgorithmen ist trotz der Verwendung von „doubledouble“ für alle Gleitpunktrechnungen immer noch wesentlich schneller als das entsprechende Matlab-Programm aus Anhang A.1.7.

3.3.4 Wahl der algorithmischen Parameter

Wir fassen die Parameter zusammen, durch die eine Variante der hier betrachteten Defektkorrekturalgorithmen spezifiziert wird, und geben an, wie diese Parameter in unseren Experimenten gesetzt worden sind:

IDeC/IIDeC, Wahl der Abszissen (2.12): Hauptsächlich sind wir an der interpolierten Defektkorrektur (IIDeC) interessiert, wobei wir für die Abszissen (2.12) sowohl Gauß- als auch RadauIIA-Abszissen betrachten. Zum Vergleich experimentieren wir aber auch mit der klassischen Defektkorrektur (IDeC).

Polynomgrad m: In den hier dokumentierten numerischen Experimenten wählen wir $m = 6$. Dazu muß gesagt werden, daß alle Experimente selbstverständlich auch mit anderen Polynomgraden durchgeführt wurden, wobei das qualitative Verhalten für alle m ähnlich ist. Die Wahl von $m = 6$ scheint vernünftig (weder zu klein noch zu groß), und irgendeine Beschränkung ist unvermeidlich, wenn die Ergebnisse der numerischen Experimente auf ökonomische Art präsentiert werden sollen.

Basisverfahren: Wir betrachten die folgenden Basisverfahren aus Abschnitt 2.1.7: Implizites Eulerverfahren, IMR, ITR, SDIRK(2) und RadauIIA(2). In Zusammenhang mit steifen Anfangswertproblemen wird es sich als zweckmäßig erweisen, IMR durch IMR2 und ITR durch ITR2 zu ersetzen. Später werden wir uns auf die stark A-stabilen Verfahren Implizites Eulerverfahren, RadauIIA(2) und SDIRK(2) beschränken, da diese Verfahren, wie wir feststellen werden, für gewisse steife Probleme gegenüber IMR2 und ITR2 Vorteile bringen.

Lokale/globale Verbindungsstrategie: Wir wählen, wenn nichts anderes gesagt wird, immer die globale Verbindungsstrategie.

3.4 Numerische Experimente (1): Skalares lineares Anfangswertproblem

In diesem Abschnitt studieren wir in numerischen Experimenten das Verhalten der verschiedenen Defektkorrekturverfahren, wenn sie auf ein skalares Anfangswertproblem der Gestalt (3.44) angewendet werden. Konkret betrachten wir das Problem

$$y' = \lambda(y - \sin t - 2) + \cos t, \quad y(0) = 2 \tag{3.91}$$

mit der exakten Lösung

$$y(t) = \sin t + 2, \tag{3.92}$$

und als Integrationsintervall wählen wir $[t_0, t_{end}] = [0, 3.6]$.

3.4.1 Nichtsteifer Fall ($\lambda = O(1)$)

Für den Parameter λ in (3.91) wählen wir zunächst den Wert $\lambda = -1$, um eine Vorstellung von der Leistungsfähigkeit der verschiedenen Defektkorrekturalgorithmen zu bekommen, wenn sie auf nichtsteife Anfangswertprobleme angewendet werden. Das so festgelegte Anfangswertproblem lösen wir numerisch mit den in Abschnitt 3.3.4 spezifizierten Defektkorrekturverfahren, wobei die Schrittweite

h des Basisverfahrens ausgehend von $h = 0.2$ viermal halbiert wird. Die erhaltenen Ergebnisse sind in den Tabellen 3.4 bis 3.18 zusammengefaßt. Diese Tabellen zeigen jeweils in der oberen Hälfte für den Basisschritt und die ersten 6 Defektkorrekturschritte (14 Defektkorrekturschritte im Fall der IIDeC mit Implizitem Eulerverfahren als Basisverfahren) den Betrag

$$|\eta_h^{[k]}(t_{end}) - y(t_{end})|$$

des globalen Fehlers an der Stelle $t_{end} = 3.6$. Im Fall der IIDeC (für das Implizite Eulerverfahren und IMR als Basisverfahren auch im Fall der IDeC) ist zum Vergleich jeweils in der letzten Spalte der Betrag

$$|\eta_h^*(t_{end}) - y(t_{end})|$$

des globalen Fehlers des Fixpunktes, d.h. des entsprechenden Kollokationsverfahrens[20] angegeben. In der unteren Hälfte der Tabellen findet man die entsprechenden, gemäß (3.90) empirisch bestimmten Konvergenzordnungen.

Basisverfahren der Ordnung $p = 1$ (Implizites Eulerverfahren): Die Ordnungsrelationen (3.84) werden für die klassische Defektkorrektur durch Tabelle 3.4 bestätigt. Im unteren Teil der Tabelle kann man für die kleineren Schrittweiten h deutlich beobachten, wie sich pro Defektkorrekturschritt die Konvergenzordnung jeweils um $p = 1$ erhöht, bis für $k = 5$ die Ordnung $m = 6$ und das Genauigkeitsniveau des Fixpunktes erreicht ist. Im Fall der Interpolierten Defektkorrektur setzt sich diese Steigerung der Genauigkeit und der Konvergenzordnung für $k \geq 5$ weiter fort, nun aber auf nicht mehr ganz so regelmäßige Weise. Für Gauß-Kollokationsabszissen kann auch nach den 14 in Tabelle 3.5 dargestellten Defektkorrekturschritten das (hohe) Genauigkeitsniveau des Fixpunktes nicht erreicht werden, für die in Tabelle 3.6 betrachteten RadauIIA-Kollokationsabszissen gelingt dies immerhin für die kleineren Schrittweiten $h = 0.025$ und $h = 0.0125$.

Basisverfahren der Ordnung $p = 2$ (IMR, ITR, SDIRK(2)): Wieder werden die Ordnungsrelationen (3.84) für die klassische Defektkorrektur bestätigt: In den entsprechenden Tabellen 3.7, 3.10 bzw. 3.13 läßt sich ablesen, wie sich nun pro Defektkorrekturschritt die Konvergenzordnung um $p = 2$ erhöht, bis für $k = 2$ die Konvergenzordnung $m = 6$ und das Genauigkeitsniveau des jeweiligen Fixpunktes erreicht ist. Im Fall der Interpolierten Defektkorrektur setzt sich diese Steigerung der Genauigkeit und der Konvergenzordnung weiter fort, und zwar erhöht sich im Fall von Gauß-Kollokationsabszissen, wie man den Tabellen 3.8, 3.11 und 3.15 entnehmen kann, auch für $k \geq 3$ die Konvergenzordnung ziemlich

[20] Man beachte, daß bei der Berechnung dieses Vergleichswertes, für das Kollokationsverfahren bzw. für das zu diesem äquivalente IRK-Verfahren die Schrittweite $H = m \cdot h = 6h$ verwendet wird.

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Fixpunkt[21]
0.2	2.07E−02	3.60E−03	2.28E−04	1.32E−06	8.30E−08	4.93E−07	7.61E−07	8.94E−07
0.1	1.02E−02	9.32E−04	3.06E−05	1.90E−07	1.38E−08	1.01E−08	2.00E−08	2.26E−08
0.05	5.06E−03	2.37E−04	3.97E−06	1.40E−08	1.07E−09	1.48E−10	3.98E−10	4.33E−10
0.025	2.52E−03	5.98E−05	5.07E−07	9.18E−10	4.73E−11	2.00E−12	7.08E−12	7.44E−12
0.0125	1.26E−03	1.50E−05	6.40E−08	5.83E−11	1.74E−12	2.75E−14	1.19E−13	1.22E−13
0.2								
0.1	1.02	1.95	2.90	2.79	2.59	5.60	5.25	5.31
0.05	1.01	1.97	2.94	3.76	3.70	6.10	5.65	5.71
0.025	1.01	1.99	2.97	3.93	4.49	6.21	5.81	5.86
0.0125	1.00	1.99	2.99	3.98	4.77	6.18	5.90	5.93

Tabelle 3.4: IDeC ($m = 6$, Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

regelmäßig pro Defektkorrekturschritt um $p = 2$, bis für $k = 5$ die Superkonvergenzordnung $2m = 12$ und das entsprechend hohe Genauigkeitsniveau des Fixpunktes erreicht wird. Für die in den Tabellen 3.9, 3.12 und 3.15 betrachteten RadauIIA-Kollokationsabszissen ist die Steigerung der Genauigkeit und der Konvergenzordnung pro Defektkorrekturschritt nicht ganz so regelmäßig und auch nicht ganz so schnell wie im Fall von Gauß-Kollokationsabszissen: Obwohl nun Konvergenzordnung und Genauigkeitsniveau des Fixpunktes etwas niedriger sind, können diese teilweise erst für $k = 6$ erreicht werden, nur im Fall von SDIRK(2) als Basisverfahren (vgl. Tabelle 3.15) sind dazu auch für die kleinste Schrittweite $h = 0.0125$ schon 5 Defektkorrekturschritte ausreichend.

Basisverfahren der Ordnung $p = 3$ (RadauIIA(2)): Bei der klassischen Defektkorrektur mit RadauIIA(2) als Basisverfahren ist, wie man in Tabelle 3.16 beobachtet, im wesentlichen nach einem Defektkorrekturschritt die Konvergenzordnung und die Genauigkeit des Fixpunktes erreicht. Im Fall der in Tabelle 3.17 betrachteten Interpolierten Defektkorrektur mit Gauß-Kollokationsabszissen erhöht sich die Konvergenzordnung weiter, aber nicht, wie man vermuten könnte, immer um $p = 3$ pro Defektkorrekturschritt, sodaß für $k = 3$ die Superkonvergenzordnung $2m = 12$ des Fixpunktes erreicht wäre. Vielmehr wird für die kleineren Schrittweiten $h = 0.025$ und $h = 0.0125$ die Konvergenzordnung und auch die entsprechend hohe Genauigkeit des Fixpunktes erst einen Defektkorrekturschritt später, nämlich für $k = 4$ erreicht. Im Fall von RadauIIA-Kollokationsabszissen (vgl. Tabelle 3.18) wird für kleinere Schrittweiten die Konvergenzordnung und die Genauigkeit des Fixpunktes überhaupt erst für $k = 5$ erreicht, was hier aber auch bei Verwendung von SDIRK(2) als Basisverfahren (vgl. Tabelle 3.15) der Fall ist. In diesem konkreten Fall bringt also die Verwendung des wesentlich aufwendigeren Verfahrens RadauIIA(2) als Basisverfahren gegenüber SDIRK(2) keinen Vorteil.

[21] Der Fixpunkt ist hier die zu den äquidistanten Kollokationsabszissen $\gamma_i := i/m,\ i = 1, \ldots, m,\ m = 6$ gehörige Kollokationslösung.

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
0.2	2.07E−02	3.60E−03	2.26E−04	2.77E−06	1.08E−06	1.61E−07	3.70E−08	1.78E−08
0.1	1.02E−02	9.32E−04	3.05E−05	2.11E−07	4.19E−08	5.37E−09	1.33E−09	4.24E−10
0.05	5.06E−03	2.37E−04	3.97E−06	1.43E−08	1.58E−09	1.43E−10	2.28E−11	4.15E−12
0.025	2.52E−03	5.98E−05	5.07E−07	9.22E−10	5.59E−11	3.01E−12	2.65E−13	2.59E−14
0.0125	1.26E−03	1.50E−05	6.40E−08	5.83E−11	1.87E−12	5.47E−14	2.54E−15	1.28E−16
0.2	1.02	1.95	2.89	3.72	4.68	4.90	4.80	5.39
0.1	1.01	1.97	2.94	3.88	4.73	5.23	5.86	6.67
0.05	1.01	1.99	2.97	3.95	4.82	5.57	6.42	7.32
0.025	1.00	1.99	2.99	3.98	4.90	5.78	6.71	7.66
0.0125								

h	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$	Gauß(6)
0.2	9.37E−09	4.78E−09	2.37E−09	1.15E−09	5.53E−10	2.63E−10	1.24E−10	1.78E−12
0.1	1.35E−10	4.26E−11	1.34E−11	4.18E−12	1.31E−12	4.09E−13	1.28E−13	4.19E−16
0.05	7.56E−13	1.37E−13	2.50E−14	4.57E−15	8.38E−16	1.54E−16	2.84E−17	1.01E−19
0.025	2.54E−15	2.48E−16	2.44E−17	2.42E−18	2.41E−19	2.41E−20	2.40E−21	2.46E−23
0.0125	6.49E−18	3.30E−19	1.69E−20	8.70E−22	4.50E−23	2.34E−24	1.17E−25	6.00E−27
0.2	6.12	6.81	7.47	8.11	8.72	9.33	9.92	12.06
0.1	7.48	8.28	9.06	9.84	10.61	11.37	12.14	12.02
0.05	8.22	9.11	10.00	10.88	11.77	12.65	13.53	12.00
0.025	8.61	9.55	10.50	11.44	12.38	13.33	14.32	12.00
0.0125								

Tabelle 3.5: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
0.2	2.07E−02	3.60E−03	2.28E−04	1.31E−06	8.18E−07	2.44E−07	8.74E−08	3.29E−08
0.1	1.02E−02	9.32E−04	3.06E−05	1.85E−07	3.52E−08	8.63E−09	2.10E−09	5.13E−10
0.05	5.06E−03	2.37E−04	3.97E−06	1.39E−08	1.46E−09	2.15E−10	3.03E−11	4.33E−12
0.025	2.52E−03	5.98E−05	5.07E−07	9.15E−10	5.40E−11	4.30E−12	3.25E−13	2.53E−14
0.0125	1.26E−03	1.50E−05	6.40E−08	5.82E−11	1.84E−12	7.60E−14	2.99E−15	1.21E−16
0.2	1.02	1.95	2.90	2.82	4.54	4.82	5.38	6.00
0.1	1.01	1.97	2.94	3.74	4.59	5.32	6.12	6.89
0.05	1.01	1.99	2.97	3.92	4.76	5.65	6.54	7.42
0.025	1.00	1.99	2.99	3.98	4.87	5.82	6.77	7.71
0.0125								

h	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$	RadauIIA(6)
0.2	1.26E−08	4.97E−09	2.04E−09	8.71E−10	3.90E−10	1.88E−10	1.02E−10	3.61E−11
0.1	1.30E−10	3.46E−11	9.55E−12	2.72E−12	7.93E−13	2.43E−13	8.36E−14	1.78E−14
0.05	6.52E−13	1.03E−13	1.70E−14	2.86E−15	4.94E−16	9.23E−17	2.32E−17	8.74E−18
0.025	2.09E−15	1.81E−16	1.62E−17	1.48E−18	1.42E−19	1.71E−20	5.50E−21	4.29E−21
0.0125	5.23E−18	2.36E−19	1.10E−20	5.28E−22	2.75E−23	3.34E−24	2.16E−24	2.10E−24
0.2	6.59	7.17	7.74	8.32	8.94	9.59	10.25	10.99
0.1	7.64	8.39	9.14	9.89	10.65	11.36	11.81	10.99
0.05	8.29	9.16	10.03	10.91	11.77	12.39	12.04	10.99
0.025	8.64	9.58	10.52	11.46	12.33	12.32	11.31	11.00
0.0125								

Tabelle 3.6: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Fixpunkt
0.2	7.37E−05	6.29E−06	4.96E−07	5.46E−07	5.46E−07	5.46E−07	5.46E−07	5.46E−07
0.1	1.74E−05	3.70E−07	6.76E−09	7.45E−09	7.45E−09	7.45E−09	7.45E−09	7.45E−09
0.05	4.29E−06	2.28E−08	1.02E−10	1.12E−10	1.12E−10	1.12E−10	1.12E−10	1.12E−10
0.025	1.07E−06	1.42E−09	1.57E−12	1.73E−12	1.73E−12	1.73E−12	1.73E−12	1.73E−12
0.0125	2.67E−07	8.88E−11	2.45E−14	2.70E−14	2.70E−14	2.70E−14	2.70E−14	2.70E−14
0.2	2.08	4.09	6.20	6.20	6.20	6.20	6.20	6.20
0.1	2.02	4.02	6.06	6.06	6.06	6.06	6.06	6.06
0.05	2.01	4.01	6.01	6.01	6.01	6.01	6.01	6.01
0.025	2.00	4.00	6.00	6.00	6.00	6.00	6.00	6.00
0.0125								

Tabelle 3.7: IDeC ($m = 6$, Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	7.37E−05	6.02E−06	2.84E−08	3.55E−10	9.13E−13	1.83E−12	1.78E−12	1.78E−12
0.1	1.74E−05	3.67E−07	4.97E−10	1.55E−12	2.98E−15	4.40E−16	4.19E−16	4.19E−16
0.05	4.29E−06	2.28E−08	8.00E−12	6.21E−15	3.51E−18	1.08E−19	1.01E−19	1.01E−19
0.025	1.07E−06	1.42E−09	1.26E−13	2.44E−17	3.58E−21	2.63E−23	2.46E−23	2.46E−23
0.0125	2.67E−07	8.88E−11	1.97E−15	9.56E−20	3.54E−24	6.42E−27	6.00E−27	6.00E−27
0.2	2.08	4.04	5.84	7.84	8.26	12.02	12.06	12.06
0.1	2.02	4.01	5.96	7.96	9.73	12.00	12.02	12.02
0.05	2.01	4.00	5.99	7.99	9.94	12.00	12.00	12.00
0.025	2.00	4.00	6.00	8.00	9.98	12.00	12.00	12.00
0.0125								

Tabelle 3.8: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	7.37E−05	6.15E−06	8.61E−09	5.73E−10	5.64E−11	3.56E−11	3.62E−11	3.61E−11
0.1	1.74E−05	3.68E−07	3.39E−10	4.66E−12	8.58E−14	1.65E−14	1.78E−14	1.78E−14
0.05	4.29E−06	2.28E−08	6.73E−12	3.83E−14	3.80E−16	4.45E−18	8.79E−18	8.74E−18
0.025	1.07E−06	1.42E−09	1.16E−13	3.20E−16	1.88E−18	7.59E−21	4.36E−21	4.29E−21
0.0125	2.67E−07	8.88E−11	1.89E−15	2.62E−18	8.40E−21	2.57E−23	2.19E−24	2.10E−24
0.2	2.08	4.06	4.67	6.94	9.36	11.07	10.99	10.99
0.1	2.02	4.01	5.65	6.93	7.82	11.86	10.98	10.99
0.05	2.01	4.00	5.86	6.90	7.66	9.19	10.98	10.99
0.025	2.00	4.00	5.94	6.93	7.80	8.21	10.96	11.00
0.0125								

Tabelle 3.9: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$

.

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	2.28E−03	6.12E−06	5.01E−07	5.23E−07	5.23E−07	5.23E−07	5.23E−07
0.1	5.70E−04	3.41E−07	7.01E−09	7.48E−09	7.48E−09	7.48E−09	7.48E−09
0.05	1.43E−04	2.07E−08	1.06E−10	1.14E−10	1.14E−10	1.14E−10	1.14E−10
0.025	3.56E−05	1.29E−09	1.64E−12	1.77E−12	1.77E−12	1.77E−12	1.77E−12
0.0125	8.91E−06	8.02E−11	2.56E−14	2.76E−14	2.76E−14	2.76E−14	2.76E−14
0.2							
0.1	2.00	4.16	6.16	6.13	6.13	6.13	6.13
0.05	2.00	4.04	6.05	6.04	6.04	6.04	6.04
0.025	2.00	4.01	6.01	6.01	6.01	6.01	6.01
0.0125	2.00	4.00	6.00	6.00	6.00	6.00	6.00

Tabelle 3.10: IDeC ($m = 6$, Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	2.28E−03	5.54E−06	2.01E−08	7.80E−10	3.72E−11	1.86E−13	1.90E−12	1.78E−12
0.1	5.70E−04	3.33E−07	4.40E−10	4.49E−12	6.33E−14	6.95E−16	4.40E−16	4.19E−16
0.05	1.43E−04	2.06E−08	7.35E−12	1.93E−14	7.24E−17	2.49E−19	1.03E−19	1.01E−19
0.025	3.56E−05	1.28E−09	1.17E−13	7.70E−17	7.37E−20	6.69E−23	2.47E−23	2.46E−23
0.0125	8.91E−06	8.01E−11	1.83E−15	3.03E−19	7.28E−23	1.67E−26	6.01E−27	6.00E−27
0.2								
0.1	2.00	4.06	5.52	7.44	9.20	8.07	12.08	12.06
0.05	2.00	4.01	5.90	7.87	9.77	11.45	12.06	12.02
0.025	2.00	4.00	5.98	7.97	9.94	11.86	12.02	12.00
0.0125	2.00	4.00	5.99	7.99	9.98	11.96	12.01	12.00

Tabelle 3.11: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	2.28E−03	5.76E−06	5.55E−08	1.45E−09	2.21E−10	1.19E−11	3.95E−11	3.61E−11
0.1	5.70E−04	3.34E−07	7.13E−10	1.40E−11	4.01E−13	9.49E−15	1.78E−14	1.78E−14
0.05	1.43E−04	2.06E−08	9.40E−12	1.42E−13	1.88E−15	1.21E−17	8.86E−18	8.74E−18
0.025	3.56E−05	1.28E−09	1.32E−13	1.26E−15	8.56E−18	4.96E−20	4.58E−21	4.29E−21
0.0125	8.91E−06	8.01E−11	1.95E−15	1.05E−17	3.61E−20	1.17E−22	2.47E−24	2.10E−24
0.2								
0.1	2.00	4.11	6.28	6.69	9.11	10.29	11.11	10.99
0.05	2.00	4.02	6.25	6.62	7.74	9.61	10.97	10.99
0.025	2.00	4.00	6.15	6.81	7.78	7.93	10.92	10.99
0.0125	2.00	4.00	6.08	6.91	7.89	8.73	10.86	11.00

Tabelle 3.12: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	6.40E−04	5.55E−07	1.91E−07	1.91E−07	1.91E−07	1.91E−07	1.91E−07
0.1	1.51E−04	4.37E−08	2.17E−09	2.18E−09	2.18E−09	2.18E−09	2.18E−09
0.05	3.67E−05	2.81E−09	2.79E−11	2.80E−11	2.80E−11	2.80E−11	2.80E−11
0.025	9.04E−06	1.76E−10	3.90E−13	3.92E−13	3.92E−13	3.92E−13	3.92E−13
0.0125	2.24E−06	1.10E−11	5.74E−15	5.77E−15	5.77E−15	5.77E−15	5.77E−15
0.2							
0.1	2.08	3.67	6.46	6.46	6.46	6.46	6.46
0.05	2.04	3.96	6.28	6.28	6.28	6.28	6.28
0.025	2.02	4.00	6.16	6.16	6.16	6.16	6.16
0.0125	2.01	4.00	6.09	6.09	6.09	6.09	6.09

Tabelle 3.13: IDeC ($m = 6$, Basisverfahren: SDIRK(2)) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	6.40E−04	7.11E−07	4.89E−11	1.46E−11	1.85E−12	1.78E−12	1.78E−12	1.78E−12
0.1	1.51E−04	4.54E−08	7.47E−12	7.44E−14	5.36E−16	4.18E−16	4.19E−16	4.19E−16
0.05	3.67E−05	2.83E−09	1.48E−13	2.92E−16	2.39E−19	1.01E−19	1.01E−19	1.01E−19
0.025	9.04E−06	1.76E−10	2.46E−15	1.11E−18	1.63E−22	2.46E−23	2.46E−23	2.46E−23
0.0125	2.24E−06	1.10E−11	3.92E−17	4.27E−21	1.41E−25	5.99E−27	6.00E−27	6.00E−27
0.2								
0.1	2.08	3.97	2.71	7.62	11.75	12.06	12.06	12.06
0.05	2.04	4.00	5.65	7.99	11.13	12.02	12.02	12.02
0.025	2.02	4.01	5.91	8.03	10.52	12.00	12.00	12.00
0.0125	2.01	4.00	5.97	8.03	10.18	12.00	12.00	12.00

Tabelle 3.14: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: SDIRK(2)) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	6.40E−04	6.19E−07	5.37E−09	2.71E−10	2.60E−11	3.66E−11	3.61E−11	3.61E−11
0.1	1.51E−04	4.42E−08	5.89E−11	1.23E−12	1.05E−14	1.84E−14	1.78E−14	1.78E−14
0.05	3.67E−05	2.81E−09	6.97E−13	6.65E−15	7.46E−17	9.76E−18	8.73E−18	8.74E−18
0.025	9.04E−06	1.76E−10	9.06E−15	3.99E−17	2.58E−19	5.99E−21	4.28E−21	4.29E−21
0.0125	2.24E−06	1.10E−11	1.27E−16	2.63E−19	8.87E−22	5.07E−24	2.09E−24	2.10E−24
0.2								
0.1	2.08	3.81	6.51	7.78	11.28	10.96	10.99	10.99
0.05	2.04	3.97	6.40	7.53	7.13	10.88	10.99	10.99
0.025	2.02	4.00	6.27	7.38	8.17	10.67	10.99	10.99
0.0125	2.01	4.00	6.16	7.24	8.19	10.21	11.00	11.00

Tabelle 3.15: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) angewendet auf (3.91) mit $\lambda = -1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.09E−04	1.23E−07	1.13E−07	1.13E−07	1.13E−07	1.13E−07	1.13E−07
0.1	1.38E−05	1.06E−09	9.14E−10	9.14E−10	9.14E−10	9.14E−10	9.14E−10
0.05	1.74E−06	1.03E−11	7.92E−12	7.92E−12	7.92E−12	7.92E−12	7.92E−12
0.025	2.18E−07	1.12E−13	7.46E−14	7.46E−14	7.46E−14	7.46E−14	7.46E−14
0.0125	2.73E−08	1.36E−15	7.84E−16	7.84E−16	7.84E−16	7.84E−16	7.84E−16
0.2							
0.1	2.98	6.85	6.95	6.95	6.95	6.95	6.95
0.05	2.99	6.70	6.85	6.85	6.85	6.85	6.85
0.025	2.99	6.52	6.73	6.73	6.73	6.73	6.73
0.0125	3.00	6.35	6.57	6.57	6.57	6.57	6.57

Tabelle 3.16: IDeC ($m=6$, Basisverfahren: RadauIIA(2)) angewendet auf (3.91) mit $\lambda=-1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end}=3.6$.

h	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	1.09E−04	2.47E−08	7.81E−11	5.93E−13	1.77E−12	1.78E−12	1.78E−12	1.78E−12
0.1	1.38E−05	2.67E−10	1.56E−13	2.23E−16	4.16E−16	4.19E−16	4.19E−16	4.19E−16
0.05	1.74E−06	3.23E−12	2.79E−16	1.87E−19	1.01E−19	1.01E−19	1.01E−19	1.01E−19
0.025	2.18E−07	4.31E−14	5.11E−19	1.20E−22	2.46E−23	2.46E−23	2.46E−23	2.46E−23
0.0125	2.73E−08	6.15E−16	9.78E−22	8.43E−26	6.01E−27	6.00E−27	6.00E−27	6.00E−27
0.2								
0.1	2.98	6.53	8.97	11.38	12.05	12.06	12.06	12.06
0.05	2.99	6.37	9.13	10.22	12.01	12.02	12.02	12.02
0.025	2.99	6.23	9.09	10.61	12.00	12.00	12.00	12.00
0.0125	3.00	6.13	9.03	10.47	12.00	12.00	12.00	12.00

Tabelle 3.17: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: RadauIIA(2) angewendet auf (3.91) mit $\lambda=-1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end}=3.6$.

h	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.09E−04	9.60E−08	4.34E−09	2.46E−10	2.60E−11	3.66E−11	3.61E−11	3.61E−11
0.1	1.38E−05	1.20E−09	3.00E−11	7.95E−13	2.32E−15	1.83E−14	1.78E−14	1.78E−14
0.05	1.74E−06	1.69E−11	2.19E−13	2.96E−15	3.07E−17	9.26E−18	8.74E−18	8.74E−18
0.025	2.18E−07	2.52E−13	1.66E−15	1.14E−17	7.27E−20	4.81E−21	4.29E−21	4.29E−21
0.0125	2.73E−08	3.84E−15	1.28E−17	4.41E−20	1.48E−22	2.60E−24	2.10E−24	2.10E−24
0.2								
0.1	2.98	6.32	7.18	8.27	13.46	10.97	10.99	10.99
0.05	2.99	6.15	7.10	8.07	6.24	10.95	10.99	10.99
0.025	2.99	6.07	7.05	8.02	8.72	10.91	10.99	10.99
0.0125	3.00	6.03	7.02	8.01	8.94	10.85	11.00	11.00

Tabelle 3.18: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: RadauIIA(2)) angewendet auf (3.91) mit $\lambda=-1$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end}=3.6$.

Der Fall $\lambda = 0$: In diesem Fall, wo (3.44) in ein Quadraturproblem übergeht, ist der Operator S_h aus (3.3) nilpotent der Ordnung $\widetilde{m} := \lceil \frac{m}{p} \rceil$, wobei mit p die Ordnung des Basisverfahrens bezeichnet wird, vgl. Abschnitt 3.2.2. Für $k \geq \widetilde{m}$ gilt also $S_h^k = 0$, und daher gilt für diese k

$$\begin{aligned} \eta_h^{[k]} &= S_h^k \eta_h^{[0]} + S_h^{k-1} v_h + S_h^{k-2} v_h + \ldots + S_h v_h + v_h \\ &= S_h^{\widetilde{m}-1} v_h + S_h^{\widetilde{m}-2} v_h + \ldots + S_h v_h + v_h \\ &= \eta_h^* \quad \big(= \text{Lösung von } (Id - S_h)\eta_h^* = v_h\big), \end{aligned}$$

d.h. nach $\widetilde{m}$ Defektkorrekturschritten erreicht die Defektkorrekturiteration ihren Fixpunkt η_h^* exakt. Im Fall der Interpolierten Defektkorrektur mit einem superkonvergenten Fixpunkt hat das die bemerkenswerte Konsequenz, daß schon die $\widetilde{m}$-te Approximation $\eta_h^{[\widetilde{m}]}$ für $h \to 0$ an den Stellen $T_\ell = t_0 + \ell \cdot H,\ H = m \cdot h$ die Superkonvergenzordnung des Fixpunktes aufweist. Wir demonstrieren dies in der folgenden Tabelle 3.19 für ITR (p=2) als Basisverfahren und Gauß(6)-Abszissen als Kollokationsabszissen (2.12):

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	1.48E−03	5.09E−06	1.52E−09	7.84E−16	7.84E−16	7.84E−16	7.84E−16	7.84E−16
0.1	3.69E−04	3.10E−07	2.34E−11	1.84E−19	1.84E−19	1.84E−19	1.84E−19	1.84E−19
0.05	9.22E−05	1.92E−08	3.64E−13	4.44E−23	4.44E−23	4.44E−23	4.44E−23	4.44E−23
0.025	2.30E−05	1.20E−09	5.69E−15	1.08E−26	1.08E−26	1.08E−26	1.08E−26	1.08E−26
0.2								
0.1	2.00	4.04	6.02	12.06	12.06	12.06	12.06	12.06
0.05	2.00	4.01	6.01	12.01	12.01	12.01	12.01	12.01
0.025	2.00	4.00	6.00	12.00	12.00	12.00	12.00	12.00

Tabelle 3.19: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = 0$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

In diesem Fall gilt $\widetilde{m} = \lceil \frac{6}{2} \rceil = 3$, und tatsächlich ist, wie man der Tabelle entnimmt, für $k = 3$ der Fixpunkt und damit dessen Konvergenzordnung $2m = 12$ erreicht.

Es stellt sich nun die Frage, inwieweit sich dieses vorteilhafte Konvergenzverhalten der interpolierten Defektkorrektur für $\lambda = 0$ auf betragskleine λ überträgt. Dazu wurde für alle λ aus einer logarithmischen Skala $\subseteq [-1, -10^{-10}]$ das jeweilige Problem (3.91) mit der interpolierten Defektkorrektur numerisch integriert, wobei als Basisverfahren ITR mit der Schrittweite $h = 0.05$ (bzw. zusätzlich mit der Schrittweite $\frac{h}{2} = 0.025$ für die Berechnung der empirischen Konvergenzordnung) und als Kollokationsabszissen (2.12) Gauß(6)-Abszissen verwendet wurden. Die dabei beobachteten globalen Fehler an der Stelle $t_{end} = 3.6$ für das Basisverfahren, die ersten 4 Defektkorrekturschritte und den Fixpunkt Gauß(6) sind in

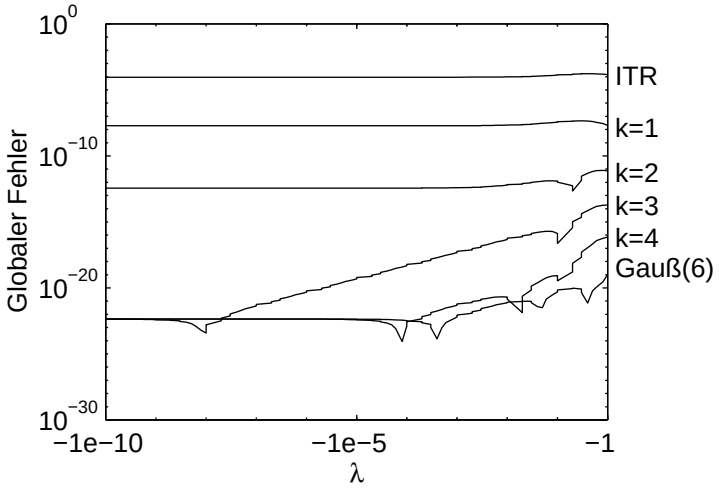

Abbildung 3.22: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR mit Schrittweite $h = 0.05$) angewendet auf (3.91) mit verschiedenen λ: Globaler Fehler an der Stelle $t_{end} = 3.6$.

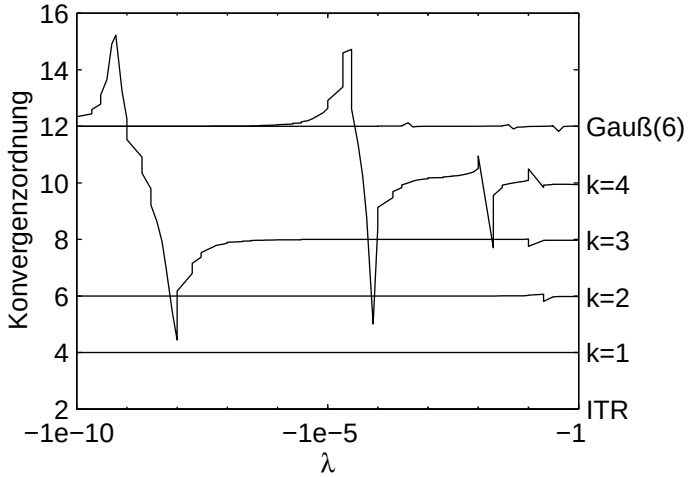

Abbildung 3.23: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR mit Schrittweite $h = 0.05$ bzw. $\frac{h}{2} = 0.025$) angewendet auf (3.91) mit verschiedenen λ: Beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

Abbildung 3.22 dargestellt, die entsprechenden, gemäß (3.90) berechneten empirischen Konvergenzordnungen in Abbildung 3.23. Erst für $|\lambda| \leq 10^{-5}$ beobachtet man hier für $k = 4$ die Konvergenzordnung 12 und für $k = 3$ überhaupt erst für $|\lambda| \leq 10^{-9}$ (für $k = 5$ ist schon mit $\lambda = -1$ die Konvergenzordnung 12 festzustellen, vgl. Tabelle 3.11). Für das Konvergenzverhalten der interpolierten Defektkorrektur im allgemeinen nichtsteifen Fall (d.h. $|\lambda|$ ist von der Größenordnung $O(1)$, aber nicht unbedingt verschwindend klein) lassen sich somit, wie dieses numerische Experiment zeigt (und analoge Experimente mit anderen m,

anderen Basisverfahren und/oder RadauIIA- statt Gauß-Abszissen), keine allzu weitreichenden Schlußfolgerungen aus dem entsprechenden Konvergenzverhalten für $\lambda = 0$ ziehen.

3.4.2 Steifer Fall ($\lambda \ll 0$)

Unser Testproblem (3.91) ist ein Mitglied der Familie von linearen Modellproblemen

$$y' = \lambda(y - g(t)) + g'(t), \quad y(t_0) = g(t_0), \tag{3.93}$$

die im Zusammenhang mit impliziten Runge-Kutta-Verfahren von A. Prothero und A. Robinson [12] betrachtet wurde. In [12] wurde zum ersten Mal für die Kollokationsverfahren Gauß bzw. RadauIIA eine Ordnungsreduktion beobachtet, wenn sie auf steife Probleme der Form (3.93) mit $\lambda \ll 0$ angewendet werden. Und zwar beobachtet man in diesem Fall nicht mehr die Superkonvergenzordnung $2m$ bzw. $2m-1$ (vgl. (3.86)) sondern jeweils nur noch die reduzierte Ordnung m.[22] Gleichzeitig ist aber für das stark A-stabile RadauIIA-Verfahren ein ε-Faktor im globalen Fehler festzustellen, d.h. für $\lambda = -\frac{1}{\varepsilon},\ 0 < \varepsilon \ll 1$ gilt

$$\eta_h^*(T_\ell) - y(T_\ell) \ = \ O(\varepsilon h^m), \qquad T_\ell = t_0 + \ell \cdot H, \quad H = m \cdot h, \tag{3.94}$$

wobei hier $\eta_h^* \in \mathcal{E}_h$ die zu RadauIIA(m)-Abszissen gehörige Kollokationslösung von (3.93) ist.[23] Dieses Resultat ist ein Spezialfall der in Abschnitt 1.2.4 erwähnten neuen Konvergenztheorie für IRK-Verfahren. Tatsächlich fällt das zu (3.93) äquivalente autonome 2-dimensionale Problem

$$\begin{pmatrix} y' \\ z' \end{pmatrix} \ = \ \begin{pmatrix} \lambda(y - g(z)) + g'(z) \\ 1 \end{pmatrix}, \quad \begin{pmatrix} y(t_0) \\ z(t_0) \end{pmatrix} \ = \ \begin{pmatrix} g(t_0) \\ t_0 \end{pmatrix}. \tag{3.95}$$

in die in dieser Theorie betrachtete Klasse steifer Systeme.[24] Das Problem (3.95) ist aber insofern speziell, daß hier die glatte Komponente, die mit der z-Komponente zusammenfällt, exakt integriert wird, woraus folgt, daß die glatte

[22] Im Fall der Gauß-Verfahren beobachtet man für ungerades m für den lokalen Fehler die Ordnung m und für den globalen Fehler die Ordnung $m+1$. Die Ursache dafür ist, daß hier die lokalen Fehler einander teilweise auslöschen.

[23] In [12] wurde der ε-Faktor nur für den lokalen Fehler hergeleitet, das Auftreten dieses Faktors im globalen Fehler läßt sich daraus aber mit Hilfsmitteln der B-Konvergenztheorie folgern.

[24] Man beachte, daß, wenn man in der Approximation, die ein IRK-Verfahren (oder wie man sich leicht überlegt auch ein hier betrachtetes Defektkorrekturverfahren) bei Anwendung auf das autonom umgeschriebene Problem (3.95) liefert, jeweils die letzte Komponente wegläßt, man dieselbe Approximation erhält, wie wenn man das Verfahren direkt auf (3.93) anwendet. Dazu muß für das IRK-Verfahren (bzw. das Basisverfahren im Fall eines Defektkorrekturverfahrens) die Butcher-Bedingung $B(1)$ erfüllt sein, was in der Regel der Fall ist.

Fehlerkomponente $\equiv 0$ ist und daher die von der Theorie vorhergesagte Konvergenzordnung $2m-1$ für diese glatte Fehlerkomponente nicht beobachtbar ist. Daher ist in diesem Spezialfall für den Gesamtfehler auch im stark steifen Fall immer nur die reduzierte Ordnung m der steifen Fehlerkomponente zu beobachten, im Gegensatz zu der Behauptung aus Abschnitt 1.2.4, daß im allgemeinen für den stark steifen Fall die klassische Ordnung $2m-1$ zu beobachten sein sollte.

In unseren numerischen Experimenten gilt es nun zu untersuchen, ob der erwähnte ε-Faktor auch im Fall der interpolierten Defektkorrektur mit RadauIIA-Abszissen als Kollokationsabszissen im globalen Fehler $\eta_h^{[k]}(T_\ell) - y(T_\ell)$ an den Stellen T_ℓ für endliches k zu beobachten ist, wenn sie auf ein Anfangswertproblem der Gestalt (3.93) mit $\lambda = -\frac{1}{\varepsilon},\ 0 < \varepsilon \ll 1$ angewendet wird. Zu erwarten ist das nur, wenn im globalen Fehler des Basisverfahrens dieser ε-Faktor auftritt. Von den Basisverfahren aus Abschnitt 2.1.7 ist das für das Implizite Eulerverfahren, SDIRK, RadauIIA(2), ITR und ITR2 der Fall.

Wir beschreiben nun geordnet nach den verwendeten Basisverfahren Experimente, die mit den Defektkorrekturalgorithmen angewendet auf (3.91) mit $\lambda \ll 0$ durchgeführt wurden.

Implizites Eulerverfahren, SDIRK(2) und RadauIIA(2): Aufgrund von Satz 3.2.2 aus Abschnitt 3.2.3 ist für diese Basisverfahren im Fall der klassischen IDeC die sehr rasche Konvergenz der Approximationen $\eta_h^{[k]}$ für $k \to \infty$ gegen den jeweiligen Fixpunkt zu erwarten. Das wird durch die folgende Tabelle 3.20 für das Implizite Eulerverfahren bestätigt, wo der Parameter λ in (3.91) auf den Wert $\lambda = -100000$ gesetzt wurde:

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Fixpunkt
0.2	3.71E−07	8.82E−11	8.79E−11	8.79E−11	8.79E−11	8.79E−11	8.79E−11	8.79E−11
0.1	2.01E−07	1.44E−12	1.36E−12	1.36E−12	1.36E−12	1.36E−12	1.36E−12	1.36E−12
0.05	1.04E−07	4.27E−14	2.05E−14	2.05E−14	2.05E−14	2.05E−14	2.05E−14	2.05E−14
0.025	5.29E−08	6.08E−15	3.13E−16	3.13E−16	3.13E−16	3.13E−16	3.13E−16	3.13E−16
0.0125	2.67E−08	1.48E−15	4.81E−18	4.81E−18	4.81E−18	4.81E−18	4.81E−18	4.81E−18
0.2	0.89	5.94	6.01	6.01	6.01	6.01	6.01	6.01
0.1	0.95	5.08	6.05	6.05	6.05	6.05	6.05	6.05
0.05	0.97	2.81	6.04	6.04	6.04	6.04	6.04	6.04
0.025	0.99	2.04	6.02	6.02	6.02	6.02	6.02	6.02
0.0125								

Tabelle 3.20: IDeC ($m = 6$, Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

Man erkennt, daß für die Schrittweiten $h = 0.2$ und $h = 0.1$ der Fixpunkt im

wesentlichen schon nach einem Defektkorrekturschritt erreicht wird, für die kleineren Schrittweiten werden dazu zwei Defektkorrekturschritte benötigt.

Für den Fall der Interpolierten Defektkorrektur sind die entsprechenden Ergebnisse in den auf den folgenden Seiten zu findenden Tabellen 3.21 bis 3.22 zusammengefaßt. Diese Tabellen zeigen zusätzlich zum globalen Fehler und der empirisch bestimmten Konvergenzordnung an der Stelle t_{end} noch den Abstand

$$|\eta_h^{[k]}(t_{end}) - \eta_h^*(t_{end})| \tag{3.96}$$

der k-ten Approximation $\eta_h^{[k]}$ vom Fixpunkt η_h^* an der Stelle t_{end} und den Quotienten

$$\frac{|\eta_h^{[k+1]}(t_{end}) - \eta_h^*(t_{end})|}{|\eta_h^{[k]}(t_{end}) - \eta_h^*(t_{end})|} \tag{3.97}$$

zweier solcher aufeinanderfolgender Abstände. In den Legenden zu den Tabellen wird dieser Quotient als Konvergenzfaktor bezeichnet: Er gibt an, um welchen Faktor sich der Iterationsfehler (3.96) pro Defektkorrekturschritt reduziert.

Im Fall von Gauß-Kollokationsabszissen (vgl. die Tabellen 3.21, 3.23 und 3.25) ist für kleinere Schrittweiten h nach einem Defektkorrekturschritt die Approximation $\eta_h^{[1]}$ besser als der Fixpunkt, was sowohl das Niveau des globalen Fehlers als auch die entsprechende Konvergenzordnung betrifft: Man beobachtet für $\eta_h^{[1]}$ die Konvergenzordnung $m+1=7$ im Gegensatz zur (reduzierten) Ordnung $m=6$ des Fixpunktes.[25] Im weiteren Verlauf der Defektkorrekturiteration werden die Approximationen $\eta_h^{[k]}$ etwas schlechter, sie nähern sich langsam (die Konvergenzfaktoren (3.97) sind nur etwas kleiner als 1) dem Fixpunkt an.

Im Fall von RadauIIA-Kollokationsabszissen (vgl. die Tabellen 3.22, 3.24 und 3.26) wird nach zwei Defektkorrekturschritten (bei RadauIIA(2) als Basisverfahren sogar schon nach einem Defektkorrekturschritt, vgl. Tabelle 3.26) das Genauigkeitsniveau und die Konvergenzordnung $m=6$ des Fixpunktes erreicht. Im Gegensatz zur klassischen Defektkorrektur (vgl. Tabelle 3.20), wo praktisch nach zwei Defektkorrekturschritten der Fixpunkt auf dem Niveau der Rechengenauigkeit erreicht wird (d.h. in unserem Fall bei Verwendung von „doubledouble“ (vgl. Abschnitt 3.3.3) auf ca. 30 Dezimalstellen genau), ist jetzt nach zwei Defektkorrekturschritten an der Stelle $t_{end}=3.6$ der Abstand (3.96) zum Fixpunkt nur um etwa eine Zehnerpotenz kleiner als der dortige Abstand des Fixpunktes zur exakten Lösung. Wie man jeweils dem unteren Teil der Tabellen 3.22, 3.24 und 3.26 entnimmt, verringert sich der Abstand (3.96) der Approximationen zum Fixpunkt in den weiteren Defektkorrekturschritten pro Defektkorrekturschritt um einen Faktor, der ziemlich genau dem Wert

$$1 - 2 \cdot \frac{m! \cdot m^m}{(2 \cdot m)!} \;=\; 1 - 2 \cdot \frac{6! \cdot 6^6}{(2 \cdot 6)!} \;=\; \frac{331}{385} \;\approx\; 0.86$$

[25] Für ungerades m beobachtet man die Ordnung $m+1$ auch für den Fixpunkt, vgl. Fußnote 22.

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauss(6)
0.2	3.71E−07	3.81E−07	4.16E−07	4.47E−07	4.75E−07	4.99E−07	5.19E−07	2.98E−07
0.1	2.01E−07	2.95E−09	3.35E−09	3.75E−09	4.14E−09	4.53E−09	4.90E−09	4.52E−09
0.05	1.04E−07	2.22E−11	2.55E−11	2.87E−11	3.19E−11	3.52E−11	3.84E−11	7.16E−11
0.025	5.29E−08	1.64E−13	1.94E−13	2.18E−13	2.43E−13	2.67E−13	2.92E−13	1.21E−12
0.0125	2.67E−08	1.70E−16	1.49E−15	1.67E−15	1.86E−15	2.04E−15	2.23E−15	2.35E−14
0.2								
0.1	0.89	7.01	6.96	6.90	6.84	6.78	6.73	6.04
0.05	0.95	7.05	7.04	7.03	7.02	7.01	7.00	5.98
0.025	0.97	7.09	7.04	7.04	7.04	7.04	7.04	5.89
0.0125	0.99	9.91	7.03	7.03	7.03	7.03	7.03	5.69
0.2	6.70E−07	8.27E−08	1.18E−07	1.49E−07	1.76E−07	2.00E−07	2.21E−07	0.0
0.1	2.05E−07	1.57E−09	1.17E−09	7.69E−10	3.78E−10	5.04E−12	3.79E−10	0.0
0.05	1.04E−07	4.94E−11	4.62E−11	4.29E−11	3.97E−11	3.65E−11	3.33E−11	0.0
0.025	5.29E−08	1.05E−12	1.02E−12	9.94E−13	9.69E−13	9.45E−13	9.20E−13	0.0
0.0125	2.67E−08	2.36E−14	2.20E−14	2.18E−14	2.16E−14	2.14E−14	2.12E−14	0.0
0.2	1.24E−01	1.42E+00	1.26E+00	1.18E+00	1.14E+00	1.10E+00		
0.1	7.67E−03	7.42E−01	6.58E−01	4.91E−01	1.33E−02	7.52E+01		
0.05	4.75E−04	9.34E−01	9.30E−01	9.25E−01	9.19E−01	9.12E−01		
0.025	1.98E−05	9.71E−01	9.76E−01	9.75E−01	9.75E−01	9.74E−01		
0.0125	8.85E−07	9.30E−01	9.92E−01	9.91E−01	9.91E−01	9.91E−01		

Tabelle 3.21: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	3.71E−07	8.17E−11	5.95E−11	6.15E−11	6.32E−11	6.47E−11	6.60E−11	7.36E−11
0.1	2.01E−07	1.34E−12	9.22E−13	9.54E−13	9.81E−13	1.00E−12	1.02E−12	1.15E−12
0.05	1.04E−07	4.12E−14	1.39E−14	1.44E−14	1.48E−14	1.52E−14	1.55E−14	1.73E−14
0.025	5.29E−08	6.06E−15	2.12E−16	2.20E−16	2.26E−16	2.31E−16	2.36E−16	2.64E−16
0.0125	2.67E−08	1.48E−15	3.26E−18	3.37E−18	3.47E−18	3.55E−18	3.62E−18	4.05E−18
0.2								
0.1	0.89	5.93	6.01	6.01	6.01	6.01	6.01	6.00
0.05	0.95	5.03	6.05	6.05	6.05	6.05	6.05	6.05
0.025	0.97	2.76	6.04	6.04	6.04	6.04	6.04	6.04
0.0125	0.99	2.04	6.02	6.02	6.02	6.02	6.02	6.03
0.2	3.71E−07	8.15E−12	1.41E−11	1.21E−11	1.03E−11	8.85E−12	7.58E−12	0.0
0.1	2.01E−07	1.94E−13	2.25E−13	1.93E−13	1.66E−13	1.43E−13	1.23E−13	0.0
0.05	1.04E−07	2.38E−14	3.40E−15	2.92E−15	2.51E−15	2.15E−15	1.85E−15	0.0
0.025	5.29E−08	5.80E−15	5.14E−17	4.41E−17	3.79E−17	3.26E−17	2.80E−17	0.0
0.0125	2.67E−08	1.47E−15	7.84E−19	6.73E−19	5.78E−19	4.97E−19	4.27E−19	0.0
0.2	2.20E−05	1.73E+00	8.56E−01	8.57E−01	8.57E−01	8.57E−01		
0.1	9.65E−07	1.16E+00	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.05	2.29E−07	1.43E−01	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.025	1.10E−07	8.86E−03	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.0125	5.52E−08	5.32E−04	8.59E−01	8.59E−01	8.59E−01	8.59E−01		

Tabelle 3.22: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: Implizites Eulerverfahren) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauss(6)
0.2	1.90E−07	3.81E−07	4.16E−07	4.47E−07	4.75E−07	4.99E−07	5.19E−07	2.98E−07
0.1	1.24E−07	2.95E−09	3.36E−09	3.76E−09	4.15E−09	4.53E−09	4.91E−09	4.52E−09
0.05	6.90E−08	2.22E−11	2.55E−11	2.87E−11	3.20E−11	3.52E−11	3.85E−11	7.16E−11
0.025	3.62E−08	1.67E−13	1.94E−13	2.19E−13	2.44E−13	2.68E−13	2.93E−13	1.21E−12
0.0125	1.84E−08	6.02E−16	1.49E−15	1.68E−15	1.87E−15	2.06E−15	2.25E−15	2.35E−14
0.2	0.61	7.01	6.96	6.90	6.84	6.78	6.73	6.04
0.1	0.84	7.05	7.04	7.03	7.02	7.01	6.99	5.98
0.05	0.93	7.06	7.04	7.04	7.04	7.04	7.04	5.89
0.025	0.97	8.11	7.02	7.02	7.03	7.03	7.03	5.69
0.0125								
0.2	1.09E−07	8.27E−08	1.18E−07	1.49E−07	1.76E−07	2.00E−07	2.21E−07	0.0
0.1	1.19E−07	1.57E−09	1.17E−09	7.66E−10	3.74E−10	9.35E−12	3.84E−10	0.0
0.05	6.89E−08	4.94E−11	4.62E−11	4.29E−11	3.97E−11	3.64E−11	3.32E−11	0.0
0.025	3.62E−08	1.05E−12	1.02E−12	9.93E−13	9.68E−13	9.44E−13	9.19E−13	0.0
0.0125	1.84E−08	2.29E−14	2.20E−14	2.18E−14	2.16E−14	2.14E−14	2.12E−14	0.0
0.2	7.59E−01	1.43E+00	1.26E+00	1.18E+00	1.14E+00	1.10E+00		
0.1	1.32E−02	7.41E−01	6.57E−01	4.88E−01	2.50E−02	4.11E+01		
0.05	7.17E−04	9.34E−01	9.30E−01	9.24E−01	9.18E−01	9.11E−01		
0.025	2.89E−05	9.74E−01	9.76E−01	9.75E−01	9.74E−01	9.74E−01		
0.0125	1.24E−06	9.61E−01	9.91E−01	9.91E−01	9.91E−01	9.91E−01		

Tabelle 3.23: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: SDIRK(2)) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.90E−07	5.28E−11	5.89E−11	6.10E−11	6.28E−11	6.43E−11	6.56E−11	7.36E−11
0.1	1.24E−07	8.43E−13	9.12E−13	9.45E−13	9.74E−13	9.98E−13	1.02E−12	1.15E−12
0.05	6.90E−08	2.17E−14	1.38E−14	1.43E−14	1.47E−14	1.51E−14	1.54E−14	1.73E−14
0.025	3.62E−08	2.85E−15	2.10E−16	2.18E−16	2.24E−16	2.30E−16	2.35E−16	2.64E−16
0.0125	1.84E−08	7.02E−16	3.23E−18	3.35E−18	3.45E−18	3.54E−18	3.61E−18	4.05E−18
0.2	0.61	5.97	6.01	6.01	6.01	6.01	6.01	6.00
0.1	0.84	5.28	6.05	6.05	6.05	6.05	6.05	6.05
0.05	0.93	2.93	6.04	6.04	6.04	6.04	6.04	6.04
0.025	0.97	2.02	6.02	6.02	6.02	6.02	6.02	6.03
0.0125								
0.2	1.90E−07	2.08E−11	1.47E−11	1.26E−11	1.08E−11	9.25E−12	7.92E−12	0.0
0.1	1.24E−07	3.04E−13	2.36E−13	2.02E−13	1.74E−13	1.49E−13	1.28E−13	0.0
0.05	6.90E−08	4.36E−15	3.55E−15	3.05E−15	2.62E−15	2.25E−15	1.93E−15	0.0
0.025	3.62E−08	2.58E−15	5.36E−17	4.60E−17	3.95E−17	3.38E−17	2.90E−17	0.0
0.0125	1.84E−08	6.98E−16	8.17E−19	6.99E−19	5.98E−19	5.12E−19	4.38E−19	0.0
0.2	1.09E−04	7.09E−01	8.57E−01	8.57E−01	8.57E−01	8.57E−01		
0.1	2.46E−06	7.74E−01	8.58E−01	8.58E−01	8.58E−01	8.58E−01		
0.05	6.32E−08	8.14E−01	8.58E−01	8.58E−01	8.58E−01	8.58E−01		
0.025	7.14E−08	2.08E−02	8.58E−01	8.58E−01	8.58E−01	8.58E−01		
0.0125	3.78E−08	1.17E−03	8.56E−01	8.56E−01	8.56E−01	8.56E−01		

Tabelle 3.24: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauss(6)
0.2	4.02E−08	3.81E−07	4.16E−07	4.47E−07	4.75E−07	4.99E−07	5.19E−07	2.98E−07
0.1	9.88E−09	2.95E−09	3.36E−09	3.75E−09	4.15E−09	4.53E−09	4.90E−09	4.52E−09
0.05	2.45E−09	2.22E−11	2.55E−11	2.87E−11	3.20E−11	3.52E−11	3.84E−11	7.16E−11
0.025	6.08E−10	1.69E−13	1.94E−13	2.18E−13	2.43E−13	2.68E−13	2.93E−13	1.21E−12
0.0125	1.51E−10	1.30E−15	1.49E−15	1.68E−15	1.86E−15	2.05E−15	2.24E−15	2.35E−14
0.2	2.02	7.01	6.96	6.90	6.84	6.78	6.73	6.04
0.1	2.01	7.05	7.04	7.03	7.02	7.01	7.00	5.98
0.05	2.01	7.04	7.04	7.04	7.04	7.04	7.04	5.89
0.025	2.01	7.02	7.03	7.03	7.03	7.03	7.03	5.69
0.0125								
0.2	3.39E−07	8.27E−08	1.18E−07	1.49E−07	1.76E−07	2.00E−07	2.21E−07	0.0
0.1	1.44E−08	1.57E−09	1.17E−09	7.67E−10	3.76E−10	7.20E−12	3.82E−10	0.0
0.05	2.52E−09	4.94E−11	4.62E−11	4.29E−11	3.97E−11	3.64E−11	3.32E−11	0.0
0.025	6.09E−10	1.04E−12	1.02E−12	9.94E−13	9.69E−13	9.44E−13	9.19E−13	0.0
0.0125	1.51E−10	2.22E−14	2.20E−14	2.18E−14	2.16E−14	2.14E−14	2.12E−14	0.0
0.2	2.44E−01	1.43E+00	1.26E+00	1.18E+00	1.14E+00	1.10E+00		
0.1	1.09E−01	7.42E−01	6.58E−01	4.90E−01	1.92E−02	5.30E+01		
0.05	1.96E−02	9.34E−01	9.30E−01	9.25E−01	9.18E−01	9.11E−01		
0.025	1.71E−03	9.76E−01	9.76E−01	9.75E−01	9.74E−01	9.74E−01		
0.0125	1.46E−04	9.92E−01	9.91E−01	9.91E−01	9.91E−01	9.91E−01		

Tabelle 3.25: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: RadauIIA(2)) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	4.02E−08	6.43E−11	5.91E−11	6.12E−11	6.30E−11	6.45E−11	6.58E−11	7.36E−11
0.1	9.88E−09	9.94E−13	9.16E−13	9.49E−13	9.77E−13	1.00E−12	1.02E−12	1.15E−12
0.05	2.45E−09	1.50E−14	1.38E−14	1.43E−14	1.48E−14	1.51E−14	1.54E−14	1.73E−14
0.025	6.08E−10	2.29E−16	2.11E−16	2.18E−16	2.25E−16	2.30E−16	2.35E−16	2.64E−16
0.0125	1.51E−10	3.51E−18	3.25E−18	3.36E−18	3.46E−18	3.54E−18	3.61E−18	4.05E−18
0.2	2.02	6.02	6.01	6.01	6.01	6.01	6.01	6.00
0.1	2.01	6.05	6.05	6.05	6.05	6.05	6.05	6.05
0.05	2.01	6.04	6.04	6.04	6.04	6.04	6.04	6.04
0.025	2.01	6.03	6.02	6.02	6.02	6.02	6.02	6.03
0.0125								
0.2	4.01E−08	9.28E−12	1.44E−11	1.24E−11	1.06E−11	9.08E−12	7.78E−12	0.0
0.1	9.88E−09	1.53E−13	2.31E−13	1.98E−13	1.70E−13	1.46E−13	1.26E−13	0.0
0.05	2.45E−09	2.31E−15	3.48E−15	2.99E−15	2.57E−15	2.21E−15	1.90E−15	0.0
0.025	6.08E−10	3.49E−17	5.27E−17	4.52E−17	3.88E−17	3.33E−17	2.86E−17	0.0
0.0125	1.51E−10	5.36E−19	8.02E−19	6.88E−19	5.90E−19	5.06E−19	4.34E−19	0.0
0.2	2.31E−04	1.56E+00	8.56E−01	8.57E−01	8.57E−01	8.57E−01		
0.1	1.55E−05	1.51E+00	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.05	9.44E−07	1.51E+00	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.025	5.73E−08	1.51E+00	8.59E−01	8.59E−01	8.59E−01	8.59E−01		
0.0125	3.55E−09	1.50E+00	8.58E−01	8.58E−01	8.58E−01	8.58E−01		

Tabelle 3.26: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: RadauIIA(2)) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

des Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ der Iterationsmatrix $\boldsymbol{S}_{\mathrm{IIDeC}}(z)$ für $z \to \infty$ entspricht, vgl. Satz 3.2.4 aus Abschnitt 3.2.3. Der Grund, daß nicht auch im Fall von Gauß-Kollokationsabszissen der entsprechende Spektralradius

$$1 - \frac{m! \cdot m^m}{(2 \cdot m)!} \;=\; 1 - \frac{6! \cdot 6^6}{(2 \cdot 6)!} \;=\; \frac{358}{385} \;\approx\; 0.93 \tag{3.98}$$

genauso deutlich als Konvergenzfaktor in den Tabellen 3.21, 3.23 und 3.25 zu beobachten ist, ist, daß im Fall von RadauIIA-Abszissen die Matrix $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z)$ diagonalisierbar ist,[26] wogegen im Fall von Gauß-Abszissen, wie mit Hilfe von Maple durchgeführte Experimente bestätigen, der Eigenwert $1-\frac{m!\cdot m^m}{(2\cdot m)!}$ der Matrix $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z)$ die geometrische Vielfachheit 1 und die algebraische Vielfachheit N zu haben scheint, d.h. diesem Eigenwert entspricht in der Jordanschen Normalform der Matrix ein Jordan-Block der Größe $N \times N$. (N ist die Anzahl der Teilintervalle der Länge $H = h \cdot m$ des Integrationsintervalls $[t_0, t_{end}]$, vgl. Abschnitt 2.1.2.)

Zur Frage nach dem oben erwähnten ε-Faktors im globalen Fehler im Fall von RadauIIA-Kollokationsabszissen betrachten wir die Tabellen 3.27, 3.28 bzw. 3.29. Hier wurde die Interpolierte Defektkorrektur auf (3.91) mit $\lambda = -\frac{1}{\varepsilon}$ jeweils für $\varepsilon = 10^{-5},\ \varepsilon = 10^{-6}$ und $\varepsilon = 10^{-7}$ angewendet. Man kann deutlich erkennen, daß eine Verkleinerung von ε um den Faktor 10, die Verkleinerung des globalen Fehlers der jeweiligen Basisapproximation und des globalen Fehlers des Fixpunktes RadauIIA(6) um denselben Faktor zur Folge hat. Die mit der Interpolierten Defektkorrektur gewonnenen Approximationen $\eta_h^{[k]}$ werden ebenfalls um den Faktor 10 genauer, und zwar für $k = 1$ der Größenordnung nach, und für $k \geq 2$ fast exakt um diesen Faktor (im Fall von RadauIIA(2) als Basisverfahren gilt das auch schon für $k = 1$).

ε	Euler	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
1E−05	1.04E−07	4.12E−14	1.39E−14	1.44E−14	1.48E−14	1.52E−14	1.55E−14	1.73E−14
1E−06	1.04E−08	2.13E−15	1.39E−15	1.44E−15	1.48E−15	1.52E−15	1.55E−15	1.73E−15
1E−07	1.04E−09	1.93E−16	1.39E−16	1.44E−16	1.48E−16	1.52E−16	1.55E−16	1.74E−16

Tabelle 3.27: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: Implizites Eulerverfahren mit Schrittweite $h = 0.05$) angewendet auf (3.91) mit $\lambda = -\frac{1}{\varepsilon}$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

[26] Zum Beweis konstruiert man für die Matrix $\lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z)$ analog wie für den Fall $N = 1$ im Beweis von Satz 3.2.3 N linear unabhängige Eigenvektoren zum Eigenwert $1-2\cdot\frac{m!\cdot m^m}{(2\cdot m)!}$ und $N \cdot (m-1)$ linear unabhängige Eigenvektoren zum Eigenwert 0, wobei die Tatsache, daß für RadauIIA-Abszissen (2.12) $\gamma_m = 1$ gilt, diese Konstruktion wesentlich erleichtert bzw. überhaupt erst möglich macht. Im Fall von Gauß-Abszissen mit $\gamma_m \neq 1$ funktioniert diese Konstruktion nicht.

ε	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
1E−05	6.90E−08	2.17E−14	1.38E−14	1.43E−14	1.47E−14	1.51E−14	1.54E−14	1.73E−14
1E−06	6.91E−09	1.33E−15	1.38E−15	1.43E−15	1.47E−15	1.51E−15	1.54E−15	1.73E−15
1E−07	6.91E−10	1.24E−16	1.38E−16	1.43E−16	1.47E−16	1.51E−16	1.54E−16	1.74E−16

Tabelle 3.28: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2) mit Schrittweite $h = 0.05$) angewendet auf (3.91) mit $\lambda = -\frac{1}{\varepsilon}$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

ε	RadauIIA(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
1E−05	2.45E−09	1.50E−14	1.38E−14	1.43E−14	1.48E−14	1.51E−14	1.54E−14	1.73E−14
1E−06	2.45E−10	1.50E−15	1.39E−15	1.43E−15	1.48E−15	1.51E−15	1.54E−15	1.73E−15
1E−07	2.45E−11	1.50E−16	1.39E−16	1.43E−16	1.48E−16	1.51E−16	1.54E−16	1.74E−16

Tabelle 3.29: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: RadauIIA(2) mit Schrittweite $h = 0.05$) angewendet auf (3.91) mit $\lambda = -\frac{1}{\varepsilon}$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

Numerische Experimente wie das eben beschriebene lassen somit darauf schließen, daß bei Verwendung von Implizitem Eulerverfahren, SDIRK(2) oder RadauIIA(2) als Basisverfahren und RadauIIA-Abszissen als Kollokationsabszissen, der Faktor ε im globalen Fehler der auf (3.93) mit $\lambda = -\frac{1}{\varepsilon}$ angewendeten Interpolierten Defektkorrektur tatsächlich vorkommt.

IMR und IMR2: Die Tabelle 3.3 aus Abschnitt 3.2.3 zeigt im Fall der klassischen Defektkorrektur nur bei Verwendung der Impliziten Mittelpunktregel (IMR) als Basisverfahren einen Wert ungleich Null für $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$. Für $m = 6$ entnimmt man der Tabelle den Wert $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) \approx 9$. Diesen Wert kann man (zumindest für die größeren Schrittweiten h) für $k \geq 2$ als Konvergenzfaktor[27] in Tabelle 3.30 wiedererkennen. Dadurch vergrößert sich für $k \geq 3$ der Abstand von $\eta_h^{[k]}(t_{end})$ vom Fixpunkt $\eta_h^*(t_{end})$ pro Defektkorrekturschritt um etwa eine Zehnerpotenz, was zur Folge hat, daß auch der Abstand $\eta_h^{[k]}(t_{end})$ von der exakten Lösung an der Stelle t_{end}, d.h. der globale Fehler an dieser Stelle, um etwa eine Zehnerpotenz pro Defektkorrekturschritt größer wird. Somit tritt bei Verwendung der IMR als Basisverfahren keine Fixpunktkonvergenz ein, für größere k zeigen die $\eta_h^{[k]}$ ein divergentes Verhalten, wobei aber auch für kleine k das Genauigkeitsniveau des Fixpunktes nicht erreicht werden kann.

Wegen Satz 3.2.5 aus Abschnitt 3.2.3 darf man erwarten, daß dieses divergente Verhalten der klassischen Defektkorrektur vermieden werden kann, wenn man das Basisverfahren IMR durch IMR2 ersetzt. Diese Erwartung wird in Tabelle 3.31

[27] Der Konvergenzfaktor sollte jetzt besser als „Divergenzfaktor“ bezeichnet werden.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Fixpunkt
0.2	2.22E−03	1.07E−05	9.57E−07	1.00E−06	6.04E−07	4.18E−06	2.80E−05	9.61E−07
0.1	5.54E−04	6.86E−07	1.68E−08	6.44E−09	2.02E−07	1.67E−06	1.51E−05	1.45E−08
0.05	1.38E−04	4.34E−08	2.86E−09	2.31E−08	2.07E−07	1.83E−06	1.62E−05	2.27E−10
0.025	3.46E−05	2.96E−09	2.20E−09	1.84E−08	1.54E−07	1.29E−06	1.07E−05	3.71E−12
0.0125	8.64E−06	3.19E−10	9.74E−10	6.12E−09	3.58E−08	1.84E−07	6.70E−07	6.73E−14
0.2								
0.1	2.00	3.96	5.83	7.28	1.58	1.32	0.89	6.05
0.05	2.00	3.98	2.56	−1.84	−0.03	−0.13	−0.10	6.00
0.025	2.00	3.87	0.38	0.33	0.43	0.51	0.60	5.94
0.0125	2.00	3.22	1.17	1.59	2.10	2.81	4.00	5.79
0.2	2.22E−03	1.17E−05	4.42E−09	3.97E−08	3.57E−07	3.22E−06	2.89E−05	0.0
0.1	5.54E−04	7.01E−07	2.33E−09	2.09E−08	1.88E−07	1.69E−06	1.51E−05	0.0
0.05	1.38E−04	4.37E−08	2.64E−09	2.33E−08	2.07E−07	1.83E−06	1.62E−05	0.0
0.025	3.46E−05	2.97E−09	2.19E−09	1.84E−08	1.54E−07	1.29E−06	1.07E−05	0.0
0.0125	8.64E−06	3.19E−10	9.74E−10	6.12E−09	3.58E−08	1.84E−07	6.70E−07	0.0
0.2	5.25E−03	3.80E−04	8.98E+00	9.00E+00	9.00E+00	9.00E+00		
0.1	1.27E−03	3.33E−03	8.97E+00	8.97E+00	8.97E+00	8.97E+00		
0.05	3.16E−04	6.04E−02	8.85E+00	8.86E+00	8.86E+00	8.85E+00		
0.025	8.58E−05	7.39E−01	8.39E+00	8.37E+00	8.36E+00	8.34E+00		
0.0125	3.69E−05	3.05E+00	6.29E+00	5.85E+00	5.13E+00	3.65E+00		

Tabelle 3.30: IDeC ($m = 6$, Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	IMR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	5.54E−04	5.01E−07	2.25E−07	2.25E−07	2.25E−07	2.25E−07	2.25E−07
0.1	1.38E−04	4.02E−08	3.52E−09	3.52E−09	3.52E−09	3.52E−09	3.52E−09
0.05	3.46E−05	2.65E−09	6.19E−11	6.19E−11	6.19E−11	6.19E−11	6.19E−11
0.025	8.64E−06	1.68E−10	1.25E−12	1.25E−12	1.25E−12	1.25E−12	1.25E−12
0.0125	2.16E−06	1.05E−11	1.91E−14	1.92E−14	1.92E−14	1.92E−14	1.92E−14
0.2							
0.1	2.00	3.64	6.00	6.00	6.00	6.00	6.00
0.05	2.00	3.93	5.83	5.83	5.83	5.83	5.83
0.025	2.00	3.98	5.63	5.63	5.63	5.63	5.63
0.0125	2.00	3.99	6.03	6.03	6.03	6.03	6.03

Tabelle 3.31: IDeC ($m = 6$, Basisverfahren: IMR2) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	2.22E−03	1.10E−05	3.35E−07	2.99E−07	2.97E−07	3.03E−07	2.85E−07	2.98E−07
0.1	5.54E−04	6.91E−07	5.34E−09	3.82E−09	6.68E−09	2.04E−09	2.45E−08	4.52E−09
0.05	1.38E−04	4.33E−08	3.25E−10	6.67E−10	2.30E−09	6.63E−09	2.03E−08	7.16E−11
0.025	3.46E−05	2.77E−09	2.07E−10	5.92E−10	1.71E−09	4.92E−09	1.42E−08	1.21E−12
0.0125	8.64E−06	2.09E−10	9.73E−11	2.28E−10	5.14E−10	1.10E−09	2.18E−09	2.35E−14
0.2								
0.1	2.00	4.00	5.97	6.29	5.48	7.22	3.54	6.04
0.05	2.00	4.00	4.04	2.52	1.54	−1.70	0.27	5.98
0.025	2.00	3.97	0.66	0.17	0.42	0.43	0.52	5.89
0.0125	2.00	3.73	1.09	1.38	1.73	2.16	2.70	5.69
0.2	2.22E−03	1.13E−05	3.65E−08	1.91E−10	1.47E−09	4.52E−09	1.39E−08	0.0
0.1	5.54E−04	6.95E−07	8.20E−10	7.06E−10	2.15E−09	6.56E−09	2.00E−08	0.0
0.05	1.38E−04	4.34E−08	2.54E−10	7.38E−10	2.22E−09	6.70E−09	2.02E−08	0.0
0.025	3.46E−05	2.77E−09	2.05E−10	5.93E−10	1.71E−09	4.93E−09	1.42E−08	0.0
0.0125	8.64E−06	2.09E−10	9.73E−11	2.28E−10	5.14E−10	1.10E−09	2.18E−09	0.0
0.2	5.09E−03	3.22E−03	5.25E−03	7.66E+00	3.08E+00	3.06E+00		
0.1	1.26E−03	1.18E−03	8.62E−01	3.05E+00	3.05E+00	3.05E+00		
0.05	3.13E−04	5.85E−03	2.91E+00	3.01E+00	3.01E+00	3.01E+00		
0.025	8.02E−05	7.41E−02	2.89E+00	2.88E+00	2.88E+00	2.88E+00		
0.0125	2.42E−05	4.66E−01	2.34E+00	2.26E+00	2.15E+00	1.97E+00		

Tabelle 3.32: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	IMR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	5.54E−04	4.34E−07	3.01E−07	3.02E−07	3.02E−07	3.02E−07	3.02E−07	2.98E−07
0.1	1.38E−04	3.92E−08	4.71E−09	4.72E−09	4.72E−09	4.72E−09	4.72E−09	4.52E−09
0.05	3.46E−05	2.63E−09	8.28E−11	8.29E−11	8.29E−11	8.29E−11	8.29E−11	7.16E−11
0.025	8.64E−06	1.67E−10	1.67E−12	1.67E−12	1.67E−12	1.67E−12	1.67E−12	1.21E−12
0.0125	2.16E−06	1.05E−11	2.56E−14	2.57E−14	2.57E−14	2.57E−14	2.57E−14	2.35E−14
0.2								
0.1	2.00	3.47	6.00	6.00	6.00	6.00	6.00	6.04
0.05	2.00	3.90	5.83	5.83	5.83	5.83	5.83	5.98
0.025	2.00	3.97	5.63	5.63	5.63	5.63	5.63	5.89
0.0125	2.00	3.99	6.03	6.03	6.03	6.03	6.03	5.69

Tabelle 3.33: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: IMR2) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	IMR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	2.22E−03	1.11E−05	9.64E−07	5.05E−06	4.20E−05	1.93E−04	1.19E−04	7.36E−11
0.1	5.54E−04	7.56E−07	1.14E−07	2.84E−06	6.41E−06	8.67E−05	5.00E−05	1.15E−12
0.05	1.38E−04	7.56E−08	3.05E−07	8.19E−07	2.61E−05	2.13E−05	1.26E−03	1.73E−14
0.025	3.46E−05	1.69E−08	3.70E−07	4.16E−06	9.89E−07	5.00E−04	3.40E−03	2.64E−16
0.0125	8.64E−06	4.25E−09	2.43E−07	8.39E−06	1.77E−04	2.04E−03	2.22E−03	4.05E−18
0.2								
0.1	2.00	3.88	3.08	0.83	2.71	1.15	1.26	6.00
0.05	2.00	3.32	−1.42	1.80	−2.03	2.02	−4.66	6.05
0.025	2.00	2.16	−0.28	−2.34	4.72	−4.55	−1.43	6.04
0.0125	2.00	1.99	0.61	−1.01	−7.48	−2.03	0.62	6.03
0.2	2.22E−03	1.11E−05	9.64E−07	5.05E−06	4.20E−05	1.93E−04	1.19E−04	0.0
0.1	5.54E−04	7.56E−07	1.14E−07	2.84E−06	6.41E−06	8.67E−05	5.00E−05	0.0
0.05	1.38E−04	7.56E−08	3.05E−07	8.19E−07	2.61E−05	2.13E−05	1.26E−03	0.0
0.025	3.46E−05	1.69E−08	3.70E−07	4.16E−06	9.89E−07	5.00E−04	3.40E−03	0.0
0.0125	8.64E−06	4.25E−09	2.43E−07	8.39E−06	1.77E−04	2.04E−03	2.22E−03	0.0
0.2	5.00E−03	8.69E−02	5.24E+00	8.32E+00	4.59E+00	6.19E−01		
0.1	1.37E−03	1.51E−01	2.50E+01	2.25E+00	1.35E+01	5.77E−01		
0.05	5.46E−04	4.04E+00	2.69E+00	3.19E+01	8.18E−01	5.90E+01		
0.025	4.89E−04	2.19E+01	1.12E+01	2.38E−01	5.06E+02	6.80E+00		
0.0125	4.92E−04	5.72E+01	3.45E+01	2.11E+01	1.15E+01	1.09E+00		

Tabelle 3.34: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: IMR) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	IMR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	5.54E−04	5.81E−07	8.52E−08	8.33E−08	8.33E−08	8.33E−08	8.33E−08	7.36E−11
0.1	1.38E−04	4.12E−08	1.75E−09	1.72E−09	1.72E−09	1.72E−09	1.72E−09	1.15E−12
0.05	3.46E−05	2.66E−09	3.46E−11	3.42E−11	3.42E−11	3.42E−11	3.42E−11	1.73E−14
0.025	8.64E−06	1.68E−10	7.33E−13	7.25E−13	7.25E−13	7.25E−13	7.25E−13	2.64E−16
0.0125	2.16E−06	1.05E−11	1.13E−14	1.12E−14	1.12E−14	1.12E−14	1.12E−14	4.05E−18
0.2								
0.1	2.00	3.82	5.61	5.60	5.60	5.60	5.60	6.00
0.05	2.00	3.95	5.66	5.65	5.65	5.65	5.65	6.05
0.025	2.00	3.99	5.56	5.56	5.56	5.56	5.56	6.04
0.0125	2.00	3.99	6.02	6.02	6.02	6.02	6.02	6.03

Tabelle 3.35: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: IMR2) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

bestätigt,[28] man beobachtet die rasche Fixpunktkonvergenz der klassischen Defektkorrektur mit IMR2 als Basisverfahren, nach zwei Defektkorrekturschritten[29] wird der Fixpunkt erreicht.

Im Fall der Interpolierten Defektkorrektur mit IMR als Basisverfahren zeigt sich ein ähnliches Bild wie bei der klassischen Defektkorrektur. Für Gauß-Kollokationsabszissen entnimmt man der Tabelle 3.3 aus Abschnitt 3.2.3 den Wert $\lim_{z\to\infty} \rho \boldsymbol{S}_{\mathrm{IIDeC}}(z)) \approx 3$, welcher sich für größere Schrittweiten h in Tabelle 3.32 als Konvergenzfaktor für $k \geq 4$ beobachten läßt. Wieder tritt keine Fixpunktkonvergenz ein, die $\eta_h^{[k]}$ zeigen für größere k ein divergentes Verhalten, wenn auch (wegen des Konvergenzfaktors 3 statt 9) nicht ganz so ausgeprägt wie im Fall der klassischen Defektkorrektur, und wieder kann auch für kleine k das Genauigkeitsniveau des Fixpunktes nicht erreicht werden.

Das unregelmäßige Verhalten der Konvergenzfaktoren in Tabelle 3.34 bei Verwendung von RadauIIA-Kollokationsabszissen deutet darauf hin, daß in diesem Fall die Iterationsmatrix $\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)$ der Interpolierten Defektkorrektur nicht diagonalisierbar ist. Die $\eta_h^{[k]}$ divergieren für $k \geq 2$ vom Fixpunkt weg, wobei besonders deutlich das Genauigkeitsniveau des Fixpunktes RadauIIA(6) verfehlt wird, welches hier wegen des ε-Faktors im globalen Fehler der RadauIIA-Verfahren besonders gut ist.

Im Fall der Interpolierten Defektkorrektur mit IMR2 als Basisverfahren gilt für den Spektralradius der Iterationsmatrix $\lim_{h\lambda\to\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)) = 1$, vgl. Satz 3.2.6. Das läßt für endliches $\lambda \ll 0$ wenn überhaupt nur eine extrem langsame Konvergenz der $\eta_h^{[k]}$ gegen die Kollokationslösung η_h^* erwarten.[30] Tatsächlich beobachtet man in den Tabellen 3.33 und 3.35, daß ab dem zweiten Defektkorrekturschritt der globale Fehler der $\eta_h^{[k]}$ stagniert, und zwar etwa auf dem Niveau des Fixpunktes der klassischen Defektkorrektur (vgl. Tabelle 3.31). Das bedeutet, daß bei Verwendung von IMR2 als Basisverfahren, die Interpolierte Defektkorrektur mit RadauIIA-Kollokationsabszissen gegenüber der klassischen Defektkorrektur bei Anwendung auf (3.91) mit $\lambda \ll 0$ keine Vorteile bringt. (Bei Verwendung

[28] Beim Vergleich der Tabellen 3.30 und 3.31 beachte man, daß ein Integrationsschritt mit IMR2 zu zwei Integrationsschritten der halben Schrittweite mit IMR äquivalent ist, d.h. die Einträge in der zweiten Spalte von Tabelle 3.31 sind gegenüber den entsprechenden Einträgen in Tabelle 3.30 um eine Zeile nach oben verschoben. Für die Tabellen 3.32 und 3.33 bzw. 3.34 und 3.35 gilt analoges.

[29] Das ist kein Widerspruch zu Satz 3.2.5, der besagt, daß die Iterationsmatrix der klassischen Defektkorrektur mit IMR2 als Basisverfahren für $h\lambda \to \infty$ nilpotent der Ordnung $\lceil \frac{m}{2} \rceil$ ist. Für $m = 6$ folgt daraus, daß für $\lambda \ll 0$ nach $\lceil \frac{6}{2} \rceil = 3$ Defektkorrekturschritten der Fixpunkt erreicht sein sollte, was aber nicht ausschließt, daß er auch schon früher erreicht werden kann.

[30] Im Grenzfall für $h\lambda \to \infty$ ist die Matrix $I_{Nm} - \boldsymbol{S} = I_{Nm} - \lim_{h\lambda\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(h\lambda)$ wegen des Eigenwerts 1 von $\boldsymbol{S} := \lim_{z\to\infty} \boldsymbol{S}_{\mathrm{IIDeC}}(z)$ singulär, woraus folgt, daß in diesem Grenzfall der Fixpunkt der Iteration (3.6), d.h. die Lösung $\boldsymbol{\eta}^*$ von $(I_{Nm} - \boldsymbol{S}) \cdot \boldsymbol{\eta}^* = \boldsymbol{v}$ nicht eindeutig bestimmt ist. Das liefert eine Erklärung dafür, daß in den Tabellen 3.33 und 3.35 dem Anschein nach für $k \geq 3$ ein Fixpunkt erreicht wird, der aber ungleich der Kollokationslösung ist.

von Gauß-Abszissen bringt hier die Interpolierte Defektkorrektur ohnehin keine Vorteile, weil in diesem Fall das Gauß(6)-Verfahren (vgl. Tabelle 3.33) und der Fixpunkt der klassischen Defektkorrektur (vgl. Tabelle 3.31) ungefähr das gleiche Genauigkeitsniveau haben.)

ITR und ITR2: Im Fall der klassischen Defektkorrektur mit der Impliziten Trapezregel (ITR) als Basisverfahren beobachtet man in Tabelle 3.36 nur für die größeren Schrittweiten $h = 0.2$ und $h = 0.1$ eine rasche Fixpunktkonvergenz innerhalb von 2 bis 3 Defektkorrekturschritten. Für die kleineren Schrittweiten $h = 0.025$ und $h = 0.0125$ ist auch nach 5 Defektkorrekturschritten ein Fixpunkt noch nicht erreicht. Die Situation verbessert sich deutlich, wenn das Basisverfah-

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	6.17E−08	1.03E−10	2.25E−11	2.22E−11	2.22E−11	2.22E−11	2.22E−11
0.1	1.53E−08	1.23E−10	1.29E−12	3.31E−13	3.36E−13	3.36E−13	3.36E−13
0.05	3.75E−09	1.19E−10	3.67E−12	7.01E−14	6.48E−15	5.29E−15	5.30E−15
0.025	8.75E−10	1.03E−10	1.27E−11	1.05E−12	6.46E−14	3.10E−15	2.25E−16
0.0125	1.73E−10	5.89E−11	2.90E−11	9.52E−12	2.34E−12	4.62E−13	7.59E−14
0.2							
0.1	2.01	−0.26	4.13	6.07	6.05	6.05	6.05
0.05	2.03	0.05	−1.51	2.24	5.70	5.99	5.98
0.025	2.10	0.20	−1.79	−3.90	−3.32	0.77	4.56
0.0125	2.33	0.81	−1.19	−3.19	−5.18	−7.22	−8.40

Tabelle 3.36: IDeC ($m = 6$, Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	ITR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.53E−08	7.72E−12	7.72E−12	7.72E−12	7.72E−12	7.72E−12	7.72E−12
0.1	3.75E−09	1.17E−13	1.17E−13	1.17E−13	1.17E−13	1.17E−13	1.17E−13
0.05	8.75E−10	1.86E−15	1.86E−15	1.86E−15	1.86E−15	1.86E−15	1.86E−15
0.025	1.73E−10	3.50E−17	3.50E−17	3.50E−17	3.50E−17	3.50E−17	3.50E−17
0.0125	3.00E−11	9.81E−19	9.81E−19	9.81E−19	9.81E−19	9.81E−19	9.81E−19
0.2							
0.1	2.03	6.05	6.05	6.05	6.05	6.05	6.05
0.05	2.10	5.97	5.97	5.97	5.97	5.97	5.97
0.025	2.33	5.73	5.73	5.73	5.73	5.73	5.73
0.0125	2.53	5.16	5.16	5.16	5.16	5.16	5.16

Tabelle 3.37: IDeC ($m = 6$, Basisverfahren: ITR2) angewendet auf (3.91) mit $\lambda = -100000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	6.17E−08	6.64E−08	1.51E−06	5.89E−06	3.13E−05	1.53E−04	7.54E−04	2.98E−07
0.1	1.53E−08	2.35E−08	9.15E−08	4.77E−07	2.27E−06	1.08E−05	5.04E−05	4.52E−09
0.05	3.75E−09	5.62E−09	2.74E−08	1.33E−07	6.24E−07	2.89E−06	1.32E−05	7.16E−11
0.025	8.75E−10	1.11E−09	5.33E−09	2.50E−08	1.15E−07	5.20E−07	2.32E−06	1.21E−12
0.0125	1.73E−10	9.44E−11	4.03E−10	1.71E−09	7.14E−09	2.97E−08	1.22E−07	2.35E−14
0.2								
0.1	2.01	1.50	4.05	3.62	3.78	3.83	3.90	6.04
0.05	2.03	2.06	1.74	1.85	1.87	1.90	1.94	5.98
0.025	2.10	2.34	2.37	2.41	2.44	2.47	2.51	5.89
0.0125	2.33	3.55	3.72	3.87	4.01	4.13	4.24	5.69
0.2	3.60E−07	2.32E−07	1.21E−06	6.18E−06	3.10E−05	1.54E−04	7.53E−04	0.0
0.1	1.98E−08	1.90E−08	9.60E−08	4.73E−07	2.28E−06	1.08E−05	5.04E−05	0.0
0.05	3.83E−09	5.55E−09	2.75E−08	1.33E−07	6.24E−07	2.89E−06	1.32E−05	0.0
0.025	8.76E−10	1.11E−09	5.33E−09	2.50E−08	1.15E−07	5.20E−07	2.32E−06	0.0
0.0125	1.73E−10	9.44E−11	4.03E−10	1.71E−09	7.14E−09	2.97E−08	1.22E−07	0.0
0.2	6.44E−01	5.22E+00	5.11E+00	5.02E+00	4.95E+00	4.90E+00		
0.1	9.56E−01	5.07E+00	4.92E+00	4.82E+00	4.74E+00	4.67E+00		
0.05	1.45E+00	4.96E+00	4.82E+00	4.71E+00	4.63E+00	4.56E+00		
0.025	1.26E+00	4.81E+00	4.69E+00	4.60E+00	4.52E+00	4.46E+00		
0.0125	5.44E−01	4.27E+00	4.23E+00	4.19E+00	4.15E+00	4.12E+00		

Tabelle 3.38: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	ITR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Gauß(6)
0.2	1.53E−08	2.74E−08	2.87E−08	2.87E−08	2.87E−08	2.87E−08	2.87E−08	2.98E−07
0.1	3.75E−09	1.60E−10	1.63E−10	1.63E−10	1.63E−10	1.63E−10	1.63E−10	4.52E−09
0.05	8.75E−10	1.70E−11	1.70E−11	1.70E−11	1.70E−11	1.70E−11	1.70E−11	7.16E−11
0.025	1.73E−10	1.08E−12	1.08E−12	1.08E−12	1.08E−12	1.08E−12	1.08E−12	1.21E−12
0.0125	3.00E−11	2.50E−14	2.50E−14	2.50E−14	2.50E−14	2.50E−14	2.50E−14	2.35E−14
0.2								
0.1	2.03	7.42	7.45	7.45	7.45	7.45	7.45	6.04
0.05	2.10	3.23	3.27	3.27	3.27	3.27	3.27	5.98
0.025	2.33	3.97	3.97	3.97	3.97	3.97	3.97	5.89
80.0125	2.53	5.44	5.44	5.44	5.44	5.44	5.44	5.69

Tabelle 3.39: IIDeC (Kollokationsabszissen: Gauß(6), Basisverfahren: ITR2) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	ITR	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	6.17E−08	7.57E−07	4.35E−06	8.10E−06	6.93E−04	1.33E−02	1.96E−01	7.36E−11
0.1	1.53E−08	4.44E−07	1.36E−05	3.16E−04	6.10E−03	1.04E−01	1.62E+00	1.15E−12
0.05	3.75E−09	2.11E−07	1.23E−05	5.04E−04	1.65E−02	4.56E−01	1.11E+01	1.73E−14
0.025	8.75E−10	9.10E−08	1.02E−05	7.82E−04	4.65E−02	2.28E+00	9.57E+01	2.64E−16
0.0125	1.73E−10	2.58E−08	5.64E−06	8.36E−04	9.45E−02	8.67E+00	6.73E+02	4.05E−18
0.2								
0.1	2.01	0.77	−1.65	−5.28	−3.14	−2.96	−3.05	6.00
0.05	2.03	1.07	0.15	−0.68	−1.43	−2.13	−2.77	6.05
0.025	2.10	1.21	0.27	−0.63	−1.49	−2.32	−3.11	6.04
0.0125	2.33	1.82	0.85	−0.10	−1.02	−1.93	−2.82	6.03
0.2	6.16E−08	7.57E−07	4.35E−06	8.10E−06	6.93E−04	1.33E−02	1.96E−01	0.0
0.1	1.53E−08	4.44E−07	1.36E−05	3.16E−04	6.10E−03	1.04E−01	1.62E+00	0.0
0.05	3.75E−09	2.11E−07	1.23E−05	5.04E−04	1.65E−02	4.56E−01	1.11E+01	0.0
0.025	8.75E−10	9.10E−08	1.02E−05	7.82E−04	4.65E−02	2.28E+00	9.57E+01	0.0
0.0125	1.73E−10	2.58E−08	5.64E−06	8.36E−04	9.45E−02	8.67E+00	6.73E+02	0.0
0.2	1.23E+01	5.75E+00	1.86E+00	8.56E+01	1.92E+01	1.47E+01		
0.1	2.90E+01	3.07E+01	2.31E+01	1.93E+01	1.71E+01	1.55E+01		
0.05	5.62E+01	5.81E+01	4.11E+01	3.27E+01	2.76E+01	2.43E+01		
0.025	1.04E+02	1.12E+02	7.69E+01	5.94E+01	4.90E+01	4.20E+01		
0.0125	1.49E+02	2.19E+02	1.48E+02	1.13E+02	9.18E+01	7.77E+01		

Tabelle 3.40: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: ITR) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler, beobachtete Konvergenzordnung, Abstand zum Fixpunkt und Konvergenzfaktor an der Stelle $t_{end} = 3.6$.

h	ITR2	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.53E−08	1.76E−07	1.75E−07	1.75E−07	1.75E−07	1.75E−07	1.75E−07	7.36E−11
0.1	3.75E−09	1.34E−09	1.34E−09	1.34E−09	1.34E−09	1.34E−09	1.34E−09	1.15E−12
0.05	8.75E−10	1.37E−12	1.37E−12	1.37E−12	1.37E−12	1.37E−12	1.37E−12	1.73E−14
0.025	1.73E−10	4.37E−13	4.37E−13	4.37E−13	4.37E−13	4.37E−13	4.37E−13	2.64E−16
0.0125	3.00E−11	1.09E−14	1.09E−14	1.09E−14	1.09E−14	1.09E−14	1.09E−14	4.05E−18
0.2								
0.1	2.03	7.03	7.03	7.03	7.03	7.03	7.03	6.00
0.05	2.10	9.94	9.94	9.94	9.94	9.94	9.94	6.05
0.025	2.33	1.65	1.64	1.64	1.64	1.64	1.64	6.04
0.0125	2.53	5.33	5.33	5.33	5.33	5.33	5.33	6.03

Tabelle 3.41: IIDeC (Kollokationsabszissen: RadauIIA(6), Basisverfahren: ITR2) angewendet auf (3.91) mit $\lambda = -10000$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

ren ITR durch ITR2 ersetzt wird.[31] Nun wird, wie man in Tabelle 3.37 beobachtet, für alle dort betrachteten Schrittweiten schon nach einem Defektkorrekturschritt der Fixpunkt erreicht. Für den großen Unterschied, der hier zwischen den Basisverfahren ITR und ITR2 besteht, obwohl für beide nach Satz 3.2.6 aus Abschnitt 3.2.3 $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IDeC}}(z)) = 0$ gilt, liefert der Vergleich der Abbildungen 3.10 und 3.13 eine Erklärung: Für die in den Tabellen 3.36 und 3.37 betrachteten Werte für $z = h\lambda$ ist der Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ im Fall von ITR als Basisverfahren (vgl. Abbildung 3.10) wesentlich größer als im Fall von ITR2 als Basisverfahren (vgl. Abbildung 3.13).

Was nun die Interpolierte Defektkorrektur betrifft, so ergibt sich mit ITR bzw. ITR2 als Basisverfahren ein ähnliches Bild wie bei Verwendung von IMR bzw. IMR2 als Basisverfahren: Wieder ist mit ITR als Basisverfahren keine Fixpunktkonvergenz festzustellen, sondern man beobachtet für Gauß-Kollokationsabszissen in Tabelle 3.38, wie der Abstand der $\eta_h^{[k]}$ vom Fixpunkt η_h^* nach und nach größer wird. Für RadauIIA-Kollokationsabszissen ist die Divergenz der $\eta_h^{[k]}$ besonders ausgeprägt. Wie man der Tabelle 3.40 entnimmt, verliert man nun für die größeren Schrittweiten h pro Defektkorrekturschritt eine Dezimalstelle an Genauigkeit, für die kleineren Schrittweiten sogar fast zwei Dezimalstellen.

Durch Ersetzen des Basisverfahrens ITR durch ITR2 kann zwar dieses divergente Verhalten verhindert werden, doch sitzt man nun anscheinend nach einem Defektkorrekturschritt auf einem Fixpunkt fest, der ungleich der jeweiligen Kollokationslösung Gauss(6) (vgl. Tabelle 3.39) bzw. RadauIIA(6) (vgl. Tabelle 3.41) ist, was mit der Tatsache vereinbar ist, daß bei Verwendung von ITR2 als Basisverfahren, $\lim_{z\to\infty} \rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z)) = 1$ gilt, vgl. Satz 3.2.8. Durch Vergleichen der Tabellen 3.39 und 3.41 mit Tabelle 3.37 erkennt man, daß bei Verwendung von ITR2 als Basisverfahren mit Hilfe der klassischen Defektkorrektur eine wesentlich höhere Genauigkeit erzielt werden kann als mit Hilfe der Interpolierten Defektkorrektur. Somit bringt die Interpolierte Defektkorrektur mit ITR2 als Basisverfahren bei Anwendung auf (3.91) mit $\lambda \ll 1$ gegenüber der klassischen Defektkorrektur keine Vorteile, das ist analog zu der Situation, die wir schon im Fall von IMR2 als Basisverfahren beobachtet haben.

3.4.3 Zusammenfassung

In diesem Abschnitt 3.4 haben wir verschiedene Varianten der Interpolierten Defektkorrektur auf skalare Anfangswertprobleme der Gestalt (3.93) sowohl im

[31] Beim Vergleich der Tabellen 3.36 und 3.37 beachte man, daß ein Integrationsschritt mit ITR2 zu zwei Integrationsschritten der halben Schrittweite mit ITR äquivalent ist, d.h. die Einträge in der zweiten Spalte von Tabelle 3.37 sind gegenüber den entsprechenden Einträgen in Tabelle 3.36 um eine Zeile nach oben verschoben. Für die Tabellen 3.38 und 3.39 bzw. 3.40 und 3.41 gilt analoges.

nichtsteifen als auch im steifen Fall numerisch angewendet, wobei sich diese Varianten im wesentlichen durch die verwendeten Basisverfahren und/oder die verwendeten Kollokationsabszissen unterscheiden. Obwohl die betrachtete Klasse von Modellproblemen sehr speziell ist, können aus den Ergebnissen dieser Experimente dennoch erste Schlußfolgerungen gezogen werden, wobei insbesondere Ergebnisse negativer Art von Bedeutung sind, weil Schwierigkeiten, die einem speziellen Defektkorrekturalgorithmus von Problemen der Gestalt (3.93) bereitet werden, erwarten lassen, daß dieser Algorithmus auch für allgemeinere Probleme nicht zufriedenstellend funktionieren wird.

Da, wie wir in Abschnitt 3.4.1 festgestellt haben, im nichtsteifen Fall alle hier betrachteten Varianten der Interpolierten Defektkorrektur zufriedenstellend arbeiten, wollen wir uns bei der Beurteilung dieser Varianten auf die Ergebnisse im steifen Fall aus Abschnitt 3.4.2 beschränken. Dabei wird besonders der Fall von RadauIIA-Kollokationsabszissen beachtet, da hier die entsprechenden Kollokationslösungen wegen des im globalen Fehler auftretenden ε-Faktors besonders gute Approximationen für die exakte Lösung sind:

- Die „klassischen“ Verfahren der Ordnung 2, nämlich IMR und ITR sind als Basisverfahren ungeeignet, da mit ihnen keine Fixpunktkonvergenz eintritt, sondern im Gegenteil ein mehr oder weniger starkes divergentes Verhalten festzustellen ist.

- Mit den Verfahren IMR2 und ITR2 als Basisverfahren kann zwar dieses divergente Verhalten verhindert werden, eine Fixpunktkonvergenz zur RadauIIA-Kollokationslösung tritt aber dennoch nicht ein, und der Genauigkeitsgrad der einzelnen Approximationen liegt tief unter jenem des RadauIIA-Verfahrens.

- Mit den übrigen hier betrachteten Basisverfahren (Implizites Eulerverfahren, SDIRK(2), RadauIIA(2)) lassen sich bei Anwendung auf steife Probleme der Gestalt (3.93) gute Ergebnisse erzielen: Nach nur ein bis zwei Defektkorrekturschritten kann die (reduzierte) Konvergenzordnung und das hohe Genauigkeitsniveau des Fixpunktes, d.h. des RadauIIA-Verfahrens erreicht werden, der ε-Faktor im globalen Fehler des Fixpunktes ist auch im globalen Fehler der einzelnen Approximationen feststellbar.

Somit sind nur die zuletzt genannten Basisverfahren im Fall von RadauIIA-Kollokationsabszissen attraktive Kandidaten für die Interpolierte Defektkorrektur, und wir werden uns im folgenden auf diese beschränken. Allerdings besteht hier bei RadauIIA(2) der Nachteil, daß damit in jedem Integrationsschritt ein algebraisches Gleichungssystem gelöst werden muß, welches von doppelt so großer Dimension ist wie jenes, das beim Implizitem Eulerverfahren gelöst werden muß, bzw. wie jene beiden, die beim diagonalimplizitem Verfahren SDIRK(2) gelöst

werden müssen. Auch die Vorteile von RadauIIA(2) gegenüber SDIRK(2), wie die etwas schnellere Fixpunktkonvergenz im nichtsteifen Fall, lassen diesen höheren Aufwand beim Lösen der algebraischen Gleichungen nicht vertretbar erscheinen. Hingegen erscheint bei Verwendung des Impliziten Eulerverfahrens als Basisverfahren die Konvergenzgeschwindigkeit gegen den Fixpunkt im nichtsteifen Fall als zu gering und als zu wenig gut vorhersagbar, sodaß wir im folgenden hauptsächlich SDIRK(2) als Basisverfahren betrachten werden.

3.5 Numerische Experimente (2): lineares Anfangswertproblem der Dimension 2

Nachdem wir nun ausführlich das numerische Verhalten der Defektkorrekturalgorithmen bei Anwendung auf ein skalares lineares Anfangswertproblem untersucht haben, wenden wir uns dem „nächst schwierigeren“ Fall, nämlich einem linearen Anfangswertproblem der Dimension 2 zu. Konkret betrachten wir auf dem Integrationsintervall $[t_0, t_{end}] = [0, 3.6]$ das Problem

$$y' \;=\; A(t) \cdot (y - g(t)) + g'(t), \quad g(t) \;=\; \begin{pmatrix} \sin t + 2 \\ \cos t + 2 \end{pmatrix}, \quad y(0) = g(0) \tag{3.99}$$

mit der exakten (von der Wahl der Matrix $A(t)$ unabhängigen) Lösung $y(t) = g(t)$, wobei wir für $A(t)$ Matrizen der folgenden Gestalt wählen:

- steife konstante Matrix $A(t) \equiv A$:

$$A \;=\; \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{\varepsilon} & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^{-1}, \tag{3.100}$$

- steife Matrix mit konstanten Eigenrichtungen und variierenden Eigenwerten:

$$A(t) \;=\; \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{\varepsilon} \cdot (\cos t + 2) & 0 \\ 0 & -(\sin t + 2) \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}^{-1}, \tag{3.101}$$

- steife Matrix mit variierenden orthogonalen Eigenrichtungen:

$$A(t) \;=\; \begin{pmatrix} \cos \omega t & \sin \omega t \\ -\sin \omega t & \cos \omega t \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{\varepsilon} & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} \cos \omega t & \sin \omega t \\ -\sin \omega t & \cos \omega t \end{pmatrix}^{-1}. \tag{3.102}$$

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	3.04E−03	6.67E−07	1.11E−08	5.54E−10	7.42E−11	9.85E−11	9.73E−11	9.73E−11
0.1	7.43E−04	5.65E−08	1.14E−10	2.11E−12	4.78E−15	5.02E−14	4.87E−14	4.85E−14
0.05	1.84E−04	3.84E−09	1.34E−12	1.02E−14	1.44E−16	1.40E−17	1.57E−17	1.97E−17
0.025	4.57E−05	2.48E−10	1.76E−14	5.77E−17	9.60E−19	5.96E−19	6.10E−19	6.87E−19
0.0125	1.14E−05	1.57E−11	2.51E−16	3.67E−19	1.08E−20	9.79E−21	1.00E−20	1.12E−20
0.2								
0.1	2.03	3.56	6.60	8.04	13.92	10.94	10.96	10.97
0.05	2.02	3.88	6.42	7.69	5.05	11.81	11.60	11.27
0.025	2.01	3.96	6.25	7.47	7.23	4.55	4.68	4.84
0.0125	2.00	3.98	6.13	7.30	6.48	5.93	5.93	5.93

Tabelle 3.42: IIDeC(Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) angewendet auf (3.99) mit $A(t)$ aus (3.100) und $\varepsilon = 10^{-8}$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	3.18E−03	2.88E−06	2.83E−09	2.15E−10	8.55E−10	8.07E−10	8.11E−10	8.10E−10
0.1	7.71E−04	1.85E−07	2.77E−11	3.86E−12	3.54E−13	1.69E−13	1.77E−13	1.77E−13
0.05	1.90E−04	1.15E−08	4.69E−14	2.26E−14	6.28E−16	8.68E−17	1.00E−16	1.04E−16
0.025	4.72E−05	7.16E−10	3.46E−15	1.30E−16	2.12E−18	5.63E−19	5.93E−19	6.62E−19
0.0125	1.17E−05	4.46E−11	9.00E−17	8.07E−19	1.35E−20	8.87E−21	9.08E−21	1.02E−20
0.2								
0.1	2.04	3.96	6.68	5.80	11.24	12.22	12.16	12.16
0.05	2.02	4.01	9.20	7.42	9.14	10.93	10.79	10.74
0.025	2.01	4.01	3.76	7.44	8.21	7.27	7.40	7.29
0.0125	2.01	4.01	5.26	7.33	7.30	5.99	6.03	6.02

Tabelle 3.43: IIDeC(Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) angewendet auf (3.99) mit $A(t)$ aus (3.101) und $\varepsilon = 10^{-8}$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

Im Fall der Matrizen (3.100) bzw. (3.101) setzen wir $\varepsilon = 10^{-8}$ und wenden auf das so spezifizierte Anfangswertproblem (3.99) die Interpolierte Defektkorrektur mit SDIRK(2) als Basisverfahren, RadauIIA-Kollokationsabszissen und verschiedenen Schrittweiten h an, wobei wir wieder den Polynomgrad $m = 6$ wählen. Die Ergebnisse für die Stelle $t_{end} = 3.6$ sind in den Tabellen 3.42 bzw. 3.43 zusammengefaßt. Man erkennt, daß hier für die Interpolierte Defektkorrektur keinerlei Schwierigkeiten auftreten, nach jeweils 5 Defektkorrekturschritten wird das Genauigkeitsniveau und die Konvergenzordnung des Fixpunktes erreicht. Für die kleineren Schrittweiten h beobachtet man für den Fixpunkt RadauIIA(6) die klassische Ordnung $2m - 1 = 11$. Das illustriert schön die Tatsache, daß auch ein Problem der Gestalt (3.99) (bei Umformulierung als autonomes Problem wie in (3.95)) in jene Klasse steifer Anfangswertprobleme fällt, die in der in den Abschnitten 1.2.4 und 3.4.2 erwähnten Konvergenztheorie für

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	2.90E−05	2.20E−03	1.71E−01	1.33E+01	1.03E+03	7.98E+04	4.14E−11
0.1	2.32E−05	2.67E−08	5.61E−07	6.56E−06	7.27E−05	8.09E−04	8.99E−03	1.08E−13
0.05	4.59E−06	3.26E−09	8.49E−10	1.06E−09	2.38E−09	5.11E−09	1.02E−08	1.75E−15
0.025	9.95E−07	2.16E−10	1.35E−11	4.89E−12	1.11E−12	3.39E−12	1.08E−12	2.79E−17
0.0125	2.30E−07	1.35E−11	6.06E−13	1.21E−13	6.46E−14	7.34E−14	6.16E−14	4.39E−19
0.2	2.52	10.09	11.94	14.67	17.48	20.28	23.08	8.58
0.1	2.34	3.03	9.37	12.60	14.90	17.27	19.75	5.95
0.05	2.21	3.92	5.97	7.76	11.07	10.56	13.21	5.97
0.025	2.12	4.00	4.48	5.33	4.10	5.53	4.13	5.99
0.0125								

Tabelle 3.44: IIDeC(Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) angewendet auf (3.99) mit $A(t)$ aus (3.102), $\omega = 0.2$ und $\varepsilon = 10^{-6}$: Globaler Fehler und beobachtete Konvergenzordnung an der Stelle $t_{end} = 3.6$.

IRK-Verfahren betrachtet wird. Die dort hergeleitete Fehlerschranke der Form $\max(O(\varepsilon h^m), O(h^{2m-1}))$ für das RadauIIA(m)-Verfahren läßt nämlich erwarten, daß sich die klassische Ordnung $2m-1$ beobachten läßt, wenn ε im Vergleich zu h hinreichend klein ist.

Wenn wir nun in (3.99) für $A(t)$ die Matrix aus (3.102) wählen, so ergibt sich ein völlig anderes Bild. In Tabelle 3.44 setzen wir $\varepsilon = 10^{-6}$, und für die „Drehgeschwindigkeit“ der Eigenrichtungen der Matrix (3.102) wählen wir den eher moderaten Wert $\omega = 0.2$. Für kleinere Schrittweiten beobachtet man im Vergleich zu Tabellen 3.42 und 3.43 eine deutlich reduzierte Konvergenzgeschwindigkeit gegen den Fixpunkt, und für größere Schrittweiten tritt überhaupt ein divergentes Verhalten in Erscheinung, welches für die größte Schrittweite $h = 0.2$ besonders deutlich ausfällt.

Diese Schwierigkeiten der Interpolierten Defektkorrektur mit variierenden (steifen) Eigenrichtungen sind typisch. Eine nähere Untersuchung dieser Schwierigkeiten und auch die Frage, wie diese überwunden werden können, sind Gegenstand des folgenden Kapitels.

Kapitel 4

Kombination der Interpolierten Defektkorrektur mit dem QR-Verfahren

4.1 Numerische Instabilität der Interpolierten Defektkorrektur bei variierender steifer Eigenrichtung

Wir wenden die Techniken aus Abschnitt 3.1 auf steife Probleme der Gestalt (3.1) an, für die die jeweilige Matrix $A(t)$ eine variierende steife Eigenrichtung hat. Konkret betrachten wir die Matrix

$$A(t) \;=\; \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{\varepsilon} & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix}^{-1} \tag{4.1}$$

aus (3.102), und als Integrationsintervall $[0, t_{end}]$ in (3.2) betrachten wir nur das erste Teilintervall $[0, H]$ der Länge $H = m \cdot h$, d.h. wir beschränken uns auf den Fall $N = 1$. Damit ist bei festem Basisverfahren, festen Kollokationsknoten (2.12), festem Polynomgrad m und fester Schrittweite h, die Iterationsmatrix (3.42) der Interpolierten Defektkorrektur aus (3.42) noch von ε und ω abhängig, und wir schreiben

$$\boldsymbol{S}_{\mathrm{IIDeC}} \;=\; \boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega). \tag{4.2}$$

Wie in Abschnitt 3.5 betrachten wir das Basisverfahren SDIRK(2) und RadauIIA-Kollokationsabszissen mit $m = 6$. Die Abbildungen 4.1, 4.2 bzw. 4.3 zeigen für die Schrittweiten $h = 0.1$, $h = 0.05$ bzw. $h = 0.025$ jeweils die Konturlinien $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega)) = \rho_0,\ \rho_0 = 0.4, 0.6, 0.8, 1, 10, 10^2, 10^3, \ldots$, und zwar für $\omega \in [0, 1]$

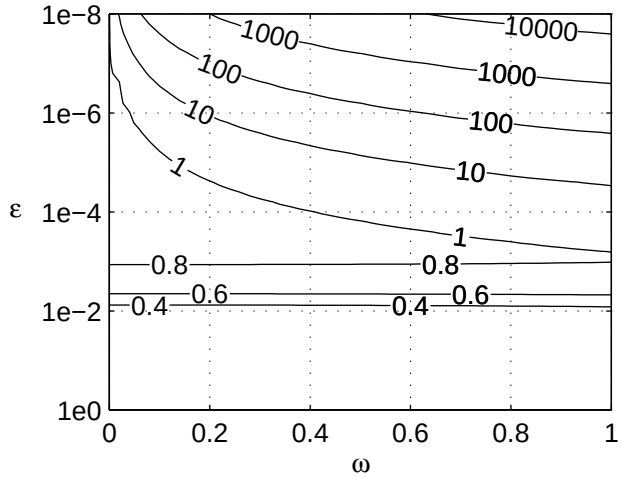

Abbildung 4.1: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega))$ bei Verwendung von SDIRK(2) mit Schrittweite $h = 0.1$ als Basisverfahren und RadauIIA(6)-Kollokationsabszissen.

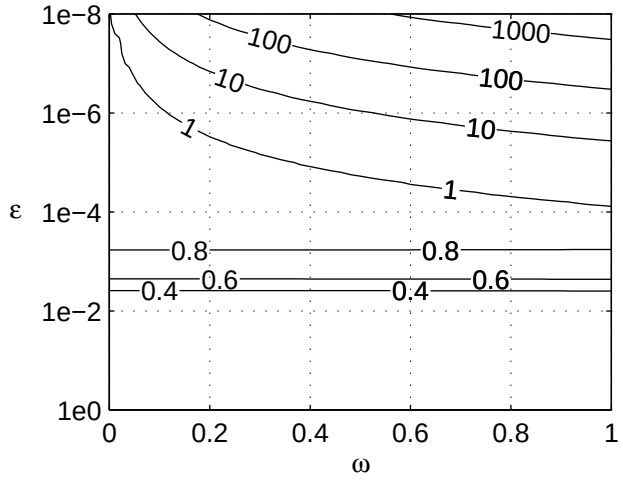

Abbildung 4.2: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega))$ bei Verwendung von SDIRK(2) mit Schrittweite $h = 0.05$ als Basisverfahren und RadauIIA(6)-Kollokationsabszissen.

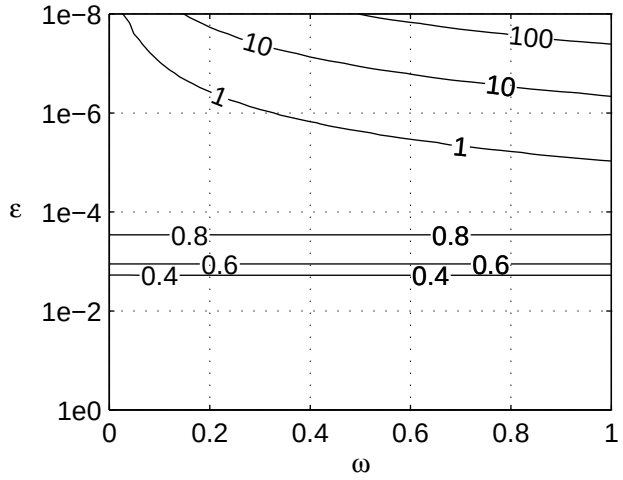

Abbildung 4.3: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega))$ bei Verwendung von SDIRK(2) mit Schrittweite $h = 0.025$ als Basisverfahren und RadauIIA(6)-Kollokationsabszissen.

und $\varepsilon \in [10^0, 10^{-8}]$, wobei die vertikale ε-Achse logarithmisch skaliert ist. Diese Kontur-Plots vermitteln einen guten Eindruck vom Verlauf des Spektralradius $\boldsymbol{S}_{\mathrm{IIDeC}}(\varepsilon, \omega)$ in Abhängigkeit von ε und ω. Außer für sehr kleines ω scheint hier bei festem ω, wie man den Abbildungen entnimmt, $\rho(S_{\mathrm{IIDeC}}(\varepsilon, \omega)) = O(\frac{1}{\varepsilon})$ zu gelten. Das hat für die entsprechenden steifen Probleme (3.1) mit $0 < \varepsilon \ll 1$ jenes instabile Verhalten der Interpolierten Defektkorrektur zur Folge, welches wir bereits in Tabelle 3.44 beobachten konnten.

4.2 Interpolierte Defektkorrektur mit transformierten Defekten (TIIDeC)

Wir betrachten lineare steife Anfangswertprobleme der Form (3.1), wobei die dortige Matrix $A(t)$ die Gestalt

$$A(t) \;=\; X(t) \cdot \Lambda(t) \cdot X(t)^{-1} \tag{4.3}$$

mit

$$X(t) \;=\; \begin{pmatrix} | & | & & | \\ x_1(t) & x_2(t) & \cdots & x_n(t) \\ | & | & & | \end{pmatrix} \tag{4.4}$$

und

$$\Lambda(t) \;=\; \mathrm{diag}\left(-\frac{c_1(t)}{\varepsilon}, c_2(t), \ldots, c_n(t)\right) \tag{4.5}$$

habe. Die Funktionen $x_i(t)$ und $c_i(t)$ seien hier hinreichend glatte Funktionen mit moderaten vom steifen Parameter $0 < \varepsilon \ll 1$ unabhängigen Ableitungen, und es gelte $c_1(t) \geq C_1$ mit einer Konstanten $C_1 > 0$.

4.2.1 Beschreibung des TIIDeC-Algorithmus

Da die Interpolierte Defektkorrektur mit RadauIIA-Kollokationsabszissen und z.B. SDIRK(2) als Basisverfahren sowohl im nichtsteifen Fall als auch im skalaren steifen Fall (vgl. Abschnitt 3.4.2) gut funktioniert, scheint die Vermutung plausibel, daß die erwähnte Instabilität der Interpolierten Defektkorrektur bei variierenden steifen Eigenrichtungen beseitigt werden könnte, wenn es gelingt, die Interpolierte Defektkorrektur in gewisser Weise separat auf die steife bzw. nichtsteife Komponente des betrachteten Problems anzuwenden. Eine Idee in dieser Richtung besteht darin, den jeweiligen n-dimensionalen Defekt $d^{[k]}(t)$ nicht komponentenweise bezüglich des Standardkoordinatensystems des $\mathbb{R}^n$ zu interpolieren, wie das bei der Interpolierten Defektkorrektur der Fall ist, sondern nun die steife bzw. nichtsteife Komponente des Defekts separat zu interpolieren. Dazu

muß der Defekt $d^{[k]}(t)$ in einem Koordinatensystem dargestellt werden, in dem der steife Eigenvektor $x_1(t)$ Basisvektor ist, was durch Multiplizieren des Defekts mit einer von t abhängigen Transformationsmatrix $Z(t)^{-1}$ gelingt, wobei die erste Spalte von $Z(t)$ ein Eigenvektor zum Eigenwert $-\frac{c_1(t)}{\varepsilon}$, d.h. ein Vielfaches von $x_1(t)$ ist. Dieser transformierte Defekt $Z(t)^{-1}d^{[k]}(t)$ wird nun genauso wie der gewöhnliche Defekt im Fall der IIDeC interpoliert, d.h. wir definieren wie in (2.37) eine stückweise Polynomfunktion $D^{[k]}(t)$ vom Grad $m-1$, für die

$$D^{[k]}(\tau_{\ell,\nu}) \;=\; Z(\tau_{\ell,\nu})^{-1} \cdot d^{[k]}(\tau_{\ell,\nu}), \qquad \tau_{\ell,\nu} \in \widehat{\Gamma}_h \tag{4.6}$$

gilt. Um nun zur Aufstellung des Nachbarproblems (2.33) eine geeignete Störung $\delta^{[k]}(t)$ zu definieren, muß dieser interpolierte transformierte Defekt $D^{[k]}(t)$, der der Darstellung einer Funktion im $\{z_1(t), z_2(t), \ldots, z_n(t)\}$-Koordinatensystem entspricht (die $z_i(t)$ sind die Spalten von $Z(t)$), im Standardkoordinatensystem des $\mathbb{R}^n$ dargestellt werden. Dies wird erreicht, indem wir $D^{[k]}(t)$ mit der Transformationsmatrix $Z(t)$ multiplizieren:

$$\delta^{[k]}(t) \;=\; Z(t) \cdot D^{[k]}(t). \tag{4.7}$$

Die Matrix $Z(t)^{-1}$ wird nur an den Punkten t des Gitters $\widehat{\Gamma}_h$ benötigt, und die Matrix $Z(t)$ bei der Lösung des Nachbarproblems (2.33) nur an den Punkten des Gitters $\widetilde{\Gamma}_h$. Wenn wir eine Vorschrift zur Berechnung der Matrix $Z(t)$ für $t \in \widehat{\Gamma}_h$ bzw. $t \in \widetilde{\Gamma}_h$ angeben, so ist dadurch ein Defektkorrekturalgorithmus der in Abschnitt 2.2.2 beschriebenen Bauart gegeben, unabhängig davon, ob die erste Spalte der jeweiligen Matrix $Z(t)$ ein Eigenvektor der Matrix $A(t)$ zum steifen Eigenwert $-\frac{c_1(t)}{\varepsilon}$ ist oder nicht. Wir bezeichnen einen solchen Algorithmus kurz als TIIDeC-Algorithmus. Beispielsweise erhält man wieder den originalen Interpolierten Defektkorrekturalgorithmus, wenn man $Z(t) = I_n$ ($I_n \ldots (n \times n)$-Einheitsmatrix) für $t \in \widehat{\Gamma}_h$ bzw. $t \in \widetilde{\Gamma}_h$ setzt.

4.2.2 Matrix $\boldsymbol{S}$ aus (3.6) im Fall von TIIDeC

Die Anwendung eines TIIDeC-Algorithmus auf ein lineares Anfangswertproblem (3.1) kann wieder als iterative Anwendung (3.3) eines affinen Operators $\eta_h \mapsto S_h\eta_h + v_h$ aufgefaßt werden. Die dem homogen linearen Operator S_h entsprechende Iterationsmatrix $\boldsymbol{S}$, die für die Interpolierte Defektkorrektur die Gestalt

$$\boldsymbol{S} \;=\; \boldsymbol{S}_{\text{IIDeC}} \;=\; I_{Nmn} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{V}} \cdot \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_1 \right) \tag{4.8}$$

hat (vgl. 3.42), hat nun, wie man sich leicht überlegt, die Gestalt

$$\boldsymbol{S} \;=\; \boldsymbol{S}_{\text{TIIDeC}} \;=\; I_{Nmn} - \boldsymbol{K} \cdot \widetilde{\boldsymbol{Z}}_1 \cdot \widetilde{\boldsymbol{V}} \cdot \widehat{\boldsymbol{Z}}_2 \cdot \left(\frac{1}{h} \cdot \widehat{\boldsymbol{W}}_1' - \widehat{\boldsymbol{A}} \cdot \widehat{\boldsymbol{W}}_1 \right), \tag{4.9}$$

wobei die Matrix $\widetilde{\boldsymbol{Z}}_1$ durch

$$\begin{aligned}\widetilde{\boldsymbol{Z}}_1 \;:=\; &\text{blockdiag}(Z(t_0+c_1h),\dots Z(t_0+c_sh),\dots \\ &\dots, Z(t_{Nm-1}+c_1h),\dots,Z(t_{Nm-1}+c_sh))\end{aligned} \tag{4.10}$$

und die Matrix $\widehat{\boldsymbol{Z}}_2$ durch

$$\widehat{\boldsymbol{Z}}_2 := \text{blockdiag}\big(Z(\tau_{0,1})^{-1},\dots,Z(\tau_{0,m})^{-1},\dots,Z(\tau_{N-1,1})^{-1},\dots,Z(\tau_{N-1,m})^{-1}\big) \tag{4.11}$$

gegeben ist.

4.2.3 Numerische Stabilität des TIIDeC-Algorithmus

Für Anfangswertprobleme (3.1), für die die Matrix $A(t)$ in faktorisierter Form (4.3) gegeben ist, können wir einfach $Z(t) \equiv X(t)$ setzen. Bei dieser Wahl von $Z(t)$ kann die Instabilität der Interpolierten Defektkorrektur bei Anwendung auf Probleme (3.1), bei denen die Matrix $A(t)$ eine variierende steife Eigenrichtung hat, tatsächlich beseitigt werden. Wir demonstrieren dies anhand der Matrix $A(t)$ aus (4.1). Wir verwenden wieder SDIRK(2) als Basisverfahren und RadauIIA-Kollokationsabszissen mit $m = 6$. Durch Festlegen der Transformationsmatrix

$$Z(t) \;:= X(t) \;=\; \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix} \tag{4.12}$$

ist damit ein TIIDeC-Algorithmus spezifiziert. Die Abbildung 4.4 zeigt Konturlinien für den Spektralradius der entsprechenden Iterationsmatrix $\boldsymbol{S}_{\text{TIIDeC}}(\varepsilon,\omega)$, wobei wir die Schrittweite $h = 0.05$ und das Integrationsintervall $[t_0, t_{end}] = [0, m\cdot h] = [0, 0.3]$ betrachten Diesen Kontur-Plot vergleichen wir mit dem entsprechenden Plot für den Fall der Interpolierten Defektkorrektur in Abbildung 4.2. Die Konturlinien für $\rho = 1, 10, 10^2, 10^3, \dots$ sind jetzt nicht mehr zu sehen, d.h. für alle hier betrachteten ε und ω ist der Spektralradius $\rho(\boldsymbol{S}_{\text{TIIDeC}}(\varepsilon,\omega))$ kleiner als 1, woraus die Fixpunktkonvergenz und numerische Stabilität des entsprechenden Defektkorrekturalgorithmus bei Anwendung auf ein lineares Problem mit Matrix (4.3) folgt.

4.2.4 Mit QR-Verfahren kombinierte Interpolierte Defektkorrektur (QR-IIDeC)

Wenn eine Faktorisierung (4.3) der Matrix $A(t)$ nicht zur Verfügung steht, so erhebt sich die Frage, wie die Transformationsmatrizen $Z(t)$ dennoch auf sinnvolle Weise definiert werden können, sodaß die numerische Stabilität des TIIDeC-Algorithmus erhalten bleibt, welche für den Fall $Z(t) \equiv X(t)$ gegeben ist. Folgende Eigenschaften von $Z(t)$ erscheinen wünschenswert, wobei die dritte Eigenschaft durch die Überlegungen aus Abschnitt 4.2.1 motiviert wird:

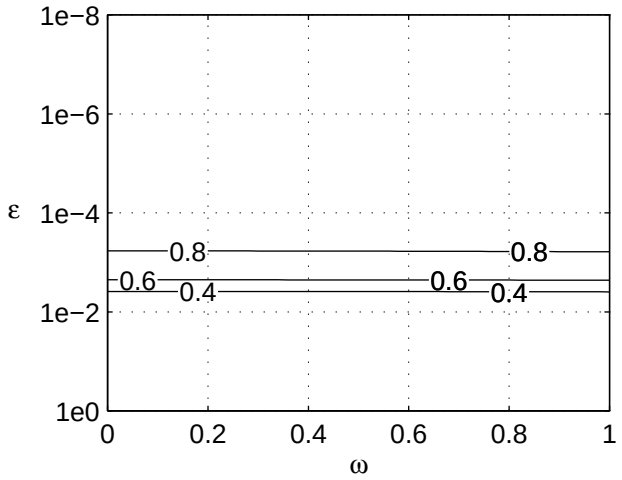

Abbildung 4.4: Spektralradius $\rho(\boldsymbol{S}_{\mathrm{TIIDeC}}(\varepsilon,\omega))$ bei Verwendung von SDIRK(2) mit Schrittweite $h = 0.05$ als Basisverfahren und RadauIIA(6)-Kollokationsabszissen.

1. Die Matrix $Z(t)$ bzw. die inverse Matrix $Z(t)^{-1}$ sollte mit möglichst wenig Aufwand berechnet werden.

2. Die Vorschrift zur Berechnung von $Z(t)$ aus den Komponenten der Matrix $A(t)$ an einer festen Stelle t sollte so beschaffen sein, daß die Einträge der Matrix $Z(t)$ stetig von den Einträgen der Matrix $A(t)$ abhängen. Dadurch wird sichergestellt, daß die Abbildung $t \mapsto Z(t)$ bzw. $t \mapsto Z(t)^{-1}$ stetig ist.

3. Die erste Spalte $z_1(t)$ von $Z(t)$ (oder eine andere feste Spalte von $Z(t)$) sollte hinreichend genau einen Eigenvektor von $A(t)$ zum steifen Eigenwert $-\frac{c_1(t)}{\varepsilon}$ approximieren.

Eine Möglichkeit zur Definition der Transformationsmatrizen $Z(t)$ ist, für $Z(t)$ jene orthogonale Matrix $Q(t)$ zu wählen, die durch eine QR-Zerlegung

$$A(t) = Q(t) \cdot R(t) \tag{4.13}$$

von $A(t)$ gegeben ist. Wir diskutieren nun die Frage, inwieweit mit dieser Wahl für $Z(t)$ die obigen drei Eigenschaften erfüllt sind:

ad 1: Im wesentlichen ist die QR-Zerlegung einer Matrix doppelt so aufwendig wie die entsprechende LU-Zerlegung ($2n^3/3+O(n^2)$ Rechenoperationen für die QR-Zerlegung im Vergleich zu $n^3/3+O(n^2)$ Rechenoperationen für die LU-Zerlegung), wobei wegen der Orthogonalität von $Q(t)$ für die Bildung

der inversen Matrix $Z(t)^{-1} = Q(t)^T$ kein zusätzlicher Rechenaufwand notwendig ist.

Bei Verwendung von SDIRK(2) (2.28) als Basisverfahren ist im linearen Fall (3.1) das Gleichungssystem (2.21) äquivalent zum folgenden linearen Gleichungssystem für die unbekannten Vektoren $Y_1, Y_2 \in \mathbb{R}^n$:

$$\begin{aligned} Y_1 &= \eta_\nu + h\gamma \cdot A(t_\nu + \gamma h) \cdot Y_1, \\ \eta_{\nu+1} = Y_2 &= \eta_\nu + h(1-\gamma) \cdot A(t_\nu + \gamma h) \cdot Y_1 + h\gamma \cdot A(t_\nu + h) \cdot Y_2 \end{aligned} \tag{4.14}$$

mit $\gamma = 1 - \sqrt{2}/2$. Durch einfache Umformungen ergibt sich, daß dieses lineare Gleichungssystem zu

$$\begin{aligned} \big[I_n - h\gamma \cdot A(t_\nu + \gamma h)\big] \cdot Y_1 &= \eta_\nu, \\ \big[I_n - h\gamma \cdot A(t_\nu + h)\big] \cdot Y_2 &= \frac{2\gamma - 1}{\gamma} \cdot Y_1 + \frac{1-\gamma}{\gamma} \cdot \eta_\nu \end{aligned} \tag{4.15}$$

äquivalent ist. Zur Bestimmung von Y_1 und Y_2 ist hier also die Auflösung zweier Gleichungssysteme erforderlich, deren Matrix in beiden Fällen die Gestalt $I_n - h\gamma A(t)$ hat, wobei die Stelle t ein Punkt des Gitters $\widetilde{\Gamma}_h$ ist. Die Transformationsmatrizen $Z(t)$ im TIIDeC-Algorithmus werden (unter anderem) ebenfalls an den Punkten des Gitters $\widetilde{\Gamma}_h$ benötigt, vgl. Abschnitt 4.2.1. Damit stellt sich die Frage, ob die Matrizen $Q(t)$ und $R(t)$ der QR-Zerlegung (4.13) von $A(t)$ für $t \in \widetilde{\Gamma}_h$ nicht auch zur Lösung der Gleichungen (4.15) verwendet werden können. Auf naheliegende Weise scheint das nicht möglich zu sein, man benötigt für die Lösung dieser Gleichungen eine Zerlegung der jeweiligen Matrizen $I_n - h\gamma A(t)$, die Zerlegung (4.13) der entsprechenden Matrizen $A(t)$ ist dazu nicht von großem Nutzen. Umgekehrt kann man sich fragen, ob es nicht auch möglich ist, für die Transformationsmatrizen $Z(t)$ des TIIDeC-Algorithmus statt der Matrizen $Q(t)$ aus (4.13) nun die orthogonale Matrix $Q(t)$ einer QR-Zerlegung

$$I_n - h\gamma A(t) = Q(t) \cdot R(t) \tag{4.16}$$

von $I_n - h\gamma A(t)$ zu betrachten, zumal ja in Hinblick auf Eigenschaft 3, die Matrizen $A(t)$ und $I_n - h\gamma A(t)$ die gleichen Eigenvektoren haben, wobei ein Eigenvektor von $A(t)$ zum Eigenwert λ ein Eigenvektor von $I_n - h\gamma A(t)$ zum Eigenwert $1 - h\gamma\lambda$ ist. Für $t \in \widetilde{\Gamma}_h$ wäre diese Zerlegung auch zur Lösung der Gleichungen (4.15) verwendbar.

ad 2: Im Algorithmus nach Householder, der üblicherweise für die QR-Zerlegung $A = Q \cdot R$ einer Matrix A angewendet wird, sind gewisse in den einzelnen Schritten des Algorithmus vorkommende Werte nur bis auf das Vorzeichen spezifiziert. Das hat zur Folge, daß die Orientierung der Spalten von Q und

jene der Zeilen von R nicht eindeutig bestimmt sind: Wenn $A = Q_0 \cdot R_0$ eine QR-Zerlegung von A ist, dann sind alle anderen QR-Zerlegungen von A durch

$$A = \underbrace{(Q_0 D)}_{Q} \cdot \underbrace{(D R_0)}_{R}, \qquad D = \mathrm{diag}(\pm 1, \ldots, \pm 1) \tag{4.17}$$

gegeben. Wenn wir die erwähnten Vorzeichen im Vorhinein festlegen würden, so wäre die damit definierte Matrix Q tatsächlich eindeutig bestimmt und stetig von den Einträgen der Matrix A abhängig. In der Praxis (d.h. in den Routinen der gängigen numerischen Programmbibliotheken, in denen die QR-Zerlegung implementiert ist) werden diese Vorzeichen aber nicht im Vorhinein festgelegt, sondern sie werden im Laufe des Algorithmus jeweils so gewählt, daß Auslöschungseffekte vermieden werden, und so die numerische Stabilität des Algorithmus gewahrt bleibt. Dadurch geht aber i.a. die stetige Abhängigkeit der Matrizen Q von den Einträgen der Matrix A verloren.

Die stetige Abhängigkeit der Transformationsmatrizen $Q(t)$ von t kann aber dennoch erreicht werden, und zwar z.B. dadurch, daß man in der QR-Zerlegung $A(t) = Q_0(t) \cdot R_0(t)$ an einer festen Stelle t, die mit Hilfe einer „Black-Box-Routine“ berechnet wird, im Nachhinein durch Multiplizieren mit einer geeigneten ± 1-Diagonalmatrix $D(t)$ wie in (4.17) die Orientierung der Spalten von $Q_0(t)$ (und entsprechend die Orientierung der Zeilen von $R_0(t)$) gemäß einer geeigneten Vorschrift neu festsetzt. Z.B. ist eine solche Vorschrift durch die Forderung gegeben, daß in jeder Spalte von $Q(t) = Q_0(t) \cdot D(t)$ das betragsgrößte Element ein positives Vorzeichen haben soll.

Eine andere Möglichkeit, die stetige Abhängigkeit der Matrix $Q(t)$ von den Einträgen der Matrix $A(t)$ zu erreichen, besteht darin, im Householder-Algorithmus auf die oben erwähnte Wahlmöglichkeit der Vorzeichen im Hinblick auf die numerische Stabilität zu verzichten, und diese Vorzeichen im Vorhinein festzulegen.[1] Diese Vorgangsweise ist nach unserer numerischen Erfahrung der oben beschriebenen nachträglichen Adaptierung der Orientierung der Spalten von $Q(t)$ vorzuziehen. Daher gehen wir auch in unseren numerischen Experimenten nach dieser Methode vor: Statt der in Matlab eingebauten Routine `qr.m` verwenden wir für die QR-Zerlegung unsere eigene Routine `qr1.m` aus Anhang A.1.8.

[1] Man kann dabei so vorgehen, daß die numerische Stabilität trotzdem nicht verloren geht, indem man die Auswirkungen, die die verschiedenen in Hinsicht auf die numerische Stabilität gewählten Vorzeichen auf die Orientierung der einzelnen Spalten von $Q_0(t)$ haben, im Nachhinein durch Multiplizieren mit der entsprechenden ± 1-Diagonalmatrix $D(t)$ wieder rückgängig macht, sodaß die so gewonnene Matrix $Q(t) = Q_0(t) \cdot D(t)$ die gleiche wie jene ist, welche man bei im Vorhinein fest gewählten Vorzeichen erhalten würde.

ad 3: Für Matrizen $A(t)$ der Gestalt (4.3) mit $0 < \varepsilon \ll 1$ ist die erste Spalte der zugehörigen orthogonalen Matrix $Q(t)$ tatsächlich eine ziemlich genaue Approximation eines Eigenvektor von $A(t)$ zum Eigenwert $-\frac{c_1(t)}{\varepsilon}$. Eine Erklärung dafür liefert die folgende kurze Diskussion des QR-Verfahrens zur Berechnung von Eigenvektoren, Details zum QR-Verfahren findet man in [5].

QR-Verfahren zur Bestimmung eines steifen Eigenvektors: Gegeben sei eine Matrix A der Gestalt

$$A = X \cdot \Lambda \cdot X^{-1} \tag{4.18}$$

mit

$$X = \begin{pmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{pmatrix} \tag{4.19}$$

und

$$\Lambda = \operatorname{diag}\left(-\frac{c}{\varepsilon}, \lambda_2, \ldots, \lambda_n\right), \tag{4.20}$$

wobei $0 < \varepsilon \ll 1,\ c > 0$ gilt, und ohne Beschränkung der Allgemeinheit, die Eigenwerte von A dem Betrag nach geordnet sind:

$$\frac{c}{\varepsilon} > |\lambda_2| > \ldots > |\lambda_n|. \tag{4.21}$$

Damit definieren wir eine Folge $\{A_k\}$ von Matrizen durch

$$A_1 := A, \tag{4.22}$$

$$A_{k+1} := R_k \cdot Q_k, \quad k = 1, 2, 3, \ldots, \tag{4.23}$$

wobei die orthogonale Matrix Q_k und die obere Dreiecksmatrix R_k jeweils durch die QR-Zerlegung von A_k,

$$A_k = Q_k \cdot R_k, \tag{4.24}$$

konstruiert werden. Weiters definieren wir eine Folge $\{\bar{Q}_k\}$ orthogonaler Matrizen durch Aufmultiplizieren der Q_k:

$$\bar{Q}_k := Q_1 \cdot Q_2 \cdots Q_k, \quad k = 1, 2, 3, \ldots. \tag{4.25}$$

Unter bestimmten Voraussetzungen konvergiert die Folge $\{A_k\}$ gegen eine obere Dreiecksmatrix R und die Folge $\{\bar{Q}_k\}$ gegen eine orthogonale Matrix Q, durch die eine Schur-Zerlegung

$$\begin{aligned} A &= \lim_{k\to\infty} \bar{Q}_k \cdot A_k \cdot \bar{Q}_k^T = Q \cdot R \cdot Q^T \\ &= \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} -\frac{c}{\varepsilon} & & & * \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \cdot \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{pmatrix}^T \end{aligned} \tag{4.26}$$

der Matrix A realisiert wird, bei der die Diagonalelemente von R, d.h. die Eigenwerte von A dem Betrag nach geordnet sind, vgl. (4.21). Die Subdiagonalelemente in der ersten Spalte von A_k konvergieren dabei wie $\left|\frac{\lambda_2}{c/\varepsilon}\right|^k$ also wie ε^k gegen 0, und entsprechend schnell konvergieren die ersten Spalten der Matrizen $\bar{Q}_k$ gegen die erste Spalte q_1 von Q, und damit gegen einen Eigenvektor von A zum steifen Eigenwert $-\frac{c}{\varepsilon}$.

Somit läßt sich ein solcher steifer Eigenvektor mit Hilfe des QR-Verfahrens auf besonders effiziente Weise berechnen. Oft genügt ein einziger Schritt des QR-Verfahrens zur hinreichend genauen Bestimmung dieses Eigenvektors. d.h. wenn die orthogonale Matrix Q durch eine QR-Zerlegung

$$A = Q \cdot R \tag{4.27}$$

von A definiert ist, so ist die erste Spalte von Q mit großer Genauigkeit ein Eigenvektor der Matrix A zum steifen Eigenwert $-\frac{c}{\varepsilon}$. Dadurch ist eine heuristische Begründung dafür gegeben, daß die Wahl $Z(t) = Q(t)$ mit $Q(t)$ aus (4.13) in Hinblick auf die Eigenschaft 3 zielführend sein könnte.

Diese Begründung kann auch für die Wahl $Z(t) = Q(t)$ mit $Q(t)$ aus (4.16) gegeben werden: Dazu setzen wir für den Startwert des QR-Verfahrens statt (4.22) nun

$$A_1 \ := I_n - \gamma h \cdot A. \tag{4.28}$$

Damit definieren wir mit Hilfe des durch (4.23), (4.24) und (4.25) gegebenen QR-Verfahrens wieder eine Folge $\{\bar{Q}_k\}$ orthogonaler Matrizen. Durch den Grenzwert in (4.26) ist damit jetzt eine Schur-Zerlegung der Matrix $I_n - \gamma h \cdot A$ gegeben,

$$\begin{aligned} A \ &= \ \lim_{k\to\infty} \bar{Q}_k \cdot A_k \cdot \bar{Q}_k^T \ = \ Q \cdot R \cdot Q^T \\ &= \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} 1+\frac{\gamma hc}{\varepsilon} & & & * \\ & * & & \\ & & \ddots & \\ & & & * \end{pmatrix} \cdot \begin{pmatrix} | & | & & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & & | \end{pmatrix}^T, \end{aligned} \tag{4.29}$$

wobei die Subdiagonalelemente in der ersten Spalte von A_k wie

$$\left|\frac{1-\gamma h\lambda_i}{1-(\gamma hc)/\varepsilon}\right|^k \quad \text{(mit jenem } i \in \{2,\ldots,n\}\text{, für das } |1-\gamma h\lambda_i| \text{ maximal ist)},$$

d.h. für $0 < \varepsilon \ll 1$ und kleines $h > 0$ im wesentlichen wie $|\varepsilon/h|^k$ gegen Null konvergieren. Für ε hinreichend klein im Vergleich zur Schrittweite h bedeutet das die rasche Konvergenz der ersten Spalten der Matrizen Q_k gegen die erste Spalte q_1 von Q, d.h. gegen einen Eigenvektor von $I_n - \gamma h \cdot A$ zum Eigenwert $1 + \frac{\gamma hc}{\varepsilon}$ und damit gegen einen Eigenvektor von A zum steifen Eigenwert $-\frac{c}{\varepsilon}$. Somit scheint in Hinblick auf Eigenschaft 3 auch die Wahl $Z(t) = Q(t)$ mit $Q(t)$ aus (4.16) vernünftig zu sein.

4.3 Numerische Experimente (3): Vergleich der verschiedenen Defektkorrekturalgorithmen bei Anwendung auf ein Problem mit variierender Eigenrichtung

In diesem Abschnitt vergleichen wir anhand des Problems (3.99) mit $A(t)$ aus (3.102) die folgenden Varianten der Defektkorrektur:

IDeC: Klassische Defektkorrektur mit SDIRK(2) als Basisverfahren und $m = 6$ als Grad der Interpolationspolynome.

IIDeC: Interpolierte Defektkorrektur mit SDIRK(2) als Basisverfahren und RadauIIA(6)-Kollokationsabszissen.

TIIDeC: Interpolierte Defektkorrektur (Basisverfahren: SDIRK(2), Kollokationsabszissen: RadauII(2)) mit transformiertem Defekt, wobei für die Transformationsmatrix $Z(t)$ in (4.6) und (4.7) die Matrix (4.12) bestehend aus den exakten Eigenrichtungen der Matrix $A(t)$ aus (3.102) verwendet wird.

QR-IIDeC (1): Interpolierte Defektkorrektur (Basisverfahren: SDIRK(2), Kollokationsabszissen: RadauII(2)) mit transformiertem Defekt, wobei für die Transformationsmatrix $Z(t)$ in (4.6) und (4.7) die Matrix $Q(t)$ einer QR-Zerlegung $A(t) = Q(t) \cdot R(t)$ von $A(t)$ verwendet wird.

QR-IIDeC (2): Interpolierte Defektkorrektur (Basisverfahren: SDIRK(2), Kollokationsabszissen: RadauII(2)) mit transformiertem Defekt, wobei für die Transformationsmatrix $Z(t)$ in (4.6) und (4.7) die Matrix $Q(t)$ einer QR-Zerlegung $I_2 - \gamma h A(t) = Q(t) \cdot R(t)$ von $I_2 - \gamma h A(t)$, $\gamma = 1 - \sqrt{2}/2$, verwendet wird.

Im Gegensatz zu den Experimenten in den Abschnitten 3.4 und 3.5, die mit Hilfe eines in C++ geschriebenen Computerprogramms unter Verwendung einer Gleitpunktarithmethik mit erweiterter Genauigkeit durchgeführt wurden (vgl. Abschnitt 3.3.3), wurden für die Experimente in diesem Abschnitt die Matlab-Programme aus Anhang A.1 verwendet. Dadurch wird es möglich, auch die mit der QR-IIDeC durchgeführten Experimente zu wiederholen und so die hier angegebenen Resultate nachzuprüfen, wobei es, wie wir gesehen haben, besonders auf die spezielle Implementierung der QR-Zerlegung ankommt, die in unserem Fall durch die Matlab-Funktion `qr1.m` aus Abschnitt A.1.8 gegeben ist. Zur Angabe der entsprechenden Matlab-Programme sind nur relativ wenig Seiten erforderlich, wogegen die Angabe der jeweiligen in C++ geschriebenen Programme jeden vernünftigen Rahmen sprengen würde.

Für die Tabellen 4.1 bis 4.4 setzen wir für die Drehgeschwindigkeit $\omega = 0.2$, und als Integrationsintervall betrachten wir $[t_0, t_{end}] = [0, 3.6]$. Wie man der Tabelle 4.1 entnimmt, arbeiten alle hier betrachteten Varianten der Interpolierten Defektkorrektur im nichtsteifen Fall $\varepsilon = 1$ einwandfrei, die nach 5 Defektkorrekturschritten erzielbare Genauigkeit ist in allen Fällen in etwa die gleiche, und zwar wesentlich besser als im Fall der klassischen Defektkorrektur und für die größeren Schrittweiten mit der Genauigkeit des Fixpunktes RadauIIA(6) vergleichbar. Für die kleineren Schrittweiten erreicht man nach 3 Defektkorrekturschritten eine Genauigkeit, die schon nahe an das Niveau der Rechengenauigkeit (ca. 15 Dezimalstellen) heranreicht, wodurch sich Rundungsfehler bemerkbar machen, und so durch weitere Defektkorrekturschritte die Genauigkeit nicht mehr gesteigert werden kann.

Für $\varepsilon = 10^{-4}$ (vgl. Tabelle 4.2) ist im Fall der IIDeC eine Instabilität noch nicht bemerkbar, die Konvergenz gegen den Fixpunkt erfolgt aber mit einer deutlich reduzierten Geschwindigkeit, nach 6 Defektkorrekturschritten kann noch nicht einmal das Genauigkeitsniveau erreicht werden, welches mit der klasischen IDeC schon nach 2 Defektkorrekturschritten erzielt wird. Durch das Transformieren des Defekts kann diese Situation wesentlich verbessert werden: Sowohl mit der TIIDeC als auch mit den beiden Varianten der QR-IIDeC erzielt man nach 6 Defektkorrekturschritten eine Genauigkeit, die schon nahe an das Genauigkeitsniveau des Fixpunktes heranreicht, wobei die größten Genauigkeitssteigerungen hier bereits in den ersten beiden Defektkorrekturschritten erfolgen. Interessanterweise erzielt man hier mit der zweiten Variante der QR-IIDeC (QR-Zerlegung der Matrix $I_2 - \gamma h A(t)$) etwas bessere Resultate als mit der ersten (QR-Zerlegung der Matrix $A(t)$).

In den Tabellen 4.3 bzw. 4.4 wird nun bei gleichbleibender Drehgeschwindigkeit $\omega = 0.2$ die Steifheit des betrachteten Problems weiter erhöht, wir setzen $\varepsilon = 10^{-6}$ bzw. $\varepsilon = 10^{-8}$. Im Fall der IIDeC macht sich jetzt die schon in Abschnitt 3.5 erwähnte Instabilität bei variierenden Eigenrichtungen bemerkbar, für kleinere Schrittweiten beobachtet man ein besonders stark ausgeprägtes divergentes Verhalten. Durch das Transformieren des Defekts kann diese Instabilität vermieden werden, die erzielte Genauigkeit ist zumindest für die kleineren Schrittweiten wesentlich höher als im Fall der klassischen Defektkorrektur, allerdings ist man nach 6 Defektkorrekturschritten noch weit vom Genauigkeitsniveau des Fixpunktes RadauIIA(6) entfernt.

In Tabelle 4.5, in der $\varepsilon = 10^{-6}$ gesetzt und die Drehgeschwindigkeit ω auf $\omega = 0.5$ erhöht wurde, beobachtet man für die beiden Varianten der QR-IIDeC ein instabiles Verhalten. Die Ursache dafür ist aber nicht die schnellere Drehung der Eigenrichtungen von $A(t)$, sondern die Tatsache, daß hier die Matrix $A(t)$ aus (3.102) in einer Umgebung von $t = \pi \in [t, t_{end}] = [0, 3.6]$ eine starke Unglattheit aufweist, wie man sofort erkennt, wenn man z.B. $A(t)$ für die drei benachbarten

Werte $t = \pi - 0.05$, $t = \pi$ und $t = \pi + 0.05$ berechnet:

$$\begin{aligned} A(\pi - 0.05) &\doteq \begin{pmatrix} -6.259 \cdot 10^2 & 2.499 \cdot 10^4 \\ 2.499 \cdot 10^4 & -9.993 \cdot 10^5 \end{pmatrix}, \\ A(\pi) &= \begin{pmatrix} -1 & 0 \\ 0 & -10^{-6} \end{pmatrix}, \\ A(\pi + 0.05) &\doteq \begin{pmatrix} -6.259 \cdot 10^2 & -2.499 \cdot 10^4 \\ -2.499 \cdot 10^4 & -9.993 \cdot 10^5 \end{pmatrix}. \end{aligned}$$

Tatsächlich ist diese Instabilität der QR-IIDeC nicht mehr beobachtbar, wenn man wie in Tabelle 4.6 das Integrationsintervall auf $[t_0, t_{end}] = [0, 2.4]$ verkleinert, sodaß der Punkt $t = \pi$ in diesem Intervall nicht mehr enthalten ist. Mit beiden Varianten der QR-IIDeC kann nun eine wesentlich höhere Genauigkeit als mit der klassischen IDeC erzielt werden, allerdings kann für die größeren Schrittweiten nach 6 Defektkorrekturschritten die Genauigkeit des Fixpunktes nicht erreicht werden. Für die kleineren Schrittweiten dringt man, was die Genauigkeit betrifft, bis in einen Bereich vor, in dem sich schon Rundungsfehler bemerkbar machen.

In Tabelle 4.7 wird schließlich die Drehgeschwindigkeit der Eigenrichtungen von $A(t)$ auf $\omega = 1$ erhöht, wobei gleichzeitig das Integrationsintervall nochmals auf $[t_0, t_{end}] = [0, 1.2]$ verkleinert wird. Dadurch wird vermieden, daß die Stelle $t = \pi/2$, an der jetzt die Matrix $A(t)$ aus (3.102) die oben beschriebene Unglattheit aufweisen würde, in diesem Intervall enthalten ist. Die erzielten Resultate sind mit denen aus Tabelle 4.6 vergleichbar: Wieder sind die mit den beiden Varianten der QR-IIDeC gewonnenen Resultate besser, als jene, die mit der klassischen Defektkorrektur berechnet wurden, wobei aber auch hier die Genauigkeit des Fixpunktes RadauIIA(6) nach 6 Defektkorrekturschritten nicht erreicht werden kann.

4.3.1 Zusammenfassung

Die numerischen Experimente in diesem Abschnitt deuten darauf hin, daß die Instabilität der IIDeC (mit SDIRK(2) als Basisverfahren und RadauIIA-Kollokationsabszissen) im Fall einer variierenden steifen Eigenrichtung mit Hilfe der beiden Varianten der QR-IIDeC tatsächlich vermieden werden können. Zwar kann damit im allgemeinen innerhalb weniger Defektkorrekturschritte die Genauigkeit des Fixpunktes, d.h. des entsprechenden RadauIIA-Kollokationsverfahrens nicht erreicht werden, aber die erzielbare Genauigkeit ist immer noch höher als im Fall der klassischen IDeC. Zu beurteilen, inwieweit das den höheren Rechenaufwand der QR-IIDeC rechtfertigt, der im wesentliche dadurch gegeben ist, daß jetzt QR-Zerlegungen anstelle von LU-Zerlegungen der jeweiligen Matrizen $I_n - \gamma h A(t)$ notwendig sind, dazu reicht unsere numerische Erfahrung zum gegenwärtigen Zeitpunkt noch nicht aus, weitere Untersuchungen sind dazu erforderlich.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	9.64E−04	5.91E−07	2.02E−07	2.02E−07	2.02E−07	2.02E−07	2.02E−07
0.1	2.36E−04	4.53E−08	2.27E−09	2.27E−09	2.27E−09	2.27E−09	2.27E−09
0.05	5.85E−05	2.88E−09	2.93E−11	2.93E−11	2.93E−11	2.93E−11	2.93E−11
0.025	1.45E−05	1.80E−10	4.06E−13	4.04E−13	4.06E−13	4.04E−13	4.03E−13

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	9.64E−04	6.56E−07	6.84E−09	3.79E−10	2.38E−11	3.73E−11	3.65E−11	3.65E−11
0.1	2.36E−04	4.58E−08	5.94E−11	1.46E−12	2.56E−14	1.60E−14	1.60E−14	1.76E−14
0.05	5.85E−05	2.89E−09	6.39E−13	2.11E−15	5.24E−15	7.85E−15	4.93E−15	4.97E−16
0.025	1.45E−05	1.80E−10	1.86E−14	1.15E−14	1.24E−14	1.58E−14	1.41E−14	8.01E−16

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	9.64E−04	6.56E−07	1.03E−08	8.71E−10	1.09E−10	3.48E−11	3.62E−11	3.65E−11
0.1	2.36E−04	4.58E−08	7.25E−11	3.41E−12	1.56E−13	1.37E−14	1.61E−14	1.76E−14
0.05	5.85E−05	2.89E−09	6.53E−13	1.53E−14	8.14E−15	5.50E−15	1.14E−14	4.97E−16
0.025	1.45E−05	1.80E−10	1.88E−14	1.56E−14	1.31E−14	1.43E−14	1.26E−14	8.01E−16

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	9.64E−04	6.56E−07	6.84E−09	3.79E−10	2.38E−11	3.73E−11	3.65E−11	3.65E−11
0.1	2.36E−04	4.58E−08	5.94E−11	1.46E−12	2.56E−14	1.60E−14	1.60E−14	1.76E−14
0.05	5.85E−05	2.89E−09	6.39E−13	2.11E−15	5.24E−15	7.85E−15	4.93E−15	4.97E−16
0.025	1.45E−05	1.80E−10	1.86E−14	1.15E−14	1.24E−14	1.58E−14	1.41E−14	8.01E−16

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	9.64E−04	6.56E−07	6.84E−09	3.79E−10	2.38E−11	3.73E−11	3.65E−11	3.65E−11
0.1	2.36E−04	4.58E−08	5.94E−11	1.46E−12	2.56E−14	1.60E−14	1.60E−14	1.76E−14
0.05	5.85E−05	2.89E−09	6.39E−13	2.11E−15	5.24E−15	7.85E−15	4.93E−15	4.97E−16
0.025	1.45E−05	1.80E−10	1.86E−14	1.15E−14	1.24E−14	1.58E−14	1.41E−14	8.01E−16

Tabelle 4.1: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 1$, $\omega = 0.2$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.31E−04	7.65E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07
0.1	2.26E−05	5.56E−08	2.10E−09	2.09E−09	2.09E−09	2.09E−09	2.09E−09
0.05	4.29E−06	3.50E−09	2.65E−11	2.65E−11	2.65E−11	2.65E−11	2.65E−11
0.025	8.72E−07	2.18E−10	3.57E−13	3.62E−13	3.61E−13	3.64E−13	3.62E−13

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.31E−04	4.86E−07	4.86E−07	3.34E−07	3.95E−07	9.66E−07	9.16E−07	5.85E−10
0.1	2.26E−05	5.42E−08	6.07E−09	2.18E−09	4.67E−10	1.54E−09	9.87E−10	1.06E−11
0.05	4.29E−06	3.47E−09	1.87E−10	4.15E−11	2.45E−11	2.45E−11	2.05E−11	1.73E−13
0.025	8.72E−07	2.16E−10	9.68E−12	1.93E−12	1.30E−12	1.12E−12	9.42E−13	4.08E−15

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.31E−04	8.28E−07	3.72E−09	2.47E−09	1.87E−09	1.65E−09	1.45E−09	5.85E−10
0.1	2.26E−05	5.61E−08	1.75E−11	3.05E−11	2.56E−11	2.25E−11	1.99E−11	1.06E−11
0.05	4.29E−06	3.51E−09	1.16E−12	1.37E−12	1.16E−12	9.88E−13	8.43E−13	1.73E−13
0.025	8.72E−07	2.18E−10	1.66E−13	1.36E−13	1.09E−13	9.28E−14	7.58E−14	4.08E−15

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.31E−04	8.28E−07	3.58E−09	2.42E−09	1.82E−09	1.61E−09	1.41E−09	5.85E−10
0.1	2.26E−05	5.61E−08	1.68E−11	3.03E−11	2.54E−11	2.23E−11	1.98E−11	1.06E−11
0.05	4.29E−06	3.51E−09	1.18E−12	1.36E−12	1.16E−12	9.86E−13	8.38E−13	1.73E−13
0.025	8.72E−07	2.18E−10	1.63E−13	1.37E−13	1.12E−13	9.46E−14	7.52E−14	4.08E−15

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.31E−04	8.27E−07	1.28E−09	1.69E−09	1.08E−09	9.97E−10	9.03E−10	5.85E−10
0.1	2.26E−05	5.61E−08	1.63E−11	2.24E−11	1.91E−11	1.72E−11	1.56E−11	1.06E−11
0.05	4.29E−06	3.51E−09	2.49E−12	1.07E−12	9.49E−13	8.08E−13	6.89E−13	1.73E−13
0.025	8.72E−07	2.18E−10	2.90E−13	1.05E−13	9.14E−14	7.29E−14	5.95E−14	4.08E−15

Tabelle 4.2: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-4}$, $\omega = 0.2$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.33E−04	7.65E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07
0.1	2.32E−05	5.56E−08	2.10E−09	2.09E−09	2.09E−09	2.09E−09	2.09E−09
0.05	4.59E−06	3.51E−09	2.62E−11	2.61E−11	2.61E−11	2.61E−11	2.61E−11
0.025	9.95E−07	2.18E−10	5.32E−13	5.31E−13	5.31E−13	5.32E−13	5.33E−13

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	2.90E−05	2.20E−03	1.71E−01	1.33E+01	1.03E+03	7.98E+04	4.14E−11
0.1	2.32E−05	2.67E−08	5.61E−07	6.56E−06	7.27E−05	8.09E−04	8.99E−03	1.08E−13
0.05	4.59E−06	3.26E−09	8.49E−10	1.06E−09	2.38E−09	5.11E−09	1.02E−08	2.36E−15
0.025	9.95E−07	2.16E−10	1.37E−11	4.84E−12	1.20E−12	3.31E−12	9.05E−13	1.26E−15

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.28E−07	3.83E−09	2.32E−09	1.73E−09	1.50E−09	1.30E−09	4.14E−11
0.1	2.32E−05	5.61E−08	2.79E−11	1.95E−11	1.57E−11	1.36E−11	1.17E−11	1.08E−13
0.05	4.59E−06	3.51E−09	7.00E−13	2.31E−13	2.54E−13	2.78E−13	2.93E−13	2.36E−15
0.025	9.95E−07	2.18E−10	1.52E−13	1.62E−13	1.60E−13	1.55E−13	1.59E−13	1.26E−15

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.28E−07	3.70E−09	2.28E−09	1.68E−09	1.47E−09	1.27E−09	4.14E−11
0.1	2.32E−05	5.61E−08	2.74E−11	1.93E−11	1.55E−11	1.35E−11	1.16E−11	1.08E−13
0.05	4.59E−06	3.51E−09	6.97E−13	2.29E−13	2.57E−13	2.77E−13	2.93E−13	2.36E−15
0.025	9.95E−07	2.18E−10	1.54E−13	1.60E−13	1.59E−13	1.59E−13	1.57E−13	1.26E−15

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.27E−07	1.51E−09	1.57E−09	9.08E−10	8.17E−10	7.06E−10	4.14E−11
0.1	2.32E−05	5.61E−08	1.14E−11	1.38E−11	1.06E−11	9.21E−12	7.98E−12	1.08E−13
0.05	4.59E−06	3.51E−09	5.68E−13	2.72E−13	2.93E−13	3.07E−13	3.18E−13	2.36E−15
0.025	9.95E−07	2.18E−10	1.57E−13	1.60E−13	1.59E−13	1.61E−13	1.63E−13	1.26E−15

Tabelle 4.3: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-6}$, $\omega = 0.2$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.33E−04	7.65E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07	1.94E−07
0.1	2.33E−05	5.56E−08	2.04E−09	2.04E−09	2.04E−09	2.04E−09	2.04E−09
0.05	4.59E−06	3.53E−09	1.35E−12	1.32E−12	1.27E−12	1.32E−12	1.30E−12
0.025	9.97E−07	2.27E−10	9.08E−12	9.09E−12	9.10E−12	9.08E−12	9.09E−12

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	2.97E−03	2.26E+01	1.73E+05	1.32E+09	1.00E+13	7.66E+16	4.10E−11
0.1	2.33E−05	8.10E−06	8.09E−03	8.06E+00	8.03E+03	8.01E+06	7.98E+09	1.95E−14
0.05	4.59E−06	1.80E−08	2.63E−06	3.38E−04	4.34E−02	5.56E+00	7.13E+02	7.02E−16
0.025	9.97E−07	1.67E−10	5.97E−10	1.05E−08	1.81E−07	3.12E−06	5.37E−05	1.29E−15

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.28E−07	3.95E−09	2.20E−09	1.61E−09	1.38E−09	1.18E−09	4.10E−11
0.1	2.33E−05	5.62E−08	8.31E−11	3.56E−11	3.95E−11	4.13E−11	4.31E−11	1.95E−14
0.05	4.59E−06	3.53E−09	2.54E−11	2.51E−11	2.50E−11	2.51E−11	2.50E−11	7.02E−16
0.025	9.97E−07	2.27E−10	9.48E−12	9.47E−12	9.46E−12	9.46E−12	9.46E−12	1.29E−15

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.28E−07	3.82E−09	2.16E−09	1.56E−09	1.34E−09	1.15E−09	4.10E−11
0.1	2.33E−05	5.62E−08	8.26E−11	3.56E−11	3.94E−11	4.14E−11	4.32E−11	1.95E−14
0.05	4.59E−06	3.53E−09	2.55E−11	2.50E−11	2.50E−11	2.51E−11	2.51E−11	7.02E−16
0.025	9.97E−07	2.27E−10	9.48E−12	9.46E−12	9.47E−12	9.47E−12	9.47E−12	1.29E−15

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.33E−04	8.27E−07	1.64E−09	1.45E−09	7.86E−10	6.96E−10	5.83E−10	4.10E−11
0.1	2.33E−05	5.62E−08	6.65E−11	4.11E−11	4.43E−11	4.60E−11	4.72E−11	1.95E−14
0.05	4.59E−06	3.53E−09	2.53E−11	2.50E−11	2.51E−11	2.51E−11	2.51E−11	7.02E−16
0.025	9.97E−07	2.27E−10	9.47E−12	9.47E−12	9.45E−12	9.46E−12	9.46E−12	1.29E−15

Tabelle 4.4: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-8}$, $\omega = 0.2$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.97E−03	5.50E−07	2.09E−07	2.11E−07	2.11E−07	2.11E−07	2.11E−07
0.1	4.81E−04	5.01E−08	2.58E−09	2.61E−09	2.61E−09	2.61E−09	2.61E−09
0.05	1.19E−04	3.40E−09	3.44E−11	3.48E−11	3.48E−11	3.48E−11	3.48E−11
0.025	2.94E−05	2.19E−10	4.73E−13	4.77E−13	4.80E−13	4.82E−13	4.82E−13

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.97E−03	1.22E−04	6.07E−02	2.96E+01	1.45E+04	7.07E+06	3.45E+09	3.10E−11
0.1	4.81E−04	2.28E−07	1.07E−05	6.80E−04	4.34E−02	2.77E+00	1.77E+02	2.06E−14
0.05	1.19E−04	3.67E−09	8.86E−10	9.26E−09	8.29E−08	7.54E−07	6.84E−06	6.28E−16
0.025	2.94E−05	2.17E−10	1.14E−11	1.06E−11	2.36E−11	3.93E−11	6.88E−11	4.97E−16

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.97E−03	6.13E−07	7.41E−09	1.76E−09	1.30E−09	1.13E−09	9.72E−10	3.10E−11
0.1	4.81E−04	5.07E−08	9.56E−11	2.98E−12	1.34E−12	1.22E−12	1.08E−12	2.06E−14
0.05	1.19E−04	3.41E−09	1.17E−12	3.87E−14	3.51E−14	3.80E−14	4.23E−14	6.28E−16
0.025	2.94E−05	2.19E−10	3.71E−14	1.67E−14	2.00E−14	2.11E−14	2.27E−14	4.97E−16

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.97E−03	8.79E−05	6.33E−05	1.17E−04	1.42E−04	2.12E−04	2.75E−04	3.10E−11
0.1	4.81E−04	7.29E−06	5.08E−06	2.33E−06	1.15E−06	5.94E−07	3.57E−07	2.06E−14
0.05	1.19E−04	3.26E−06	1.08E−06	2.35E−06	1.63E−06	3.52E−07	4.52E−07	6.28E−16
0.025	2.94E−05	5.72E−08	7.64E−08	4.87E−08	2.74E−08	1.49E−08	8.02E−09	4.97E−16

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.97E−03	8.93E−05	1.23E−04	4.97E−04	1.23E−03	3.71E−03	1.03E−02	3.10E−11
0.1	4.81E−04	4.45E−06	1.38E−05	1.84E−06	2.17E−05	1.73E−05	4.78E−05	2.06E−14
0.05	1.19E−04	2.45E−07	5.08E−06	9.01E−06	1.79E−05	3.96E−05	9.11E−05	6.28E−16
0.025	2.94E−05	8.66E−07	2.57E−06	8.64E−06	2.81E−05	9.22E−05	3.02E−04	4.97E−16

Tabelle 4.5: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-6}$, $\omega = 0.5$: Globaler Fehler an der Stelle $t_{end} = 3.6$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	1.95E−03	4.76E−07	1.09E−07	1.11E−07	1.11E−07	1.11E−07	1.11E−07
0.1	4.82E−04	1.70E−08	1.59E−09	1.62E−09	1.62E−09	1.62E−09	1.62E−09
0.05	1.20E−04	7.64E−10	2.29E−11	2.34E−11	2.34E−11	2.34E−11	2.34E−11
0.025	2.99E−05	3.98E−11	2.91E−13	2.95E−13	2.95E−13	2.93E−13	3.00E−13

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.95E−03	1.80E−04	8.87E−02	4.33E+01	2.12E+04	1.03E+07	5.05E+09	5.48E−11
0.1	4.82E−04	3.22E−07	2.02E−05	1.29E−03	8.22E−02	5.25E+00	3.35E+02	6.29E−14
0.05	1.20E−04	2.19E−10	1.85E−09	1.92E−08	1.73E−07	1.57E−06	1.42E−05	8.88E−16
0.025	2.99E−05	4.71E−11	4.02E−11	3.77E−11	8.41E−11	1.39E−10	2.44E−10	0.00E+00

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.95E−03	4.49E−07	1.01E−08	3.34E−09	3.02E−09	2.62E−09	2.27E−09	5.48E−11
0.1	4.82E−04	1.67E−08	1.19E−10	3.64E−11	3.21E−11	2.75E−11	2.36E−11	6.29E−14
0.05	1.20E−04	7.60E−10	1.16E−12	1.18E−13	8.33E−14	4.60E−14	1.28E−14	8.88E−16
0.025	2.99E−05	3.98E−11	6.19E−14	4.61E−14	5.00E−14	5.24E−14	5.13E−14	0.00E+00

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.95E−03	4.48E−07	1.15E−08	2.93E−09	2.97E−09	2.55E−09	2.21E−09	5.48E−11
0.1	4.82E−04	1.67E−08	1.23E−10	3.53E−11	3.17E−11	2.72E−11	2.33E−11	6.29E−14
0.05	1.20E−04	7.60E−10	1.17E−12	1.17E−13	7.65E−14	4.31E−14	1.61E−14	8.88E−16
0.025	2.99E−05	3.98E−11	6.52E−14	5.24E−14	5.51E−14	4.51E−14	5.28E−14	0.00E+00

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	1.95E−03	4.28E−07	3.49E−08	4.12E−09	3.58E−09	8.36E−10	1.31E−09	5.48E−11
0.1	4.82E−04	1.66E−08	2.54E−10	3.35E−12	2.15E−11	1.49E−11	1.32E−11	6.29E−14
0.05	1.20E−04	7.60E−10	1.90E−12	1.02E−13	2.54E−14	5.16E−14	7.17E−14	8.88E−16
0.025	2.99E−05	3.98E−11	6.21E−14	4.91E−14	5.10E−14	5.42E−14	5.05E−14	0.00E+00

Tabelle 4.6: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-6}$, $\omega = 0.5$: Globaler Fehler an der Stelle $t_{end} = 2.4$.

IDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
0.2	7.87E−04	1.98E−06	1.16E−07	1.14E−07	1.14E−07	1.14E−07	1.14E−07
0.1	2.05E−04	1.24E−07	8.94E−10	8.46E−10	8.45E−10	8.45E−10	8.45E−10
0.05	5.21E−05	7.60E−09	7.50E−12	6.62E−12	6.62E−12	6.61E−12	6.61E−12
0.025	1.31E−05	4.69E−10	7.58E−14	6.00E−14	6.38E−14	6.20E−14	5.96E−14

IIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	7.87E−04	3.08E−04	6.28E−01	1.30E+03	2.69E+06	5.56E+09	1.15E+13	9.56E−11
0.1	2.05E−04	5.64E−07	1.72E−04	4.39E−02	1.12E+01	2.87E+03	7.33E+05	1.29E−13
0.05	5.21E−05	6.45E−09	3.03E−08	1.01E−06	3.35E−05	1.11E−03	3.69E−02	1.83E−15
0.025	1.31E−05	4.86E−10	4.22E−11	9.29E−11	5.75E−10	2.81E−09	1.42E−08	6.28E−16

TIIDeC:

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	7.87E−04	2.03E−06	1.74E−08	1.42E−08	1.25E−08	1.10E−08	9.67E−09	9.56E−11
0.1	2.05E−04	1.24E−07	1.82E−10	1.22E−10	1.06E−10	9.18E−11	7.96E−11	1.29E−13
0.05	5.21E−05	7.60E−09	1.98E−12	1.02E−12	9.02E−13	8.01E−13	7.17E−13	1.83E−15
0.025	1.31E−05	4.69E−10	2.01E−14	7.17E−15	7.17E−15	7.76E−15	1.48E−14	6.28E−16

QR-IIDeC (1):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	7.87E−04	2.03E−06	1.98E−08	1.42E−08	1.22E−08	1.08E−08	9.49E−09	9.56E−11
0.1	2.05E−04	1.24E−07	1.90E−10	1.21E−10	1.05E−10	9.09E−11	7.88E−11	1.29E−13
0.05	5.21E−05	7.60E−09	2.00E−12	1.01E−12	8.99E−13	7.98E−13	7.13E−13	1.83E−15
0.025	1.31E−05	4.69E−10	1.67E−14	1.06E−14	8.16E−15	1.29E−14	1.34E−14	6.28E−16

QR-IIDeC (2):

h	SDIRK(2)	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	RadauIIA(6)
0.2	7.87E−04	2.00E−06	5.69E−08	1.55E−08	1.36E−08	7.17E−09	7.07E−09	9.56E−11
0.1	2.05E−04	1.24E−07	4.62E−10	7.97E−11	8.37E−11	5.79E−11	5.19E−11	1.29E−13
0.05	5.21E−05	7.60E−09	3.62E−12	6.43E−13	6.37E−13	5.44E−13	4.93E−13	1.83E−15
0.025	1.31E−05	4.69E−10	1.10E−14	6.76E−15	9.16E−15	1.58E−14	1.10E−14	6.28E−16

Tabelle 4.7: Vergleich verschiedener Defektkorrekturalgorithmen (Kollokationsabszissen: RadauIIA(6), Basisverfahren: SDIRK(2)) bei Anwendung auf (3.99) mit $A(t)$ aus (3.102) und $\varepsilon = 10^{-6}, \omega = 1$: Globaler Fehler an der Stelle $t_{end} = 1.2$.

Anhang A

A.1 Matlab-Programme für die verschiedenen hier betrachteten Defektkorrekturalgorithmen

In diesem Abschnitt geben wir Matlab-Programme an, mit denen die klassische und die interpolierte Defektkorrektur getestet werden kann. Wie diese Programme angewendet werden, erfährt man am besten durch Betrachtung des konkreten Beispiels in Abschnitt A.1.9.

A.1.1 `Gauss.m`

Diese Funktion berechnet Gauß-Abszissen (2.12).
Algorithmus: Berechnung der Nullstellen des transformierten Legendre-Polynoms (2.14).

```
function gamma=Gauss(m)
%gamma=Gauss(m)
%  Berechnet die auf das Intervall [0,1] normierten
%  Abszissen des m-stufigen Gauss-Verfahrens.

if (m<1) error('m muss groessergleich 1 sein.'); end;
p0=[1 -1 0];
p=p0;
for i=2:m
   p=conv(p,p0);
end;
for i=1:m
   p=polyder(p);
```

```
end;
gamma=sort(roots(p));
```

A.1.2 RadauIIA.m

Diese Funktion berechnet RadauIIA-Abszissen (2.12).
Algorithmus: Berechnung der Nullstellen des Polynoms (2.15).

```
function gamma=RadauIIA(m)
%gamma=RadauIIA(m)
%  Berechnet die auf das Intervall [0,1] normierten
%  Abszissen des m-stufigen RadauIIA-Verfahrens.

if (m<1) error('m muss groessergleich 1 sein.'); end;
if m==1
   gamma=1;
else
   p0=[1 -1 0];
   p=p0;
   for i=2:m-1
      p=conv(p,p0);
   end;
   p=conv(p,[1 -1]);
   for i=1:m-1
      p=polyder(p);
   end;
   gamma=sort(roots(p));
end;
```

A.1.3 colloc_meth.m

Diese Funktion berechnet die Koeffizienten (2.19) von zu Kollokationsverfahren äquivalenten IRK-Verfahren.

```
function M=colloc_meth(c);
%M=colloc_meth(c)
%  Liefert in M eine Struktur mit den Koeffizienten jenes
```

```
%  IRK-Verfahrens, welches zu dem zu den in c gegebenen
%  Abszissen gehoerigen Kollokationsverfahren aequivalent
%  ist.

M.s=length(c);
c=reshape(c,M.s,1);

V=fliplr(vander(c));
C=zeros(M.s);
for i=1:M.s
   C(:,i)=c.^i/i;
end;
M.A=C/V;
e=ones(1,M.s)./[1:M.s];
M.b=e/V;
M.c=c';
M.d=e/C;
M.name=sprintf('colloc(%i)',M.s);
```

A.1.4 base_meth.m

Diese Funktion liefert für die IRK-Verfahren aus Abschnitt. 2.1.7 eine Struktur, die die Koeffizienten (2.19) des jeweiligen IRK-Verfahrens enthält.

```
function B=base_meth(name);
%B=base_meth(name)
%  Liefert in B eine Struktur mit den Koeffizienten
%  des durch name bezeichneten IRK-Verfahrens.
%  name muss einer der folgenden Strings sein:
%  'Euler'       ... Implizites Eulerverfahren
%  'IMR'         ... Implizite Mittelpunktsregel
%  'IMR2'
%  'ITR'         ... Implizite Trapezregel
%  'ITR2'
%  'SDIRK(2)'    ... 2-stufiges L-stabiles Singly-Diagonalimplizites Verfahren
%  'RadauIIA(2)' ... 2-stufiges RadauIIA-Verfahren

switch (name)
case 'Euler',
   B.s=1;B.A=1;B.b=1;B.c=1;B.d=1;
case 'IMR',
```

```
   B.s=1;B.A=.5;B.b=1;B.c=.5;B.d=2;
case 'IMR2',
   B.s=2;
   B.A=[.25 0; .5 .25];
   B.b=[.5 .5];
   B.c=[.25 .75];
   B.d=[-2 2];
case 'ITR',
   B.s=2;
   B.A=[0 0; .5 .5];
   B.b=[.5 .5];
   B.c=[0  1];
   B.d=[0  1];
case 'ITR2',
   B.s=3;
   B.A=[0 0 0; .25 .25 0; .25 .5 .25];
   B.b=[.25 .5 .25];
   B.c=[0 .5 1];
   B.d=[0 0 1];
case 'SDIRK(2)',
   B.s=2;
   gamma=1-sqrt(2)/2;
   B.A=[gamma 0; 1-gamma gamma];
   B.b=[1-gamma gamma];
   B.c=[gamma 1];
   B.d=[0 1];
case 'RadauIIA(2)'
   B.s=2;
   B.A=[5/12 -1/12; .75 .25];
   B.b=[.75 .25];
   B.c=[1/3 1];
   B.d=[0 1];
otherwise
   error('Nicht bekanntes IRK-Verfahren.');
end;
B.name=name;
```

A.1.5 IRK_Run.m

Diese Funktion berechnet numerische Lösungen von Anfangswertproblemen mit Hilfe von beliebigen IRK-Verfahren. Sie dient hier hauptsächlich zur Gewinnung von Kollokationslösungen, mit denen die mit den verschiedenen Defektkorrekturalgorithmen gewonnenen Approximationen verglichen werden.

```
function yend=IRK_Run(IRK,odefile,t0,y0,h,steps,varargin)
%yend=IRK_Run(IRK,'odefile',t0,y0,h,steps,...)
%  Liefert in yend die numerische Loesung an der Stelle t0+h*steps
%  der durch 'odefile' gegebenen Differentialgleichung mit dem
%  Anfangswert y0 an der Stelle t0.
%IRK   ... Struktur, die die Koeffizienten des IRK-Verfahrens
%  enthaelt. Die Funktionen base_meth.m und colloc_meth.m dienen
%  zur Generierung dieser Struktur.
%'odefile' ... Name einer Matlabfunktion
%  in der die rechte Seite der Differentialgleichung und
%  deren Jacobi-Matrix gegeben ist. Diese Funktion sollte
%  die folgende Form haben:
%
%    function out=odefile(t,y,flag,...)
%    if isempty(flag)
%      out=[...]; % rechte Seite der DGL
%    elseif strcmp(flag, 'jacobian')
%      out=[...]; % Jacobi-Matrix
%    end;
%
%t0,y0 ... Anfangswert
%h     ... Schrittweite
%steps ... Anzahl an Integrationsschritten der Schrittweite h
%Alle weiteren Argumente werden an die Funktion
%  'odefile' uebergeben.

n=length(y0); %Dimension der DGL
eps=1e-13;     %Genauigkeit des Newtonverfahrens,
               %fuer lineare AWP: setze eps=realmax;
max_newton=15; %Maximale Anzahl an Newtonschritten,
               %fuer lineare AWP: setze max_newton=1

f=zeros(n*IRK.s,1);
J=zeros(n,IRK.s*n);
hA_kron_ones = h*kron(IRK.A,ones(n));
hA_kron_eye  = h*kron(IRK.A,eye(n));
d_kron_eye   = kron(IRK.d,eye(n));

t=t0;y=y0;
for nu=1:steps
   z=zeros(n*IRK.s,1); %Startwerte fuer Newton-Iteration
   for newton_step=1:max_newton  %Newton-Iteration...
      %Iterationsmatrix generieren:
      for i=1:IRK.s
```

```
        f((i-1)*n+1:i*n,:)=feval(odefile,t+h*IRK.c(i), ...
                    y+z((i-1)*n+1:i*n,:),'',varargin:);
        J(:,(i-1)*n+1:i*n)=feval(odefile,t+h*IRK.c(i), ...
             y+z((i-1)*n+1:i*n,:),'jacobian',varargin:);
      end;
      JJ=eye(IRK.s*n)-hA_kron_ones.*kron(ones(IRK.s,1),J);
      hff=hA_kron_eye*f;
      dz=JJ\(-z+hff);
      z=z+dz;
      nn=norm(dz,inf);
      if nn<eps break; end;
   end;
   if nn>eps
      error(strcat(sprintf('In Integrationsschritt %i, ',nu),...
                   sprintf('Newton-Verfahren konvergiert nicht.\n'),...
                   sprintf('(Nach %i Newtonschritten: |dz|=%e).',n)));
   end;
   y=y+d_kron_eye*z;
   t=t+h;
end;
yend=y;
```

A.1.6 DC_Init.m

Diese Funktion dient zur Initialisierung der verschiedenen in DC_Run.m (vgl. Abschnitt A.1.7) implementierten Defektkorrekturalgorithmen. Insbesonders werden hier im Fall der klassischen Defektkorrektur die Interpolationsgewichte $\widetilde{w}_{i,r,j}$ (2.63) und $\widetilde{w}'_{i,r,j}$ (2.64) bzw. im Fall der interpolierten Defektkorrektur die Interpolationsgewichte $\widehat{w}_{i,j}$ (2.69), $\widehat{w}'_{i,j}$ (2.70) und $\widetilde{v}_{i,r,j}$ (2.74) berechnet.

```
function DC=DC_Init(type,g,conn,base);
%DC=DC_Init(type,g,conn,base);
%  Liefert in DC eine Struktur von Daten, die einen bestimmten
%  Defektkorrekturalgorithmus spezifizieren. Diese Struktur
%  wird von DC_Run.m benoetigt.
%type muss einer der folgenden Strings sein, wobei der Parameter
%  g in Abhaengigkeit von type die angegebene Bedeutung hat:
%  'IDeC'      ... klassische Defektkorrektur mit
%                  Grad der Interpolationspolynome m=g.
%  'IIDeC'     ... interpolierte Defektkorrektur mit den in g
%                  gegebenen Defektinterpolationsabszissen
%  'TIIDeC'    ... interpolierte Defektkorrektur mit den in g
```

```
%                   gegebenen Defektinterpolationsabszissen, wobei
%                   der Defekt zuerst transformiert wird und dieser
%                   transformierte Defekt dann interpoliert wird.
%                   Die Transformationsmatrizen muessen dabei
%                   vom Benutzer bereitgestellt werden (siehe
%                   DC_run.m).
%  'QR-IIDeC'  ... Mit QR-Verfahren kombinierte Interpolierte
%                   Defektkorrektur, wobei die jeweilige
%                   Transformationsmatrix Z durch die orthogonale
%                   Matrix Q der QR-Zerlegung der entsprechenden
%                   Jacobi-Matrix A gegeben ist
%  'QR-IIDeC2' ... Mit QR-Verfahren kombinierte Interpolierte
%                   Defektkorrektur, wobei die jeweilige
%                   Transformationsmatrix Z durch die orthogonale
%                   Matrix Q der QR-Zerlegung der Matrix I-h*c(1)*A
%                   gegeben ist. Das ist nur fuer singly-diagonalimplizite
%                   Basisverfahren wie Euler und SDIRK(2) sinnvoll.
%conn muss einer der folgenden Strings sein:
%  'local'  ... lokale Verbindungsstrategie
%  'global' ... globale Verbindungsstrategie
%base ist eine Struktur, die die Koeffizienten des Basisverfahrens
%  enthaelt. Die Funktionen base.m und colloc_meth.m dienen zur
%  Generierung dieser Struktur.

DC.base=base;
if strcmp(conn,'local')
   DC.conn=1;
elseif strcmp(conn,'global')
   DC.conn=2;
else
   error(sprintf('Unbekannte Verbindungsstrategie: "%s"',conn));
end;
if strcmp(type,'IDeC')
   DC.type=1;
   DC.m=g;
   DC.W0=zeros(DC.m*DC.base.s,1);
   DC.W1=zeros(DC.m*DC.base.s,DC.m);
   DC.Wd0=zeros(DC.m*DC.base.s,1);
   DC.Wd1=zeros(DC.m*DC.base.s,DC.m);
   DC.V=[];
   for j=0:DC.m
      denom=prod(j-[0:j-1,j+1:DC.m]);
      for i=1:DC.m
         for r=1:DC.base.s
            num=prod(i-1+DC.base.c(r)-[0:j-1,j+1:DC.m]);
```

```
            if (j==0)
               DC.W0((i-1)*DC.base.s+r)=num/denom;
            else
               DC.W1((i-1)*DC.base.s+r,j)=num/denom;
            end;
            num=0;
            for k1=[0:j-1,j+1:DC.m]
               num=num+prod(i-1+DC.base.c(r)- ...
                  [0:min(j,k1)-1,min(j,k1)+1:max(j,k1)-1,max(j,k1)+1:DC.m]);
            end;
            if (j==0)
               DC.Wd0((i-1)*DC.base.s+r)=num/denom;
            else
               DC.Wd1((i-1)*DC.base.s+r,j)=num/denom;
            end;
         end;
      end;
   end;
elseif strcmp(type,'IIDeC')    | strcmp(type,'TIIDeC') |...
       strcmp(type,'QR-IIDeC') | strcmp(type,'QR-IIDeC2')
   DC.m=length(g);
   DC.gamma=g;
   DC.W0=zeros(DC.m,1);
   DC.W1=zeros(DC.m,DC.m);
   DC.Wd0=zeros(DC.m,1);
   DC.Wd1=zeros(DC.m,DC.m);
   DC.V=zeros(DC.m*DC.base.s,DC.m);
   for j=0:DC.m
      denom=prod(j-[0:j-1,j+1:DC.m]);
      for i=1:DC.m
         num=prod(DC.m*DC.gamma(i)-[0:j-1,j+1:DC.m]);
         if (j==0)
            DC.W0(i)=num/denom;
         else
            DC.W1(i,j)=num/denom;
         end;
         num=0;
         for k1=[0:j-1,j+1:DC.m]
            num=num+prod(DC.m*DC.gamma(i)- ...
               [0:min(j,k1)-1,min(j,k1)+1:max(j,k1)-1,max(j,k1)+1:DC.m]);
         end;
         if (j==0)
            DC.Wd0(i)=num/denom;
         else
            DC.Wd1(i,j)=num/denom;
```

```
         end;
      end;
   end;
   for j=1:DC.m
      denom=prod(DC.m*DC.gamma(j)-DC.m*DC.gamma([1:j-1,j+1:DC.m]));
      for i=1:DC.m
         for r=1:DC.base.s
            num=prod(i-1+DC.base.c(r)-DC.m*DC.gamma([1:j-1,j+1:DC.m]));
            DC.V((i-1)*DC.base.s+r,j)=num/denom;
         end;
      end;
   end;
   if strcmp(type,'IIDeC')
      DC.type=2;
   else
      if strcmp(type,'TIIDeC')
         DC.type=3;
      elseif strcmp(type,'QR-IIDeC')
         DC.type=4;
      elseif strcmp(type,'QR-IIDeC2')
         DC.type=5;
      end;
      % Die folgenden Matrizen DC.U0 u. DC.U1
      % entsprechen den Matrizen DC.W0 u. DC.W1
      % im Fall der klassischen IDeC
      DC.U0=zeros(DC.m*DC.base.s,1);
      DC.U1=zeros(DC.m*DC.base.s,DC.m);
      for j=0:DC.m
         denom=prod(j-[0:j-1,j+1:DC.m]);
         for i=1:DC.m
            for r=1:DC.base.s
               num=prod(i-1+DC.base.c(r)-[0:j-1,j+1:DC.m]);
               if (j==0)
                  DC.U0((i-1)*DC.base.s+r)=num/denom;
               else
                  DC.U1((i-1)*DC.base.s+r,j)=num/denom;
               end;
            end;
         end;
      end;
   end;
else
   error(sprintf('Unbekannter Defektkorrekturtyp: "%s"',type));
end;
```

A.1.7 DC_Run.m

In dieser Funktion sind nun die verschiedenen Varianten der Defektkorrektur aus Kapitel 2 und Kapitel 4 implementiert. In ihrer Struktur folgt diese Funktion ziemlich genau dem Pseudocode-Algorithmus 2.4.1.

```
function eta_m=DC_Run(DC,odefile,t0,y0,H,steps,dc_steps,varargin)
%yend=DC_Run(DC,odefile,t0,y0,H,steps,dc_steps,...)
%  Liefert in yend eine Matrix mit dc_steps+1 Spalten.
%  1.Spalte: Basisapproximation an der Stelle t0+H*steps
%  fuer die exakte Loesung der in 'odefile' gegebenen
%  Differentialgleichung mit Anfangswert y0 an der Stelle t0.
%  Die weiteren Spalten enthalten die durch Defektkorrektur
%  verbesserten Approximationen an der Stelle t0+H*steps.
%DC ... Struktur, durch die der Defektkorrekturalgorithmus
%  spezifiziert wird. Diese Struktur wird von der Funktion
%  IDeC_Init.m generiert.
%'odefile' ... Name einer Matlabfunktion
%  in der die rechte Seite Differentialgleichung und
%  deren Jacobi-Matrix gegeben ist. Diese Funktion sollte
%  die folgende Form haben:
%
%    function out=odefile(t,y,flag,...)
%    if isempty(flag)
%      out=[...]; % rechte Seite der DGL
%    elseif strcmp(flag, 'jacobian')
%      out=[...]; % Jacobi-Matrix
%    end;
%
%  Im Fall der Interpolierten Defektkorrektur mit transformierten
%  Defekten (TIIDeC) sollte diese Funktion die folgende Form haben:
%
%    function out=odefile(t,y,flag,...)
%    if isempty(flag)
%      out=[...]; % rechte Seite der DGL
%    elseif strcmp(flag, 'jacobian')
%      out=[...]; % Jacobi-Matrix
%    elseif strcmp(flag, 'trafo1')
%      out=[...]; % Transformationsmatrix Z(t)
%    elseif strcmp(flag, 'trafo2')
%      out=[...]; % Transformationsmatrix Z(t)^(-1)
%    end;
%
%t0,y0    ... Anfangswert
```

```
%H        ... Schrittweite (des ZIEL(!)verfahrens)
%steps    ... Anzahl an Integrationsschritten der Schrittweite H
%dc_steps ... Anzahl an Defektkorrekturschritten
%Alle weiteren Argumente werden an die Funktion
%  'odefile' uebergeben.

steps=round(steps);
dc_steps=round(dc_steps);

linear=1;       %fuer nichtlineare Probleme setze linear=0;
n=length(y0);   %Dimension der DGL
if (linear==1)  %Fuer lineare Probleme jeweils nur eine
   eps=realmax; %Newton-Iteration notwendig
   max_newton=1;
else
   eps=1e-13;     %Genauigkeit des Newtonverfahrens
   max_newton=25; %Maximale Anzahl an Newtonschritten
end;
h=H/DC.m;      %H...Schrittweite des Zielverfahrens
               %h...Schrittweite des Basisverfahrens

%Koeffizienten- und Gewichts-Matrizen
%auf die richtigen dimensionsabhaengigen Groessen bringen
hA_kron_ones = h*kron(DC.base.A,ones(n));
hA_kron_eye  = h*kron(DC.base.A,eye(n));
d_kron_eye   = kron(DC.base.d,eye(n));
W0  = kron(DC.W0,eye(n));
W1  = kron(DC.W1,eye(n));
Wd0_div_h = kron(DC.Wd0/h,eye(n));
Wd1_div_h = kron(DC.Wd1/h,eye(n));
if (DC.type==2)
   V = kron(DC.V,eye(n));
end;
if (DC.type>=3)
   U0  = kron(DC.U0,eye(n));
   U1  = kron(DC.U1,eye(n));
end;
eta_0=kron(ones(1,dc_steps+1),y0);
s=eta_0;

for l=0:steps-1
   if (l>=1)
      if (DC.conn==1) %lokale Verbindungsstrategie
         s=kron(ones(1,dc_steps+1),eta_m(:,dc_steps+1));
         eta_0=s;
```

```
   else            %globale Verbindungsstrategie
      s=pi_m;
      eta_0=eta_m;
   end;
end;

%***** Basisschritt: *****
t=t0;y=s(:,1);
for nu=1:DC.m
  %Startwerte fuer Newton-Iteration: Bessere Werte moeglich,
  %vgl. [9, p.120]
  z=zeros(n*DC.base.s,1);
   for newton_step=1:max_newton  %Newton-Iteration...
      %Iterationsmatrix generieren:
      for i=1:DC.base.s
         f((i-1)*n+1:i*n,:)=feval(odefile,t+h*DC.base.c(i),...
                         y+z((i-1)*n+1:i*n,:),'',varargin:);
         J(:,(i-1)*n+1:i*n)=feval(odefile,t+h*DC.base.c(i),...
                   y+z((i-1)*n+1:i*n,:),'jacobian',varargin:);
      end;
      JJ=eye(DC.base.s*n)-hA_kron_ones.*kron(ones(DC.base.s,1),J);
      %Speichere LU-Zerlegung von JJ fuer die spaeteren
      %Defektkorrekturschritte:
      [LL(:,:,nu),UU(:,:,nu)]=lu(JJ);
      hff=hA_kron_eye*f;
      dz=UU(:,:,nu)\(LL(:,:,nu)\(-z+hff));
      z=z+dz;
      nn=norm(dz,inf);
      if nn<eps break; end; %primitives Abbruchkriterium
                            %fuer Newton-Iteration
   end;
   if nn>eps
      error(strcat(sprintf('In Integrationsschritt %i, ',l*DC.m+nu),...
            sprintf('Defektkorrekturschritt %i: ',0),        ...
            sprintf('Newton-Verfahren konvergiert nicht.\n'),...
            sprintf('(Nach %i Newtonschritten: |dz|=%e).',newton_step,nn)));
   end;
   zz(:,nu)=z; %Speichere Newton-Startwerte fuer
               %die spaeteren Defektkorrekturschritte
   y=y+d_kron_eye*z;
   eta_base((nu-1)*n+1:nu*n,:)=y;
   t=t+h;
end;
pi_m(:,1)=y;
eta_m(:,1)=y;
```

```
eta=eta_base;

%***** Defektkorrekturschritte: *****
if (DC.type>=3) % Interpolierte Defektkorrektur mit Defektkttransformation:

   % Transformationsmatrizen generieren...
   V=zeros(DC.m*DC.base.s*n,DC.m*n);
   P=reshape(W0*eta_0(:,1)+W1*eta,n,DC.m);
   for nu=1:DC.m
      if (DC.type==3)        %TIIDeC
         X2(:,:,nu)=feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),P(:,nu),'trafo2',varargin:);
      elseif (DC.type==4)    %QR-IIDeC
         X2(:,:,nu)=qr1(feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),P(:,nu),'jacobian',varargin:));
      elseif (DC.type==5)    %QR-IIDeC2
         X2(:,:,nu)=qr1(eye(n)-DC.base.c(1)*h*feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),P(:,nu),'jacobian',varargin:));
      end;
   end;
   P=reshape(U0*eta_0(:,1)+U1*eta,n,DC.base.s,DC.m);
   for nu=1:DC.m
      for i=1:DC.base.s
         ii=(nu-1)*DC.base.s+i;
         if (DC.type==3)      %TIIDeC
            X1(:,:,ii)=feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),P(:,i,nu),'trafo1',varargin:);
         elseif (DC.type==4)  %QR-IIDeC
            X1(:,:,ii)=qr1(feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),P(:,i,nu),'jacobian',varargin:));
         elseif (DC.type==5)  %QR-IIDeC2
            X1(:,:,ii)=qr1(eye(n)-DC.base.c(1)*h*feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),P(:,i,nu),'jacobian',varargin:));
         end;
      end;
   end;
   for i=1:DC.m*DC.base.s
      for j=1:DC.m
         V((i-1)*n+1:i*n,(j-1)*n+1:j*n)=DC.V(i,j)*X1(:,:,i)*X2(:,:,j);
      end;
   end;

end;

for k=1:dc_steps
```

```
%Defekt:
if (DC.type==1) %klassische Defektkorrektur
   P =reshape(W0       *eta_0(:,k) +W1       *eta, n,DC.base.s,DC.m);
   Pd=reshape(Wd0_div_h*eta_0(:,k) +Wd1_div_h*eta, n,DC.base.s,DC.m);
   for nu=1:DC.m
      for i=1:DC.base.s
         delta(:,i,nu)=Pd(:,i,nu)-feval(odefile,  ...
            t0+h*(nu-1+DC.base.c(i)),P(:,i,nu),'',varargin:);
      end;
   end;
else            %interpolierte Defektkorrektur
   P =reshape(W0       *eta_0(:,k) +W1       *eta, n,DC.m);
   Pd=        Wd0_div_h*eta_0(:,k) +Wd1_div_h*eta;
   for nu=1:DC.m
      F((nu-1)*n+1:nu*n,:)=feval(odefile,t0+H*DC.gamma(nu), ...
         P(:,nu),'',varargin:);
   end;
   delta=reshape(V*(Pd-F),n,DC.base.s,DC.m);
end;
t=t0;y=s(:,k+1);
for nu=1:DC.m
   z=zz(:,nu); %Startwerte aus Basisschritt fuer Newton-Iteration
   for newton_step=1:max_newton  %Newton-Iteration...
      for i=1:DC.base.s
         f((i-1)*n+1:i*n,:)=feval(odefile,t+h*DC.base.c(i), ...
            y+z((i-1)*n+1:i*n,:),'',varargin:)+delta(:,i,nu);
      end;
      hff=hA_kron_eye*f;
      %LU-Zerlegung der Iterationsmatrix aus Basisschritt:
      dz=UU(:,:,nu)\(LL(:,:,nu)\(-z+hff));
      z=z+dz;
      nn=norm(dz,inf);
      if nn<eps break; end;
   end;
   if nn>eps
      error(strcat(sprintf('In Integrationsschritt %i, ',l*DC.m+nu),...
            sprintf('Defektkorrekturschritt %i: ',k),        ...
            sprintf('Newton-Verfahren konvergiert nicht.n'),...
            sprintf('(Nach %i Newtonschritten: |dz|=%e).',newton_step,nn)));
   end;
   y=y+d_kron_eye*z;
   pi((nu-1)*n+1:nu*n,:)=y;
   t=t+h;
end;
pi_m(:,k+1)=y;
```

```
      eta=eta_base-(pi-eta);  %verbesserte Approximation
      eta_m(:,k+1)=eta((DC.m-1)*n+1:DC.m*n,:);
   end;
   t0=t0+H;
end;
```

A.1.8 qr1.m

Für die in der obigen Funktion `DC_Run.m` implementierten QR-IIDeC benötigen wir eine Routine für die QR-Zerlegung $A = Q \cdot R$ einer Matrix A, bei der die Matrix Q stetig von den Einträgen der Matrix A abhängt. Dies wird durch die folgende Funktion `qr1.m` gewährleistet. Bei exakter Arithmetik würde der in `qr1.m` implementierte Algorithmus dasselbe Resultat liefern, wie wenn der `if`-Block nicht vorhanden wäre, in dem (im Fall `R(k,k)<0`) das Vorzeichen von `sigma` geändert wird.

```
function [Q,R]=qr1(A)
%QR-Zerlegung der Matrix A, wobei hier im Gegensatz zur in
%Matlab eingebauten Routine qr.m die Matrizen Q und R
%stetig von den Eintraegen der Matrix A abhaengen.
%Diese Funktion basiert auf der Routine "qrdcmp" aus
%  W.H. Press e.a.: Numerical Recipes in C, The Art of Scientfic
%  Computing, 2nd editition, Cambridge University Press.

n=size(A,1);
c=zeros(n,1);
d=zeros(n,1);
xx=zeros(n,1);
Q=zeros(n);
R=A;
dd=ones(1,n);

for k=1:n-1
   scale=0.0;
   for i=k:n
      scale=max([scale,abs(R(i,k))]);
   end;
   if scale==0.0
      c(k)=0.0;
      d(k)=0.0;
   else
      for i=k:n
```

```
         R(i,k)=R(i,k)/scale;
      end;
      sum=0.0;
      for i=k:n
         sum=sum+R(i,k)*R(i,k);
      end;
      sigma=sqrt(sum);
      if R(k,k)<0.0
         sigma=-sigma;  %Das hat die Umorientierung der k-ten und
         dd(k)=-dd(k);  %der letzten Spalte von Q zur Folge, was ganz
         dd(n)=-dd(n);  %unten wieder rueckgaengig gemacht wird.
      end;
      R(k,k)=R(k,k)+sigma;
      c(k)=sigma*R(k,k);
      d(k)=-scale*sigma;
      for j=k+1:n
         sum=0.0;
         for i=k:n
            sum=sum+R(i,k)*R(i,j);
         end;
         tau=sum/c(k);
         for i=k:n
               R(i,j)=R(i,j)-tau*R(i,k);
         end;
      end;
   end;
end;
d(n)=R(n,n);

for i=1:n
   for j=1:n
      Q(i,j)=-R(i,1)*R(j,1)/c(1);
   end;
   Q(i,i)=Q(i,i)+1.0;
end;

xx=zeros(n,1);
for j=2:n-1
   for i=1:n
      sum=0.0;
      for k=j:n
         sum=sum+R(k,j)*Q(i,k);
      end;
      xx(i)=sum;
   end;
```

```
   for i=1:n
      for k=j:n
         Q(i,k)=Q(i,k)-xx(i)*R(k,j)/c(j);
      end;
   end;
end;
for j=1:n
   R(j,j)=d(j);
   for i=j+1:n
      R(i,j)=0.0;
   end;
end;
%nachtraegliches Rueckgaengigmachen
%der oben erwaehnten Umorientierung
%der Spalten von Q und der Zeilen von R:
D=diag(dd);
Q=Q*D;
R=D*R;
```

A.1.9 Exemplarische Anwendung der obigen Programme

Als Differentialgleichung betrachten wir die 2-dimensionale lineare Differentialgleichung (3.99) mit der Matrix $A(t)$ aus (3.102). Dazu erstellen wir die folgende Datei mit dem Namen `ode_bsp.m`:

```
function out = ode_bsp(t,y,flag,epsilon,omega)
if isempty(flag)
   s=sin(t);c=cos(t);
   ss=sin(omega*t);cc=cos(omega*t);
   out=[(-cc*cc/epsilon-ss*ss)*(y(1)-s-2)+(cc/epsilon*ss-ss*cc)*(y(2)-c-2)+c;
      (cc/epsilon*ss-ss*cc)*(y(1)-s-2)+(-ss*ss/epsilon-cc*cc)*(y(2)-c-2)-s];
elseif strcmp(flag, 'jacobian')
   ss=sin(omega*t);cc=cos(omega*t);
   out=[-cc*cc/epsilon-ss*ss, cc/epsilon*ss-ss*cc;
         cc/epsilon*ss-ss*cc, -ss*ss/epsilon-cc*cc];
elseif strcmp(flag, 'trafo1')
   ss=sin(omega*t);cc=cos(omega*t);
   out=[cc,ss;-ss,cc];
elseif strcmp(flag, 'trafo2')
   ss=sin(omega*t);cc=cos(omega*t);
   out=[cc,-ss;ss,cc];
```

```
end;
```

Diese Differentialgleichungs-Datei ist von einer solchen Form, daß sie auch für die Matlab-eigenen Differentialgleichungslöser wie z.B. `ode15s` oder `ode23s` verwendet werden kann.

Wir betrachten nun einen interaktiven Dialog mit Matlab, wobei zunächst alle Variablen gelöscht werden, und das Ausgabeformat festgelegt wird:

```
> clear; format compact; format short g
```

Für die Parameter der Differentialgleichung setzen wir $\varepsilon = 10^{-6}$ und $\omega = 0.2$, und als Anfangswertbedingung nehmen wir $y(0) = (2, 3)^T$:

```
> epsilon=1e-6; omega=0.2;
> t0=0; y0=[2; 3];
```

Das so spezifizierte Anfangswertproblem integrieren wir mit Hilfe von 5 Defektkorrekturschritten der klassischen Defektkorrektur für $t \in [0, 2.4]$, wobei der Polynomgrad $m = 6$, die globale Verbindungsstrategie und als Basisverfahren SDIRK(2) mit Schrittweite $h = H/m = 0.05$ verwendet wird:

```
> DC=DC_Init('IDeC',6,'global',base_meth('SDIRK(2)'));
> H=.2; steps=12; dc_steps=5;
> y_dc=DC_Run(DC,'ode_bsp',t0,y0,H,steps,dc_steps,epsilon,omega)
y_dc =
       2.6755       2.6755       2.6755       2.6755       2.6755       2.6755
       1.2626       1.2626       1.2626       1.2626       1.2626       1.2626
```

Die erhaltenen Approximationen für die Lösung an der Stelle $t = 2.4$ vergleichen wir mit der exakten Lösung $(2 + \sin(2.4), 2 + \cos(2.4))^T$:

```
> y_ex=[2+sin(H*steps);2+cos(H*steps)]
y_ex =
       2.6755
       1.2626
> y_dc-kron(ones(1,dc_steps+1),y_ex)
ans =
  1.3162e-005 -3.0836e-010  1.1915e-012  1.1924e-012  1.1946e-012  1.1928e-012
  2.5257e-005 -5.9216e-010  2.2884e-012  2.2895e-012  2.2933e-012  2.2904e-012
```

Wir lösen nun das Anfangswertproblem mit Hilfe der Interpolierten Defektkorrektur, wobei als Defektinterpolations-Abszissen (2.12) die Abszissen des 6-stufigen RadauIIA-Verfahrens verwendet werden und alle anderen Parameter unverändert bleiben:

```
> DC=DC_Init('IIDeC',RadauIIA(6),'global',base_meth('SDIRK(2)'));
> y_dc=DC_Run(DC,'ode_bsp',t0,y0,H,steps,dc_steps,epsilon,omega)
y_dc =
       2.6755       2.6755       2.6755       2.6755       2.6755       2.6755
       1.2626       1.2626       1.2626       1.2626       1.2626       1.2626
```

Das Ergebnis vergleichen wir wieder mit der exakten Lösung:

```
> y_dc-kron(ones(1,dc_steps+1),y_ex)
ans =
  1.3162e-005 -3.0109e-010  4.0059e-011 -1.8266e-011  2.1111e-011 -2.2176e-011
  2.5257e-005 -5.7821e-010  7.6945e-011 -3.5086e-011   4.055e-011 -4.2598e-011
```

Man beobachtet, daß im Laufe der Defektkorrekturiteration die Fehler wieder größer werden. Das ist ein Anzeichen für die Instabilität der Interpolierten Defektkorrektur bei variierenden steifen Eigenrichtungen. Durch Verwendung der QR-IIDeC sollte dieses Ergebnis stark verbessert werden, das wird durch das folgende bestätigt:

```
> DC=DC_Init('QR-IIDeC',RadauIIA(6),'global',base_meth('SDIRK(2)'));
> y_dc=DC_Run(DC,'ode_bsp',t0,y0,H,steps,dc_steps,epsilon,omega)
y_dc =
       2.6755       2.6755       2.6755       2.6755       2.6755       2.6755
       1.2626       1.2626       1.2626       1.2626       1.2626       1.2626
> y_dc-kron(ones(1,dc_steps+1),y_ex)
ans =
  1.3162e-005 -3.0854e-010 -1.7319e-014  5.7732e-015  7.9936e-015  7.5495e-015
  2.5257e-005 -5.9252e-010 -3.3973e-014  1.0214e-014  1.4433e-014  1.3101e-014
```

Zuletzt vergleichen wir noch die mit der QR-IIDeC gewonnenen Approximationen mit der durch Anwendung des entsprechenden 6-stufigen RadauIIA-Verfahrens gewonnenen Approximation:

```
> y_coll=IRK_Run(colloc_meth(RadauIIA(6)),'ode_bsp',t0,y0,H,steps,epsilon,omega)
y_coll =
     2.6755
     1.2626
> y_dc-kron(ones(1,dc_steps+1),y_coll)
ans =
  1.3162e-005 -3.0854e-010 -1.7764e-014  5.3291e-015  7.5495e-015  7.1054e-015
  2.5257e-005 -5.9252e-010 -3.4195e-014   9.992e-015  1.4211e-014  1.2879e-014
```

Man beobachtet die rasche Fixpunkt-Konvergenz durch die QR-IIDeC gewonnenen Approximationen gegen die Kollokationslösung.

A.2 Matlab-Skript zur Generierung der Abbildungen 3.1–3.21

Die Abbildung 3.3 (Implizites Eulerverfahren als Basisverfahren und RadauIIA(m)-Abszissen (2.12), $m = 3, 4, 5, 6$) wurde mit Hilfe des folgenden Matlab-Skripts erzeugt (für die anderen der Abbildungen 3.1–3.21 wurde jeweils die fünfte Zeile des Skripts entsprechend abgeändert):

```
z=-10.^[-5:.05:10]; %logarithmische z-Skala
rho=zeros(1,length(z));
hold off
for m=3:6
   DC=DC_Init('IIDeC',RadauIIA(m),'global',base_meth('Euler'));
   for k=1:length(z)
      rho(k)=DC_scalar_rho(DC,z(k));
   end;
   semilogx(-z,rho); %-z, damit z->0 links und z->inf rechts ist
   text(-z(end),rho(end),sprintf(' m=%i',m));
   hold on;
end;
set(gca,'XTICK',[1e-5 1 1e5 1e10])
set(gca,'XTICKlabel','-1e-5|-1|-1e5|-1e10')
xlabel('\itz');
ylabel('\rho(\itS(\itz))');
```

Dieses Skript verwendet die Funktion `DC_Init.m` aus Abschnitt A.1.6 und die folgende Funktion `DC_scalar_rho.m`, die zur Berechnung des Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IDeC}}(z))$ bzw. $\rho(\boldsymbol{S}_{\mathrm{IIDeC}}(z))$ der Matrix $\boldsymbol{S}_{\mathrm{IDeC}}(z)$ bzw. $\boldsymbol{S}_{\mathrm{IIDeC}}(z)$ aus Abschnitt 3.2 für gegebenes $z = h\lambda$ dient:

```
function rho=DC_scalar_rho(DC,z);
%rho=IDeC_scalar_rho(DC,z)
%  Berechnung des Spektralradius der Iterationsmatrix
%  von auf  y'=lambda*y angewendetetn Defektkorrektur-
%  algorithmen. Dieser Spektralradius ist vom Produkt
%  z=h*lambda abhaengig, wobei h die Schrittweite des
%  Basisverfahrens ist.
%DC ... Struktur, durch die der Defektkorrektur-
```

```
%  algorithmus spezifiziert wird. Diese Struktur wird
%  von der Funktion DC_Init.m generiert.
%z  ... das oben erwaehnte Produkt h*lambda

r=zeros(1,DC.base.s);
denom=det(eye(DC.base.s)-z*DC.base.A);
M=[eye(DC.base.s)-z*DC.base.A;-z*DC.base.b];
r0=det([M,ones(DC.base.s+1,1)])/denom;
for j=1:DC.base.s
   r(j)=det([M,[DC.base.A(:,j);DC.base.b(j)]])/denom;
end;
K=eye(DC.m);
x=kron(ones(DC.m-1,1),r0).^([1:DC.m-1]');
for i=1:DC.m-1
  K(i+1:end,i)=x(1:DC.m-i);
end;
K=kron(K,r);
if (DC.type==1) %klassische Defektkorrektur
   S=eye(DC.m)-K*(DC.Wd1-z*DC.W1);
else            %interpolierte Defektkorrektur
   S=eye(DC.m)-K*DC.V*(DC.Wd1-z*DC.W1);
end;
rho=norm(eig(S),inf); %rho = Spektralradius von S
```

A.3 Matlab-Skript zur Generierung der Abbildungen 4.1–4.3

Die Abbildungen 4.1–4.3 wurden mit Hilfe des folgenden Matlab-Skripts erzeugt:

```
h=0.1; % fuer Abb. 4.2 und 4.4. Abb. 4.1: h=0.05, Abb. 4.3: h=0.025
m=6;
epsilon=1e-5;
DC=DC_Init('IIDeC',RadauIIA(m),'global',base_meth('SDIRK(2)'));
%fuer Abb 4.4 setze:
%DC=DC_Init('TIIDeC',RadauIIA(m),'global',base_meth('SDIRK(2)'));
odefile='ode_bsp';n=2;
x=0:.02:1; % ... omega
y=0: .2:8; % ... -log10(epsilon)
[X,Y]=meshgrid(x,y);
Z=zeros(size(X));
for i=1:length(x)
   i
```

```
   for j=1:length(y);
      omega=x(i);
      epsilon=10.^(-y(j));
      Z(j,i)=DC_rho(DC,n,ode_bsp,0,h,epsilon,omega);
   end;
end;
subplot(2,2,1);
hold off
[cs,hh]=contour(X,Y,Z,[ [.4:.2:.8] 1 10 100 1000 10000 100000],'k');
hold on
ylabel('
epsilon');
xlabel('
omega');
set(gca,'YTICK',[0 2 4 6 8]);
set(gca,'YTICKlabel','1e0|1e-2|1e-4|1e-6|1e-8');
clabel(cs,hh,'manual');
```

Dieses Skript verwendet die Funktion `DC_Init.m` aus Abschnitt A.1.6 und die folgende Funktion `DC_rho.m`, die zur Berechnung des Spektralradius $\rho(\boldsymbol{S}_{\mathrm{IIDeC}})$ bzw. $\rho(\boldsymbol{S}_{\mathrm{TIIDeC}})$ der Matrix $\boldsymbol{S}_{\mathrm{IIDeC}}$ aus Abschnitt 3.1 bzw. $\boldsymbol{S}_{\mathrm{TIIDeC}}$ aus Abschnitt 4.2.2 für eine beliebige Matrix $A(t)$ in (3.1) dient. Diese Matrix ist für die Abbildungen 4.1–4.4 die Matrix (3.102), welche von der oben angegebenen Matlab-Differentialgleichungsdatei `ode_bsp.m` als Jakobimatrix bereitgestellt wird.

```
function rho=DC_rho(DC,n,odefile,t0,h,varargin)
%rho=DC_rho(DC,n,odefile,t0,h,...)
%  Berechnung des Spekralradius der Iterationsmatrix
%  von auf y'=A(t)*y angewendeten Defektkorrektur-
%  algorithmen.
%DC ... Struktur, durch die der Defektkorrektur-
%  algorithmus spezifiziert wird. Diese Struktur
%  wird von der Funktion DC_Init.m generiert.
%n  ... Dimesnion der Differentialgleichung
%'odefile' ... Name einer Matlabfunktion,
%  in der die Matrix A(t) gegeben ist. Diese
%  Funktion sollte die folgende Form haben:
%
%    function out=odefile(t,y,flag,...)
%    if strcmp(flag, 'jacobian')
%      out=[...]; % A(t)
%    end;
%
%  Im Fall der Interpolierten Defektkorrektur mit transformierten
```

```
%  Defekten (TIIDeC) sollte diese Funktion die folgende Form haben:
%
%    function out=odefile(t,y,flag,...)
%    if strcmp(flag, 'jacobian')
%      out=[...]; % A(t)
%    elseif strcmp(flag, 'trafo1')
%      out=[...]; % Transformationsmatrix Z(t)
%    elseif strcmp(flag, 'trafo2')
%      out=[...]; % Transformationsmatrix Z(t)^(-1)
%    end;
%
%t0 ... Anfang des Integrationsintervalls
%h  ... Schrittweite des Basisverfahrens
%Alle weiteren Argumente werden an die Funktion
%  'odefile' uebergeben.

%Koeffizienten- und Gewichts-Matrizen
%auf die richtigen dimensionsabhaengigen Groessen bringen
hA_kron_eye  = h*kron(DC.base.A,eye(n));
hb_kron_eye   = h*kron(DC.base.b,eye(n));
W1  = kron(DC.W1,eye(n));
Wd1_div_h = kron(DC.Wd1/h,eye(n));
s=DC.base.s;
m=DC.m;
if (DC.type==1)
   error('In DC_rho.m ist die klassische Defektkorrektur nicht implementiert.');
elseif (DC.type==2) % Interpolierte Defektkorrektur
   V = kron(DC.V,eye(n));
elseif (DC.type>=3) % Interpolierte Defektkorrektur mit Defektkttransformation:
   % Transformationsmatrizen generieren...
   V=zeros(DC.m*DC.base.s*n,DC.m*n);
   for nu=1:DC.m
      if (DC.type==3)        %TIIDeC
         X2(:,:,nu)=feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),0,'trafo2',varargin:);
      elseif (DC.type==4)    %QR-IIDeC
         X2(:,:,nu)=qr1(feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),0,'jacobian',varargin:));
      elseif (DC.type==5)    %QR-IIDeC2
         X2(:,:,nu)=qr1(eye(n)-DC.base.c(1)*h*feval(odefile,...
            t0+h*DC.m*DC.gamma(nu),0,'jacobian',varargin:));
      end;
   end;
   for nu=1:DC.m
      for i=1:DC.base.s
```

```
         ii=(nu-1)*DC.base.s+i;
         if (DC.type==3)      %TIIDeC
            X1(:,:,ii)=feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),0,'trafo1',varargin:);
         elseif (DC.type==4)  %QR-IIDeC
            X1(:,:,ii)=qr1(feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),0,'jacobian',varargin:));
         elseif (DC.type==5)  %QR-IIDeC2
            X1(:,:,ii)=qr1(eye(n)-DC.base.c(1)*h*feval(odefile,...
               t0+h*(nu-1)+h*DC.base.c(i),0,'jacobian',varargin:));
         end;
      end;
   end;
   for i=1:DC.m*DC.base.s
      for j=1:DC.m
         V((i-1)*n+1:i*n,(j-1)*n+1:j*n)=DC.V(i,j)*X1(:,:,i)*X2(:,:,j);
      end;
   end;

end;

A_nu=zeros(s*n);
for nu=1:m
   t_nu=t0+h*(nu-1);
   for j=1:s
      A_nu((j-1)*n+1:j*n,(j-1)*n+1:j*n)=feval(odefile,...
      t_nu+h*DC.base.c(j),0,'jacobian',varargin:);
   end;
   J(:,:,nu)=eye(n)+hb_kron_eye*A_nu*( (eye(n*s)-hA_kron_eye*A_nu) ...
                                       \kron(ones(s,1),eye(n)));
   K(:,:,nu)=hb_kron_eye+hb_kron_eye*A_nu*( (eye(n*s)-hA_kron_eye*A_nu) ...
                                       \ hA_kron_eye );
end;
KK=zeros(n*m,m*s*n);
for j=1:m
   JJ=K(:,:,j);
   KK((j-1)*n+1:j*n, (j-1)*n*s+1:j*n*s)=JJ;
   for i=j+1:m
      JJ=J(:,:,i)*JJ;
      KK((i-1)*n+1:i*n, (j-1)*n*s+1:j*n*s)=JJ;
   end;
end;
AA=zeros(m*n);
for nu=1:m
   AA((nu-1)*n+1:nu*n,(nu-1)*n+1:nu*n)=feval(odefile,...
```

```
   t0+h*m*DC.gamma(nu),0,'jacobian',varargin:);
end;
S=eye(m*n)-KK*V*(Wd1_div_h-AA*W1);
rho=max(abs(eig(S)));
```

A.4 Maple-Arbeitsblatt zur Berechnung der Daten aus Tabelle 3.3

```
> restart;
  with(linalg):
  alias(Id=&*());
  specR:=S->norm(evalf(Eigenvals(S)),infinity);
  Digits:=20;

Warning, new definition for norm
Warning, new definition for trace

                                I, Id

          specR := S -> norm(evalf(Eigenvals(S)), infinity)

                              Digits := 20

> m:=5;
  basis:='Euler';
  gamma_gauss:=sort(map(x->.5*x+.5,[fsolve(orthopoly[P](m,x),x)]));
  gamma_radau:=sort(map(x->.5*x+.5,[fsolve(orthopoly[P](m-1,1,0,x),x),1.]));

                                m := 5

                             basis := Euler

  gamma_gauss := [.04691007703066800360, .23076534494715845448, .5,
        .76923465505284154552, .95308992296933199640]

  gamma_radau := [.05710419611451768219, .27684301363812382768,
        .58359043236891682006, .86024013565621944785, 1.0]

> if basis='Euler' then
    s:=1;
    A:=array([[1]]);
```

```
    b:=array([[1]]);
    c:=[1];
  elif basis='IMR' then
    s:=1;
    A:=array([[1/2]]);
    b:=array([[1]]);
    c:=[1/2];
  elif basis='IMR2' then
    s:=2;
    A:=array([[1/4,0],[1/2,1/4]]);
    b:=array([[1/2,1/2]]);
    c:=[1/4,3/4];
  elif basis='ITR' then
    s:=2;
    A:=array([[0,0],[1/2,1/2]]);
    b:=array([[1/2,1/2]]);
    c:=[0,1];
  elif basis='ITR2' then
    s:=3;
    A:=array([[0,0,0],[1/4,1/4,0],[1/4,1/2,1/4]]);
    b:=array([[1/4,1/2,1/4]]);
    c:=[0,1/2,1];
  elif basis='SDIRK2' then
    s:=2;
    g:=1-sqrt(2)/2;
    A:=array([[g,0],[1-g,g]]);
    b:=array([[1-g,g,]]);
    c:=[g,1];
  elif basis='RadauIIA(2)' then
    s:=2;
    A:=array([[5/12,-1/12],[3/4,1/4]]);
    b:=array([[3/4,1/4]]);
    c:=[1/3,1];
  fi;
```

$$s := 1$$

$$A := [1]$$

$$b := [1]$$

$$c := [1]$$

```
> R:=unapply(normal(det(augment(stackmatrix(Id-z*A,-z*b),
  array(1..s+1,1..1,[seq([1],i=1..s+1)])))/det(Id-z*A)),z);
```

```
  r:=limit(R(z),z=infinity);
  for i from 1 to s do
    R.i:=unapply(normal(det(augment(stackmatrix(Id-z*A,-z*b),
    stackmatrix(submatrix(A,1..s,i..i),submatrix(b,1..1,i..i))))
    /det(Id-z*A)),z);
    r.i:=limit(z*R.i(z),z=infinity);
  od;
```

$$R := z \to -\frac{1}{z-1}$$

$$r := 0$$

$$R := z \to -\frac{1}{z-1}$$

$$r1 := -1$$

```
> K:=array(1..m,1..s*m):
  for i from 1 to m do
     for j from 1 to m do
       for k from 1 to s do
          if (i>j) then
             K[i,(j-1)*s+k]:=r^(i-j)*r.k;
          elif (i=j) then
             K[i,(j-1)*s+k]:=r.k;
          else
             K[i,(j-1)*s+k]:=0;
          fi;
        od;
      od;
  od;
> for j from 0 to m do
    L.j:=unapply(mul((t-k/m)/(j/m-k/m),k=0..j-1)*
          mul((t-k/m)/(j/m-k/m),k=j+1..m),t
         );
  od:
> W:=array(1..m*s,1..m):
  for i from 1 to m do
  for k from 1 to s do
    for j from 1 to m do
       W[(i-1)*s+k,j]:=L.j((i-1+c[k])/m);
    od;
```

```
  od;
  od;
> W_gauss:=array(1..m,1..m):
  W_radau:=array(1..m,1..m):
  for i from 1 to m do
    for j from 1 to m do
       W_gauss[i,j]:=L.j(gamma_gauss[i]);
       W_radau[i,j]:=L.j(gamma_radau[i]);
    od;
  od:
> for j from 1 to m do
    L_gauss.j:=unapply(mul((t-gamma_gauss[k])/
    (gamma_gauss[j]-gamma_gauss[k]),k=1..j-1)
    *mul((t-gamma_gauss[k])/(gamma_gauss[j]-gamma_gauss[k]),k=j+1..m),t);
  od:
> V_gauss:=array(1..s*m,1..m):
  for i from 1 to m do
    for k from 1 to s do
      for j from 1 to m do
        V_gauss[(i-1)*s+k,j]:=L_gauss.j((i-1+c[k])/m);
      od;
    od;
  od;
> for j from 1 to m do
    L_radau.j:=unapply(mul((t-gamma_radau[k])/
    (gamma_radau[j]-gamma_radau[k]),k=1..j-1)
    *mul((t-gamma_radau[k])/(gamma_radau[j]-gamma_radau[k]),k=j+1..m),t);
  od:
> V_radau:=array(1..s*m,1..m):
  for i from 1 to m do
    for k from 1 to s do
      for j from 1 to m do
        V_radau[(i-1)*s+k,j]:=L_radau.j((i-1+c[k])/m);
      od;
    od;
  od;
> S_IDEC:=evalm(Id+K&*W):
  S_gauss:=evalm(Id+K&*V_gauss&*W_gauss):
  S_radau:=evalm(Id+K&*V_radau&*W_radau):
> specR(S_IDEC);specR(S_gauss);specR(S_radau);

                                0

                  .89666005291005291017
```

.79332010582010582022

Literaturverzeichnis

[1] R.N. de Andrade-Leal, *COLSYS-IDC: A Collocation Solver for Systems of Boundary Value Problems Using the Method of Iterated Defect Correction*, Dissertation, Institut für Angewandte und Numerische Mathematik, Technische Universität Wien, 1999.

[2] W. Auzinger, R. Frank, and G. Kirlinger, *Asymptotic error expansions for stiff equations: applications*, Computing 43 (1990) pp. 223–253.

[3] W. Auzinger, R. Frank, and G. Kirlinger, *Extending Convergence Theory for Nonlinear Stiff Problems. Part I*, BIT 26 (1996) pp. 635–652.

[4] W. Auzinger, A. Eder, and R. Frank, *Convergence Theory for Implicit Runge-Kutta Methods Applied to a One-Parameter Family of Stiff Autonomous Differential Equations*, Report 123/1998, Institute for Applied Mathematics and Numerical Analysis, Vienna University of Technology (1998).

[5] J.W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[6] A. Eder, *Konvergenztheorie für implizite Runge Kutta Verfahren bei hoch nichtlinearen steifen Differentialgleichungssystemen*, Dissertation, Institut für Angewandte und Numerische Mathematik, Technische Universität Wien, 1997.

[7] R. Frank and C.W. Ueberhuber, *Iterated defect correction for Runge–Kutta methods*, Report No. 14/75, Institute for Applied Mathematics and Numerical Analysis, Vienna University of Technology (1975).

[8] R. Frank and C.W. Ueberhuber, *Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations*, Report No. 17/76, Institute for Applied Mathematics and Numerical Analysis, Vienna University of Technology (1976).

[9] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II; Stiff and Differential-Algebraic Problems*, Springer, 1996.

[10] H. Hofstätter, *Defektkorrektur zur iterativen Realisierung superkonvergenter Kollokationsverfahren*, Diplomarbeit, Institut für Angewandte und Numerische Mathematik, Technische Universität Wien, 1996.

[11] K. Nipp and D. Stoffer, *Invariant manifolds and global error estimates of numerical integration schemes applied to stiff systems of singular perturbation type,* Numer. Math. 70 (1995) pp. 245–257.

[12] A. Prothero and A. Robinson, *On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations*, Math. Comp. 28 (1974) pp. 125–162.

[13] K.H. Schild, *Gaussian collocation via defect correction*, Numer. Math. 58 (1990) pp. 369–386.

[14] H.J. Stetter, *The defect correction principle and discretization methods,* Numer. Math. 29 (1978) pp. 425–443.

[15] P.E. Zadunaisky, *On the estimation of errors propagated in the numerical integration of ODEs*, Numer. Math. 27 (1976) pp. 21–39.